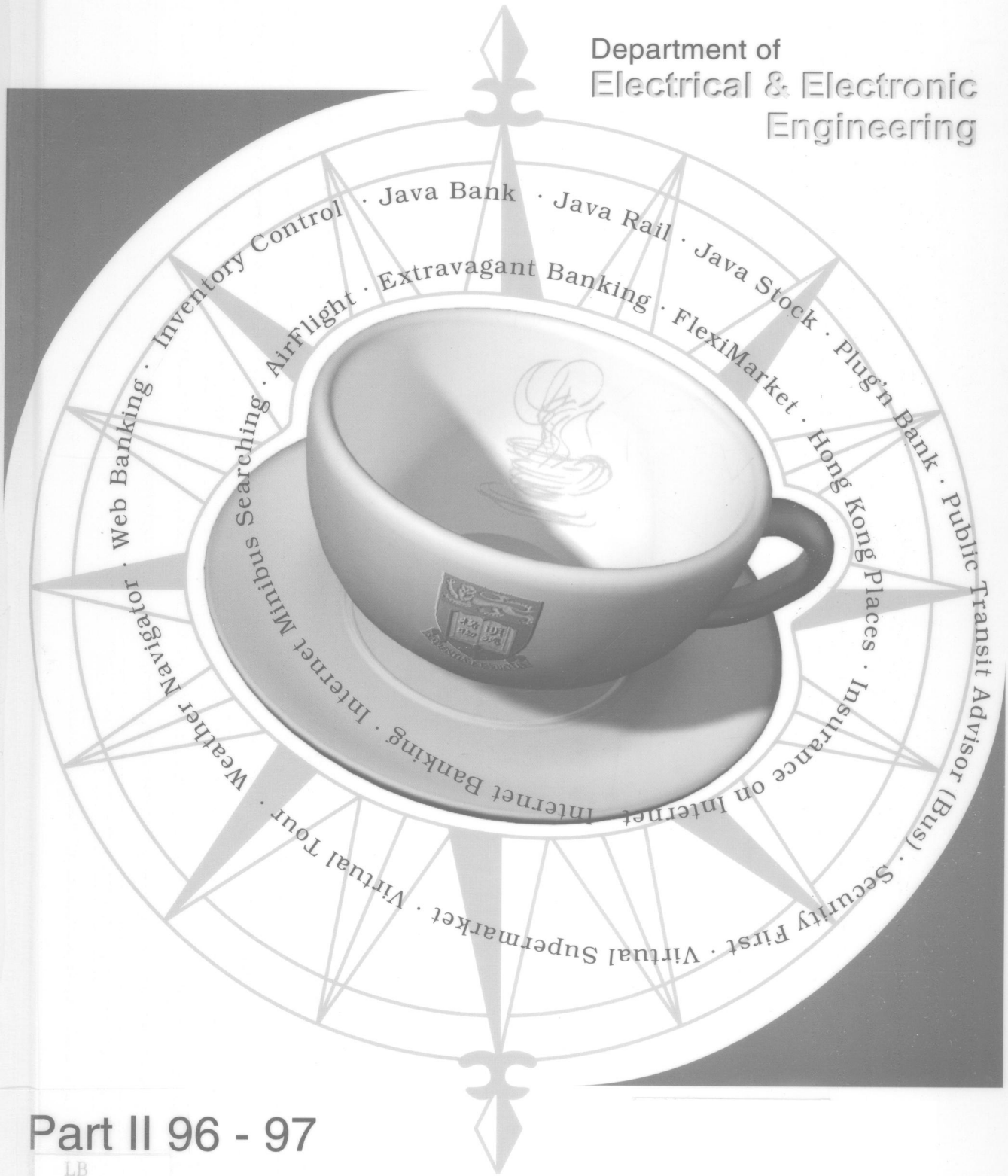
JAVA Projects

THE UNIVERSITY OF HONG KONG

Department of
Electrical & Electronic
Engineering



Part II 96 - 97

http://www.eee.hku.hk/~62269-96

Preface

Nowadays, the development of the World Wide Web on the Internet has become one of the fastest progressing industries in the world. *Java*, a new object-oriented programming language, is about to revolutionize the software development industry. It began its life as more than a programming language with the merits of its machine independent, object-oriented, secure, multithreaded and distributed natures. The fact that being an architectural neutral language makes Java an ideal platform for programming on the Internet.

In view of this, the Department of Electrical and Electronic Engineering of the University of Hong Kong has been offering a Java course for second year undergraduate students since 1996 as an elective(course code: 62269). Students enrolled in this course are required to complete a group project and they are given a variety of topics to choose from. Some of the topics originate from the idea of real life applications in different sectors, such as air-flight, banking, insurance, tourism, transportation and finance, etc. Upon completion of the project, students will get acquainted with the frontier technologies of Java and the Internet.

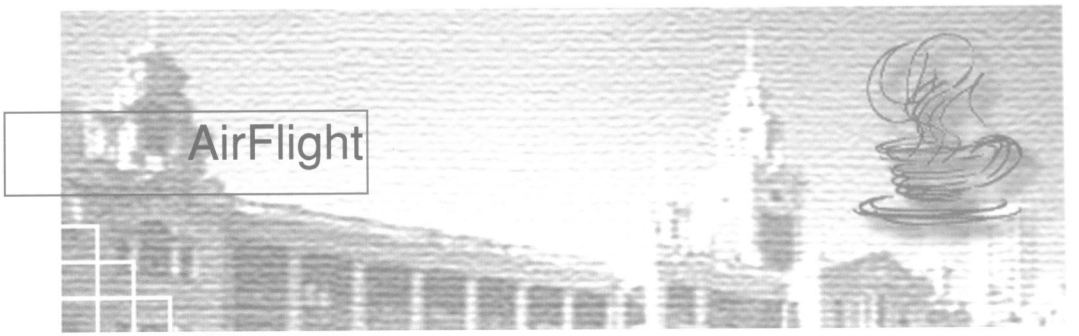
Thanks to the development of computer technology, not only that people do enjoy a higher standard of living, but also a lower cost of transaction. Nowadays, many of the transactions can be made through the Internet. Do you want to book a ticket, order your goods, transfer your money or acquire stock options at home? All you need is just an internet-ready computer and a phoneline.

This book describes all the projects that this batch of second year undergraduates have developed in the year 1996-1997. You are most welcome to visit the website at "http://www.eee.hku.hk/~62269-96/" to get a taste of all these Java applications.

Acknowledgements

Instructor:	Mr. Chris S.K. Yeung	Lecturer
Chief Editor:	Mr. Kevin K.F. Lam	Part II
Associate Editors:	Miss. Rita B.K. Woo	Part I
	Mr. Anthony F. Lam	Part I
Homepage Designers:	Mr. Lawrence M.H. Ko	Part II
	Mr. Kevin K.F. Lam	Part II

Contents



Authors

Au Fuk Kwai
So Hoi Lung
Ng Tat Shing
Wong Chi Ngong
Huen Man Wai

Abstract

The objective of this project is to build an air-flight booking system with the Java programming language.

Introduction

In the past several years, there have been an explosion in the usage of the Internet. This exponential growth is mainly attributed to the advent of the World Wide Web at the European particle physics research laboratory CERN and the graphical interface Mosaic, developed at the National Centre for Supercomputing Application in Illinois. The World Wide Web took arcane commands and plain text screens, and turned them into a graphical, formatted document where pointing and clicking was the main navigational tool. This is the first revolution of Internet.

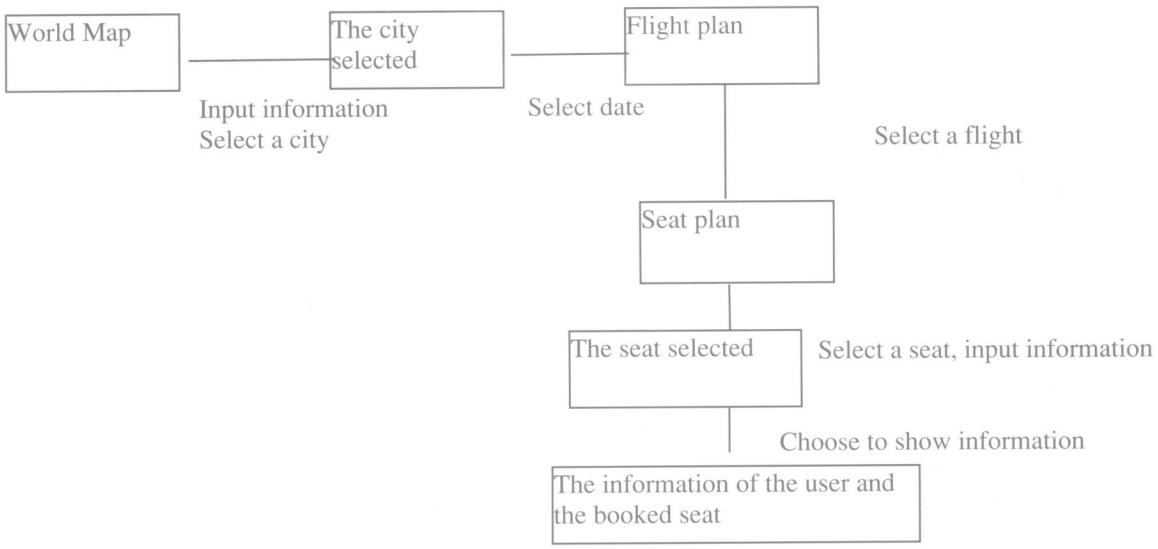
Then comes the second revolution of Internet: *Java* ! Java provides a completely new thinking to distributed computing. Before Java was invented, everything presented in a Web page was completely static. You were essentially bringing up an electronic version of a printed document. However, computers are about dynamic interactions, where the user receives immediate feedback. Java provides this function by allowing the execution of code that can be distributed across the Internet in a portable, robust, secure, high-performance environment.

Before a vacation, there are so many people rushing to the travel agencies and search for travelling information, which is very time consuming. For people who travels frequently, it is also time consuming to book an air-flight frequently. As a result, an air-flight booking system is needed so that people can get the air-flight information and booking on the Internet.

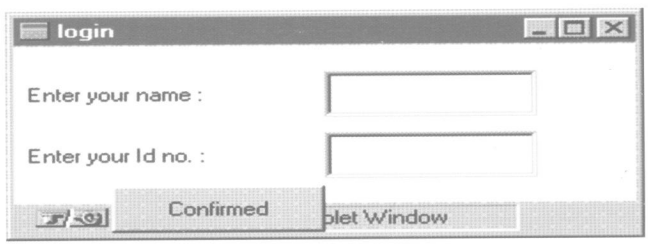
The primary step in software design is to find out the functional requirement of the software. This project should provide the following functions:

1. It shows a world map which can be used to retrieve a city's information by clicking the corresponding city.
2. It provides the information of the selected city such as hotel and restaurants, etc.
3. It provides an airline ticket booking system.
4. It shows the program is user-friendly.

Flow Chart of Main Function

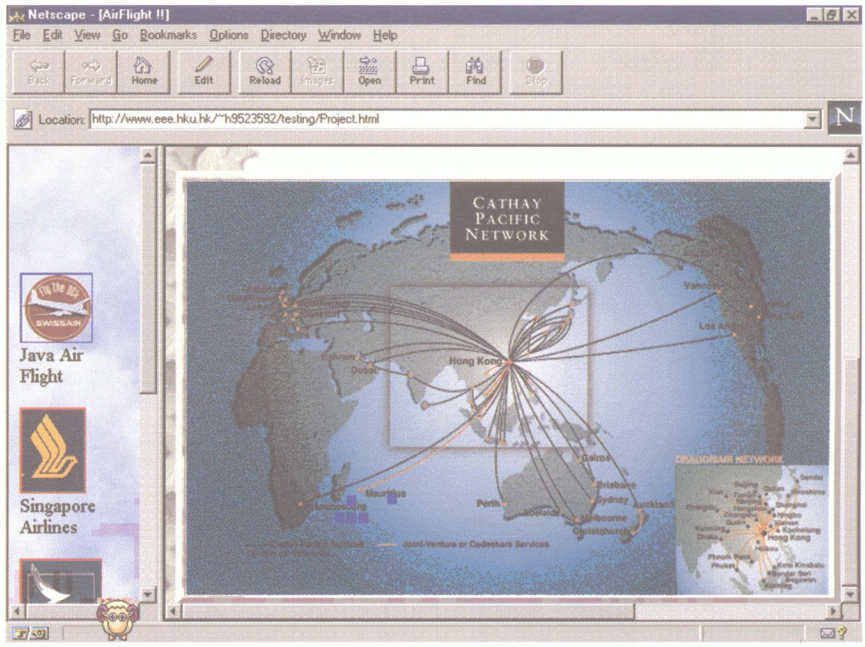


First, the customer has to login before using our system. On entering into our project, the "Project.html" will be loaded. The customer has to enter his name and his identity card number. After pressing the "confirm" button, he can find that there will be two frames in this page.



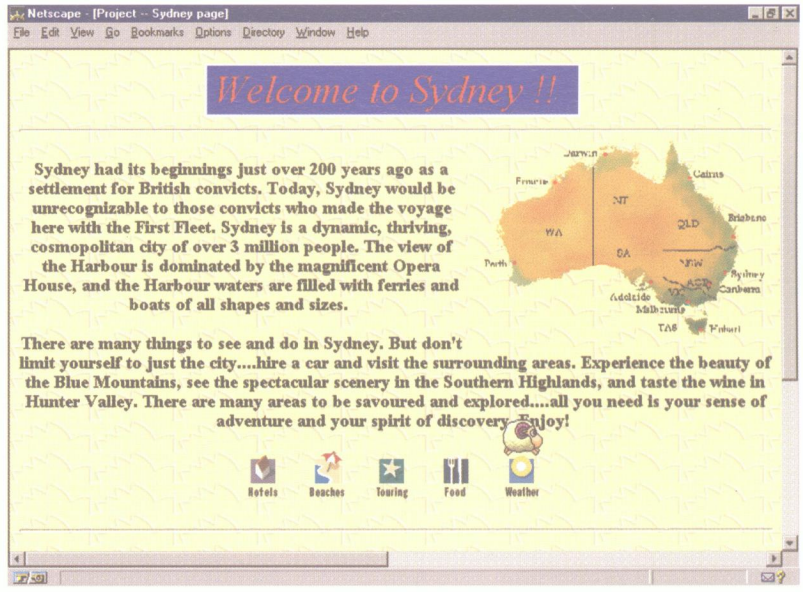
Login Box

On the left hand side, he can link to other airline companies by pressing different icons. On the other side, there is a clock on the top and a world map in the centre. In the world map, there are red spots, which can be selected.



World Map for selected place.

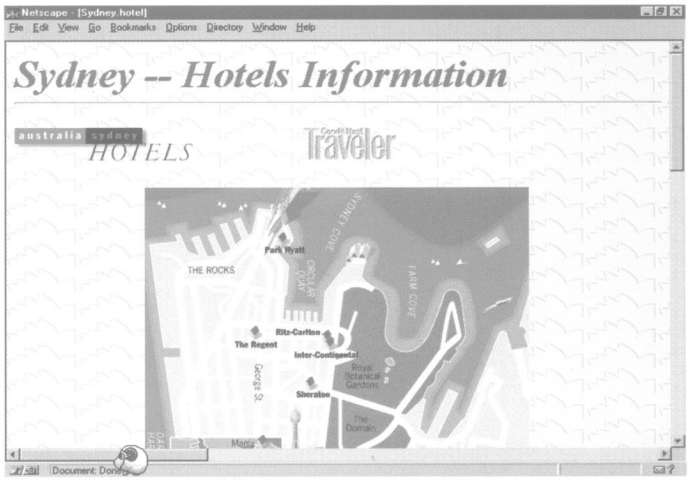
Let's take Sydney as an example:

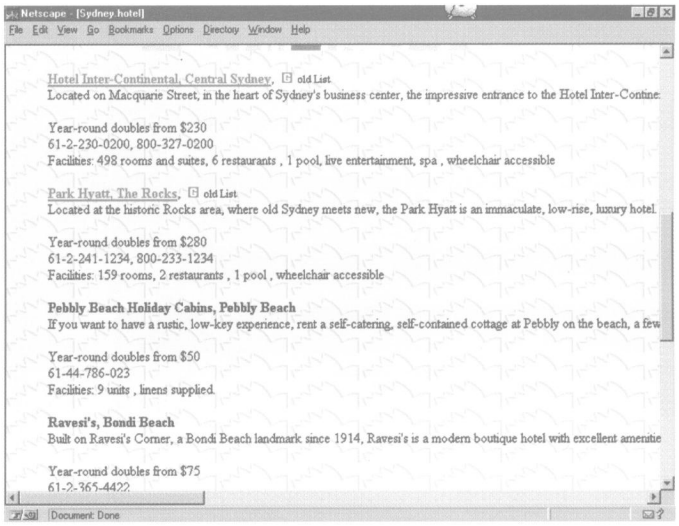


Then the "Sydney.html" will be loaded for different selections. Those pages show us the information about the tourist spots, restaurants and hotels for the cities.

For example:

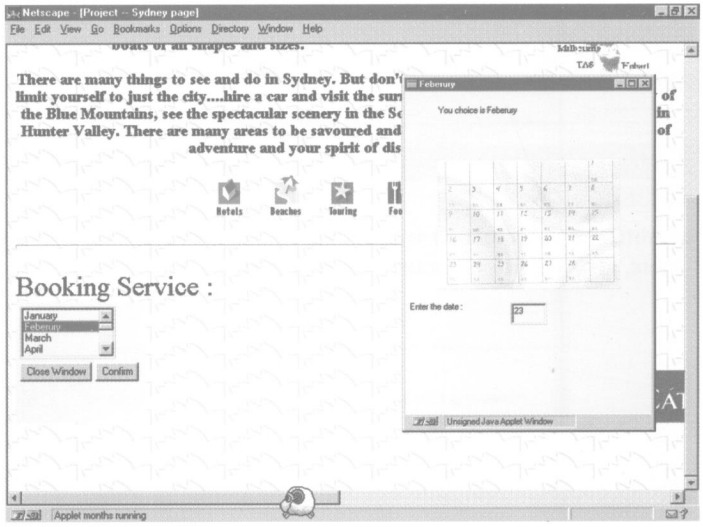
Hotel Information



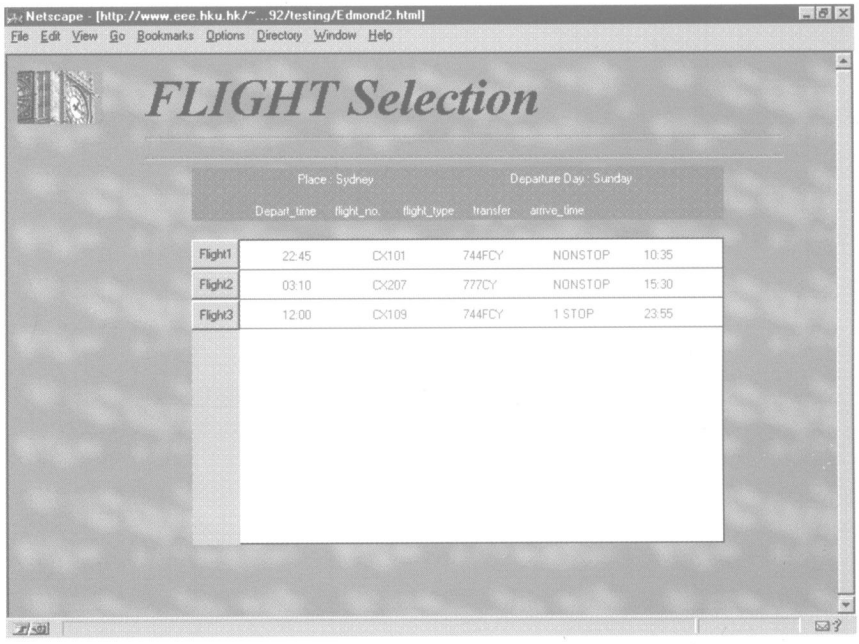


Links to other homepages for extra-information

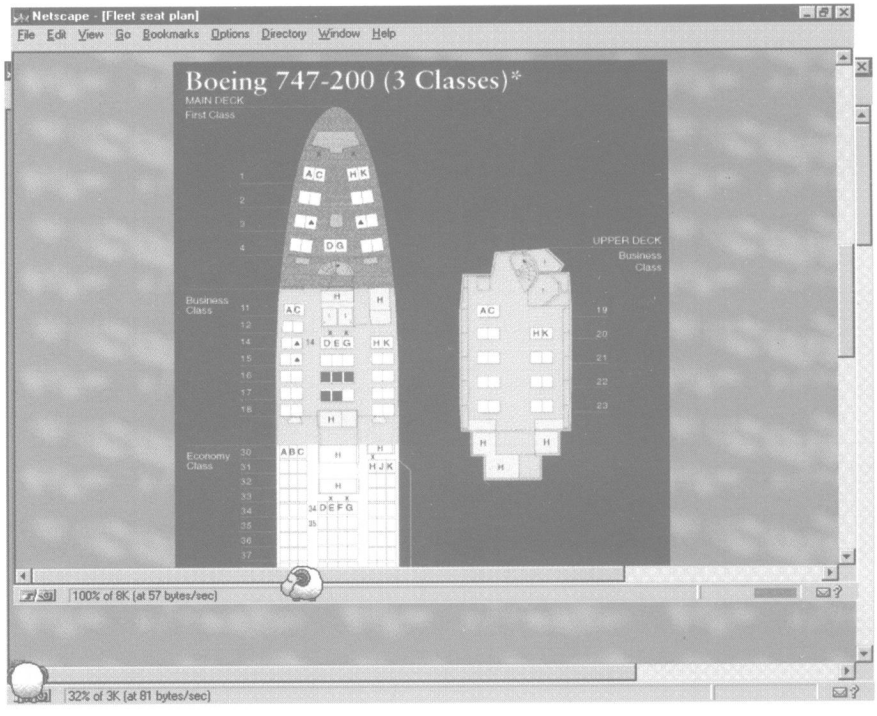
Moreover, the booking service is located at the bottom of these pages. We just need to select the month we want for travelling from the list, then another window will be displayed. The new window contains the calendar and the textfield where we need to enter the date. After pressing the "confirm" button, a new page will be loaded.



The following page is "Edmond2.html". According to the date we entered, the list of aircraft will be displayed. When we choose one of them, the seat reservation page for the aircraft will then be loaded.



In the "744FCY.html" page, we can download the picture about the 747-200 type aircraft. The seat plan is located at the centre of this page which allows reservation of seats, then the corresponding colour of the seat will be changed to blue, indicating that the seat is selected.. Besides, there is a check box button at the bottom of this page. It shows the data about the customer who has logged in.



When we select "file" in the menu bar, there are two items for "show booked seat" and "Quit". If we select the "show booked seat", the seat plan will be displayed again and we can find out which seats we had selected, which serves as a reminder. However, if we choose the "Quit" item, then the check list will be closed.

Msql Database

The need for a Database:

Since our project provides the Flight planner and booking service, a database is therefore required to store the client information.

Below are the corresponding information:

1. Client Name
2. Client ID number
3. Client selected destination, that is, the country he want to visit
4. Travel Date, that is, the Month and the Date
5. The type of Flight he selected, that is, whether it is 747 , 777 and 744 etc
6. The Flight Number
7. Departure Time
8. Arrival Time
9. The Seat where the client has selected which is store in co-ordinate form

Database Structure:

I

A table called "UserLogin", which has two fields, one is the User Name and the other their corresponding ID number. This table has only one row and will be updated every time an user logs in

UserLogin

User Name	ID Number
Xxxxxxxxxx	Xxxxxxxxxxxx

IIb

There is another table called "UserInfo", which stores the clients' specific information. They can be shown to the client to check if it is right.

UserInfo

Name	Id-No	Destination	Month	Date	Flight-Type	Flight-No	Depart Time	Arrival Time
Leon	123	London	3	12	777	Cx344	9:00am	12:00pm
Xxxx	xxxx	xxxxx	xx	xx	xxx	xxxx	xxxxx	Xxxx
Xxx	xxxxx	xxxxxx	xxxx	xx	xxxxx x	Xxx	xxx	Xxxx
Xxxxx	xxxxx	xxxxxxx	xx	Xx	xxx	xxxx	Xxx	xxx

III

Finally there is a table called "UserSeat" which stores the Flight Type, Flight Number and the booked seats coordinates. The Width and the Height fields are needed to show the booked seat to the user after his selection.

UserSeat

User Id	Flight-Type	Flight-no	X-coor	Y-coor	Width	Height
123	777	Cx344	10	23	10	10
123	777	Cx344	20	23	10	10
123	777	Cx344	30	23	10	10
123	777	Cx344	40	23	10	10
457	xxx	xxxxx	Xxx	xx	Xx	xx
xxx	xxx	xxxx	Xxx	xx	Xx	xx

Main function of the project

At the first page of our Web page, a world map is shown. The user can select a city that he wants to go to by pressing the corresponding icon for the city, then the page of that city will be shown. If the user wants to know the flight information at a particular date, he can select the date in the calendar. The next page will be the flight information table. He can select a flight by pressing the flight number, the corresponding seat plan of the selected flight is shown. He can select a seat by pressing the seat and the seat that selected will change to blue. If he wants to check the information of the booked seat, he can press the 'show data' button, then the information will be shown.

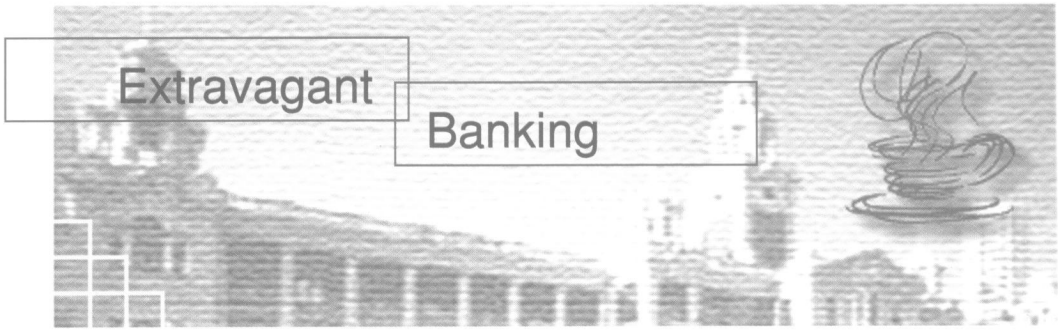
Conclusion

The Java language has generated much excitement in its ability to allow programmer to create and compile codes which can be executed on multiple platforms. It is a simple, objected-oriented, robust language that has been adopted in our project to develop a Web system in the Internet.

The Javatized Web System we have developed in this year has shown to be user-friendly, animated, and interesting, though there are still grounds for improvement, like system performance and art works. It has greatly improved our knowledge in object-oriented and multithreading programming, and software development.

Since the ideas and development tools for multimedia applications are still in the development stage, this system may be improved in the future to give a more intelligent system to the users.

It is our first time to handle such a large software project. In this project, we have learnt a lot about project scheduling and management.



Authors

Au Kin Wai
Chan Hung Yu
Cheung Wai On
Cheung Yee Him
Yau Chun Lim

Abstract

This report describes the overall structure and implementations of the 'Internet Banking Application'. The project is designed for a practical and real-world context after researches into the banking services.

Introduction

Speed, efficiency and flexibility are the key factors for successful personal banking service nowadays. With 'Internet' and its companion 'JAVA', the hottest internet programming language, we can now bring the whole banking service to every networked computer in the world. 'Internet Banking' can surely transform the traditional banking service and plays a leading role in the banking industry in the foreseeable future. It can help a bank to expand its business without much investment. Also, the bank can provide better customer services and hence increases its competitive power.

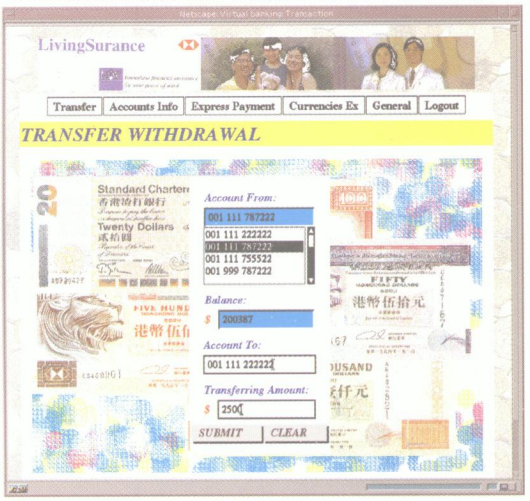
This project is aimed to implement a primary model of internet banking which provides most of the general banking services through the use of the JAVA programming language. Services provided for non-account-holders include 'General Query' and 'Account Application'. Services for account holders include 'Transfer Withdrawal', 'Personal Accounts Information', 'Foreign Currency Exchange', 'Autopay', 'Express Payment Terminal' and 'Ordering of Cheque Books and Gift Coupons'. The ideas of these functions will be further explained in the following sessions.

1. Interfacing Details

Transfer Withdrawal

The transfer withdrawal function allows a user to transfer his deposits from one account to another with a user-friendly interface. Once logged in, the program will search for all users' Hong Kong Dollar accounts from the tables 'Saving_acc', 'Current_acc', 'Deposit_acc' and 'Credit_acc', according to the 'PIN' (Personal Identity Number) input during the login process. All the accounts will then be shown on a list.

The user can choose one account from the list for Transfer Withdrawal and the balance of that account will be shown automatically. The user should then enter the amount to transfer and the destination account for Transfer Deposit. Once completed, the user should click the 'SUBMIT' button to confirm and send out the transaction request.



Upon receiving the transaction request, error checks will be performed on the input data. Errors will be identified and sometimes reported to the user for the following cases.

(1) Amount to transfer larger than the balance of the selected account,
(2) Entry for 'Amount' not of number format, for example, contains an alphabet, and
(3) Destination account not exist.

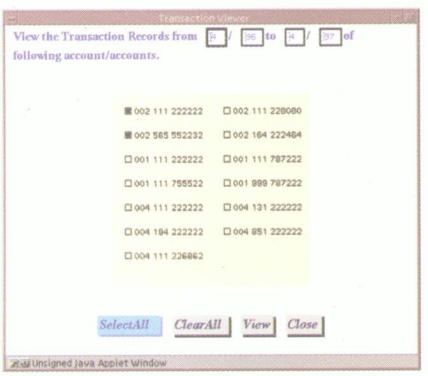
When all entries are correct, the program will subtract the transferring amount from the withdrawal account and add the transferring amount to the deposit account. Transaction details will also be appended to the 'Transaction Records' (say 'Transac_0001' is the transaction record for a user with PIN '0001') of both the sender and the receiver.

When the transaction is successful, a dialog box will pop out to inform the user.

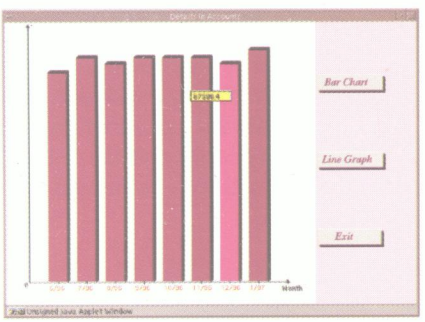
A 'CLEAR' button is also available to clear all the current entries.
* For details of the tables and database, please. refer to 'Database Organization'

Personal Accounts Information

The Account Information function consists of two parts — 'Transaction Information' and 'Balance information'. Once the 'Account Info' button is clicked, the buttons for the transaction information and balance information will appear. The user can click the button for the transaction information and a frame for account selection will appear. The information of the transaction is in the form of a table. The user can select the range of date by filling in the textfield (The default range is the previous twelve months). The user can select the account/accounts that he/she wants to view from the checkbox. The user can choose all the accounts shown in the checkbox by clicking the 'SELECT ALL' button or unselect by clicking the 'CLEAR ALL' button. To view the table, The user can clip the 'VIEW' button and the frame of the table will appear. If there is no such transaction information that the user wants to view, the frame with the sentence 'no such information' will appear.



On the other hand, once the user clicks the button for the balance information, a frame of balance information will appear. The user should choose the range of the date and the account number that they want to view. Then, the user can either click the 'OVERVIEW' button to show the balance table in a frame or the 'PLOT' button to open the frame for plotting graph. Also, the user can view the bar chart or line graph by clicking the 'Bar chart' button and the 'line graph' button respectively.



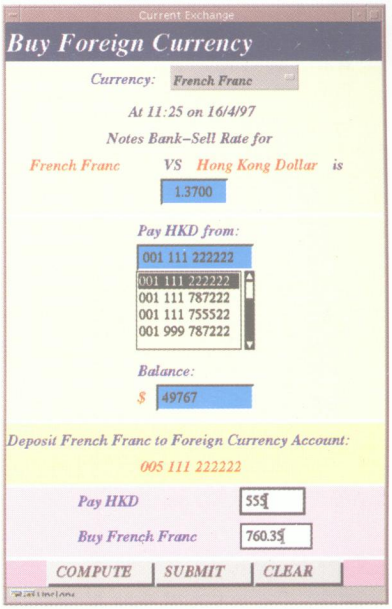
Foreign Currency Exchange

This function allows The users to buy foreign currencies through their own HKD accounts and 'Foreign Currency Account'.

Similar to 'Transfer Withdrawal', once logged in, all the accounts of the user will be listed. The user can choose one account from the list for Payment and the balance in that account will be shown.

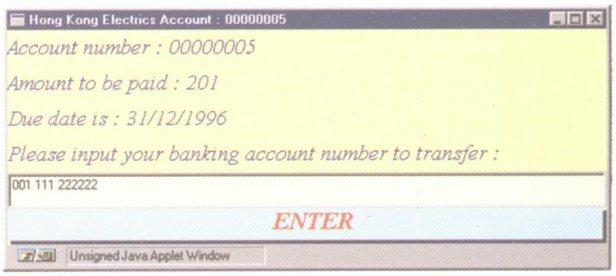
Then, the user can choose one of the 15 available foreign currencies from the list and the corresponding exchange rate will be displayed. Next, the user should enter the transaction amount in HKD. By clicking the 'COMPUTE' button, the exchanged amount in terms of the chosen currency will be computed automatically.

A 'CLEAR' button is available to clear all the entries.
* Note: The part of updating the database is not yet completed.



Express Payment Terminal

'Express Payment Terminal' is set up in order to save clients' time to pay the bills, like those for towngas, water, electric power, etc.



In our system, clients can pay charges to five companies, namely the Hongkong Telecom, China Light and Power Company, Hong Kong Electric, Hong Kong and China Gas Company and Office of Water Authority. In the main menu of express payment terminal, client can select one from these companies. Then the system will search the bill the client should pay after he/she had input the account number of the specified company. If the bill has been paid, a message will be displayed to inform the client. Provided that the bill exists, then client should key in the saving account or current account number want to transfer the amount of money to pay the charges. Next, a new frame will be displayed to tell the client the information of the transfer that includes information of the balance before transfer, the amount of charges should be paid, and the updated balance after payment. The client can select continue to process or cancel. If process is selected, the database is updated and the payment is completed.

Autopay, Cheque Books and Gift Coupons applications

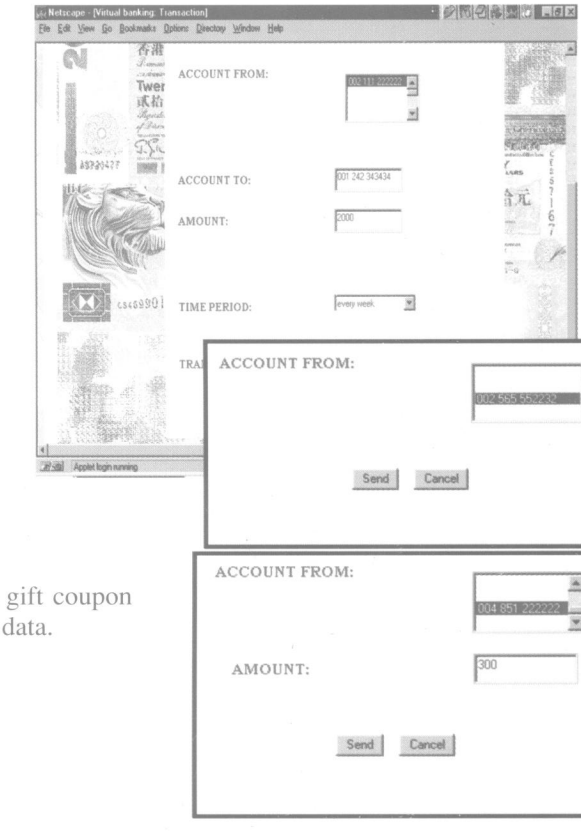
After entering the main page of Extravagant banking, clicking the fifth button 'General' on the top will go into the general application page. In this page, three general applications can be done with much convenience. These include the application of:

a. Autopay
b. Cheque book
c. Gift coupon

By clicking one of the three button at the bottom of the page, the desired application can be performed.

a. Autopay

Here, the customer can choose the account he wants to transfer money from on his own account list. Then he can enter the received account, amount of money transferred, time period and date for the autopay. After all, clicking the 'Send' button will finish the application and all the data will be sent to the database. Clicking the 'Cancel' button will go back to the general application page.



b. Cheque book

Similarly, the customer can choose which current account he wants to apply for a cheque book from his own account list. Then clicking the 'Send' button will make the application and data will be sent to the database.

c. Gift coupon

After the account was chosen and the value of the gift coupon was entered, clicking the 'Send' button will send the data.

2. Database Design

The database is built, actually based on the need of every module in our java program and it can be divided into four parts:

(i) Personal information of registered users
(ii) Ordinary account records, for example, saving account, current account, foreign account etc.
(iii) Personal account/transaction records
(iv) Express payment records
(v) Autopay records
(vi) Other services

Personal Information

Within this table, several personal details of the users are stored and these include the fields of:

1. PIN - Personal identification number that used in login
2. Pass - Password for the user
3. Common details - Name, HKID, Sex, Age, Address, Region, Phone
4. LastUpdate - Date of application of this online service

Ordinary Account Database

In our project, four types of accounts are created and illustrated:

1. Saving account
2. Current account
3. Deposit account
4. Foreign currency account

Saving Account, Current Account and Deposit Account
In these table, fields of AccNum (account number), PIN, BranchCode, DateOpen and Balance are included. As for a particular user, he/she can have more than one of these accounts and therefore, for a particular PIN, it may appears many times within each account table.

Foreign currency account
In this table, fields of AccNum, PIN, BranchCode and DateOpen are recorded. One thing different from the account types above is that for a particular user should have only one foreign currency account.

Personal Account/Transaction Records

Personal Account Records
Personal account records actually describe the details for a certain account, which in this program, is the foreign currency account. Within this table, Date, Time, CurrencyType and Amount are stored. This table tells you how many types and amount of currency you have in that account

Personal Transaction Records
Personal transaction records store all the transactions for that user. It records Date, Time, Type, Amount, HostAcc and TranAcc. This table is tailor-made for all users that can see their past transaction records whenever they want.

Express Payment Records

In this part, payment to some government departments and companies can be made as request and the database stores all the necessary information on payment status. It includes Account_no, Amount, Date_of_payment, Due_date and PayOrNot. Based on these parameters, express payment can be made.

Autopay Records

This part of database contains the information of registered autopay user. This includes the fields of PIN, AccFr, AccTo, Amount, Period1 (for example, every week), Period2 (for example, every Monday), Date and Time.

Other Services

These services include the application of cheque book and gift coupon. Based on the requests, our bank will send the right things to our customers.

Cheque Book
This includes PIN, Account, Date and Time. Whenever a cheque book is applied for a particular current account, a corresponding details are written into the database.

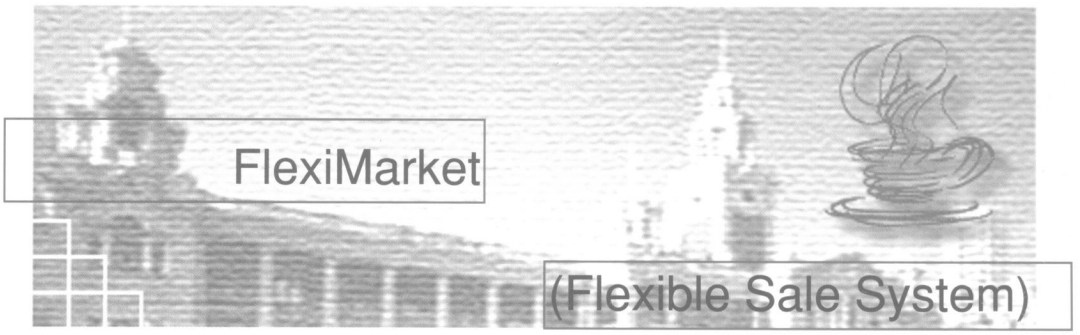
Gift Coupon
This part stores the buy of the gift coupons and it records the PIN, AccFr, Amount, Date and Time.

Conclusion

This project can successfully provide most of the banking services. Besides, it is equipped with the advantages of platform independence and flexibility as inherited from the JAVA programming language.

With platform independence, the project can be developed and readily run on a wide range of computers (for example, PCs, Mac, Sun, HP, etc.) and operating systems without the problem of compatibility. This ensures that the application project is highly accessible and can reach most customers regardless of the systems they are using. Hence, the project will be more cost effective.

Also, the project is implemented by 'Object Oriented Programming' which allows new functions and features to be added without much changes in original codes. Thus, the bank can always keep abreast with the latest business trend and puts forward new schemes and services on the internet as fast as possible with the least resources.



Authors

Cheung Chun Lung, Parker
Cheung Ka Lung
Lee Ngo Chim, Leo
Lo Hoi Kong, Thomas

Abstract

We describe in this report the java project of a supermarket web page. This web page could make buying things in real supermarket by only clicking the mouse at home possible. This project was also implemented with large number of sales strategies. These strategies could enhance the purchase made by customers. The customers can buy things even faster and more convenient than going to a nearby supermarket.

Introduction

This java web page consists of four parts : *Main Menu, Shopping, Management, Club.*

The user can go to Shopping, Management and Club at any time they want by just clicking the corresponding links in the Main Menu. He can choose the products that he wants to buy in the page Shopping. After finishing his purchases, he can browse the products that he has chosen and the amount he has to pay will be shown.

He can also apply for a membership in the page Club. He can enjoy many discounts and benefits in buying if he becomes a member. A colorful bar chart showing his purchases in the past twelve months can be checked in the page Club.

On the other hand, the management team can also change the prices and the products in the page Management. This page can save a lot of time for the manager in 'editing' the supermarket.

1. ActiveMenu

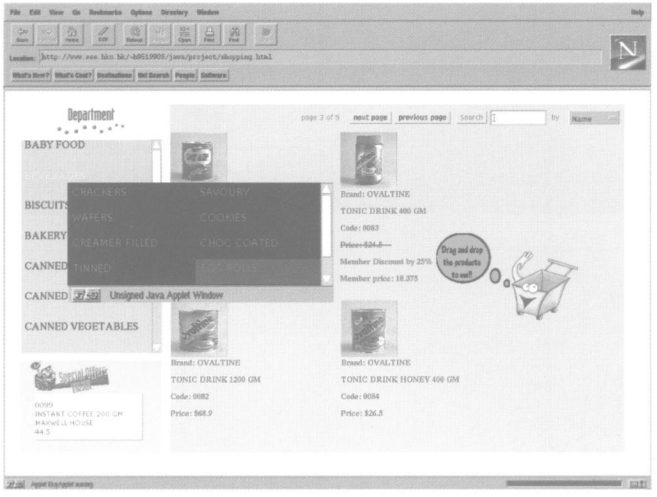


Figure 1. ActiveMenu

ActiveMenu is a menu system that incorporates a map with submenu maps. All the maps are divided into regions where labels are displayed. Each region can trace the mouse actions and perform corresponding functions.

ActiveMenu is derived from ImageMap.

Applied in this software project, the main menu displays a set of labels known as the departments in the FlexiMarket. The labels in the submenus are known as categories.

When the cursor points to a department, the label (for example, Canned Meat) is highlighted and the corresponding submenu of categories is displayed. Once the user clicks on a category, the BuyApplet (introduced later) will respond by loading into the required category.

JavaScript Cookies

JavaScript was written to enhance the ActiveMenu. The Cookie protocol (defined by Netscape) enables certain script languages to be stored and retrieve pieces of data on the client side.

Cookies are a general mechanism which server side connections (such as CGI scripts and JavaScripts) can use to both store and retrieve information on the client side of the connection

After selecting a category, the category name (for example, Bread in Bakery department) will be stored as a cookie called FSS Category. This category will be loaded when the user visits the page next time.

DiscountAdv

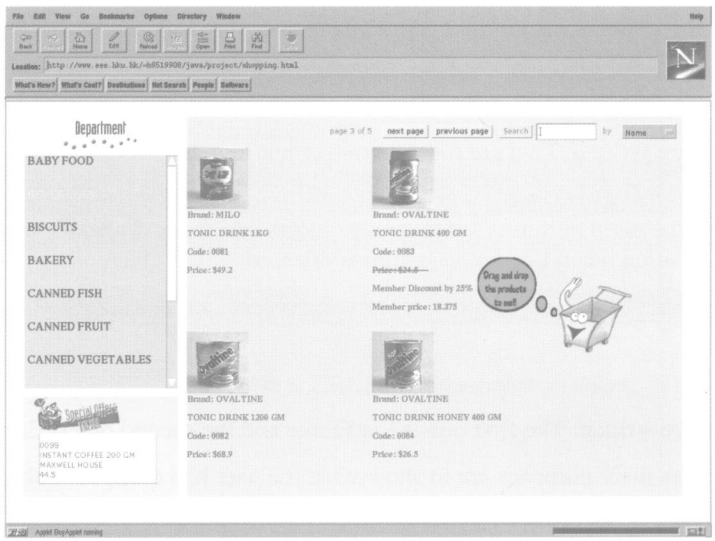
To save customers' time, the information of all discount products is continuously displayed in the applet called DiscountAdv (discount advertisement). The products are displayed one by one with the scrolling effect.

When the mouse moves on to the applet, scrolling will be paused. The customer can click the mouse button to see the details of the product on the BuyApplet.

2. BuyApplet

In order to create a user-friendly buying environment, a drag and drop buying page is used. In this page, customer can buy whatever he wants by dragging the product image and dropping it to the product bag which involves only simple mouse actions. Figure 2 shows the drag and drop buying page.

Figure 2. The drag and drop buying page



From Figure 2, the page is divided into two main parts, the active menu panel and the drag and drop buying panel. Both panels are written in Java applet. For the active menu panel, it shows a list of category names such as biscuits and baby food. For the drag and drop buying page, there are several parts. There are product images and their information, product bag, search panel and the "page up" and "page down" buttons.

The page interacts with the users. Users can find the products he want by two means. The first means is to select the category from the active menu and the second means is to use the search panel which can search the product by names, product codes, prices and Brands. Then, the drag and drop buying page will search and load the required product information from the database and the loaded images which are saved locally.

The users can buy the product he want by just dragging and dropping the product image to the product bag, any successful drop will be indicated by a change of the product bag.

The user can examine what he has bought by click the product bag once, a frame which shows the buying list will be displayed on the screen.

Figure 3 shows the buying list frame. The user can also end shopping in this buying list frame and an e-mail which contains the bill will be sent to the user (Figure 4).

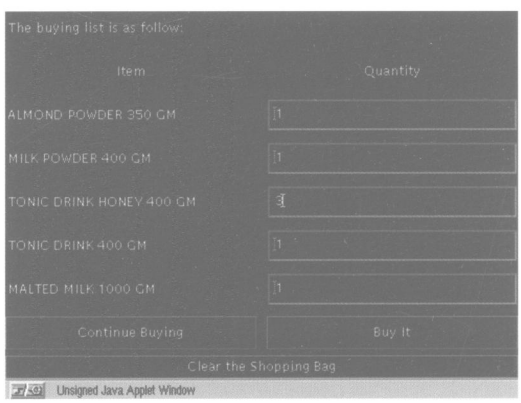


Figure 3. The buying list

Figure 4. The bill

In the BuyApplet, some objects are created such as Bag and Product which will be explained later. In addition, there are DraggableImage and the TargetImage which are the draggable product image and the product bag respectively.

The BuyApplet is flexible in which the management team can change some features of the page by just changing a few lines of the program. The features include the number of images per screen, the positions of the images, the product information as well as the order of the product information.

The drag and drop buying panel is made up of three main parts, *the drag and drop, frame and product information.*

Drag and Drop

In this part, four classes are written. Two are for the product image and the other two are for the product bag.

For the first two classes, the main purposes are to create objects which are draggable and the objects will return to their original position when the mouse button is released.

For the remaining two classes, the main purpose is to create objects which can detect which product has been dropped on it.

Frame

In this part, two classes are written. The first one is ListFrame and the second one is ErrorDialog1.

For the class ListFrame, its main purposes are to show what the user had dropped to the product bag and the amount of the corresponding products. In the list, user can change the number of product dropped by changing the number in the text box.

Moreover, user can interact with the list by clicking the buttons on the bottom of the list. By clicking the buttons, user can empty the product bag, continuing the buying process and end shopping. When the user chooses to end shopping, a bill will be displayed showing what the user had bought and the total price.

For the class ErrorDialog1, its main purpose is to tell the user when he types wrongly in the text box such as typing non-number characters.

Product Information

In this part, two class are written which are ProductDetails and the DrawCross

For the ProductDetails, the main use is to show the product information of the product. In the page, product information such as product names, product brands, product prices and product codes are displayed. Moreover, strategy of the product will also be displayed if any.

For the DrawCross, its main use is to cross out the original price of the product if there is special strategy such as discount.

3. Club

The users can apply for a membership in this page by filling in the personal information. This page includes a club form and a bar chart.

For the club form, a class Clubclass is written and for the bar chart, two classes Chart2 and DataChart are also written.

The bar chart can show the purchase of the customers in the last twelve months. It is very colorful. The bar chart get the purchase data from the data base and draw the twelve bars automatically. The bar chart is located horizontally.

The colors of the bars and the purchases of any months can be changed with very little amendments in the source code.

The bar chart can give a very good picture to the customers so that they can know the amount they have bought in our page. This is not available in a usual real supermarket.

4. Supermarket Model

The supermarket model is divided into two parts, namely the supermarket objects and the sale strategies. While the BuyApplet will directly instantiate the supermarket objects, the strategies are indirectly called to compute the prices.

The sale strategy classes may be amended or added from time to time.

Supermarket Objects

Class	Use
Storage	The superclass for Bag and Product. Mainly derived from the Hashtable class.
Bag	A class for storing selected products (inside categories) using the product code as key for retrieval.
Product	A class for storing a number of items. Each product (for example, orange) has only ONE instance of Product.
Item	A class for storing the price of different items in a Product (for example, apple).

Table 1. Classes for Supermarket Objects.

Sale Strategies

Class	Use
Strategy	The supeclass of all sale strategies.
DefaultStrategy	The default strategy used prior to other strategies. Applied to all products.
ByPercent	Discount each product by a fixed percentage.
ByValue	Discount each product by a fixed value.
ByVolume	The superclass of ByVolumePercent and ByVolumeValue.
ByVolumePercent	Discount the products by a fixed percentage when the quantity has met the required amount.
ByVolumeValue	Discount the products by a fixed value when the quantity has met the required amount.
ByPackage	The superclass of ByPackagePercent and ByPackageValue. A package is defined as a combination of products with different specified quantities.
ByPackagePercent	Discount the products by a fixed percentage when the combination of products has met the required package.
ByPackageValue	Discount the products by a fixed value when the combination of products has met the required package.
ByClubPercent	Discount each product by a fixed percentage if the user is a Club member.
ByClubValue	Discount each product by a fixed value when the user is a Club member.

Table 2. Classes for Sale Strategies.

5. Database Structure

There are three main parts in the database, namely SUPERMARKET, CLUB and STRATEGIES. They show the logical view of SUPERMARKET which is divided into Departments and each Department is subdivided into Categories.

Figure 6 and Figure 7 show the logical views of CLUB and STRATEGIES respectively.

Figure 8 shows the logical view of the database tables, with the names of tables the fields are in shade.

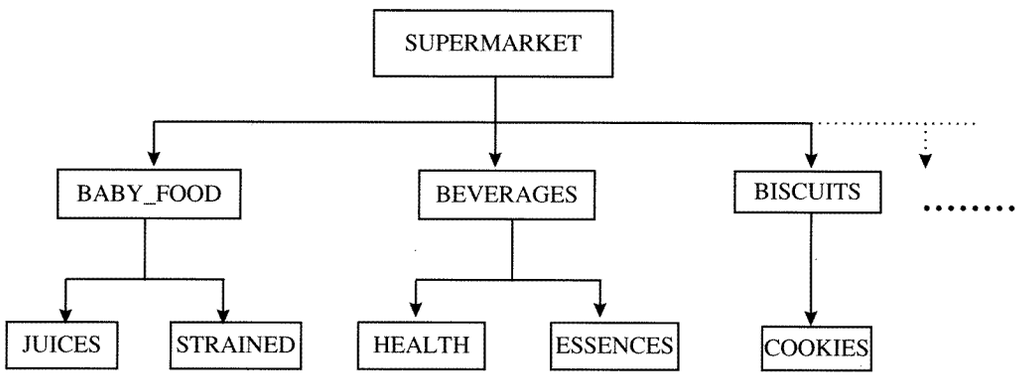


Figure 5. Logical view of SUPERMARKET

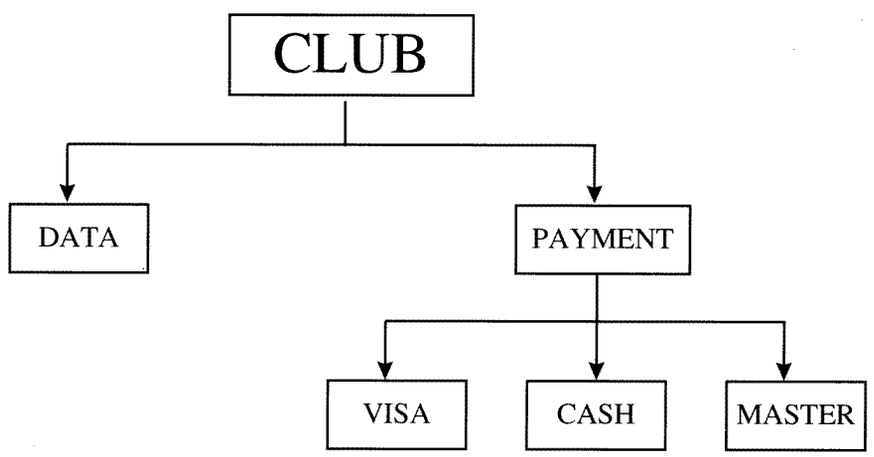


Figure 6. Logical view of CLUB

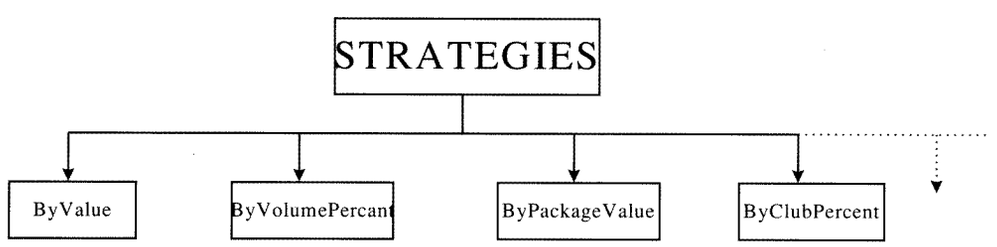
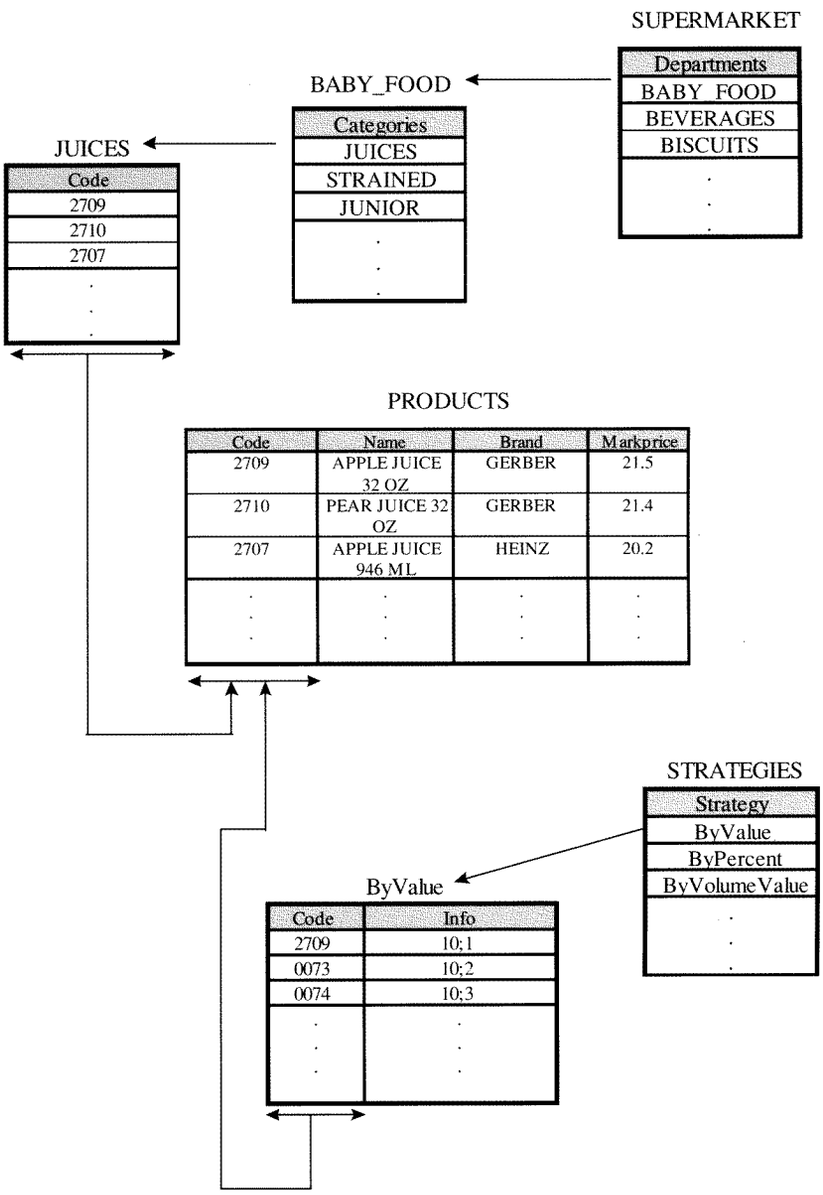


Figure 7. Logical view of STRATEGIES


SUPERMARKET
Departments
BABY_FOOD
BEVERAGES
BISCUITS
BABY_FOOD
Categories
JUICES
STRAINED
JUNIOR
JUICES
Code
2709
2710
2707
PRODUCTS
Code
Name
Brand
Markprice
2709
APPLE JUICE 32 OZ
GERBER
21.5
2710
PEAR JUICE 32 OZ
GERBER
21.4
2707
APPLE JUICE 946 ML
HEINZ
20.2
STRATEGIES
Strategy
ByValue
ByPercent
ByVolumeValue
ByValue
Code
Info
2709
10;1
0073
10;2
0074
10;3

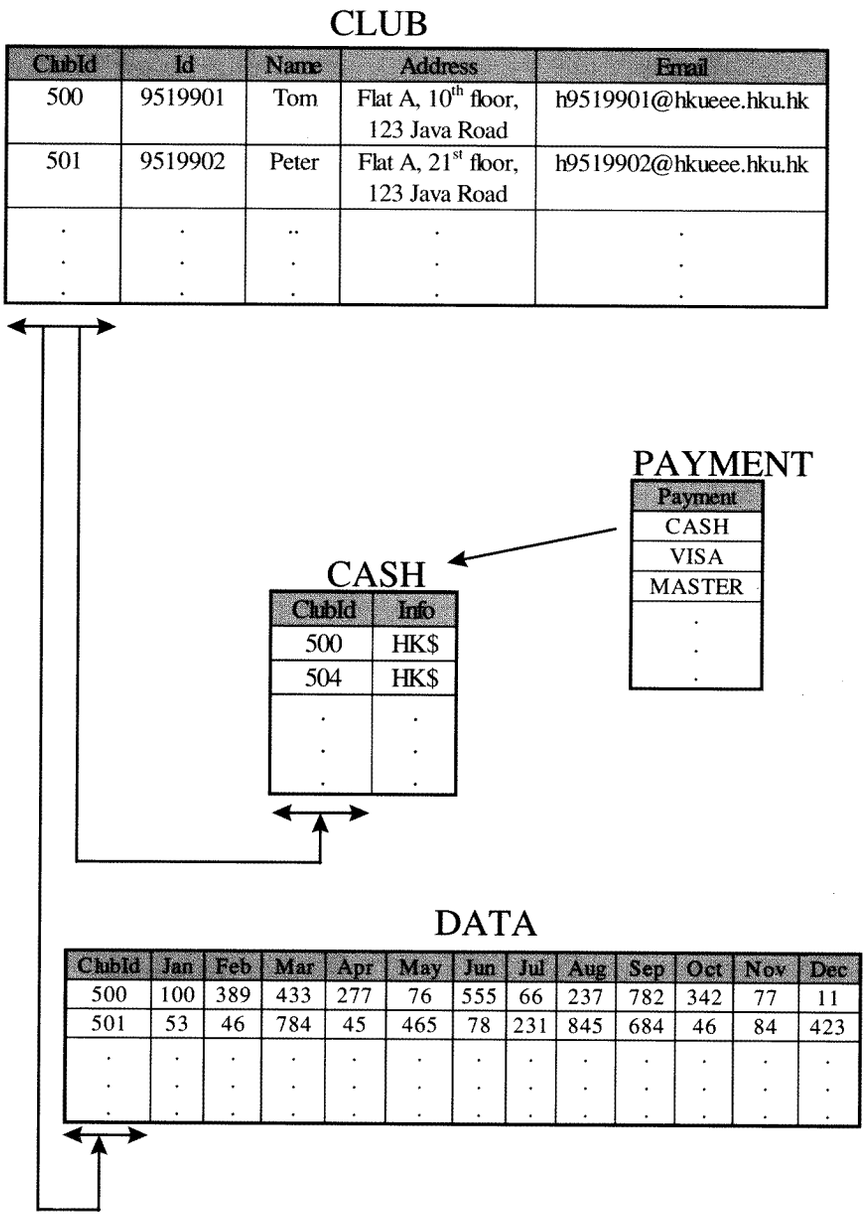
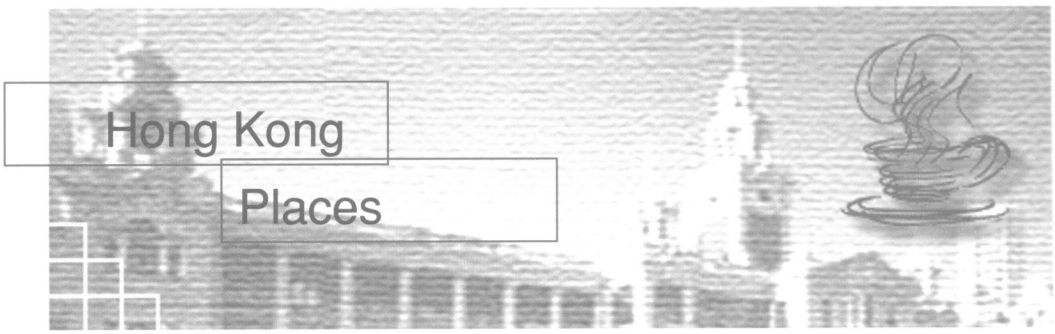


Figure 8. Physical view of database tables

Conclusion

In this project, we have designed and implemented an online sale system using Java. The system is flexible and expandable as the Java language is object-oriented. Through the internet, the platform-independence and a distributed client-server nature of the system make it more advantageous than the traditional sale means.

The system is highly applicable to new style supermarkets to create an efficient and a highly maintainable place for business. In future versions, more features will be implemented, namely the auto-pay system and the shopping-schedule system.



Authors

Chan Chun Chung
Lam Chung Ngok
Lai Ka Ping
Ko Wing Biu

Abstract

The objective of this project is to design a homepage, "Hong Kong Places" which introduces some of the famous places to people who are interested in Hong Kong, and to provide a fast, user-friendly search engine with all the necessary information about some activities in Hong Kong.

Introduction

Hong Kong is one of the most famous places in Asia, and thousands of people come to this beautiful city every year. In fact, tourism is the second major business in tertiary industry in Hong Kong. Thus, we need to try our best to serve the tourists in order to make Hong Kong an attractive place.

For those who want to visit Hong Kong, they may want to know something about Hong Kong before coming to Hong Kong. But how can they get the latest and accurate information about Hong Kong when they are staying overseas? The solution is easy and simple! They can get the updated information through the most advanced technology — the Internet. Actually, the Internet and World-Wide-Web technologies allow everybody to get instant information at different places and times. Thus, it makes our initiation to design a homepage so that anyone living abroad can access the homepage through the Internet at any places and times.

The purpose of the project is to design a homepage using a powerful computer language — Java. The title of the homepage is called "Hong Kong Places". It has two main functions. One is the map searching, the other is the database searching. The map searching is used to provide a search using a map of Hong Kong. Using the map searching, people can get the information about the district that they want to know by simply clicking the point on the map. Apart from it, the database searching is used to provide a searching service by clicking the interested activities and places. Using the database searching, people can get much detailed information about some activities in Hong Kong by simply clicking the activities and places that are concerned. All of these will be demonstrated below.

1. Demonstration of the Homepage

Our project can be operated by using any internet browser which supports Java applets. Figure 1 below depicts the initial layout of the page loaded through a web browser.

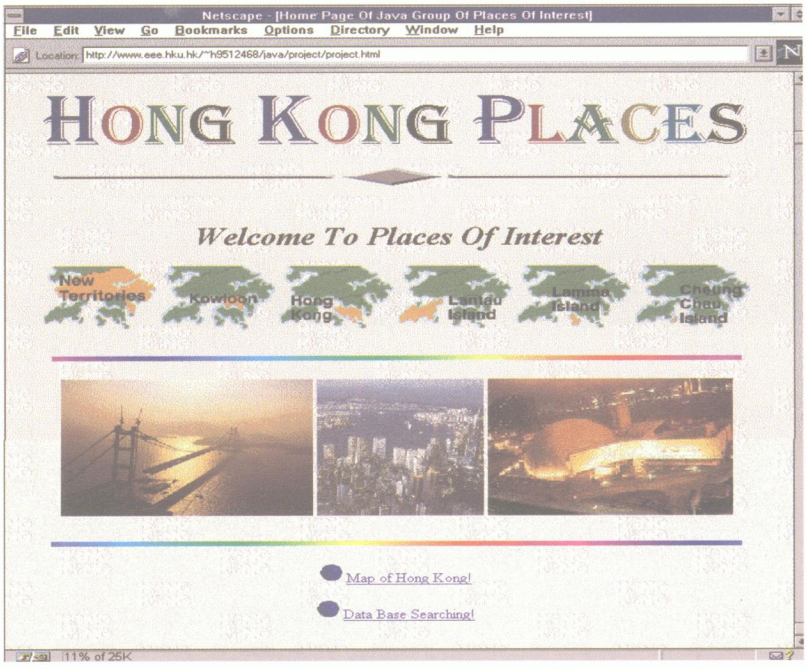


Figure 1 The Initial Layout of the Page

The page has two main functions. One is *map searching*, the other is *database searching*. The user can click on any one of these to activate the function.

(i) Map Searching

By clicking on the choice Map of Hong Kong, the map of Hong Kong will be displayed. People can get information about the district that they would like to know by simply clicking the point on the map. Figure 2 shows the map of Hong Kong.

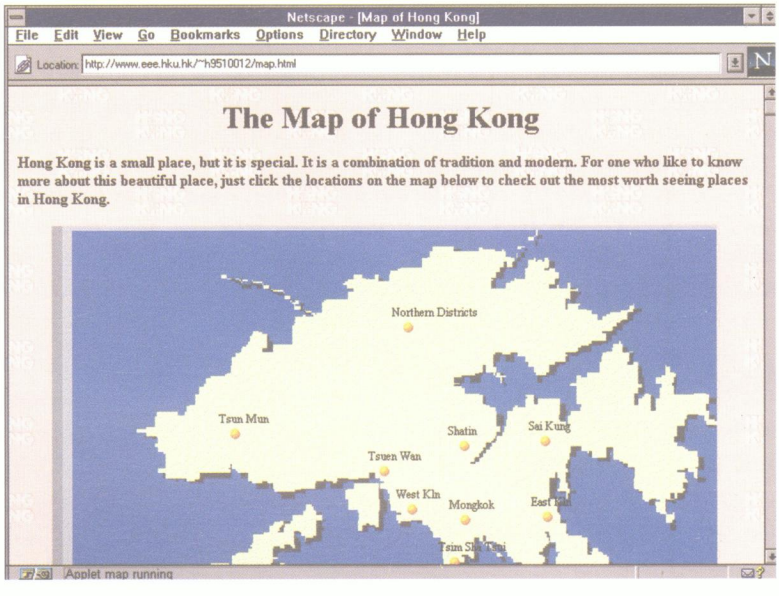


Figure 2 The Map of Hong Kong

For example, when the user click on the dot 'Tsim Sha Tsui' on the map, a detailed description about Tsim Sha Tsui will be displayed. The following figure shows the information about Tsim Sha Tsui.

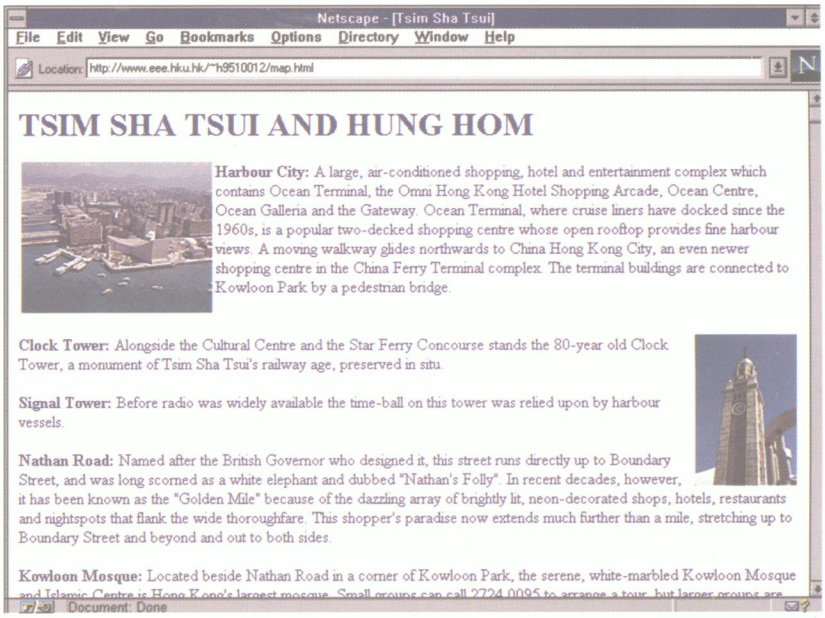


Figure 3 The Information about Tsim Sha Tsui

The description is detail and includes beautiful pictures. As shown in Figure 3, it provides the information of the famous places in Tsim Sha Tsui. It not only provides descriptive text about the place but also the photographs. This can attract the user to read through the text and become interested in the place.

(ii) Database Searching

When the user click on the choice Database Searching, another applet is loaded and a searching application is started. Figure 4 shows the searching application.

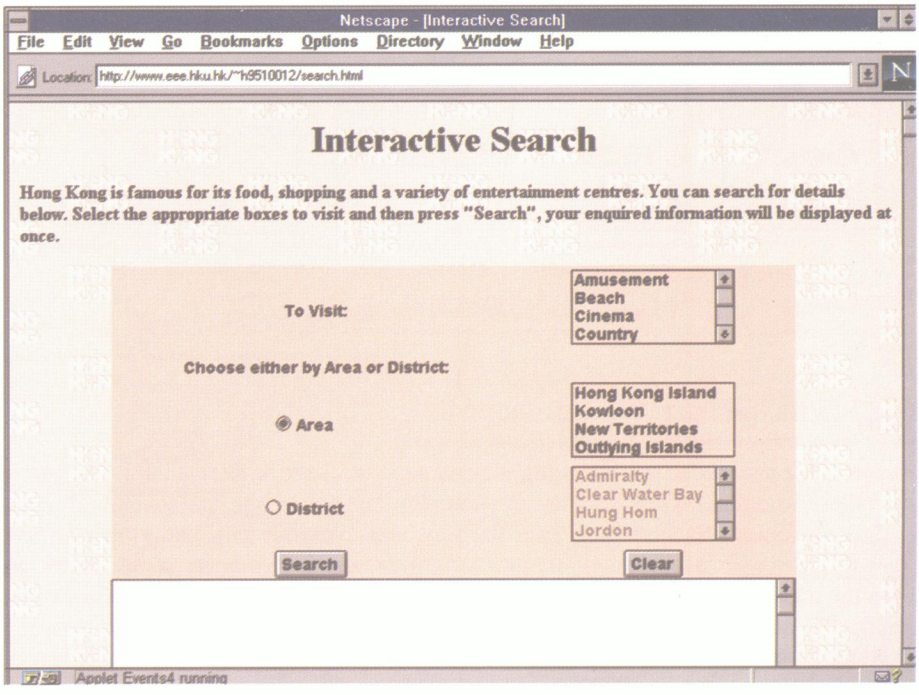


Figure 4 The Searching Application

This applet provides some dialog boxes for the user to choose. The user can search his interests by highlighting the place he would like to visit, and also the area or the district that the activity is located. Then, after clicking on the Search button, the required information will be displayed in the provided box a few seconds later.

2. Java Language

In this project, the homepage is designed with the language — Java. For the map searching and the database searching, four main classes are constructed. They include *map.class, area.class, search.class* and *event.class*. First, the map.class is used to display the map of Hong Kong on the homepage at the desired position and size. Second, the area.class is used to display yellow dots on the map of Hong Kong at the desired position and size with the name of the district. Third, the class search.class is used to display the dialog box with several activities inside the dialog box. Besides, it provides an access to the database and searches the necessary information in the database, and thus displays the result in the dialog box. Lastly, the event.class is used to display the dialog box with scroll bars on the homepage at the desired position and size for displaying the searching result in the dialog box.

3. Database Organization

Our database was written by using the MSQL. Totally, we had inputted at least 15 tables into the database called HKPlaces.1. All of these tables have the columns *Name, Addr, Area, District* and *Tel.* Some of them may also contain the column called *Type*.

In all of these tables, the columns Area and District were used as the key for searching. When we access the database, the user have to provide one of these keys, the area or the district, to search the information. Then, all of the columns of the matched results will be displayed. The structure of our database design is shown as in Figure 5.

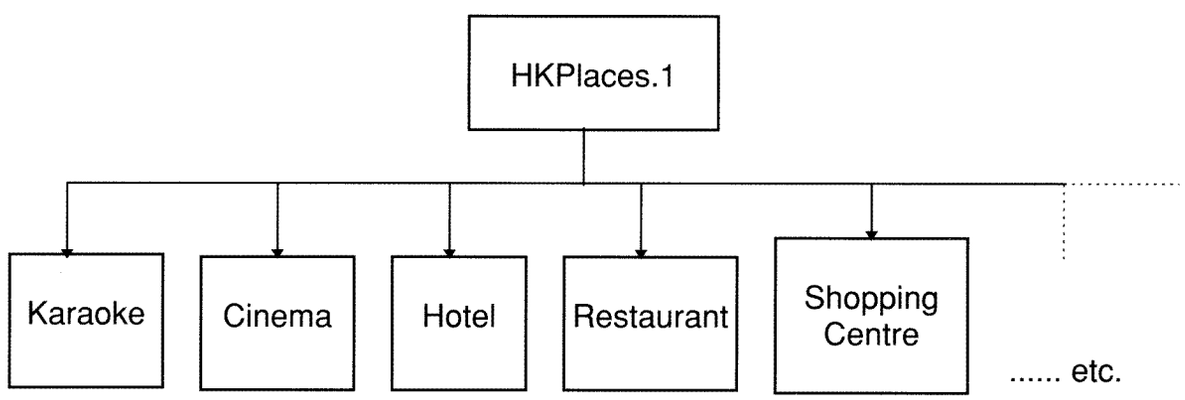
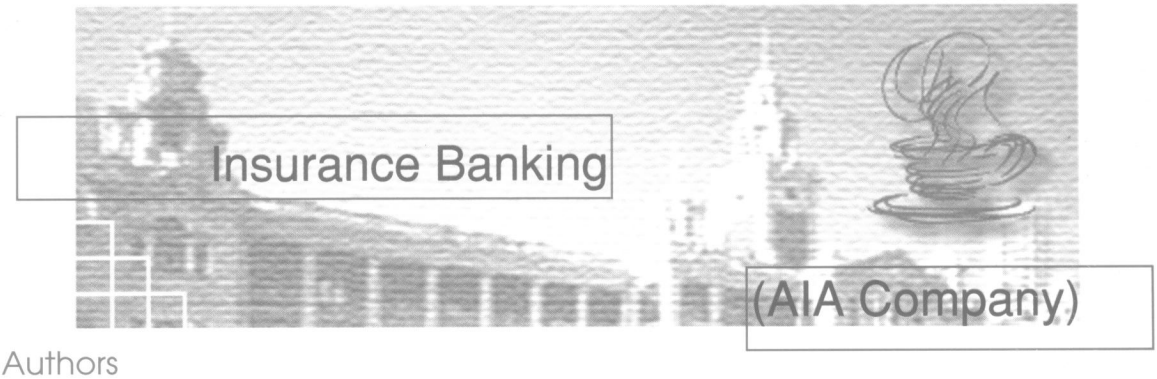


Figure 5 The Structure of the Database Design

Conclusion

In this project, we succeeded in designing a homepage with the implementation of the map searching and the database searching over the internet. Actually, our homepage has several advantages. Firstly, it provides fast searching. Second, it is user-friendly and everyone can know how to use it no matter he/she can operate with a computer or not. Moreover, the information is the latest, updated and is accurate as the database can be updated easily. Lastly, it has a nice and attractive appearance which can attract users. We hope that this homepage with the searching function can attract people who intend to come to Hong Kong. We believe that they will appreciate our user-friendly homepage as they can get the latest, updated and accurate information.



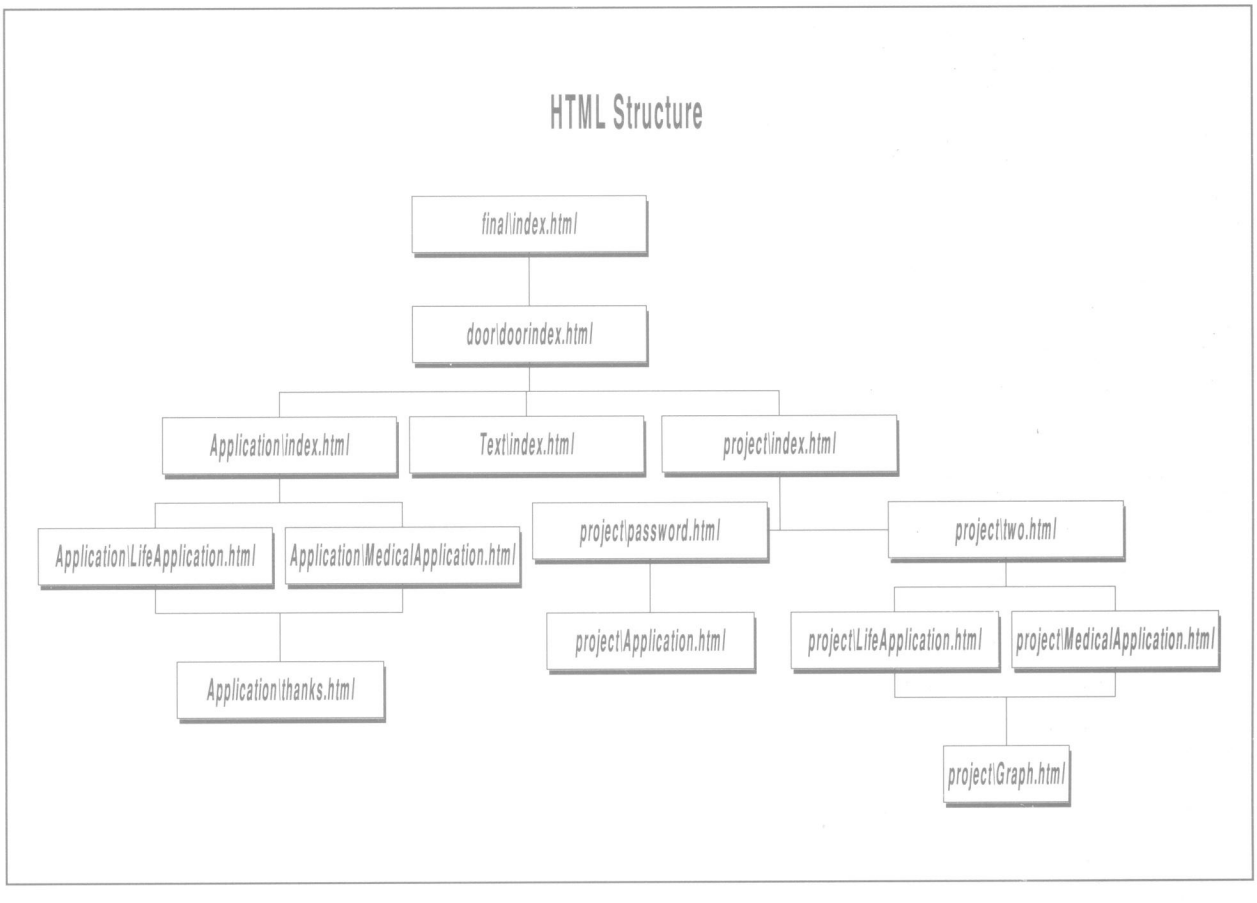
Authors

Lau Wing Lok, Billy
Li Ka Nam, Paul
Kuan Chi Chung, Raymond
Yu Yun Wai, Stanley

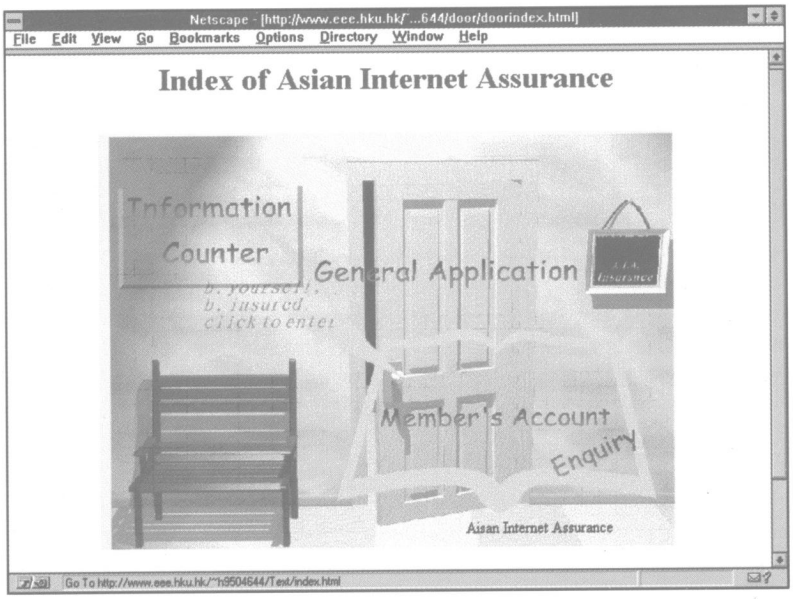
Introduction

Our scope on this project is to enhance the insurance activities over the internet. Since our group think that the internet based insurance activities are not enough at present, we would like to base our project on application as well as member's enquiry.

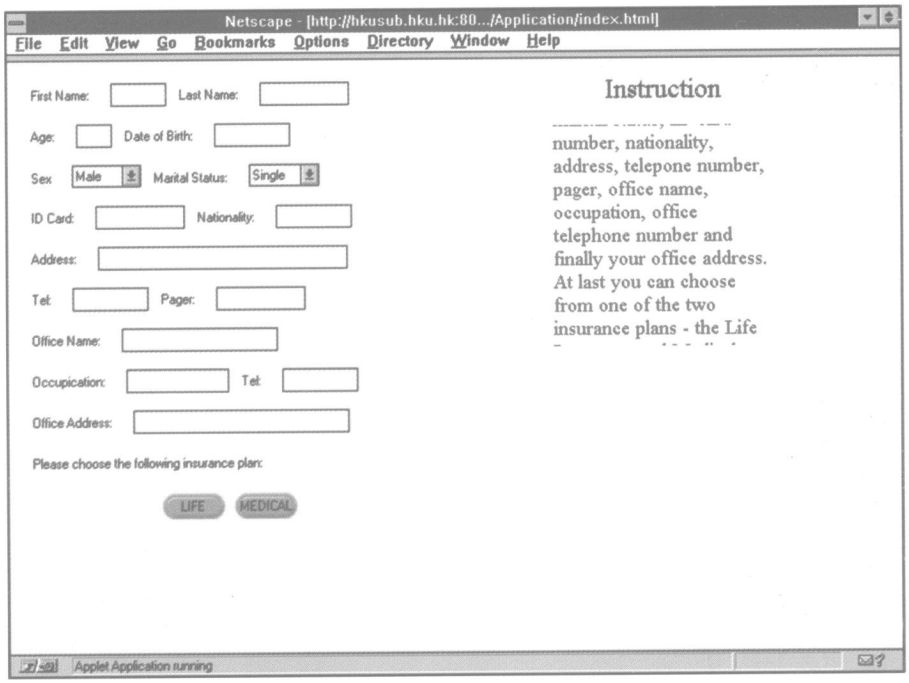
1. Program Layout



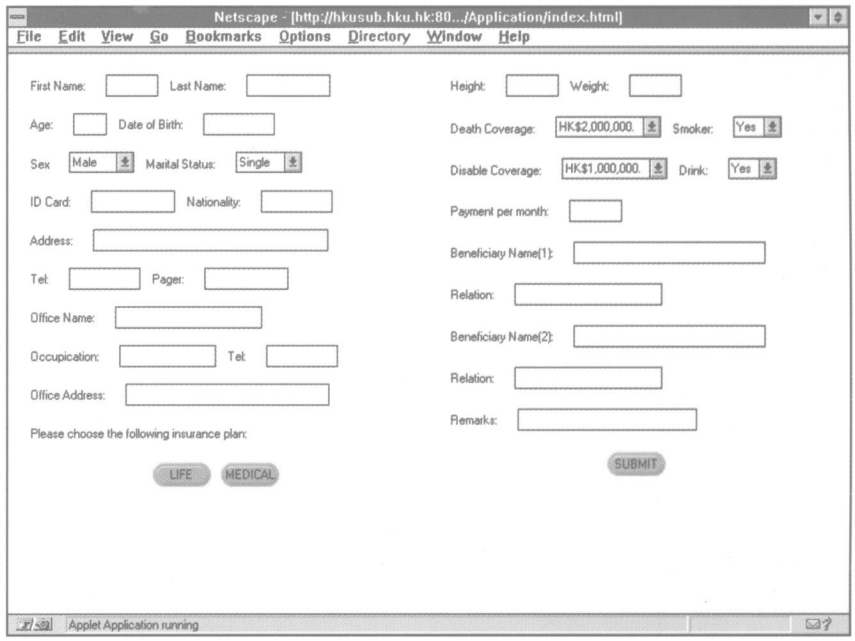
2. Chief Functions of the Main Title Page



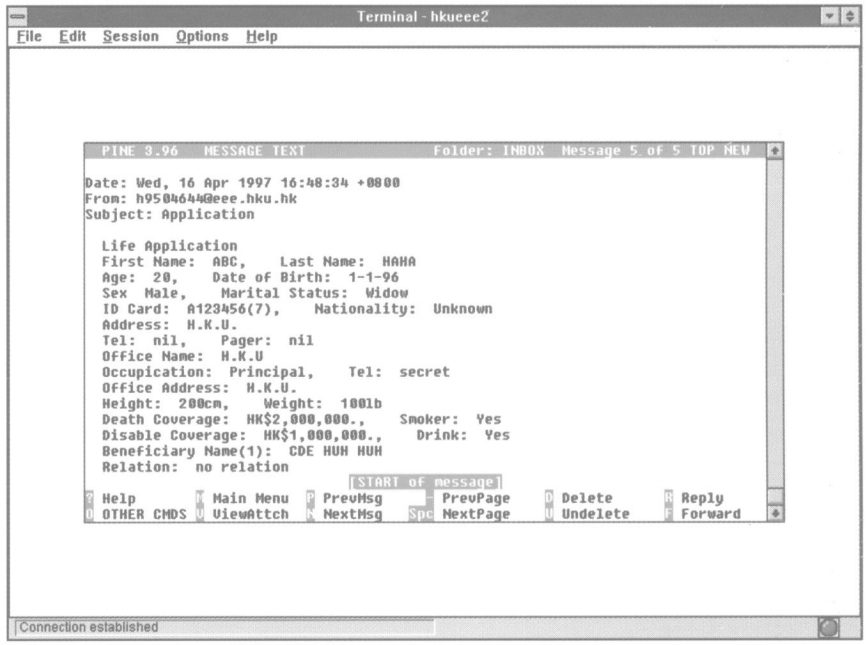
Above shows the overview of our company's main page. It contains three main parts: *Information Counter*, *General Application* and *Member's Account Enquiry*. A click on any of the above field will lead to the following page:



Above shows the Application Form of our company's insurance Scheme. They are Life Assurance and Medical Insurance. After you have entered the required information, you may just simply click the scheme button at the bottom left hand corner. The second part of the form will appear as shown ..... (Besides, the above Application Form provides you an instruction section as well which guides you to complete the form.)

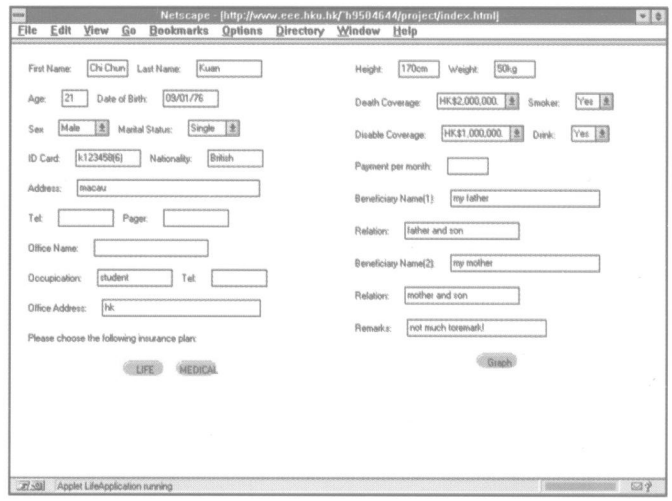


After pressing the Scheme button (Say LIFE). The Life Application form will appear on the right hand side, again, you just need to fill in the required information and press "SUBMIT". Then your application will be accepted and processed as an e-mail.

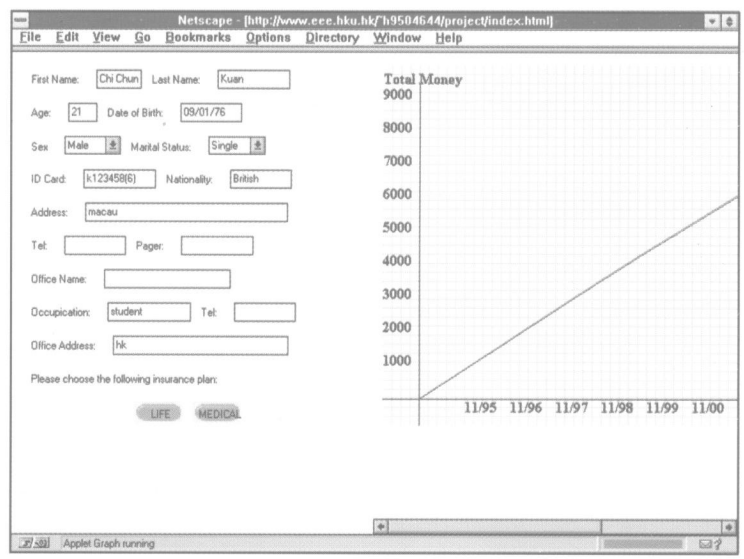


The second part of our program is the Member's Account Enquiry. The first thing you need to do is to enter your ID number and your password.

After both pieces of input data are checked correctly, your information will be displayed .....

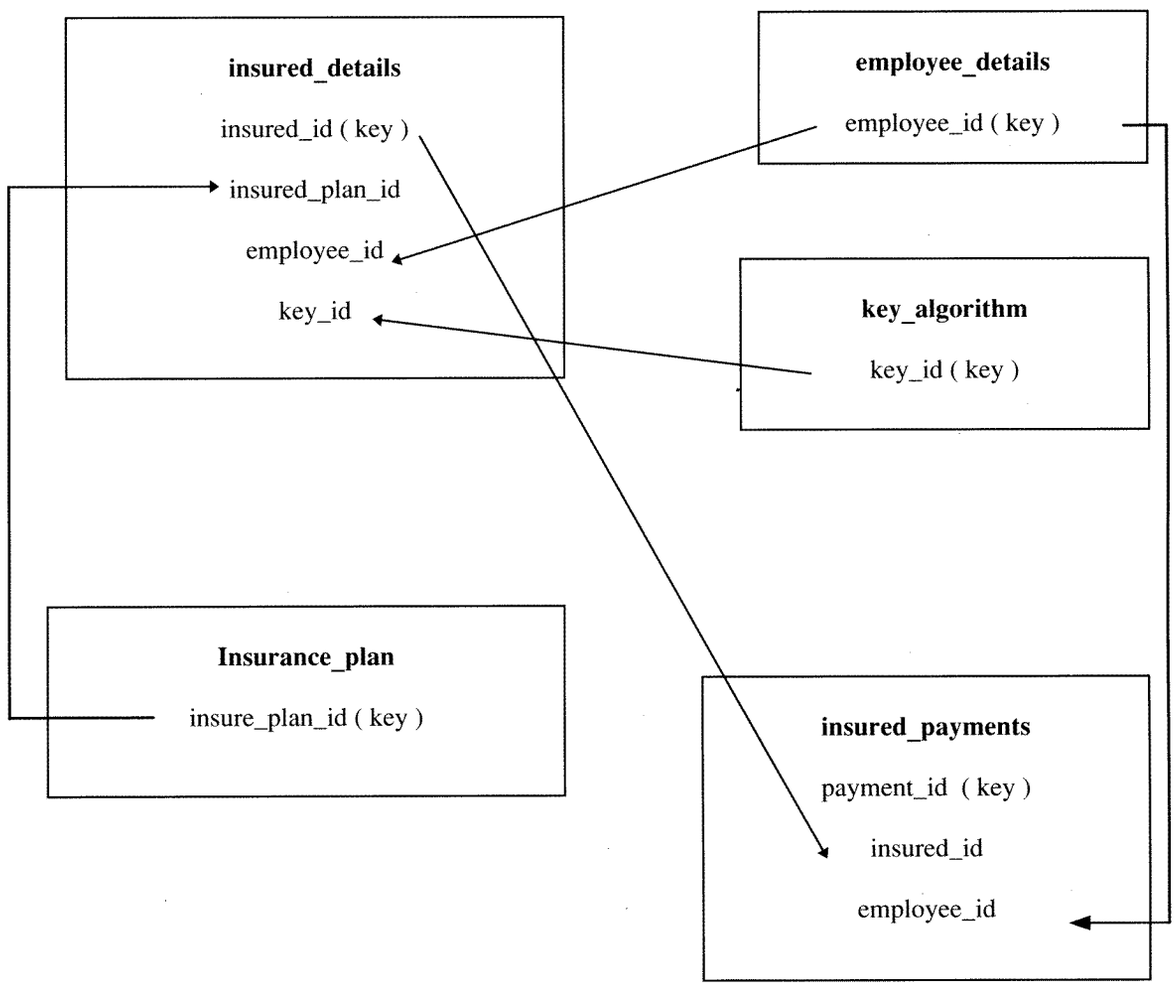


From here, you can see your own information, the most important information is about the payment of your plan. You can see on the bottom right hand corner, there is a button “GRAPH”. After this button is clicked, the payment information (in the format of a graph) will be shown .....



3. Overall Database Structure

As a compact and efficient database design is crucial to the operation of the whole system, much effort has been made on the setting up of various fields and tables for this insurance project. This includes the sorting of fields in various tables so that data will not have an unessential duplication. As the msql is of the relational database type, extensive use of the relation between tables is employed. However, as the data in this database is rather data intensive with only a few computational dynamic values, a sluggish and complicated relation between tables will be a far cry for the server end and the DataBase Jet Engine will be bound to be turtle like. Hence those large fields are indexed in order to put some work load on the client side to relieve the server CPU utilization and only the essential ones are set up between tables. The hierarchy between tables is shown below for reference. For those exact fields in various tables, they could be looked up directly in the listings of the Insurance Project.msql file.



Conclusion

As it can be seen, performing insurance transaction on the internet is easy, convenient and time- saving. However, the work behind is very much. Since, the on-line application and information enquiry are not well-developed. We need to put a lot of effort to create our own program. Since the resources back-up is not enough, it is time consuming to collect information.

The difficulties encountered are as follows:

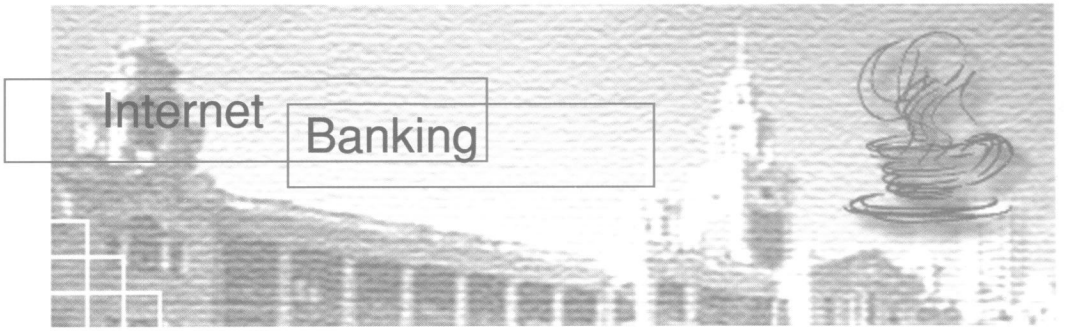
Difficulty in collecting information

Since the information relating to our project is not enough over the internet. That's why we sometimes need to create our own idea to continue our job. For example, we should not put too much information about our client on the internet (through the browser), and we need to select some essential but not critical ones for display.

The failure of the msql server

Since the msql server we are using is not very stable. Sometimes, we couldn't read information from it. As a result, this may interrupt our progress.

After working on this JAVA project, some other JAVA knowledge, for example, Java Script, can be learnt through browsing others' homepages which is helpful in our project. Last but not least, we need to express our gratitude to Mr Chris Yeung, who has put a lot of effort in helping us to finish our project. We finally look forward to having a chance to work together again next year.



Authors

Wung Ho Yin
Yeung Chik Ki
Yeung Man Chan

Introduction

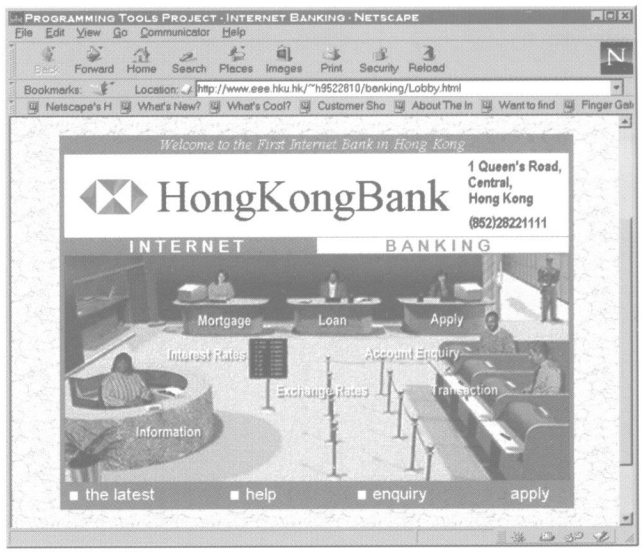
In this ever-changing world, managing banking affairs can sometimes be very time-consuming. To keep in pace with the contemporary world and to facilitate people in managing their banking affairs, we would like to introduce a remarkable idea in banking services - Internet Banking. With these services, people can handle their banking affairs at any time and space they like through the Internet. These include account application, enquiries on interest rates, exchange rates and other aspects such as account transactions, together with the settlement of bills.

In our research project, we have tried to implement a variety of banking services by using a powerful and an object-oriented programming language for the Internet - Java™. The applets written in Java can be embedded in the web pages through which they will be activated whenever the web pages are accessed by a Java-enabled browser, and they are independent of the Operating System platform on which the web browser is running.

1. A Preview of the Future Banking Services

The followings illustrate the possible ways of future banking services by making use of the Internet:

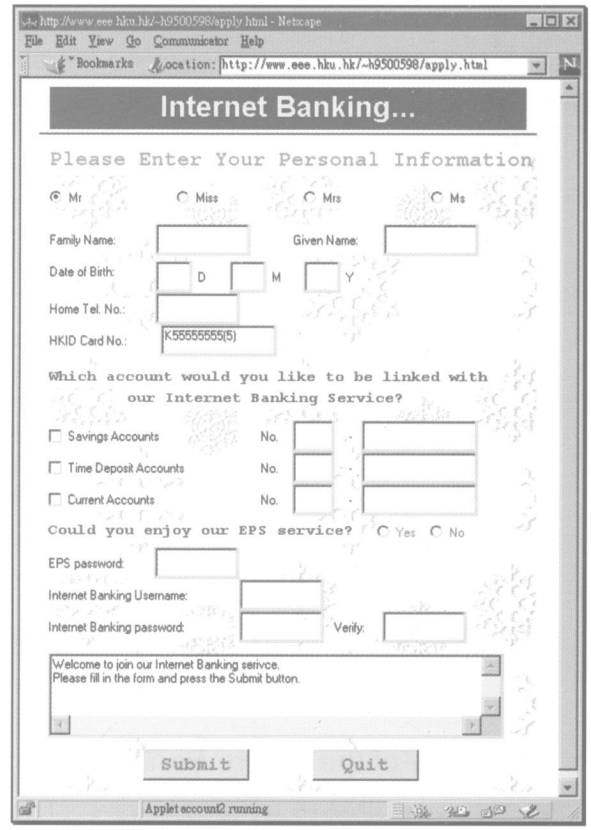
1. The Main Lobby



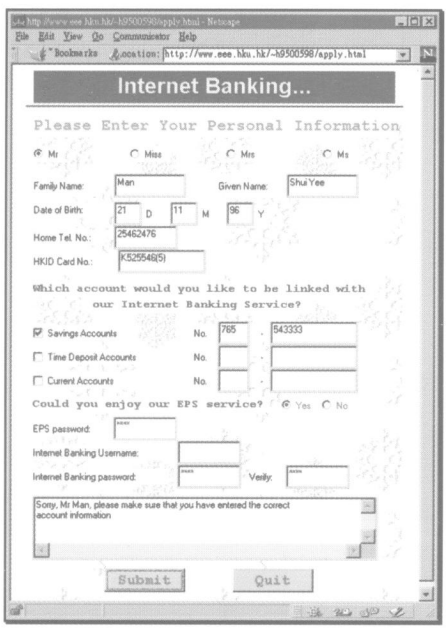
The above diagram depicts a page loaded using a web browser. It resembles the lobby of a bank in which the user can choose a number of services they need by placing the mouse pointer over the icons in the diagram. Potential services will be highlighted and they can be activated simply by a mouse click. Besides, the user can go back to this page at any time and choose another service they want.

2. Application Form for the Internet Banking Services

The user has to apply for the Internet Banking services before he or she can use them and the application can be done over the Internet. The figure on the right shows an initial layout of the application form.



Since this application form is written in Java, it is interactive and the user will be guided to fill the form according to the instructions displayed in the message box. Once the "Submit" button is pressed, the information entered will be checked with those in our database in order to verify the identity of the user. If any incorrect information are found, the user will be asked to correct it. For example, if the user has entered a wrong savings account number, a message will appear in the message box as shown below:



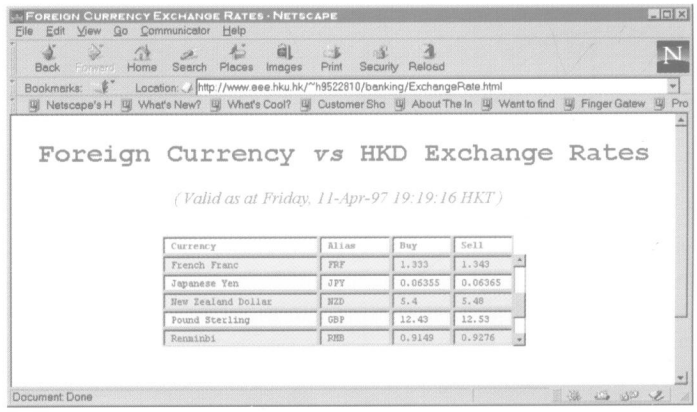
The EPS password is used as a means to verify the identity of the user. If the EPS password and all the other information entered are correct, the user will be able to use the Internet Banking services instantly and a message notifying successful application will be prompted. On the other hand, if the user does not have an EPS password, his/her application will be processed later and he will be informed by mail about his/her application. Again a message will be given.

The above checking process of applying for the Internet Banking services requires the user to wait for just a few seconds.

3. Enquiry on the Interest Rates

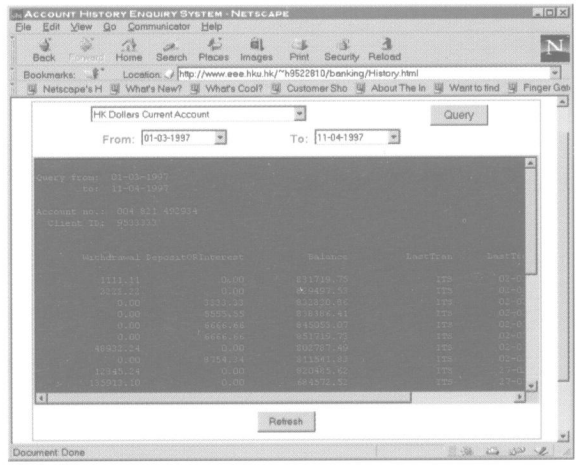
The user can obtain the interest rates for different types of Hong Kong dollar accounts as well as the foreign currency accounts. The data for the interest rates are retrieved from the database that resides in hkueee5.eee.hku.hk whenever the web page is loaded.

4. Enquiry on the Exchange Rates



Likewise, a communication link will be established with the database server once this service is selected. Data for exchange rates will be retrieved from the database and displayed on the screen. The user can visualize the names of the foreign currencies and their corresponding aliases, the bank buy rates as well as sell rates so that he/she can make transactions based on these information.

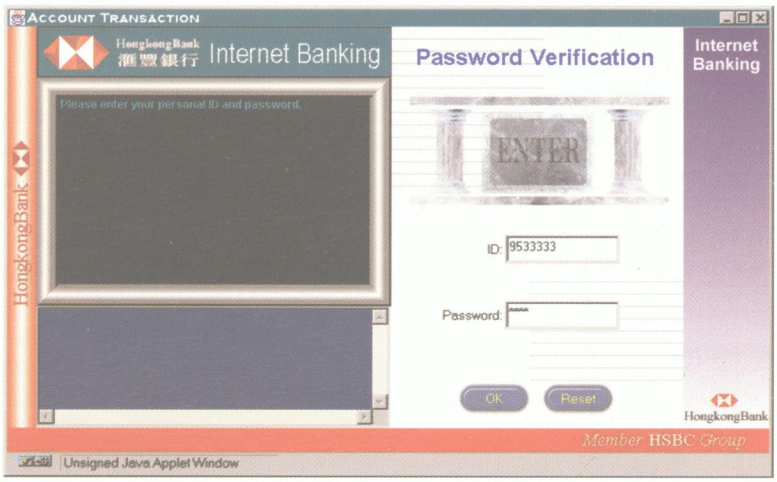
5. Account Enquiry - the Account History Enquiry System



The user is also able to check the history of transactions in his/her account by selecting the service concerned and entering a correct ID as well as password. After verification, he/she will be given a chance to choose the type of the target account and the period to check. Five types of accounts and a maximum of two years' records can be checked using this enquiry system. After querying into the database, a summary of the transactions in the selected account will then be displayed in a format similar to that in an actual passbook. Error message will also be displayed if anything is wrong.

6. Account Transactions - the Internet Transaction System (ITS)

Password Verification

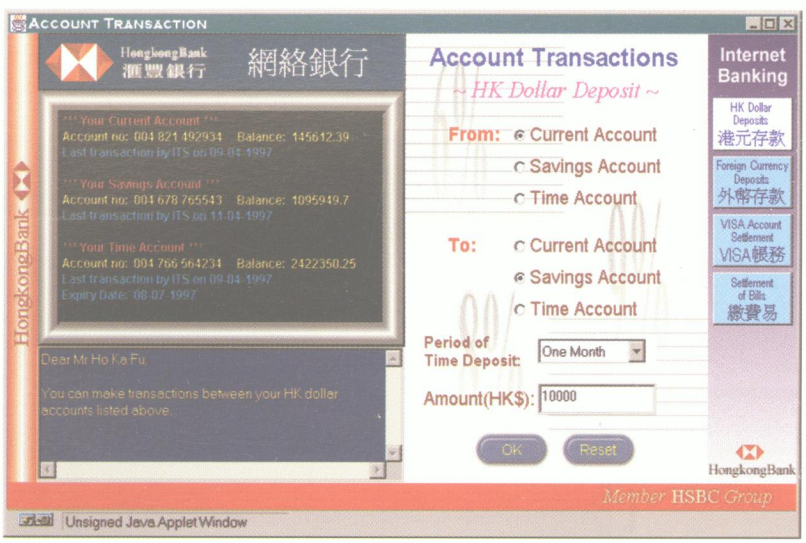


Before performing any transactions, the user is required to enter his/her ID and password correctly in order to login the transaction system. Instructions will be given in the message box if the login is successful. Otherwise, an error message will be shown. The system will reject the user to login (by disabling all the buttons) if he/she has entered an invalid ID or password for three times or more.

Accuracy Enhancement - the "Maths" Class

In this transaction system, a specially designed calculation program has been used. This program is capable of performing arithmetic calculations (including integers and decimal numbers of any range) without any truncation or rounding off errors. Using this program, it enables calculations to be exact. Since accuracy is especially important for banking services, all the calculations in this transaction system are handled using this program.

Account Transactions - Hong Kong Dollar Deposit

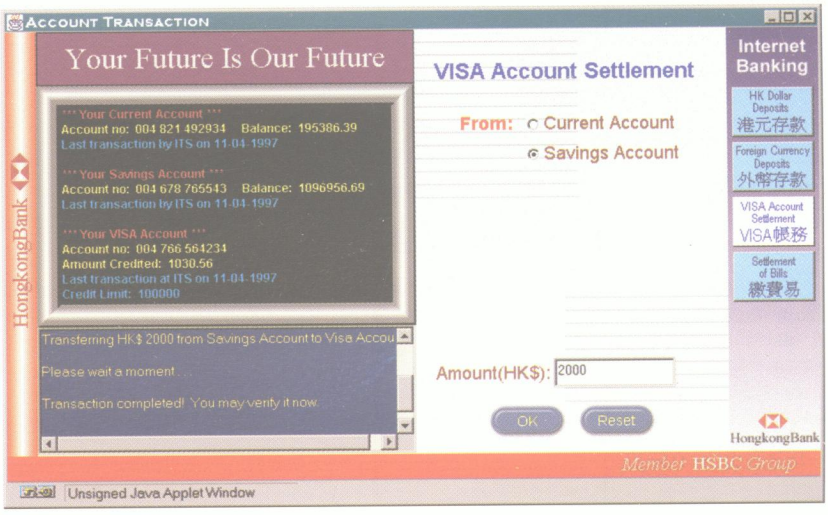


The user can transfer money between his/her Hong Kong dollar current account, savings account and time account. The message box will indicate the action requested by the user as well as the status of the transaction. The information of the user's accounts, both in the database and on the screen, will be updated whenever any transactions are made so that the user can check the balance of his/her accounts at any time. In case of the transaction that involves the time account, the interest rate will also be shown in the message box according to the period selected and money cannot be withdrawn from a time account if the expiry date has not been reached.

Account Transactions - Foreign Currency Deposit

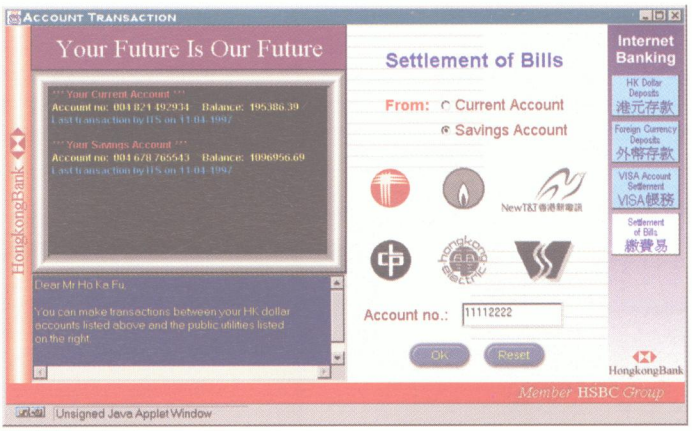
The user can transfer money between one of his/her Hong Kong dollar current account and savings account, and one of his/her foreign currency savings account and time account. The amount of transaction is calculated according to the currency selected and the corresponding exchange rate. The message box will indicate the action requested by the user, the exchange rate as well as the status of the transaction. The information of the user's accounts, both in the database and on the screen, will be updated whenever any transactions are made so that the user can check the balance of his/her accounts at any time. In case of the transaction that involves the time account, the interest rate will also be shown in the message box according to the period selected and money cannot be withdrawn from a time account if the expiry date has not been reached.

VISA Account Settlement



The user can debit the VISA account directly from his/her Hong Kong dollar current account or savings account. The message box will indicate the action requested by the user as well as the status of the transaction. The information of the user's accounts, both in the database and on the screen, will be updated whenever any transactions are made so that the user can check the balance of his/her accounts at any time.

Settlement of Bills

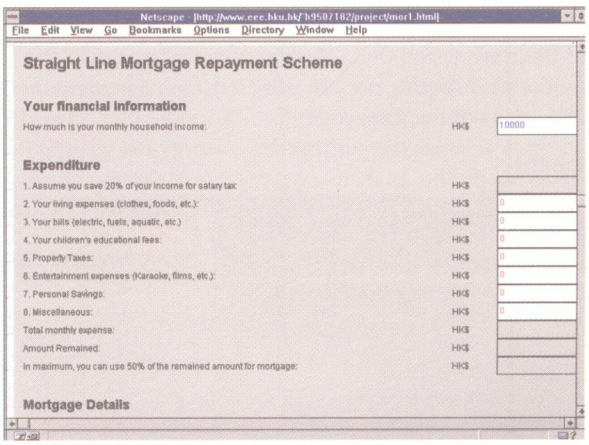


The user can also settle the bills issued from the six public utilities (whose logos are shown on the above diagram) by directly debiting money from his/her Hong Kong dollar current account or savings account. However, the user has to enter his/her account number in the selected company so that the user's information (including the amount of money that is required to debit) can be retrieved from the database. The message box will indicate the action requested by the user as well as the status of the transaction. The information of the user's accounts, both in the database and on the screen will be updated whenever any transactions are made so that the user can check the balance of his/her accounts at any time.

7. Mortgage Calculations

The user can perform mortgage calculations in three different ways by selecting the corresponding service:

Straight Line Mortgage Repayment Scheme



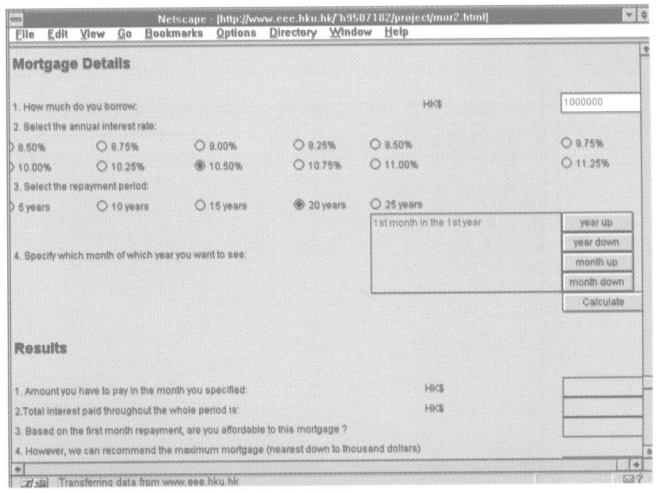
With this scheme, the borrower can pay constantly every month throughout the whole repayment period. The formula for the calculation is as follows:

$$R = \frac{P \cdot \left(1+\frac{r}{12}\right)^{k} \cdot \frac{r}{12}}{\left[\left(1+\frac{r}{12}\right)^{k} - 1\right]}$$

where r is the annual interest rate in decimal (for example, 9% = 0.09),
k is the repayment period in months,
P is the principal (the amount of money borrowed),
R is the amount to be paid back to the bank every month.

For example, if someone borrows HK$1,000,000 for 5 years at an annual interest rate of 8.5 %, then he/she has to pay HK$20516.53 every month.

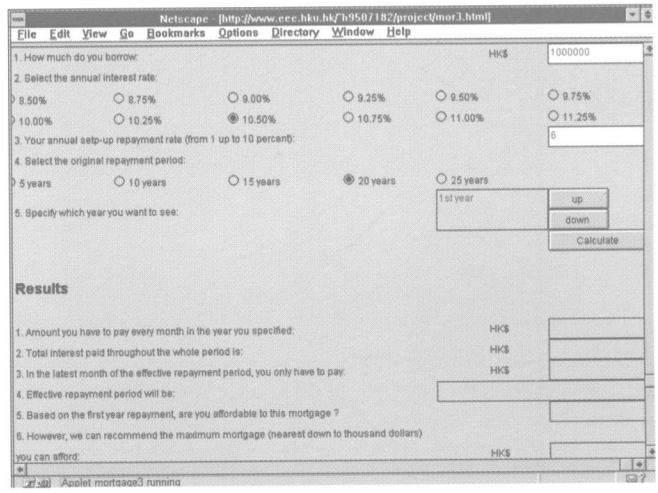
Reducing Balance Mortgage Repayment Scheme



In this scheme, the borrower pays for the principal for a constant amount but not the interest since the principal is decreasing month by month. For example, if someone borrows HK$1,000,000 for 5 years under this scheme, then the repayment period will be 60 months and he/she has to pay 1,000,000 ÷ 60 = HK$16666.67 for the principal in addition to the interest for that specific month.

In other words, the borrower has to pay the most in the first month of the first year for the whole period and he/she has to pay the least in the last month of the last year.

Step-up Mortgage Repayment Scheme



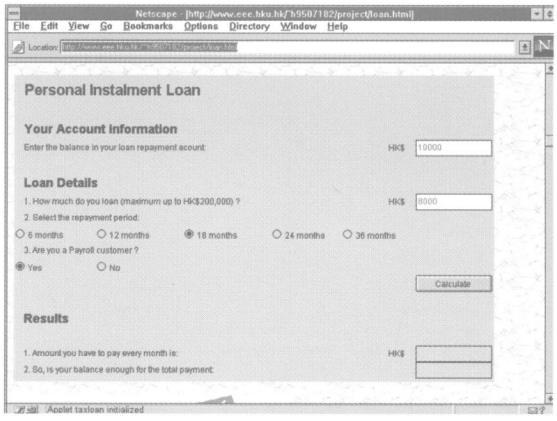
In this scheme, the calculation of repayment is the same as that in Straight Line Scheme in the first year. However, there is a specific rate called the annual step-up rate, meaning the percentage of repayment to be raised every year. For example, if someone has to pay HK$20,000 every month in the first year and he/she choose 6% as the step-up rate, then he/she has to pay HK$21,200 every month in the second year, HK$22,472 every month in the third year and so on. Since the repayment is not regular in this scheme, the actual repayment period will be shorter than the chosen period. Note that the step-up rate should be within the range between 1 and 10% annually and decimal percentage is also accepted.

It should be noted that all mortgage repayment schemes are calculated on the basis of compound interest.

8. Loan Calculations

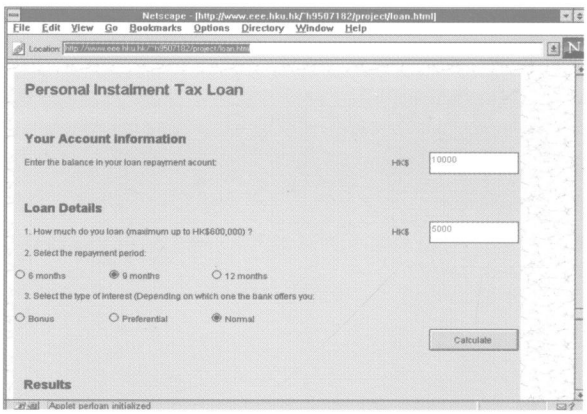
The user can also perform loan calculations by selecting the corresponding service:

Personal Loan



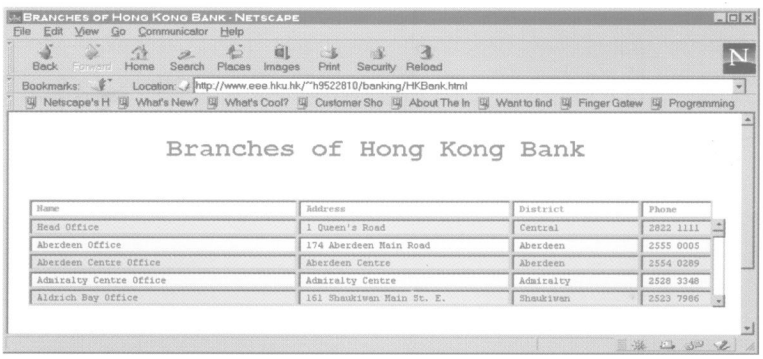
This loan scheme is for private purposes. There are only two interest rates for selection, which depend on whether the borrower has an account in the bank or not. If the user does, the interest rate will be lower.

Personal Instalment Tax Loan



This loan scheme is for paying various kinds of taxes. There are three interest rates for selection, depending on how much the borrower has to pay for tax. If he/she borrows HK$50,000 or below, then the annual interest rate will be 6%. If he/she borrows an amount between HK$50,001 and HK$500,000, then the interest rate will be 9% and for HK$500,001 and thereafter, the interest rate will be 12%.

9. Information of the Bank



The information of bank, namely the addresses and telephone numbers of the various branches, can also be found out online by choosing the corresponding service in the 'lobby'.

2. Database Organization

The project made use of a public domain database engine - mSQL, whose server daemon is run on hkueee5.eee.hku.hk. It accepts relational SQL statements and returns the results of query accordingly. Communication between the client software (that is, the various Java classes mentioned above) and the mSQL server occurs over a TCP connection which is established and managed using the MsqlJava class library. The followings is a list of the tables in the database concerned:

I. Tables holding the records of transactions

Name	Usage
CurrentAcct	Holds the latest transaction records of the clients' Hong Kong dollar current accounts.
SavingAcct	Holds the latest transaction records of the clients' Hong Kong dollar savings accounts.
TimeAcct	Holds the latest transaction records of the clients' Hong Kong dollar time accounts.
ForeignSavingAcct	Holds the latest transaction records of the clients' foreign currency savings accounts.
ForeignTimeAcct	Holds the latest transaction records of the clients' foreign currency time accounts.
VisaAcct	Holds the latest transaction records of the clients' VISA accounts.
HKTran9501, HKTran9502, . . .	Holds all transaction records of the various Hong Kong dollar accounts within a month specified by the table name.
ForeignTran9501, ForeignTran9502, . . .	Holds all transaction records of the various foreign currency accounts within a month specified by the table name.

II. Tables holding important information of the bank

Name	Usage
HKBank	Holds the names, addresses and phone numbers of different branches.
AcctInfo	Holds the interest rates of different types of accounts with different currencies.
ExchangeRate	Holds the exchange rates for different currencies, including the bank buy rates and bank sell rates.
ClientInfo	Holds the personal information of the clients as well as their statuses on using the Internet transaction system (ITS).

III. Tables holding clients' information of the various companies

Name	Usage
HKTelecom	Holds the clients' information of the Hongkong Telecom, including the amounts of the bills to be paid.
Towngas	Holds the clients' information of the Hong Kong and China Gas Co. Ltd., including the amounts of the bills to be paid.
NewTT	Holds the clients' information of the New T&T Co. Ltd., including the amounts of the bills to be paid.
ChinaLightPower	Holds the clients' information of the China Light and Power Co. Ltd., including the amounts of the bills to be paid.
HKElectric	Holds the clients' information of the Hongkong Electric Co. Ltd., including the amounts of the bills to be paid.
WaterDept	Holds the clients' information of the Water Department, including the amounts of the bills to be paid.

Conclusion

In this project, we have demonstrated different aspects of banking services that may be provided over the Internet. Users may look forward to handling their banking affairs conveniently through the Javatized web pages of the banks in the future.

Appendix

- ✯ Project URL: http://www.eee.hku.hk/~62269-96/iBanking/Lobby.html
- ✯ Some test cases for the Internet Transaction System and the Account History Enquiry System:

ID: 9533333 Password: pass

ID: 9288888 Password: pass

Authors

Cheung Lai Ching
Cheng King Wai
Kan Kar Yin
Lam Hey Tak
Tse Ching Hok

Introduction

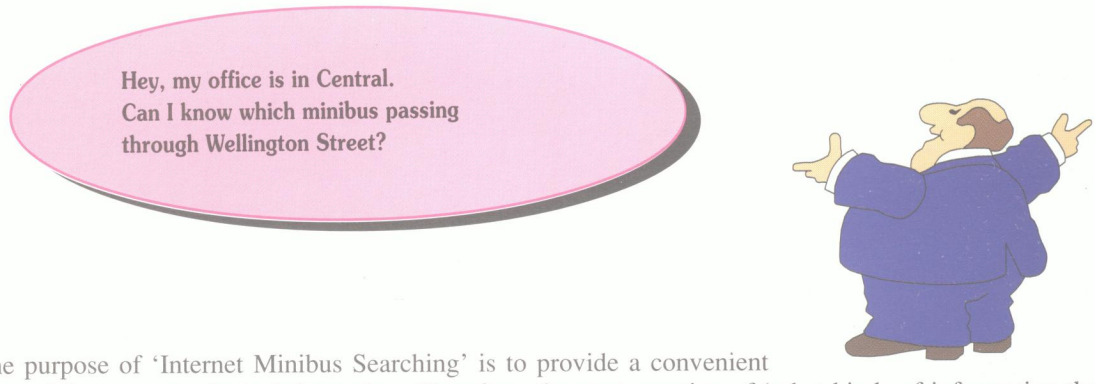
One of the most important aspects in our daily life is travelling. We spend a lot of time on travelling everyday. Public transport is usually the dominant mode of transport for most Hong Kong people. It includes buses, trams, MTR, KCR, ferries, taxis and mini-buses.

Among all kinds of public transport, mini-bus provides a fast, comfortable and economical way for travelling. However, the information about mini-bus is scarce. This is mainly because, unlike buses and MTR, there are no fixed substations for mini-bus. Most people will know how much they should pay and where the mini-bus is going to only when they get into the mini-bus, As a result,a system searching information about mini-bus is needed.

The title of our project is "Internet Mini-bus Searching". The objective of our project is to provide an efficient and convenient way for searching information about mini-bus.

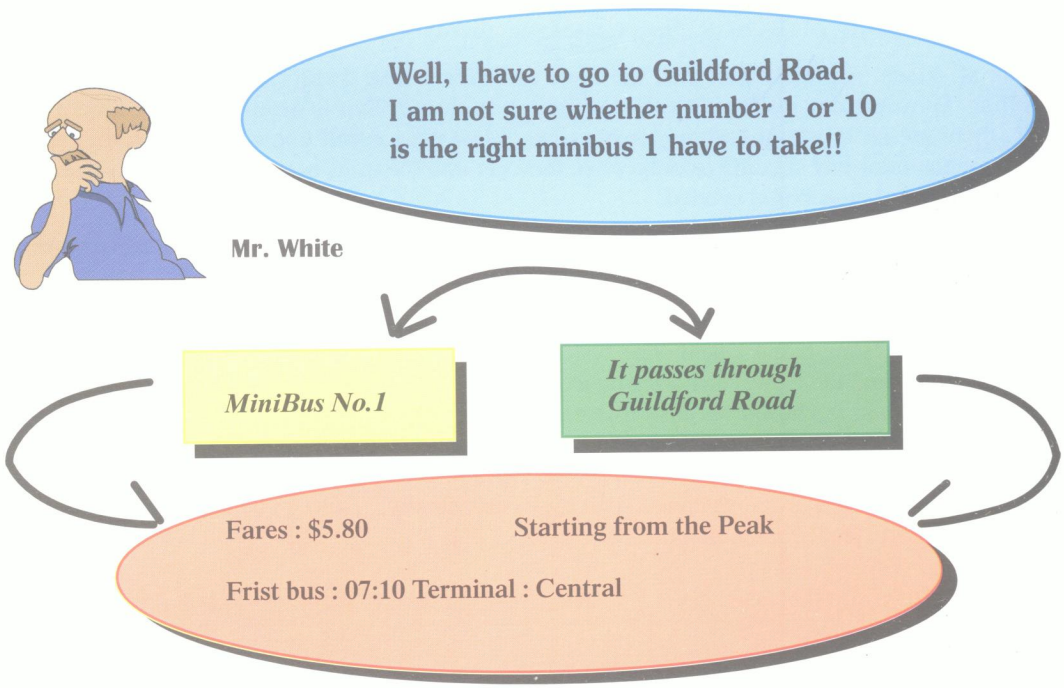
From our programs, information such as routes and fares can be obtained for the mini-bus with specific route number on Hong Kong Island. The system can be extended to illustrate the information for mini-bus in Kowloon and the New Territories by expanding the database. Only mini-bus with specific route number (green bus) is chosen because it is easier to define its starting and destination point. For the mini-bus with circulatory route (red bus), another implementation should be used since there are no clear definition for their routes and fares. Since our aim is to provide a convenient way for searching information, our programs are written so that it is user-friendly. This can be achieved by two searching methods: text input and graphic input.

1. Database Structure



The purpose of 'Internet Minibus Searching' is to provide a convenient channel for users to obtain information. Therefore, the next question of 'what kinds of information the users would like to search for' was raised.

Besides simple searching such as the travelling routes of the minibuses or the fares, users can also search for any relevant information by just knowing the minibus number or the streets that the bus pass through.



In order to provide a convenient searching for the users, a user-friendly program would be needed. Therefore, a large and rich database would be necessary.

The following tables were put in database for providing detailed information for the searching:

Number	*Fares*	*Terminal 1*	*Terminal 2*	*Starting*	*Ending*	*Period*
1	$5.80	Peak	Central Tamar St.	07:10	24:25	5 min
2	$4.30	Central Tamar St.	Mid Levels Old Peak Rd	07:00	19:00	10 min
3	$4.30	Central Tamar St.	Mid Levels Po Shan Rd.	07:00	24:00	10 min
.....			.....			

Table 1. Minibus

Number	*Street*	*Rank*
1	Peak	1
1	Peak Rd.	2
1	Guildford Rd.	3
1	Mansfield Rd.	4
.....	.....	.....

Table 2. Node

Suppose Mr. White has to go to Jaffe Road in Wanchai, but he is not sure which minibus he can take. By 'Internet MiniBus Searching', he can simply enter the street he has to go, that is, Guildford Road, then the program will search for this street in Table 2. When it is found that Guildford Road appears in the column of minibus 'No. 1', it will then switch to Table 1 and look for the detailed information of this bus. Mr. White can now know that bus no. 1 is the right bus to take and the fare is $5.80 starting from the Peak.

Moreover, Mr. White can also simply type in bus 'No.1' or 'No.10', the program will search for the information about these two buses in Table 1 directly and tell him which bus is the right one.

I planned to go to the Hong Kong Zoological & Botanical Garden with my children this Sunday. Do I have to take minibus No.1 ?

Mr. Brown

Mr. Brown would like to go to the Hong Kong Zoological & Botanical Garden. By entering bus No.1, the program will automatically search for the relevant information in Table 3.

Street	*Description*
Peak	Victoria Peak Garden
Guildford Road	Villa Verde
Mansfield Road	Mansfield Bldg.
Garden Road	HK Zoological & Botanical Garden
Edinburgh Place	Jardin House
.....	.....

Table 3. Street

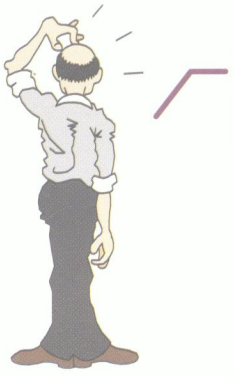
By searching in Table 3, it can be seen that HK Zoological & Botanical garden is at Garden Road. Therefore, by referring back to Table 2, bus No.1 is the right bus and more detailed information can then be obtained in Table 1.

The fares of minibus are different along the route, so the subfare is also provided by the following table:

Number	*Direction*	*Starting-order*	*Ending-order*	*Subfare*
1	F	5	8	$3.00
1	B	5	1	$3.00
1A	F	3	5	$2.20
1A	B	3	1	$2.20

Table 4. Division

For example, the fare of bus No.1A changes from $3.00 to $2.20 at Magazine Gap Road.(The streets are represented by the orders and they can be searched out in Table 2.) The forward direction is indicated by 'F' and the backward direction is indicated by 'B'.



Do I have to type 'Peak' or 'The Peak'?

The program has considered the case that there will be different names for one street. The users may have doubts about words such as 'Road', 'Rd.' or 'The'. No matter what versions of the name of a street the users enter, the program is able to search for the information of the right bus by using the following table:

Street	*Reference name*
Peak	*Peak*
Peak	*The Peak*
Peak	Tai Ping Shan
Victoria Peak Garden	*Victoria Peak Garden*
Victoria Peak Garden	*Peak Garden*
.....	.....

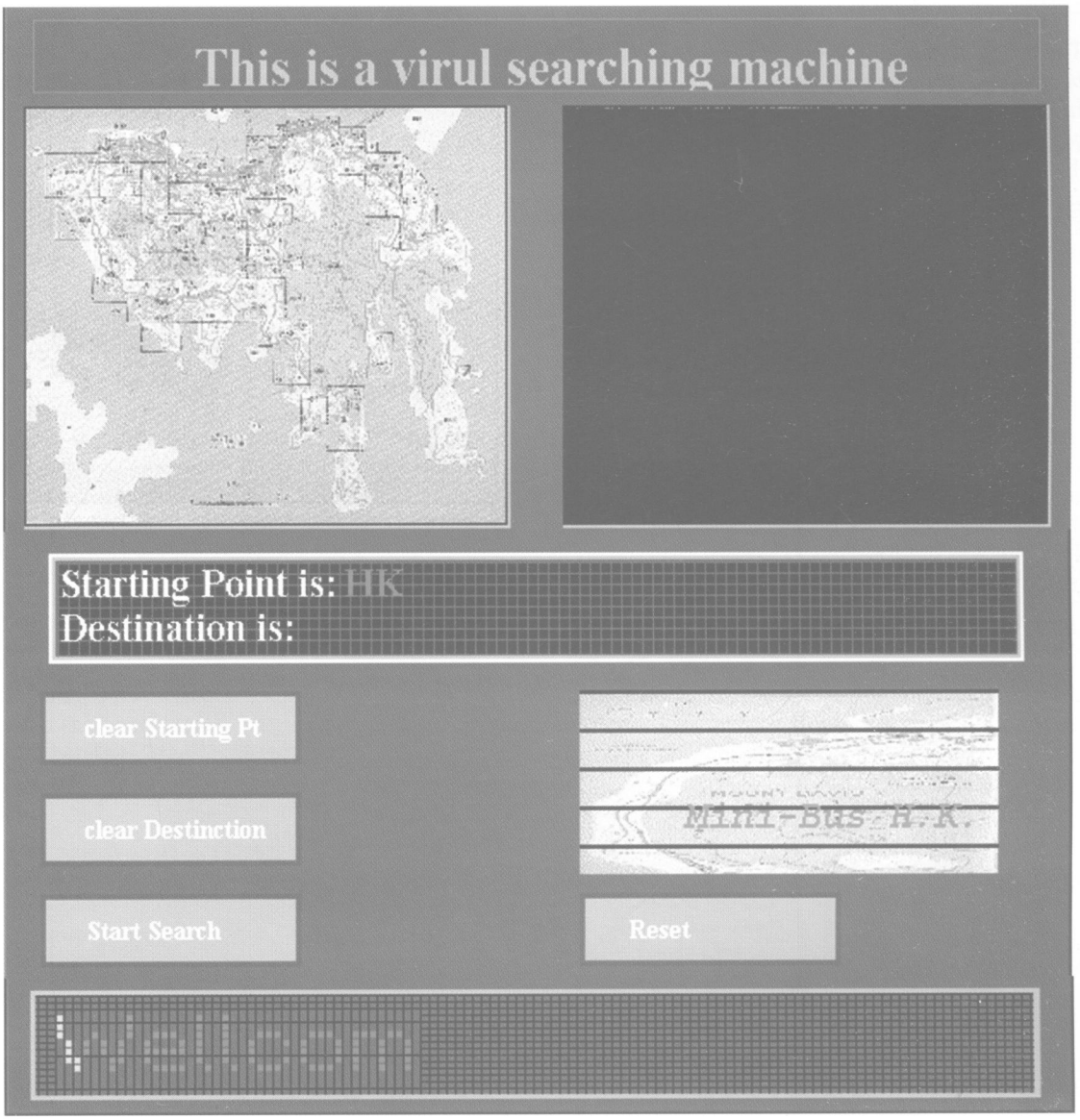
Table 5. Reference

If someone does not know the name 'The Peak' and enter 'Tai Ping Shan', the program will search for the right name in this table. After knowing the user wants to search for 'Peak', the program will switch to other relevant tables immediately.

I don't know the exact name of the street. I just know where it is. It is somewhere near Admiralty!!

If the user does not want to type in any word, he can search for the information by clicking points on the map. There are about 200 maps stored in the database so that users can find the street easily and quickly. The details of Graphical Input' will be discussed in the next section.

2. The interactive searching interface



Interactive interface use in the searching of mini-bus to suit their needs.

The window consists of two main windows. The upper left window is an imagemap and the upper right window (which is now dark in color) is a preview window which previews the area pointed by the mouse inside the imagemap. The top scroll text window displays the instruction of how to use this interface. The buttons, which are named, will perform the function as indicated. The bottom LED window is used to display some general information about the company that provides these services. The general purpose of this interface is for users who are not familiar with computers.

Principle of Design

The program was designed in such a way that for any users who do not know the place he/she wants to go. They can choose the point by simply clicking on the map provided in the interface. The first scene of the window inside the interface will be the map of Hong Kong island. Then the user can decide which district he/she wants to go and click on the corresponding part inside the window (map). After that, another map which is a zoom of that part will be displayed on the window. The users can then follow the same procedure and find his/her final point which may either be the starting point or the end point.

The program also provides preview function. Before the user clicks on the map, a zoom view of that part will be displayed in next window. Hence the user can have a look of where he will go before clicking on the map.

There are four buttons inside the interface, namely, the *start search button*, the *clear starting point button*, the *clear destination button* and the *reset button*. The functions of the buttons can be revealed from their names. There is a button that needed to be mentioned here, the reset button. It was used when users find that the program is not functioning or responding well. Once the reset button is pressed, the program resets all its parameters and enters the state as if it is just loaded.

Extension of function in future

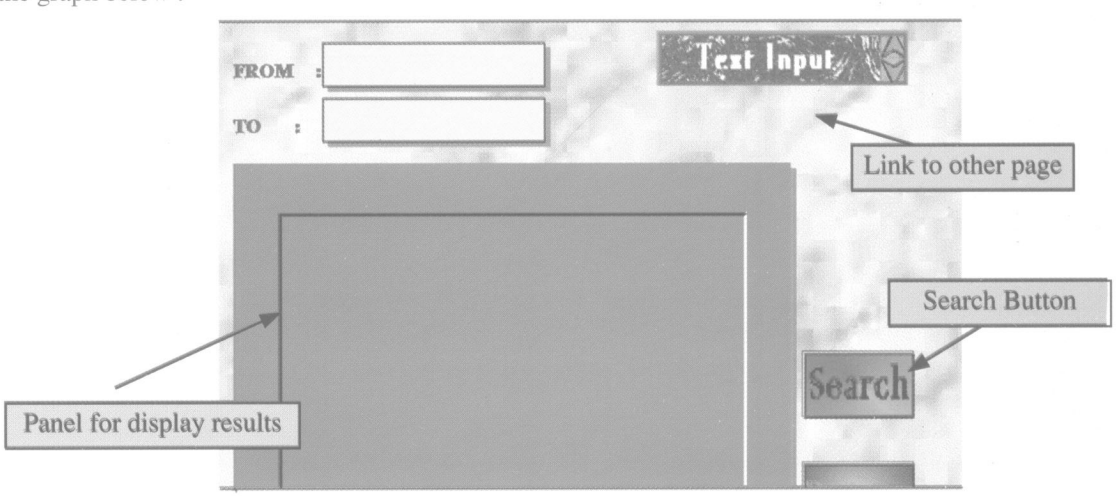
This program can extend its function so that it can cover all the places in Hong Kong. The procedure needed is simply to include all the maps inside Hong Kong and set the parameter corresponding to the maps in the HTML text. The most difficult task may be to scan maps and set parameters correctly.

Bug reported

The program was found to have a serious bug. During testing that the program cannot function properly after the clear starting point button or the clear destination button were pressed more than three times. The previous imagemap display on the window will continue its function though another map is loaded and covered on it. Thus, the points we located on the imagemap will catch the wrong parameter(which is the parameter of the previous map). And hence the program fails to function what it is designed for. This bug was not resolved and it will be resolved in the next version of this program.

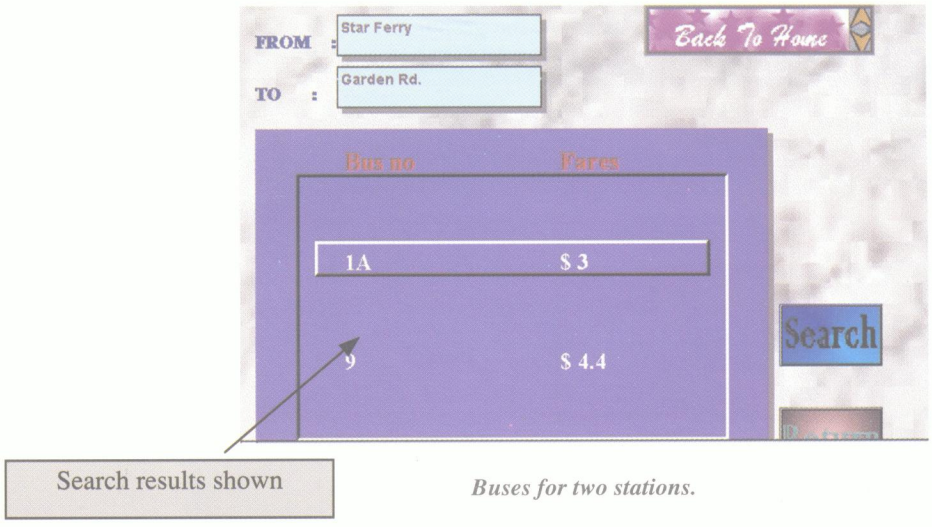
3. Text Input

Besides graphical input, the user can also choose the text input. For some users, they know the name of the place they are in and where they want to go, so if they use graphical input, they may think it is quite time consuming. Text input provides an alternative for them. The scene of the interface is as shown in the graph below :

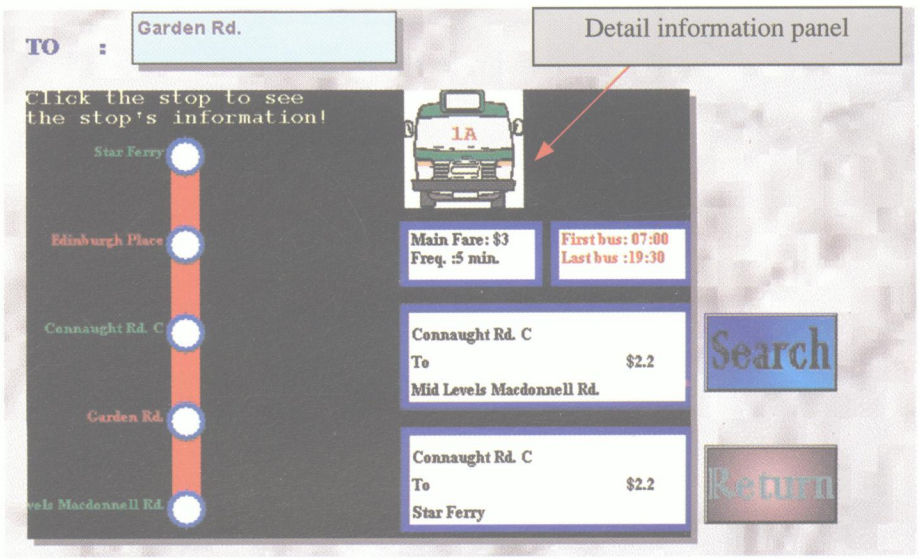


Layout of the text input interface.

In order to provide the easiest way to use text input format, the program supports both upper and lower cases. Besides, the user does not need to type in the exact name of the street. For example, "Garden Road" and "Garden Rd." are both acceptable. When the user has typed by both stations, he can get all the buses he can take and the corresponding fares by pressing the "Search" button. As passengers may be interested in more detailed information of a particular bus in the searching list, so there will be a frame showing the details as if the user presses the bus number once.

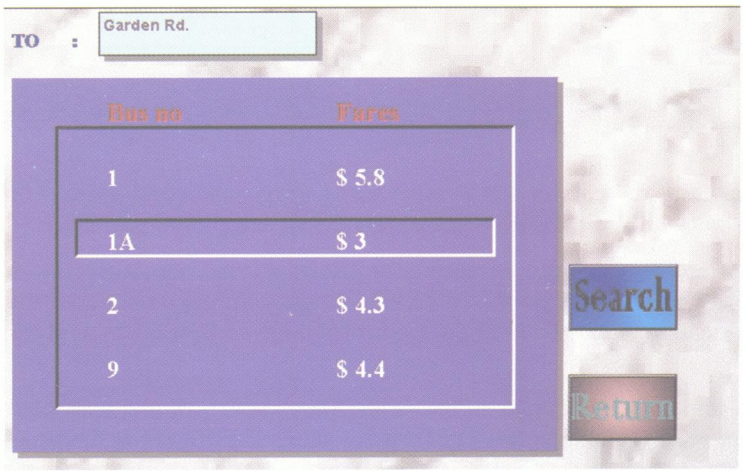


Buses for two stations.



Another panel when select a particular bus number.

In some cases, the user may only be interested in what buses they can take in one place, and how many buses are there in one station, this is also considered in the program. Users can type either one of the textfields and leave the other one blank, the search result will be the buses in a particular place.



Searching buses of a station.

The text input method is in fact just an assistant to the graphical input method, so links to the graphical input are also included. Besides, user will also be notified when there is no bus matching or when wrong street name is typed.

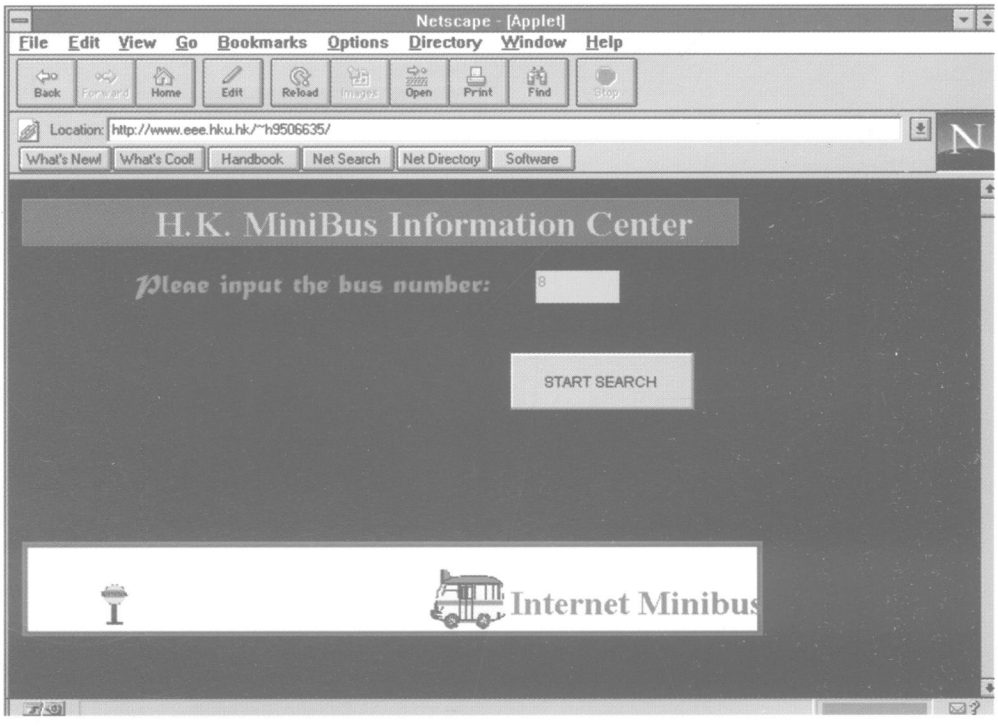
4. Bus Number Information Searching

Besides the route search, there is a case that users may only want to know the information of a certain bus. This program also provides this Bus Number Information Searching Engine. This searching engine is very powerful as it provides all the fundamental information that users need. Information such as main fare, sub-fare, the location that the sub-fare started, frequency of the bus, the time slot of the bus, etc. The most important thing is we also provide the bus stops that the bus passes and each bus stop's simple description. These simple descriptions are usually some famous buildings or parks and they can help the users to identify the location of the bus stop more easily.

The interface is very simple. On the top, there is a scroll text showing the title while at the bottom, there is a simple input area which allows users to input the bus number. After the user has typed in the bus number, he can press the large search button to start searching. Since the display of the bus information will not be shown unless the bus number is typed in, we have provided an animated picture to user to tell the user that this is a minibus searching engine.

If the user inputs a wrong bus number or press the search button before inputting the bus number, a warning message will be displayed and users can type in the information again.

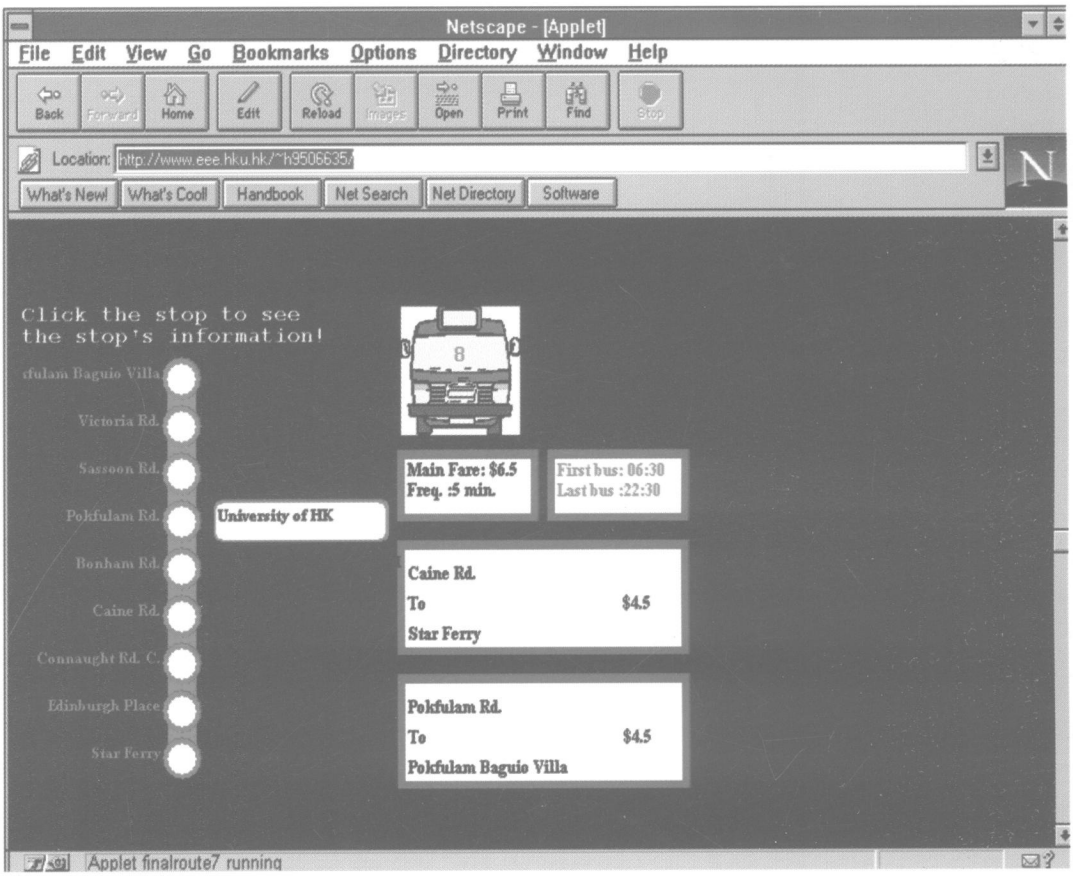
Below is the input field area:



After inputting the correct bus number and pressing the 'start search' button, the information of the selected bus will be displayed which covers the animation shown before.

The information displayed is very user friendly. All the basic information of the bus which is mentioned above will be displayed on the left so that users do not need to scroll to see the information. The information is displayed in the form of a table and users can find the information they need very easily.

The bus stop passed is shown on the right. They are shown in a way similar to that of MTR and KMB and users must be familiar with this. If the users want to know the simple description of a particular stop, users can simply click the stop once(the circle will change color once the user clicks it) and the description will be shown next to the stop.

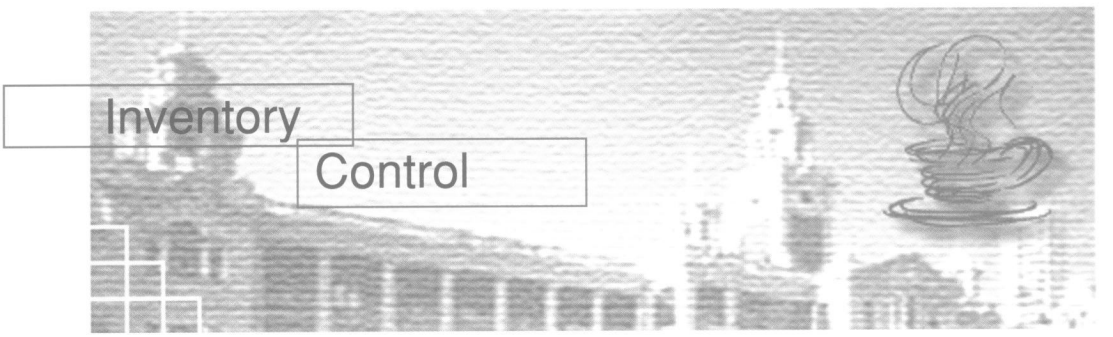


Above is the area of the display of the information of the selected bus number:

5. Further Development

To provide a better service, several improvements can be made. From the view of interface, audio part can be added to the program to facilitate the users. More animation can be included to make it more lively.

For the functional part, we can include more routes such as minibus on the Kowloon peninsula. This can easily be done by expanding the database. At this moment, the destination cannot be reached by one single minibus. This search engine should be improved in order to show multiple minibuses. To achieve this improvement, a small part of the program will only have to be changed to implement the "Select" method.



Authors

Choi Yui Fung, Jeremy
Chow Mei Sim, Simmy
Lee Chun Kwong, Matthew
Chan Sun Yin, Lincoln

Introduction

Inventory control is vital to a supermarket as there are so many different kinds of items sold. A good inventory control could help optimizing profit. Our program is designed to assist the supermarket to conduct inventory control in a much easier way.

This software integrates the organization of the products' database and the statistics of the products. The manager can easily modify the database by using the user-interface. This software has the following main functions:

1. Editing of items' information, and ordering of the items by only clicking the menu bar and filling the information in the table. It is not necessary to access the msql directly.
2. By using the function of our software, the over-stock and under-stock of goods can easily be found, by simply clicking the shortcut key bar.
3. Helping the manger to decide if he should change the ordering amount, this software can plot the past and the predicted selling figures, so as to provide a clear visual products' information.

Specifications

The software is built based on Java and Msql languages. Java is used for building up the interface and Msql is used for storing data. The user requires no Java and Msql knowledge to access the database.

Java Interface

1. Build up user interface.
2. Create user friendly environment.
3. Show pictures of various products.
4. Communicate with the database to convey data.
5. Easy pull-down menu for various selections.

Msql Database

1. Store all data of different products.
2. Contain a wide range of product information.
   Products information includes
   1. Barcodes
   2. Selling figures
   3. Products names
   4. Products companies
   5. Ordering figures
   6. Stocks
   7. Misc.

Users are allowed to create new information subjects.

The software is designed to help controlling inventory of a store, for example, a supermarket. It keeps records and provides a managing environment of each product such that the user can keep track on the stock index of the products. Whenever the stock of a product is over storing or in a shortage, the user will be alerted to put up his own strategy to keep up with an optimum profit.

Since the software is used in commercial aspect, it gives a secure privacy to each user. Whenever the user calls on to the system, it will prompt for his/her identity and password. Only those have been verified can access the program and hence the database. After logon to the program, a window with a pull-down menu will be shown. The menu contains four main items : *Files, Store, Utility, System.* A product diagram image map, which shows the product classification is also revealed. By clicking on an area, a list of various products will appear with names and barcodes respectively.

By Highlighting the item, a picture of the product will be shown. After selecting a product, the user is able to use the functions provided by the software.

1. File

There are five subitems:

a. Edit

A window containing items for different information of the product is shown. The user can modify the values like selling figures, prices, simply by typing in new numbers. The software will automatically renew the data in the database.

b. Insert

A window like the 'Edit' window is shown. It allows the user to create and record new items. All fields in the window will be left blank and to be filled by users. The software will again store the data in the database.

c. Delete

Delete will remove the item and all of its details from the database. No recovery can be made once deleted.

d. Open

A previous stored file which enhances access time of a product without accessing the database will be shown.

e. Save As

Information that are frequently used are advised to save as a file which can be retrieved without fetching from the database. The file can be called upon again using the Open item. A blank field is reserved for the user when a new file is created.

2. Store

There are two subitems:

a. Alarmed item

The manager should define a upper limit and a lower limit for the products, which should be based on the past statistics of the selling figures. Once this piece of data is entered to the database(by using File-Edit), the details of the alarmed item will be shown when it is clicked. For example , if there are two hundred cans of Coke but the lower limit is three hundred, the barcode, and the supplier of Coke will be shown together with a warning "Out of Stock with a difference = 100". This function has a shortcut icon under the menu bar since it is used very frequently.

b. Index

Return to the index page containing Names and Barcodes of various products.

3. **Utility**
   There are four subitems:
   a. Selling Figures
      A plot of selling figures versus months over a period (default setting is twelve months) of the selected product is shown. It also allowed products comparison by highlighting two products before choosing this subitem. The comparison is best used when two similar products are being checked for their selling figures. The result is mostly suitable for ordering process.

   b. Estimation
      With this function, it is possible to analyze the past selling figures of each product and to predict a possible selling trend for the next few months (default setting is 3 months). A comparing estimation plot is also available for any two products.

   c. Email
      This email function has the usual usage as other email programs. On selecting a barcode, a click on this menu will show an email window with the email address provided(search from the msql).

   d. Ordering
      User should select a barcode and then click ordering. Then the products' details including suppliers will be shown. The user can type in the amount he would like to order. This provides a faster way for ordering through the email.

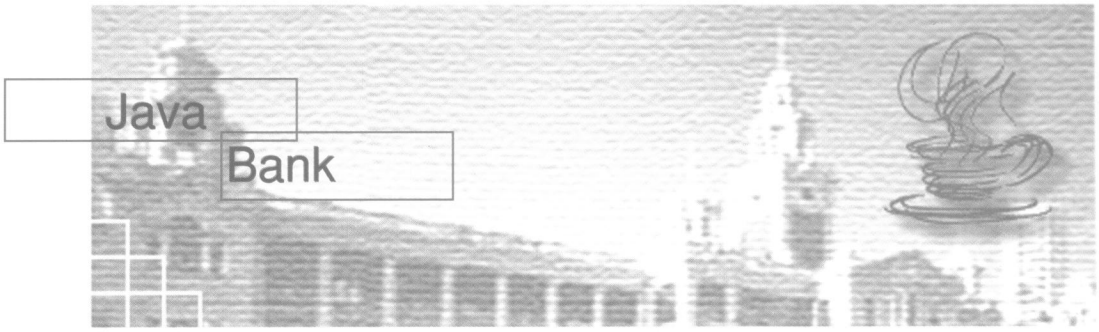
4. **System**
   There are two subitems:
   a. Logout
      Log out from the software.

   b. Add Users
      Allow access of the software and database for new comers.

Recommendations

Several functions are suggested to build in after a detailed revision of the program.

1. Search function is advised for easier fetching of an item.
2. Audio signals can be added if error or other alerting processes are encountered.
3. Profit strategy is greatly required. It would be better to develop onwards from the estimation. Price can then be adjusted for each product in the coming months so as to keep up with an acceptable profit and to relieve the over stocking problem.
4. A more complicated estimation function should be constructed with varying factors including price sales, season factors and competitions from other stores.



Authors

Kwok Hon Wa
Lee Kin Yip
Wong Him Ting

Introduction

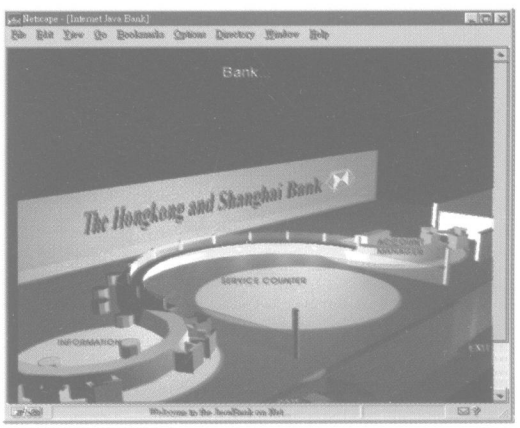
Thanks to the development of computer technology, people are now able to communicate by computers through the Internet. Internet is an enormous network connecting worldwide computers together and is able to transfer online data. Internet brings a lot of convenience to us, we can play, study, search information by just sitting in front of a computer. Among all the applications of the Internet, the most commonly used function is the world wide web. We can visit the homepages of different organizations, companies, universities or even anyone's homepage to access the newest information in the world. The only thing that is unsatisfactory is that the pages are still in the very beginning. However, there are still a lot of limitations in using the internet, and people are trying to make the pages interactive. Then CGI, VRML, JAVA and other tools came, which enable us to see a moving picture, to make decision and to communicate with one another. In this project, we are going to show the power of one of those tools, JAVA.

Nowadays, many tasks and services can be accessed through the Internet, including the banking service. However, most of them are not well-functioning, not user-friendly enough and not secure enough. To develop a perfect banking system, JAVA is the best choice because of its ability to work on Internet, as well as a large package of prepared user- friendly -components like menu, button, dialog, scroll bar, etc. which we always use in windows programs.

Through this powerful language, we have written a powerful banking query program that can be used through Web browsers in any personal computer. With it, banking information can easily be obtained. Also, we no longer have to queue up for half an hour. Instead, we can enjoy banking services comfortably at home.

1. Program Description

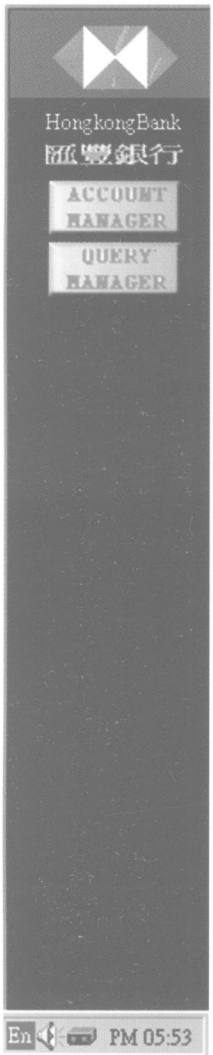
After connecting to internet and type in the address "http://www.eee.hku.hk/~62269-96/Java_Bank/java.html", a fantastic looking bank counter will appear.

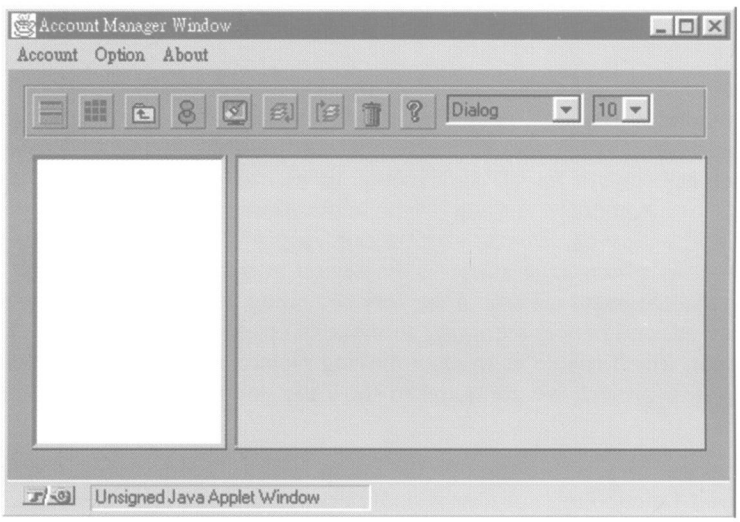


There are counters called *Account manager* and *information counter*, they can be pressed *and Account manager* and *Query manager* will appear to serve you. In the Account manager, one can have a look on how much money is inside one's account, and other related information is also available. In the Query manager, one can see the general information about the banking service like Credit card, Life Insurance, etc. If the job is finished, a click on EXIT can stop the program.

Besides, there is a more convenient way to do the same job, move the mouse to the right and a startbar as shown in the right will appear. By clicking the Account manager button, the Account manager window will appear and the same can be done for Query manager.

By dragging and moving the startbar, it can be relocated from the right to the left. Besides, a click on the Hong Kong Bank Logo can remove it.

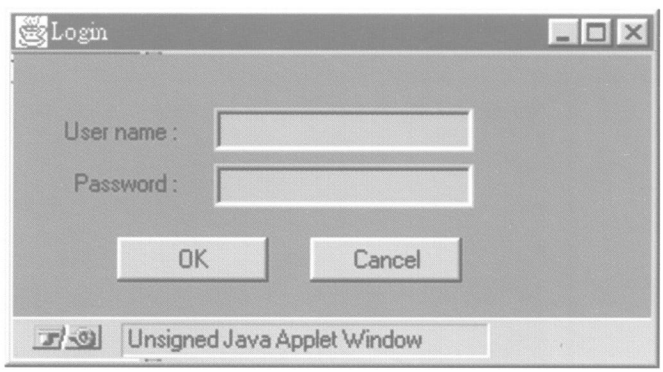




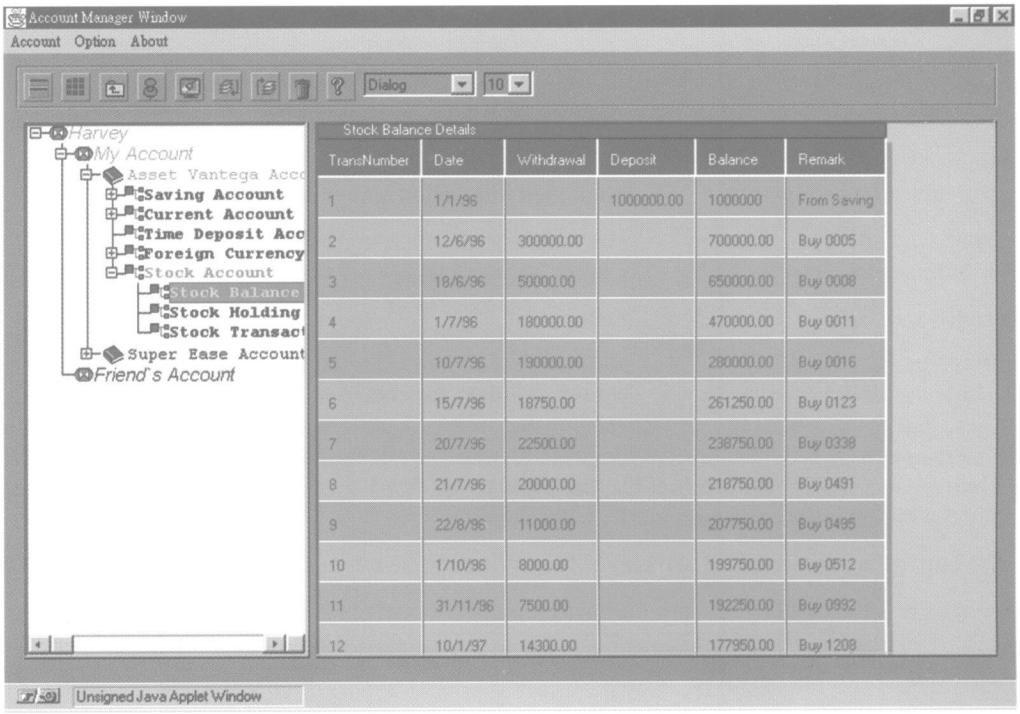
Let's have a look at the Account manager first, which is shown in above. There are three menus, namely, Account, Option and About, a set of buttons and Choice Boxes. The white area in the left is used for tree display and the right gray area is for table display.

Under Account menu, we can choose New Account, Close Account, User Login, User Logout and Exit. The item New Account and Close Account is reserved for further development. The item User Login can open existing database while the User Logout can close it. Exit is used to close the windows.

If User Logout is pressed, a dialog box will appear to ask for user name and password.

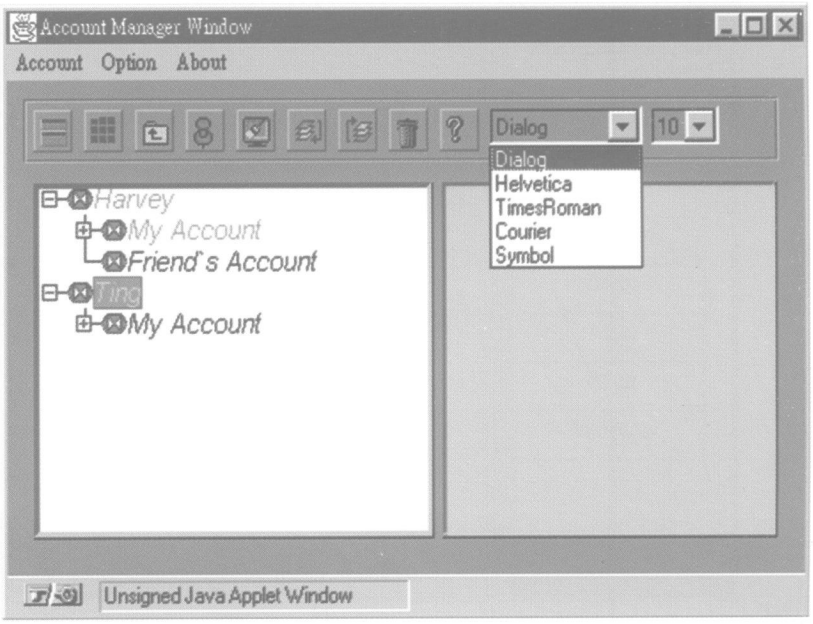


If the password is valid after checking, then a node will appear in the white area as the starting point.



If you click on it, it will show the items under it, for example, as shown above, a user called Harvey is using this program. Under the node 'Harvey' is 'My Account' and 'Friend's Account', which is designed for further development for transaction functions. Under 'My Account' is two Integrate accounts, 'Asset Vantage Account' and 'Super Ease Account'. And under 'Asset Vantage Account' is 'saving account', 'current account', 'stock account' and etc. Under 'stock account' you will see 'Stock balance', 'Stock Holding' and 'Stock transaction'. By this kind of structure, we can inspect our account easily. Once we clicked on the item, the relevant information will appear in table form in the gray area.

If more than one user have logged in, there will be multiple tree structures which enable further development like making transactions between two users' account. This function can be easily implemented under this tree structure representation.



In the menu Option, there are five entries, *'Show Upper Level'*, *'Remove Old paper'*, *'History record'*, '*Change…*', and *'Save…'*.

'Show Upper Level' means that only the node with higher level than the current selected node will be displayed. In fact, the tree structure may cause trouble when many nodes are opened due to its bulky nature. Disabling the 'Show Upper Level' function will set the tree starting with the node just selected.

'Remove Old paper' means that the previous table will be closed whenever each new table is opened. If several tables are required to be compared, then this option should be disabled.

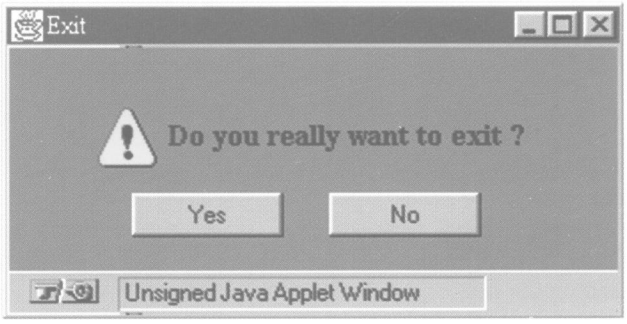
'History Record' means that the program will record which nodes have been pressed, it will change the color of those nodes to light gray so as to let the user know which node have been browsed.

'Change' and 'Save' are for further development.

For the buttons, the one at the left is to cascade the table when multiple tables are shown. The next one is to hide the table. The one with an arrow is to select the node just above the previous selected one. The pin is to pin the table so that it will remain there even if we press the CLEAR_ALL_TABLE button. The next one is to switch between 'single table' mode and 'multiple tables' mode. And then the next two buttons are to send the tables to the back and to the front. The garbage bar is to clear all tables. And finally the button with a question mark means help.

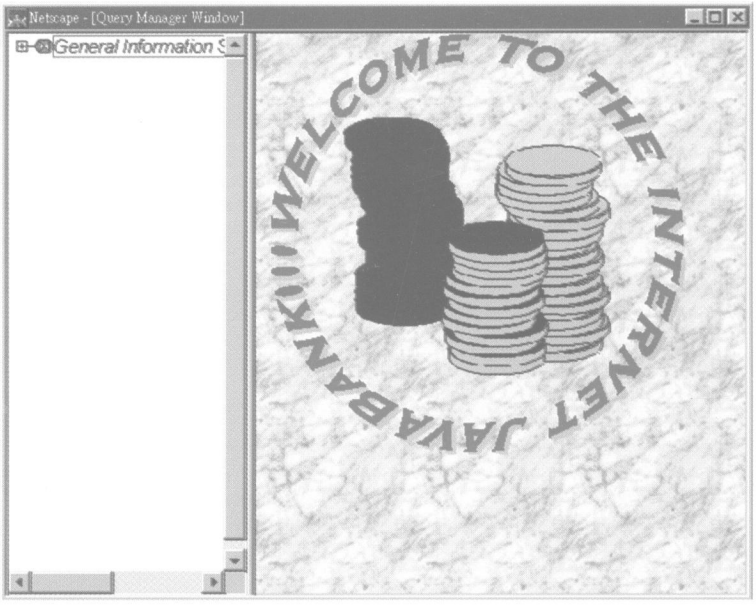
Those buttons are followed by two choice boxes that can be used to change font type and font size.

Once we press the exit key, the program will prompt for confirmation.

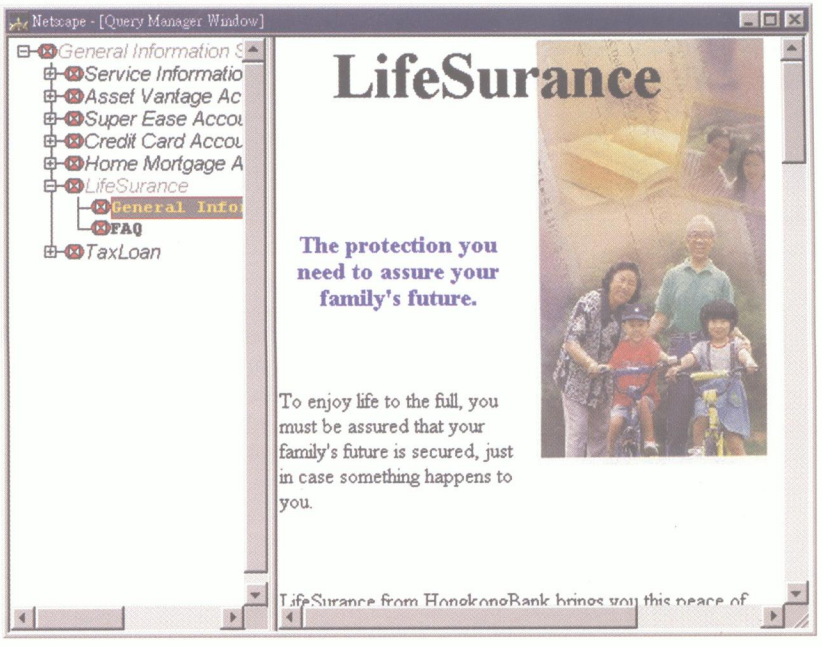


If we really want to exit, the window will be closed.

Let's open the Query manager now.



In Query manager, there is also a white area on the left to show the tree structure of the information and we can navigate through the tree.



There is no special function for the Query manager as it is already very user friendly. What we have shown above is the page about the Insurance Service. After we have selected the item, there will be a beautiful page about the item on the right.

2. Database Structure

The data format adopted are all in table format (that is, two dimensional array) for the ease of the implementation of mSQL. They can be categorized into three different classes, *UserTable*, *NodeTable* and *DisplayTable*.

UserTable format

This table format is used to store the basic data for the JavaBank system user. Each user after registration will have a field entry in this table and each time when the user login to the JavaBank system, the program will first come to here to see whether the user is a valid user of the system. The table format is shown in below.

root

User name	Password	Details	Icon Type
Harvey	123456	U1	Man
Patrick	ABCDEF	U2	Man
Ting	1w23efa	U3	Man
May	244562	U4	Woman

User Name: The field storing the name of the user

Password: The field for the storage of the password of a specific user

Details: This field stores the name of the table with NodeTable format which in turn stores the details of the user. (for example, the account information)

Icon Type: Each user can be represented by an icon which will be displayed in the tree diagram.

NodeTable format

This table format is used to store a small data item (called node). These data items will be displayed in the users' screen as a tree structure. Corresponding to each node, it may or may not has a flexible dimension table (called Display Table and will be discussed later) to show the detailed description about this node. The table structure is shown in below.

U1

SubItem	Name	Description	Icon Type
U1N1	Saving Account	U1Saving	Account
U1N2	Current Account	U1Current	Account

Subitem: It stores the name of the table with NodeTable format. In the display, these subitems are the children of a node. That means they are in one level lower than that node and so, they can be displayed unless that node (their parent) is displayed. In the coming section, there will be an example about the tree structure display.

Name: It is the name of a subitem and this name is displayed on the screen to represent this data item.

Description : Usually, there will be a descriptive table shown on the right hand side Panel in the display interface whenever this node is selected. This field acts as a pointer and it stores the name of the table with DisplayTable table format which effectively pointing to the database with data structure specified in the DisplayTable format.

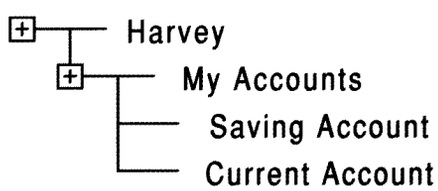
Icon Type: Similar to that of the UserTable, it stores the graphical representation of this node.

DisplayTable format

This is a very flexible table format which allow all dimension of the table because this type of table is used to display the general information of something, for example the account information, users information and some query information. The format is as below.

Item 1	Item 2	.....	Item n
Xxx	aaa	.....	bbb
Yyy	ccc	.....	ddd
Zzz	eee	.....	fff

To illustrate the working mechanism of the database structure, here is an example concerning a simple account information display of a user. Assume the following tree structure display is shown on the left hand side panel of the JavaBank client program display.



The actual database will be as follows:

root

User name	Password	Details	Icon Type
Harvey	123456	U1	Man

U1

SubItem	Name	Description	Icon Type
U1N1	My Accounts	U1AccDescription	Account

U1N1

SubItem	Name	Description	Icon Type
	Saving Account	U1Saving	Account
	Current Account	U1Current	Account

U1AccDescription

Account name	Net balance
Saving Account	$1000
Current Account	$1300

U1Saving

Date	Deposit	Withdraw	Balance
1/2/97	$100		$1000
2/3/97		$800	$200
3/3/97	$800		$1000

U1Current

Date	Deposit	Withdraw	Balance
23/2/97		$1000	$900
1/4/97	$400		$1300

With the above database, the JavaBank program will first get the user name and password of the user and send these data to the server program to check whether they are valid. The server program will access the root table and search whether there is a match with the inputted one. In this case, if the user name is inputted as "Harvey" and the password is inputted as "123456", the access is permitted and the server program will go to U1 (the table name is stored in the field Details) to get the data for the user "Harvey". The client program will display all the data in the table U1 in the second level of the tree while the first level is displaying the name of the user. In this case, the string "My Accounts" is printed on the screen. Then, when the item "My Accounts" is selected, the program will try to find the information which is lower level below "My Accounts" by looking at the field "subitem" in the table "U1N1".

In the meanwhile, the information for the item "My Accounts" is printed on the other side of the screen by displaying the table "U1AccDescription" (the table name is stored in the field "Description"). Since there are two "subitems", the program will display these two items, "Saving Account" and "Current Account" just below the level of "My Accounts". Similarly, as the "Saving Account" is clicked, the data for saving account stored in "U1Saving" will be displayed. However, since in this case there is no subitems needed for "Saving Account" further, there is no need to have a table for "Saving Account".

This is the rough mechanism for the communication between the database and the program. For the information which has more level of details needed to be displayed, the idea is similar, it just requires applying the above procedure repeatedly and recursively.

3. Class Diagram

Three diagrams are shown in the following pages. Diagram 1 shows the class relationship. In each class, it may import some other JavaBank classes. This diagram shows exactly what classes will be imported in each class. Diagram 2 and Diagram 3 are the class hierarchy diagram. These two diagrams show clearly the classes are extended from which classes and are implementing on which classes.

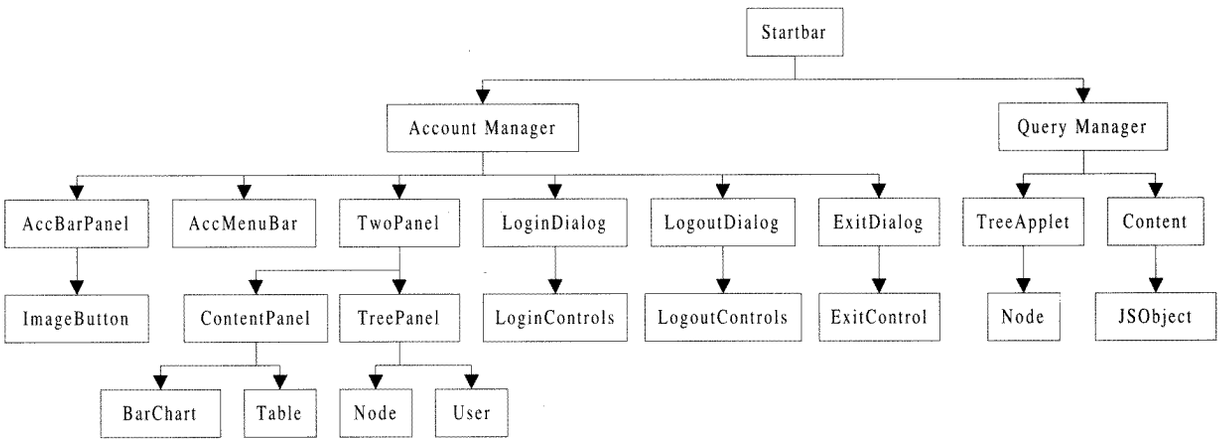


Diagram 1 : The Class relationship diagram

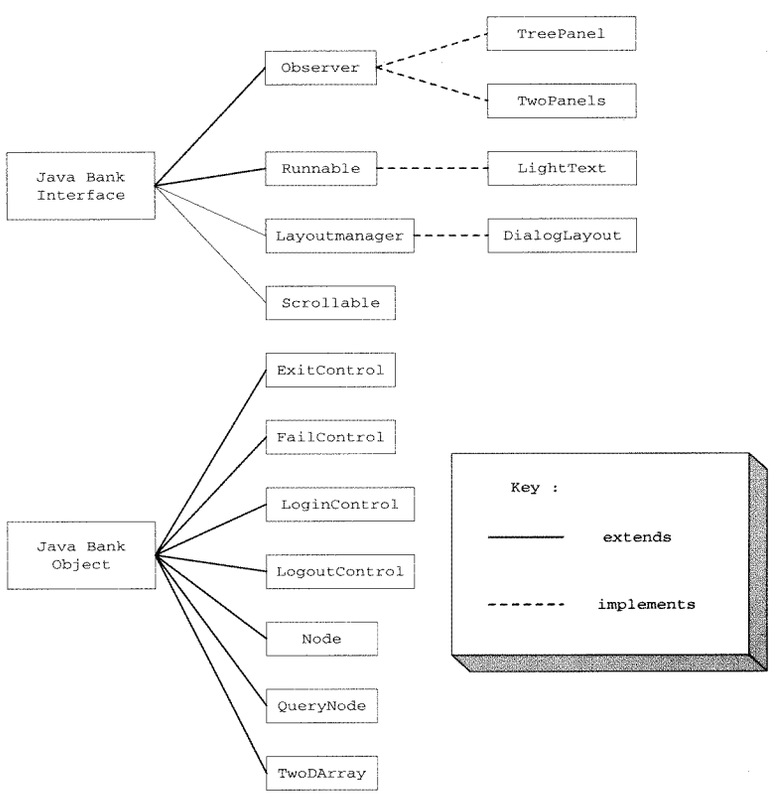


_Diagram 2 : The class hierarchy diagram A

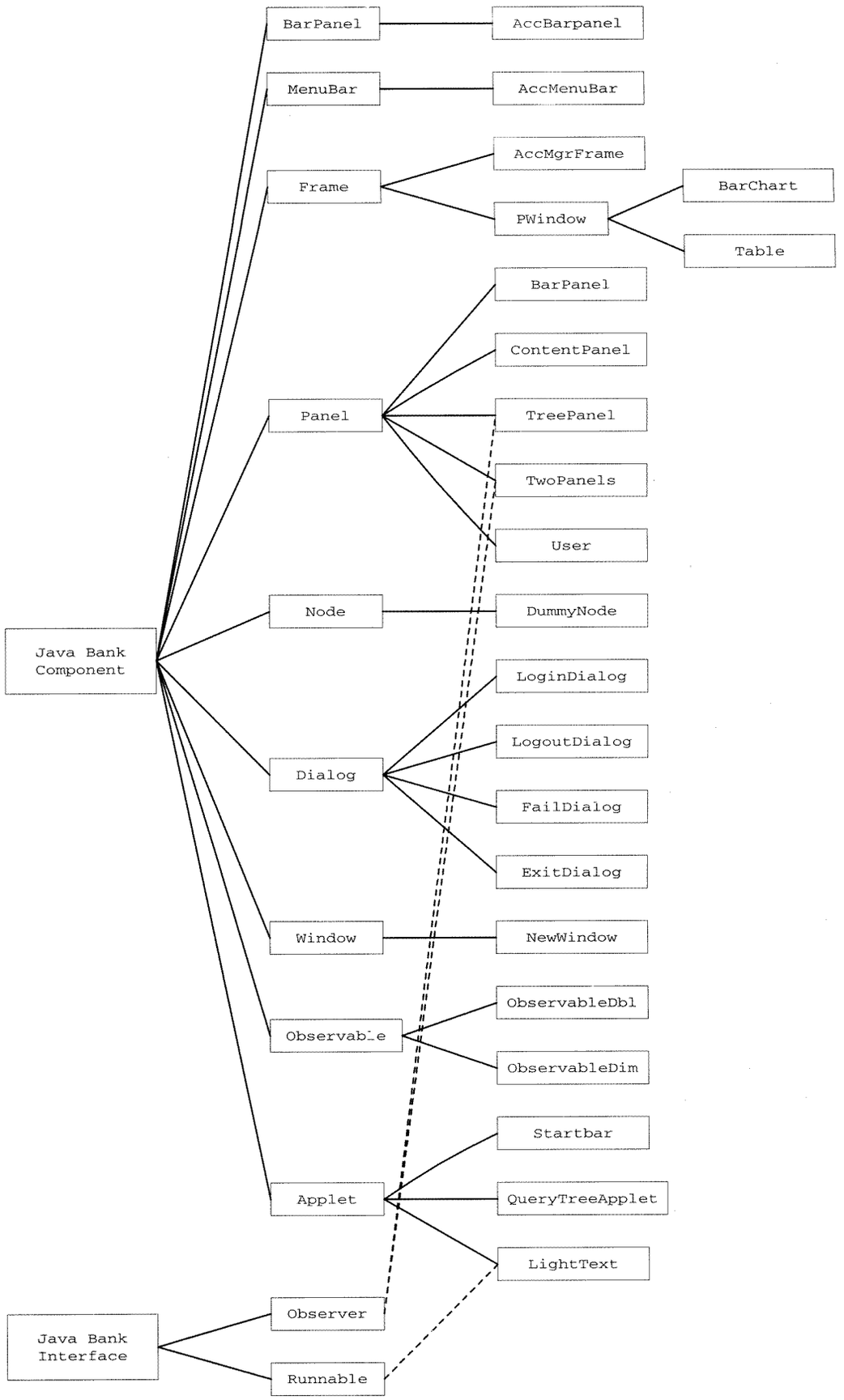


Diagram 3 : The class hierarchy diagram B

Discussion

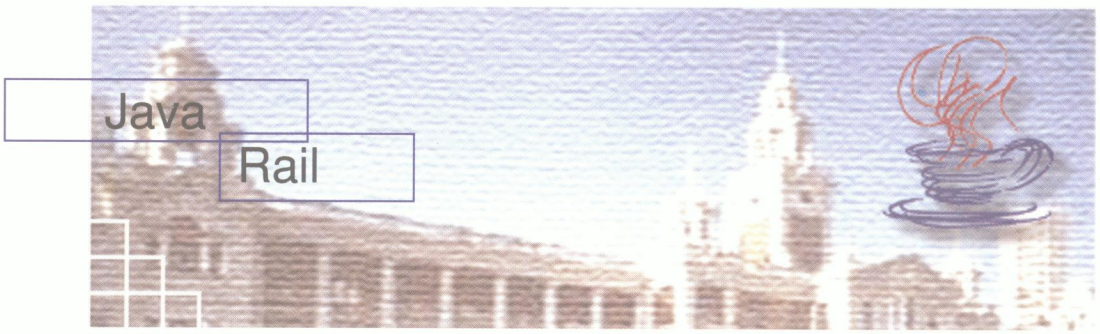
There are several difficulties in doing this project. First, for the part of "Account Manager", the goal is to follow the style of the Microsoft windows program manager so that the user interface is well understood and there is no need to memorize some operational commands. In order to follow this style, we need to develop the tree panel as the file manger. Since the tree structure is recursive in nature, we naturally code the program in a recursive way by starting programming with the class "node". So, in the following programming, the code becomes harder to debug and usually takes more time to debug. To enhance the convenience of the user, a number of special features are added, the keyboard input, drag and drop functions, multiselection, ... Additional to the tree is a button bar on the top which will take some simple actions on the tree or the displaying information. (for example, unselect all selected items, aligning the information in specific order) Moreover, the working environment besides the tree is multi-documented. That means users can select more than one items and all the information are displayed simultaneously beside each other. We then allow the information to be displayed as a sheet and can be moved freely on the screen. To make the interface more flexible, the size of the tree panel and the working environment on the right hand side can be rescaled by sliding the central axis and scrollbars that are added on them. Afterwards, different forms of sheet is developed, table sheet and bar chart sheet. When we add the features of the tree panel one by one, we find that the code becomes more and more condense and difficult to modify because usually the features are inter-related that many special consideration have to be taken.

In the "Account Manager", apart from the interface development, database is needed to be established for the user to get their data. Our database design is well designed and organized to fit our program as presented in the previous section. Certainly, before the successful model comes out, we have try numerous fail models. Finally, we still can find out the most suitable one for the program so that the coding in the "Account Manager" is easy to debug and the data are easily accessed through the tree. We have spent a lot of time to develop the whole database system because the server program for the mSQL is always down so that we cannot take a check on the database.

For the "Query Manager", the difficulty is in using JavaScript. After a great search on the internet, it seems to be a must that we need to use JavaScript to call the browser to display a HTML file. However, we still put a great effort on how to communicate between JavaScript and JAVA. Fortunately, we got a package from our supervisor where this package is used to call the JavaScript command in the JAVA code. After solving this problem, the remaining time is spent for typing the HTML file. It really demands a lot of time.

Conclusion

For security consideration, there is a series of package developed for JAVA by SUN and other companies. This language is also platform-independent and supported by the well-developed GUI package. So, it is really a good and suitable language to perform banking service on internet browser. Moreover, SUN is going to develop some low energy consumption chips which are specially designed for JAVA-written software. This has brought about the evolution of the banking system. In the future, the bank may issue some Smart cards or other electronic products with chips programmed by JAVA to extend their banking services.



Authors

Fung Wai Lung
Law Wai Fun
Lee For Lun
Ng Wing Shan
Yuen Chi Pang

Introduction

The JAVA RAIL is a web page which aims at providing a comprehensive enquiry system for the railway service in Hong Kong, namely, the Mass Transit Railway(MTR) and the Kowloon-Canton Railway(KCR). In Hong Kong, millions of people are using the railway system every day, it is becoming part of the life. This web page aims at giving all the information that a passenger needs. Written in Java, the web page is graphics oriented, and special visual effect is added to the design. There are five parts in the web page: *Route Advisor, Recommendation of Exit, Ticket Economizer, Shops* and *Special Features.*

The address of the web site is:

http://www.eee.hku.hk/~62269-96/JavaRail/java.html

Welcome Scene:

The welcome scene shows an interactive map of a concourse of the station. There is a television at the center showing snapshots of the railway system. By navigating the mouse pointer on the map, various highlighted parts can be located. To enter a particular part, click on the highlighted zone and a frame will be popped out.

1. Route Adviser

In this area, you can get information about fare, time duration, first train, last train and the suggested path for your journey by simply selecting the initial station and the destination station of any MTR or KCR station.

Operating Method

There are two ways for selecting the stations:

1. Move the mouse pointer to the wanted station and click the highlighted station.
2. Choose the MTR/KCR checkboxes in the above lists and then select the wanted stations. After selecting the initial station and the destination station, the selected information will be shown on the text area by clicking the buttons.

There are five buttons in this application:

1. Show Price

   This gives the information of fare of the journey from the initial station to the destination station. If the journey is from a MTR station to a KCR station, or the other way round, both the fare value of the MTR journey and that of the KCR journey will be shown and the total fare will be calculated.

   There are some options for this function:

   a. Adult / Student / Concessionary

      This checkboxgroup will indicate the type of fare for the journey. Usually the fare for "Adult" will be double that for "Concessionary". For the type "Student", it will be regarded as "Concessionary" for MTR journey but "Adult" for KCR journey.

   b. Single Value / Stored Value

      This checkboxgroup indicates whether a stored value ticket is used. Usually a few cents as discount will be given if a stored value ticket is used.

   c. MTR Congestion Time

      This checkbox will indicate whether the MTR journey begins within the Congestion Time, that is, 8:00am - 9:00am. For journey travelling from the Yellow Region to Purple Region, or the other way round in the period, "Student" and "Concessionary" would also be charged at the same rate as "Adult". (Note that the boundary of the Yellow/Purple Region is between Jordan and Yau Ma Tei as well as Quarry Bay and Lam Tin)

   d. MTR Stragger Time

      This checkbox indicates whether the MTR journey begins within the Stragger Time, that is, before 8:00am or 9:00am - 9:30am. For journey starting within this period, 30% discount will be given if a stored value ticket is used.

   e. KCR Sunrise Time

      This checkbox indicates whether the KCR journey begins within the Sunrise Time, that is, before 7:45am. For journey starting within this period, discounts will be given if a stored value ticket is used.

2. Show Time

   This gives the time duration of the journey from the initial station to the destination station. If the journey is from a MTR station to a KCR station, or the other way around, both the time duration of the MTR journey and the time duration of the KCR journey will be shown and the total time duration will also be calculated.

3. Show Train

   This gives the time of the first Train and the last Train of the journey from initial station to the station of destination. If the journey is from a MTR station to a KCR station, or vice versa, both the time of the first train / last train from the initial station to Kowloon Tong of the MTR journey and

that for the first train / last train from Kowloon Tong to the destination of the KCR journey will be shown.

4. Show Route

   This gives the shortest and fastest path for the journey from the starting station to the destination station. The interchange stations will be shown and the whole route will be highlighted on the map.

5. Reset

   This resets the previous selections.

2. Recommendation of Exit

This part provides information of the peripheral of each station. By clicking on the "Recommendation of Exit" in the welcome screen, a map showing the route of the railway system will be displayed. By clicking on the destination station on the map, another frame about that station will be shown.

This frame contains a peripheral map of the station, with three buttons labelling "bus", "taxi" and "maxicab". There is a text field for input and a text area showing the searched result.

By navigating the mouse pointer on the map, we can find the locations of the bus, taxi and maxicab terminals, they will be displayed when the mouse is pointing at them. Or, we can click on the "vehicle" button to show all the locations of the terminals of that public transport. The route number will also appear on the text area.

To find the right exit of a place near the station, just type the name of the place in the text field and press "Search". The exit code will be shown on the text area.

3. Stored Value Ticket Economizer

It helps the user to use the common stored value tickets (CSVT) in a more efficient way, since users can use the stored value ticket for any single travel for the last trip even if there is only $0.1 left, and therefore, it can economize the use of CSVT by finding out the way to use the CSVT so that the least value would be left based on the information of the user's usual trip and the type of ticket used.

From your usual trip, it will find out the number of times you can travel by using this CSVT. After a series of calculation, it will tell you how to plan the last one or two travels so that less than $1 will be left. In most cases, it succeeded. However, there are few exceptional cases.

Operating Method

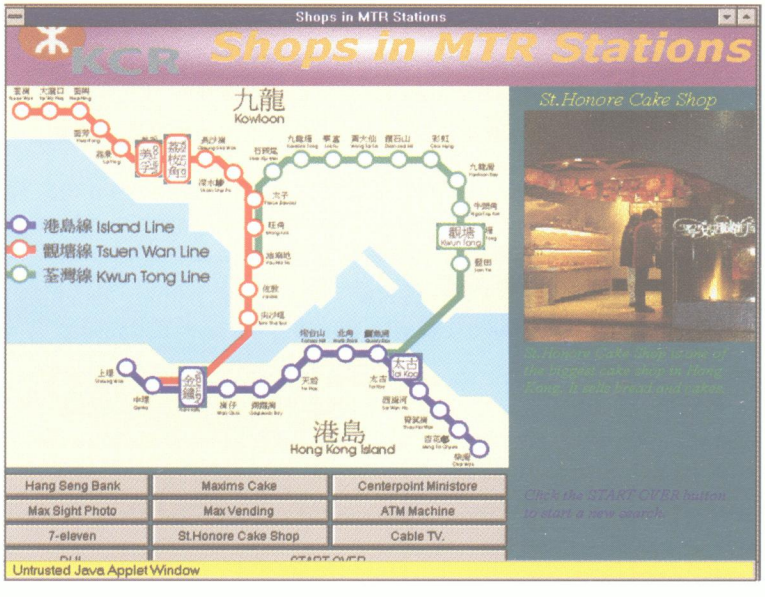
Input your usual travel by clicking your usual starting station and then your usual destination on the map.

Select the common stored value ticket (Adult $200, Student $50 …) that you are using.

Then the number of times of travel of your usual route will be displayed.

It will suggest how to plan the last one or two trips so that less than $ 1 of the ticket will be used for the last travel.

4. Shops in MTR Stations



There are a lot of shops in the MTR stations, such as the Hang Seng Bank, Maxim's Cake Shop, 7-11, etc. This program can locate and provide relevant information of the various shops.

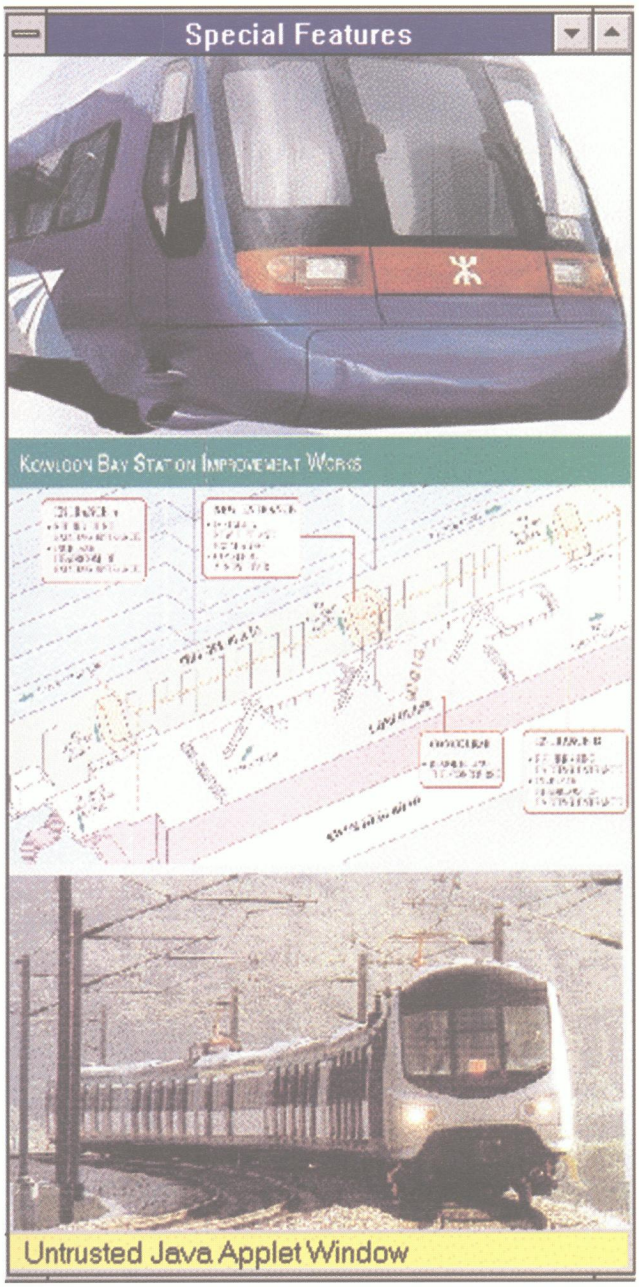
It can help you to search your desired shops by two methods. The first one is sorting by stations and the second one is sorting by shops. Also, some brief information about the shops will be provided.

Operating Method

If you click a station on the map, the photos of the shops located in that station will be displayed.

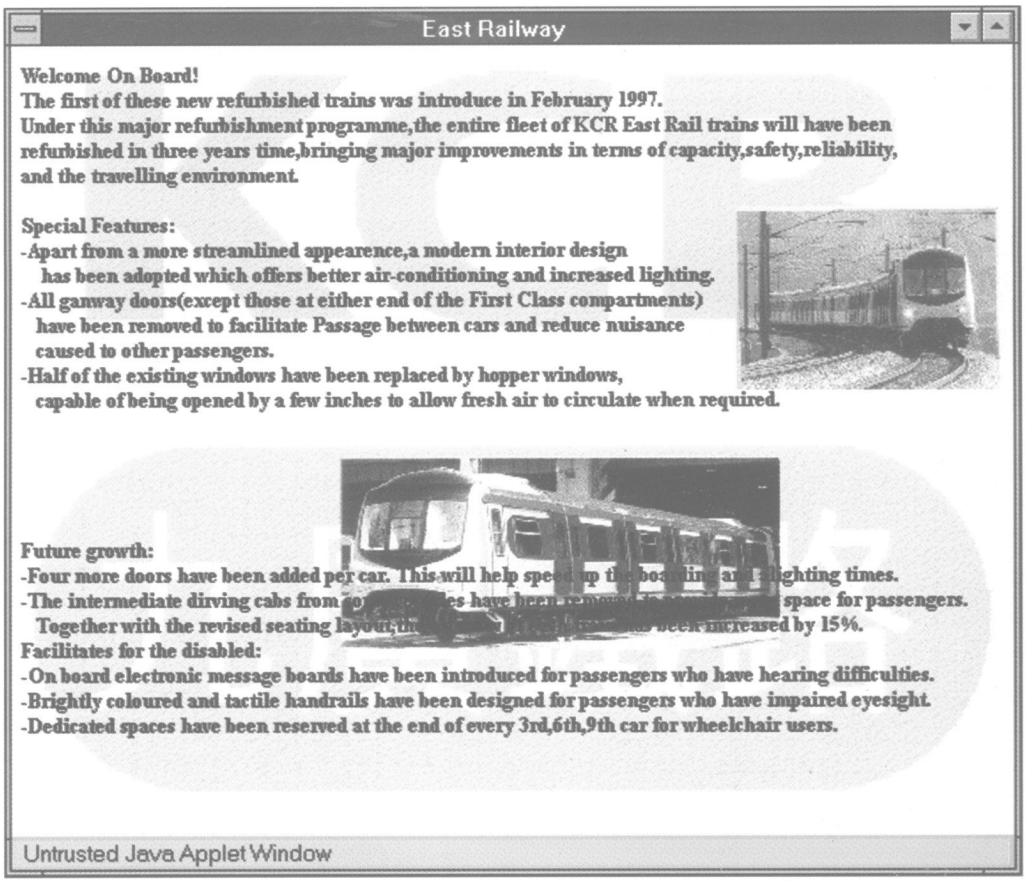
If you click a shop button on the shop panel, the stations have those shops will be highlighted on the map.

5. Special Features



The last part of the project describes the special features, which contains all recent information about MTR and KCR. They include the future development project of the Chek Lap Kok airport, improvement project of Kowloon Bay Station, expansion of the East railway and so on, so that passengers can know what facilities will be provided or improved in the near future. Besides, this part provides a channel for MTR and KCR to communicate with passengers by putting in their advertisements.

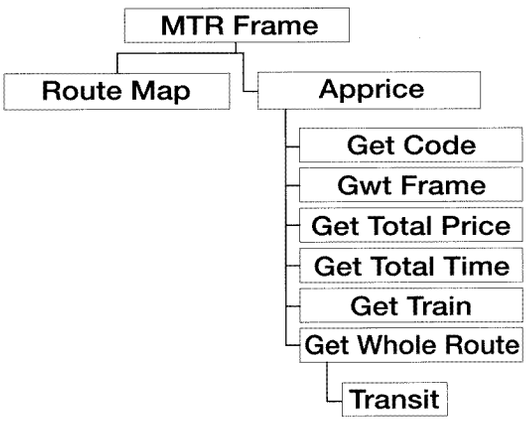
Different special features are expressed in frames. An example is shown in the figure on next page. Each time when you choose to read a piece of news, a new frame for that piece of news will be produced without closing the previous frames so that users can read any piece of news at anytime. However, this will not be possible with HTML alone. Each frame is a mixture of text and images. The text and images of the news are stored in the msql database with their location on the frame so that one can place the text and images anywhere he likes. In addition, text and images are not stored directly in the source code so that it can be changed or updated easily without modifying the source code.

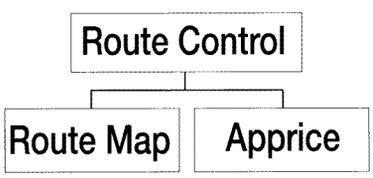


Program Structure

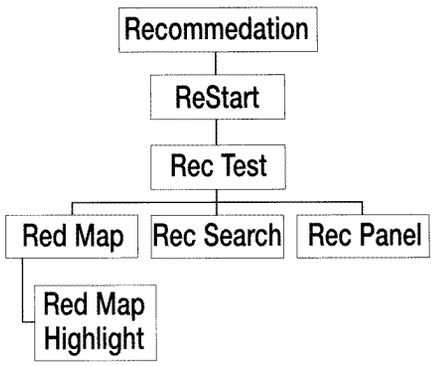
Java language is object-oriented, so, each part of the programme can be represented individually by classes. The following charts show the main structure of the programme.

Adviser

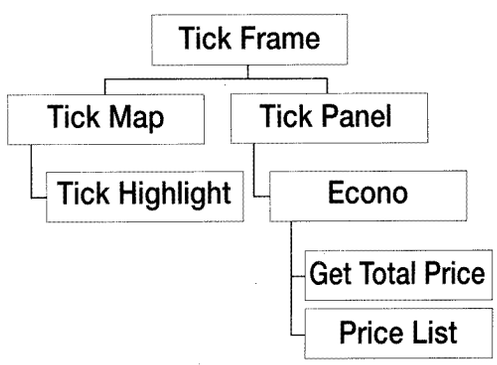




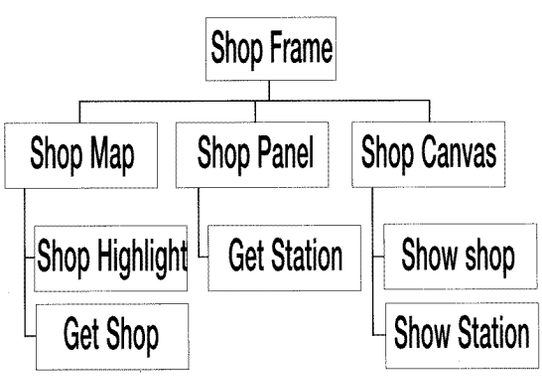
Recommendation of Exit



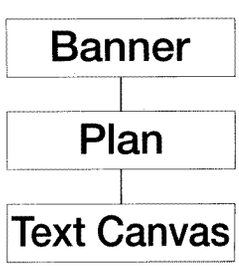
Economizer

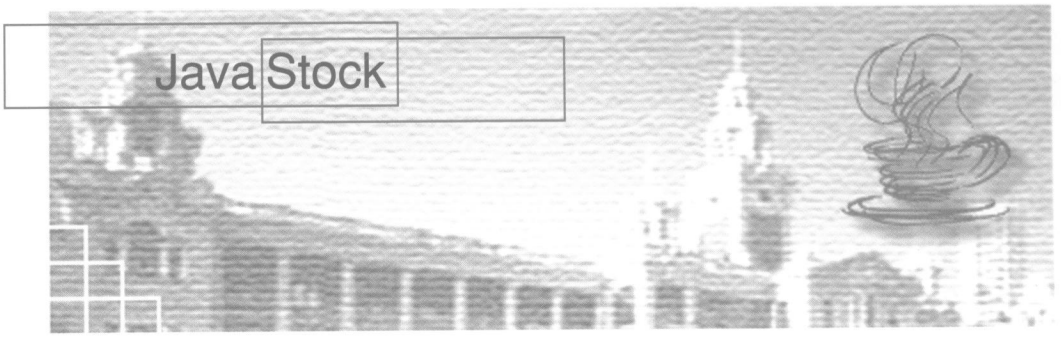


Shops



Special Features





Authors

Lee Wing Hei
Li Tsan Hang
Li Ka Lun

Abstract.

We describe in this paper the implementation of an internet-based stock price reporting and analyzing program.

Introduction

The Hong Kong Stock Market is one of the biggest stock market in the world. To become a better investor, one needs to obtain the latest information and trend of the market.

With the explosive growth of the Internet and world-wide-web technologies, it is feasible and convenient for people to access the latest market situation on the Internet.

The objective for this project is to provide a handy tool for Hong Kong Stock investors to obtain the latest information and determine the trend of the market easily at any part of the world.

To enable users to access our program at any platform, JAVATM language is used to implement the program. So, the only requirement for our program is a JAVA TM enabled browser and Internet accesses.

1. Program Briefs

In this part, some general information about the program, namely, *system requirements, main functions, program size* and *memory requirements* will be introduced.

1. System requirements
   - Java-enabled browser
   - 28.8k Internet access (ISDN or T1 leased line are recommended)
2. Main functions
   - Real-time stock reporting for Hong Kong Stock Market
   - Technical analysis tools
   - Past data searching for Hong Kong Stock Market (from Jan 96 to March 97)
3. Program size
   - The size for source code and image files equal 651,172 bytes (excluding the database)
4. Memory requirement
   - The maximum memory usage of the program is 14312K on a Window NT 4.0 workstation.

2. Functions Demonstration

In this part, three main functions of the program will be demonstrated.

1. Real-time price reporting

Click this button to invoke the stock price-reporting panel. ①

Type the name of the stock you want in this textfield and press enter. ⑤

Click on the stock you want to get the stock price. ③

② The Hang Seng Index, Chinese Enterprise Index, etc are shown on this panel one by one using animation.

④ The price of the stock is shown on the wooden bar.

After you have entered the main working environment of the program, you should click on the 1st button in Chinese character to call for the real-time price reporting function.(See ①)

Then, you will see the value of Hang Seng Index, China Enterprise Index, and Market Turnover, Mid Cap 50 Index and Volume one by one on the upper part of the window. (See ②)

To find out a particular stock, you have two methods.

You may use the selection scroll bar (See ③) to scroll to your desired stock and click on it, then you may see the stock price on the wooden bar. (See ④)

The second method is by typing in the name of the stock on the textfield provided. (See ⑤)

2. Technical Analysis

Click this button to invoke the panel ❶

Click this button to choose analysis method ❸

Use these buttons to draw trendline, horizontal and vertical line ❺

This is the trend line. ❻

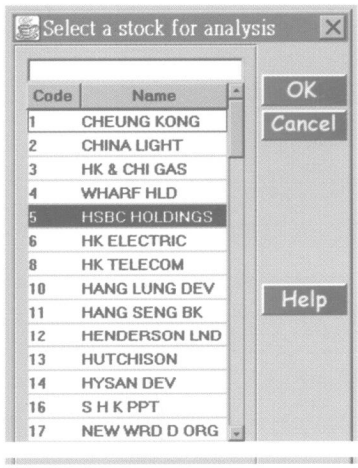
Use this vertical line to locate the date of the price. ❼

❿ This is the volume of the stock.

❾ This is the discrete Fourier transform of the closing price

❹ These are the long period and short period moving average.

❽ The RSI curve



To invoke the technical analysis panel, you may click on the second button on the top of the frame. (See ❶)

After selecting the desired stock using the dialog box provided (see ❷), you may see the working panel as shown above.

To analyse the stock, you have three methods, namely, *moving average, relative strength index* and *discrete Fourier transform.* You may choose any one of them by using the first button in the side panel. (See ❸)

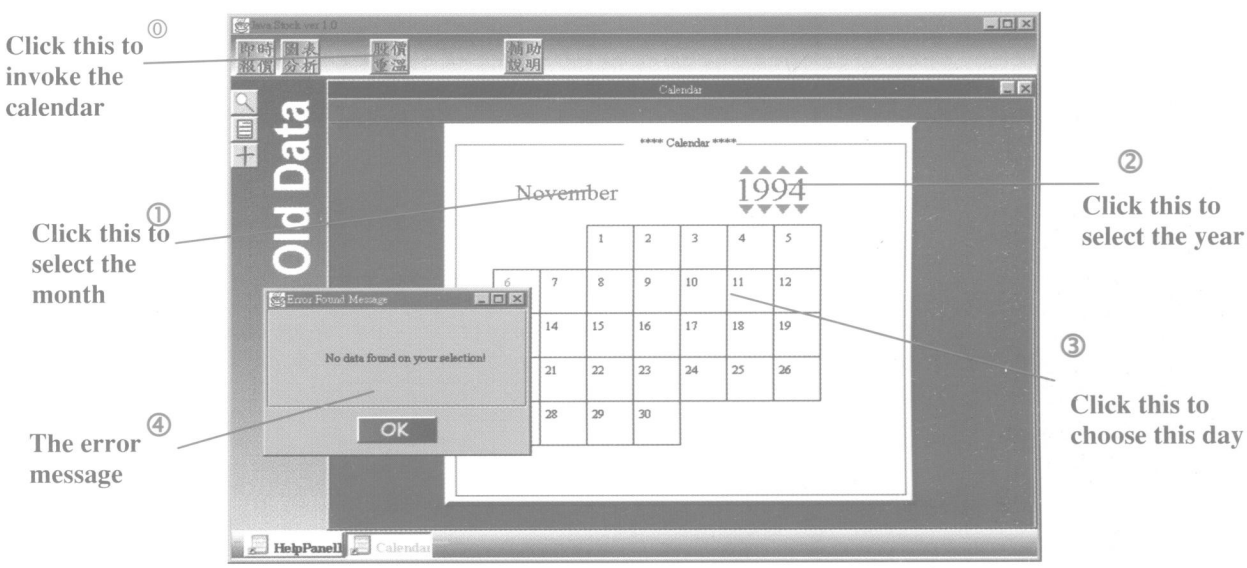
Using moving average analysis, you may choose to draw the long period moving average and short period moving average as shown above (See ❹)

You may use the drawing tools provided on the side panel to draw the trendline, horizontal line and vertical line. (See ❺, ❻, ❼). Using the Relative strength analysis, you may draw two horizontal lines at 20 and 80 to determine the buy and sell point. (See ❽)

Using Discrete Fourier Transform, you may see the smooth trend of the price of stocks. (See ❾)

On the lower part of the panel, you may see the volume of the selected stock. (See ❿)

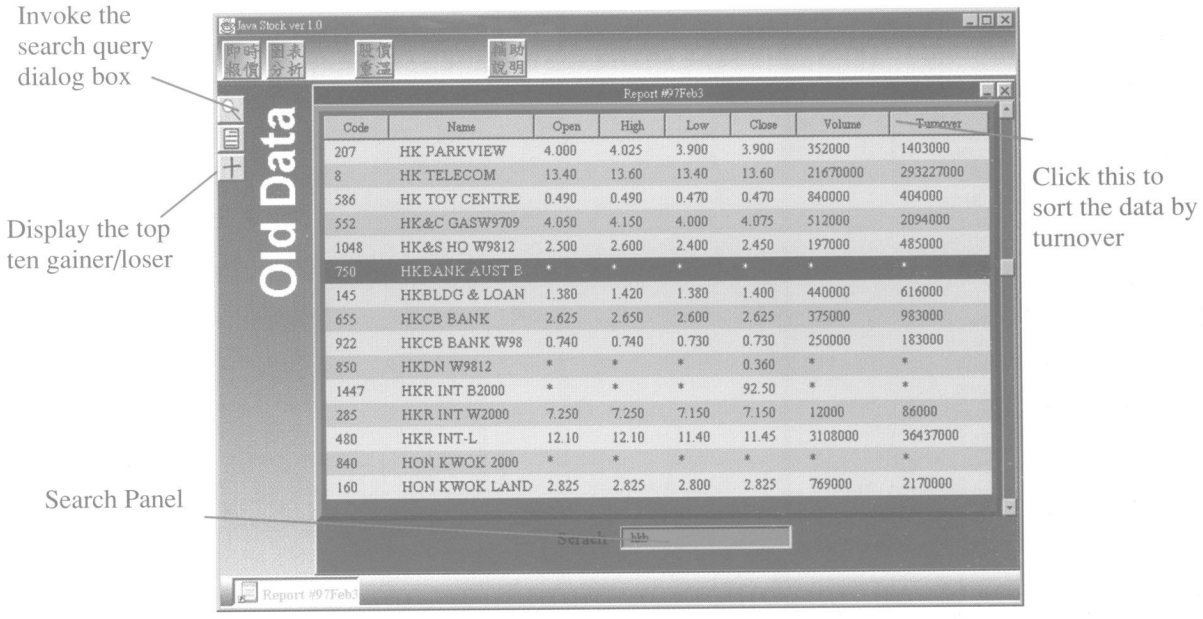
3. Past Data Searching



Click the third button on the top of the window to invoke a calendar. (See ⓪)

The above figure showed the appearance of a calendar. We can select the report simply by clicking the corresponding date. The design of this calendar considers the easiness of operation. The date can be changed easily by using the mouse. You need not to enter the date by typing. (See ①, ② and ③)

An error message will appear when no data can be found in the database. Otherwise, the full report will be loaded. (See ④)



This is the appearance of the full report. There are eight fields in the report: *code, name, opens, high, low, close, volume* and *turnover.*

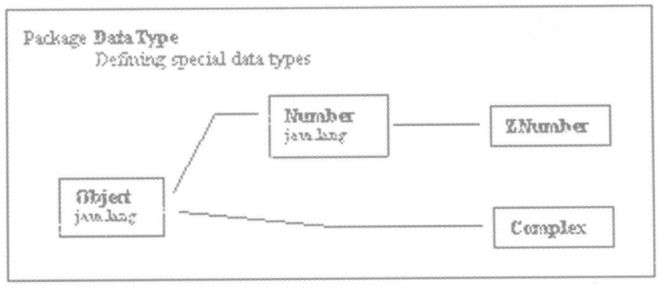
All fields can be sorted by both ascending and descending order. In order to achieve a fast operation of sorting, quick sort algorithm was chosen because it is the fastest method.

There is also another function to increase the efficiency of searching. It is a searching function by entering the text. It can search the record by entering the prefix of the stock's name. The record will be highlighted when there is a record start with your entering string. It is useful when the users doesn't know the exact spelling of the stock.

3. Design Details

In this part of the paper, the design detail about the program will be discussed. It will be divided into parts, namely, *special data types, reading and writing streams, handling real time stream format, database organization, technical analysis of stock, real time data demonstration, dummy operation system, past data searching* and *the help system.*

1. Special Data Types



Special Data types used:

ZNumber

a ZNumber object is a decimal number in the form of SZZZ...ZZZ9.99...999

where '9' stands for digit 0 to 9
'Z' stands for blank or digit 0 to 9
'S' stands for null or -

Storage method

interprete the number as mantissa x 10 ^ (- negExponent)

mantissa = SZZZ...ZZZ999...999

negExponent = no. of digits after decimal point

Precision

precision is guaranteed for 18 digits since 64 bits are used in storing mantissa.

Usage

It is more convenient to store decimal fixed point numbers in this format to save bandwidth in realtime transmission and space in database.

Complex

Using both rectangular form and polar form to store complex number.

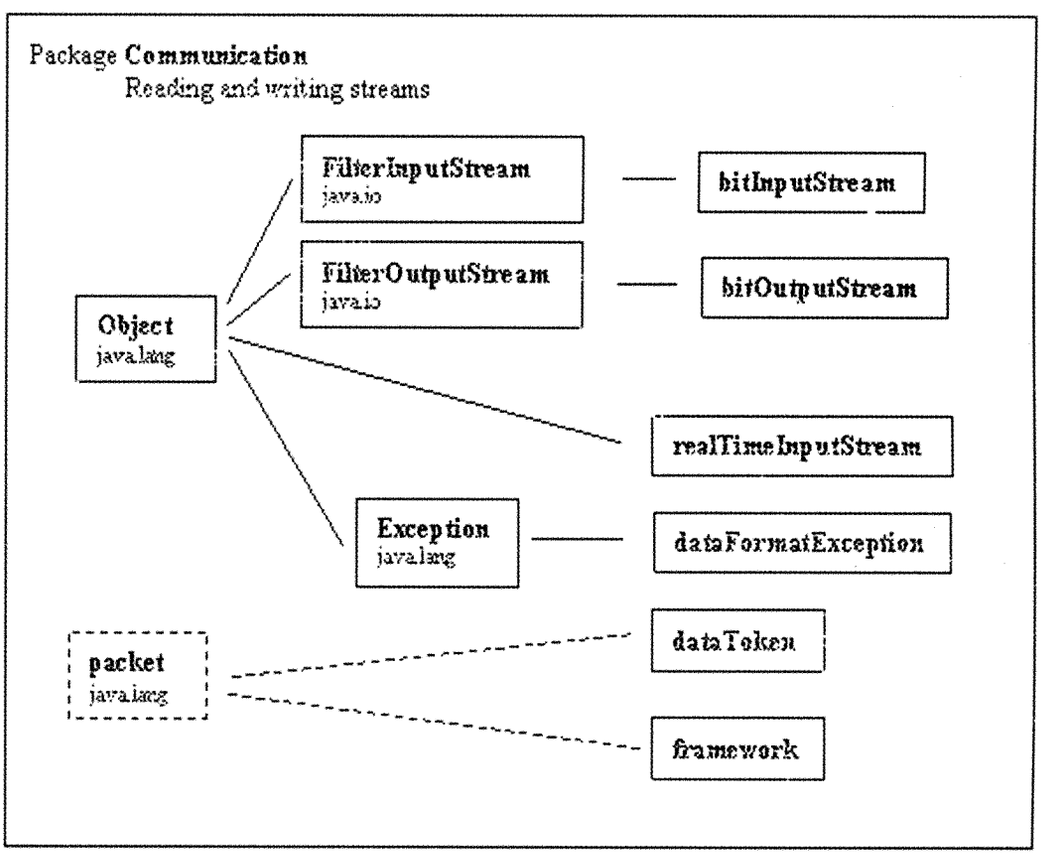
Advantage

It enhances the function calculation since multiplication is easier in polar domain while adding is easier in rectangular domain.

Usage

It serves as a basic object in manipulating calculation involving complex numbers, for example, in calculating discrete fourier transform.

2. Reading and writing streams



bitInputStream, bitOutputStream

To save bandwidth in real time transmission, data of numbers and strings are compressed in both runlength and fixed length formats and the length are measured either in bits or in bytes to further save space and increase data rate.

The class bitOutputStream is used to write strings, integers and Znumbers in compressed form to the outputstream while the class bitInputStream is used to read them out from a inputstream. The compression scheme is listed as follows:

1. Compression of string

- 6 bits each character + 6 bits for terminating
- each character is restricted to the ascii range from 32(space) to 90(Z)
- 000000 represents sp

...

111010 represents Z
111011 to 111101 are reserved for future use
111110 represents unknown(will be decoded as '^')
111111 is the terminating character

- lower case will be converted to upper case

2. Compression of Znumber

- format : mantissa * 10^(-negExponent)
-16 bits normal,4 bits for unknown or representing a predefined value usually appears
- 4 significant digits of mantissa will be stored 0000 -> 9999 (10)
- neg Exponent ranges from 0 -> 5 (10)
- result : 00000 -> 59999(10)
0000 -> EA5F (16) (16 bits used normally)
- unknown/predefined value will be represented by F (only 4 bits used in this case)

3. Compression of integer

- 4 bits for unknown,20 bits for normal,40 bits for huge
- range is restricted from 0 to 2031613 (10)
- unknown will be represented by F
- '0' to '983038' will be represented by 00000 to EFFFE
- '983039' to '2031613' will be represented by EFFFF00000 EFFFFFFFFE

packet, dataToken, framework

In decoding the real time stream containing real time information like Hang Seng Index and humorous amount of stock data, dataToken and framework define the fundamental objects in reading the stream.

A dataToken is the smallest element in a real time stream data. A framework is a collection of dataTokens and/or framework. We used 'packet' which is an interface to represent a dataToken or a framework.

For instance, in the Hang Seng Index Information, there are two fields, namely *TIME* and *HSI* which store the time and index value recorded in this information. Then TIME and HSI are two dataTokens while the information, being a collection of dataTokens, is a framework. (Note that the stream format is defined in the package streamFormat)

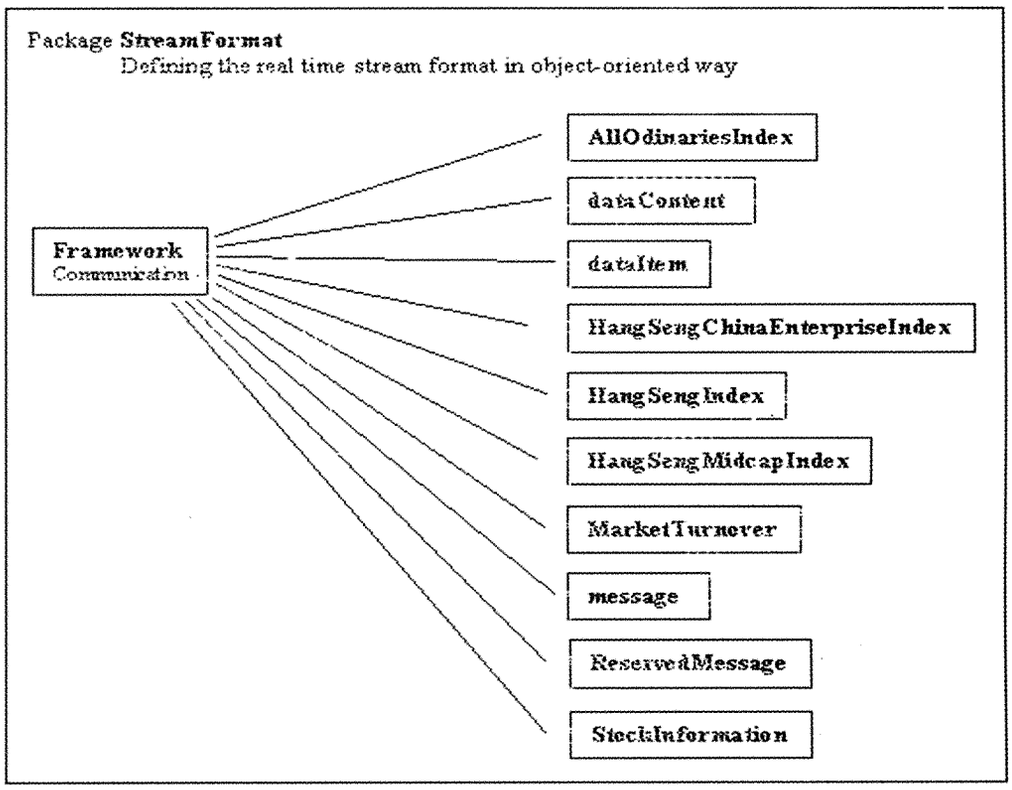
realTimeInputStream

Before decoding the individual frameworks, we must first read the stream in a startByte-content-endByte manner to determine where the framework(generally, it can be a packet) starts and ends. It is done in the class realTimeInputStream. In this class, apart from cutting the stream into frameworks, it can handle errors such as startByte appearing before endByte, etc.

dataFormatException

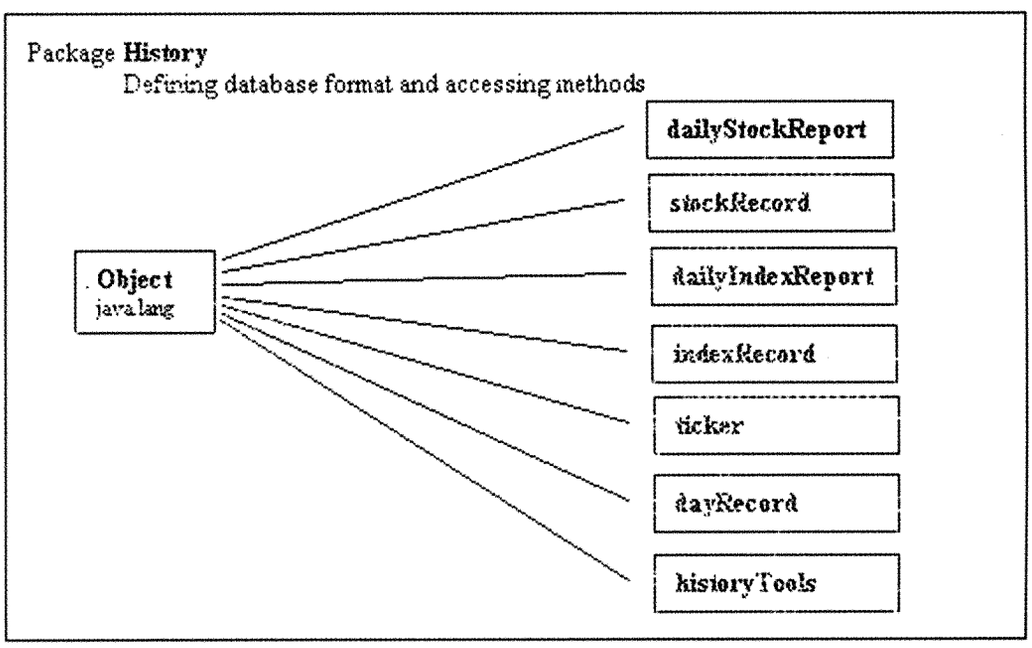
An exception to be thrown whenever there are errors in reading the stream or errors in decoding it.

3. Handling Real Time Stream Format



In this package, the real time stream format is defined in an object-oriented way such that the specification of stream can be varied time by time. Basically, the classes in this package are frameworks and each of them contains some dataTokens as well as other frameworks.

4. Database Organization



In real time application, speed is very important. To avoid busy database query and the unstable database performance, we have defined three kinds of data structure to store the daily stock information, daily index information and past history of a stock in the previous year in a file storage manner rather than storing the data in a database and accessing it by a database engine.

The daily Stock Information
Daily stock information is stored in files with extension '.dsr' . The specifications are listed below:

FileHeader

"DSR V1.0" 8 bytes

field definition

Compressed String of "CODE"	30 bits
Compressed String of "NAME"	30 bits
Compressed String of "OPEN"	30 bits
Compressed String of "HIGH"	30 bits
Compressed String of "LOW"	24 bits
Compressed String of "CLOSE"	36 bits
Compressed String of "VOLUME"	42 bits
Compressed String of "TURNOVER"(optional)	54 bits

records

Code	integer	12 bits
Name(<=13 characters)	compressed string	6 - 84 bits
Open	compressed ZNumber	4 or 16 bits
High	compressed ZNumber	4 or 16 bits
Low	compressed ZNumber	4 or 16 bits
Close	compressed ZNumber	4 or 16 bits
Volume	compressed Integer of (Volume/1000)	4 or 20 bits
Turnover(optional)	compressed Integer of (Turnover/1000)	4 or 20 bits or 40 bits

repeat until there are no more records
(if the number of bits is not a multiple of 8, extend it to 8n by adding zero bits here)

fileFooter

"EOF DSR" 7 bytes

The class dailyStockReport constructs a complete information about all the stocks at a specific date. It first reads the corresponding .dsr file by the class bitInputStream and stores the records in an array of objects called stockRecord. The structure of a stock record is:

stockRecords							
Stock Code#	Name	Open	High	Low	Close	Volume	Turnover

The daily Index Information
Daily index information is stored in files with extension '.dir' . The specification of it is listed below:

fileHeader "DIR V1.0" 8 bytes

field definition

Compressed String of "NAME"	30 bits
Compressed String of "OPEN"	30 bits
Compressed String of "HIGH"	30 bits
Compressed String of "LOW"	24 bits
Compressed String of "CLOSE"	36 bits
Compressed String of "PV CLOSE"	54 bits

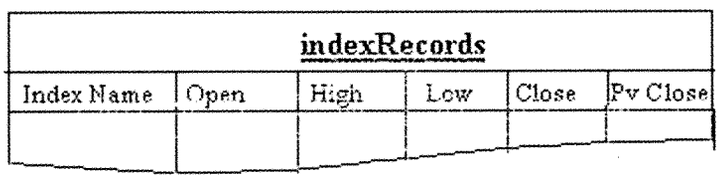
records

Name(<=15 characters)	compressed string	6 - 96 bits(usually)
Open	compressed Integer	4 or 20 bits
High	compressed Integer	4 or 20 bits
Low	compressed Integer	4 or 20 bits
Close	compressed Integer	4 or 20 bits
Pv Close	compressed Integer	4 or 20 bits

repeat until no more records
(if the number of bits is not a multiple of 8, extend it to 8n by adding zero bits here)

fileFooter	"EOF DIR"	7 bytes

The class dailyIndexReport does similar things to dailyStockReport. The structure of an index record is:



indexRecords					
Index Name	Open	High	Low	Close	Pv Close

The past history of a stock
Stock history information is stored in files with extension '.tic' . The specifications are listed below:

fileHeader	"TIC V1.0"	8 bytes
Name	compressed string	30 bits

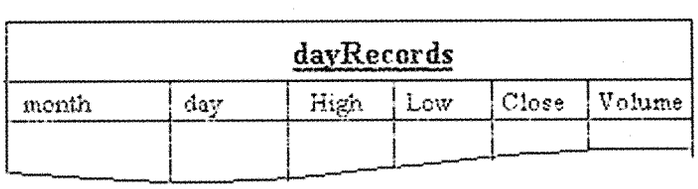
records(from Jan2 to Dec31) (4 bits for null otherwise 48 bits)

High	compressed ZNumber	16 bits
Low	compressed ZNumber	16 bits
Close	compressed ZNumber	16 bits

repeat until no more records
(if no. of bits is not multiple of 8, extend it to 8n by adding zero bits here)

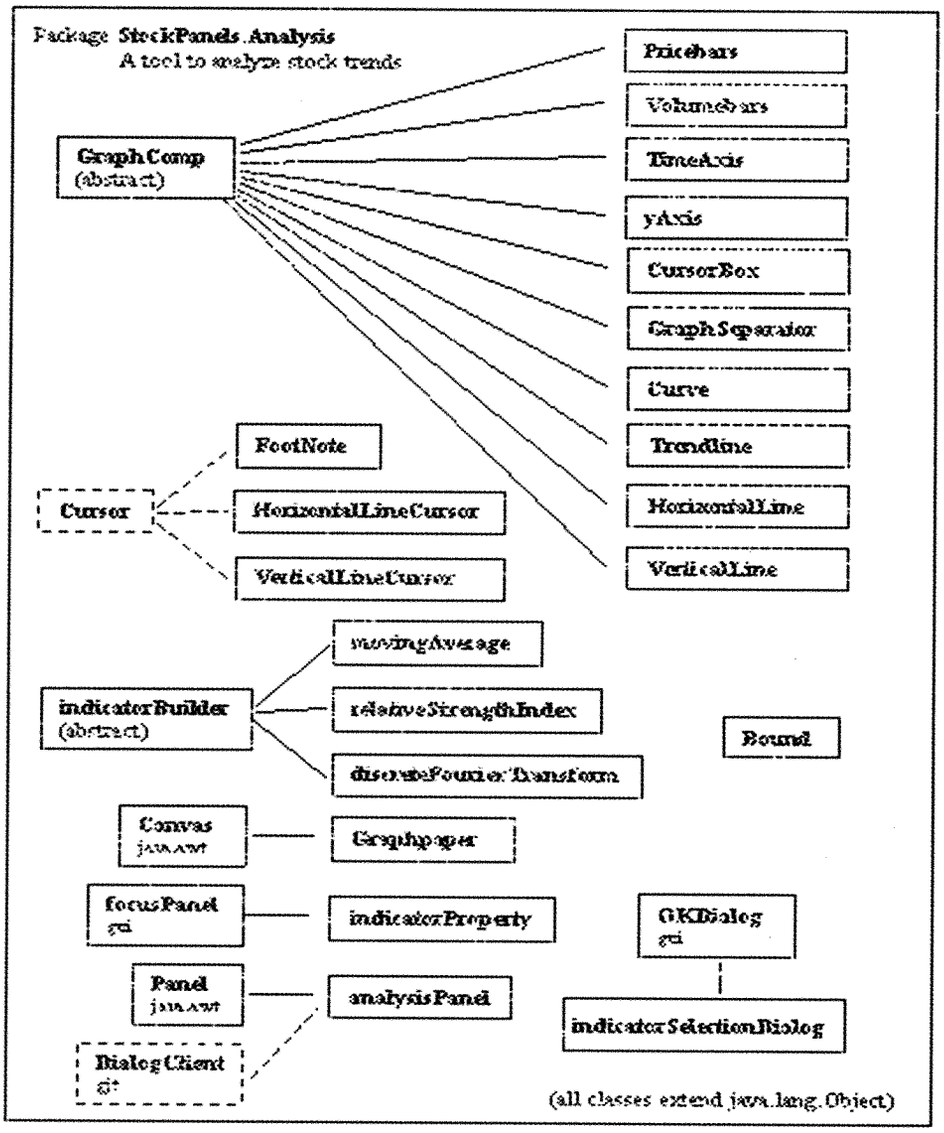
fileFooter	"EOF TIC"	7 bytes

The class ticker reads from the .tic file and stores the information in an array of dayRecords. The structure is listed below:



dayRecords					
month	day	High	Low	Close	Volume

5. Technical Analysis of Stock



The analysis Panel

The class analysisPanel serves between the dummy operating system and the graphpaper which is situated at this panel. When there are events concerning this panel, for example, pressing buttons on the side panel to add a trendline in the graph paper, this class will call for the appropriate method in the class Graphpaper.

Moreover, when the user wants to adds an indicator to analyze the price trend, a dialog will appear to let the user to choose which indicator and its property. The class indicatorSelectionDialog will construct this dialog which contains an array of properties which are defined by the class indicatorProperty. The class indicatorProperty extends focusPanel (extends Panel) provide the ability of letting the users to switch the focus between various properties and buttons by the key 'tab' on the keyboard.

The graph paper

The class Graphpaper is responsible for adding, removing, drawing all the components of the graphpaper and handling the event on it. When it receive an event, it will distribute it to the right component to continue to process the event.

In handling events and drawing components, different components have different co-ordinate systems of different locations and different scaling factors. All these mathematics transformation is done in the class Bound. Once the component is constructed, it must have a object of class Bound adhered to it helping it to do co-ordinates transformation.

Once the graphpaper receives an event like mouseDrag or mouseDown, generally, it will first check the last active component is null or not and the current component under action is null or not to divide it up to 4 cases and then process the event according different conditions.

The graph components
There are many graph components available in this application. Basic components in a graph paper are Pricebars, Volumebars, yAxis, timeAxis, etc. which must be drawn in the graphpaper. Other useful components includes trendlines, curves(shape of an indicator) which will be drawn in the graph paper under user's request.

The class GraphComp defines all common behavior of the graph components. It can handle event which is distributed by the class Graphpaper. How can it determine whether a co-ordinate is inside the component? What should the component react if it is requested to be dragged on the whole, or only one of the nodes of the component? What should it react if the cursor enter the component(may change the shape of the cursor)? All these general behaviors are defined in this class and these methods can be overridden when actual components are defined by extending this abstract class.

Cursors
The class Cursor in this package in fact defines all shapes adhering to the real mouse cursor. It includes footnotes (bubbles) and other forms of cursors.

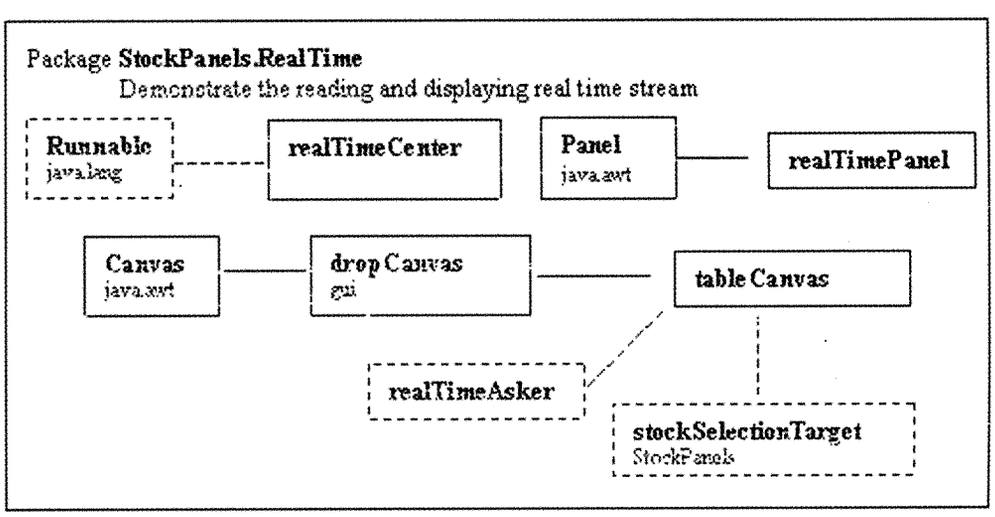
Indicators
Classes like movingAverage defines the algorithm of this indicator and the class indicatorBuilder defines the general behavior of all the indicators.

Bottleneck of Refreshing Speed
In drawing the graphpaper, rapid refresh of screen is a necessity. However, it will be very slow by using double buffering technique. In this application, the characteristics of the graphpaper is that in any time, there is only one component which is moved by the mouse cursor so that we can adopt the technique of drawing in XOR mode which is supported by the current version of Java.

In XOR mode, when any graphic objects are drawn on the same position twice, it will return to the background so that the object originally masked on the background disappears automatically.

However, the XOR mode in drawing text is not supported well in every platform and browser. For example, in a NT 4.0 IE 3.1, the XOR mode in drawing text results in gears shape of the text. While in Netscape, the XOR mode is completely unsupported since when the text is drawn on the same position twice, the text will not disappear as expected.

6. Real time data demonstration

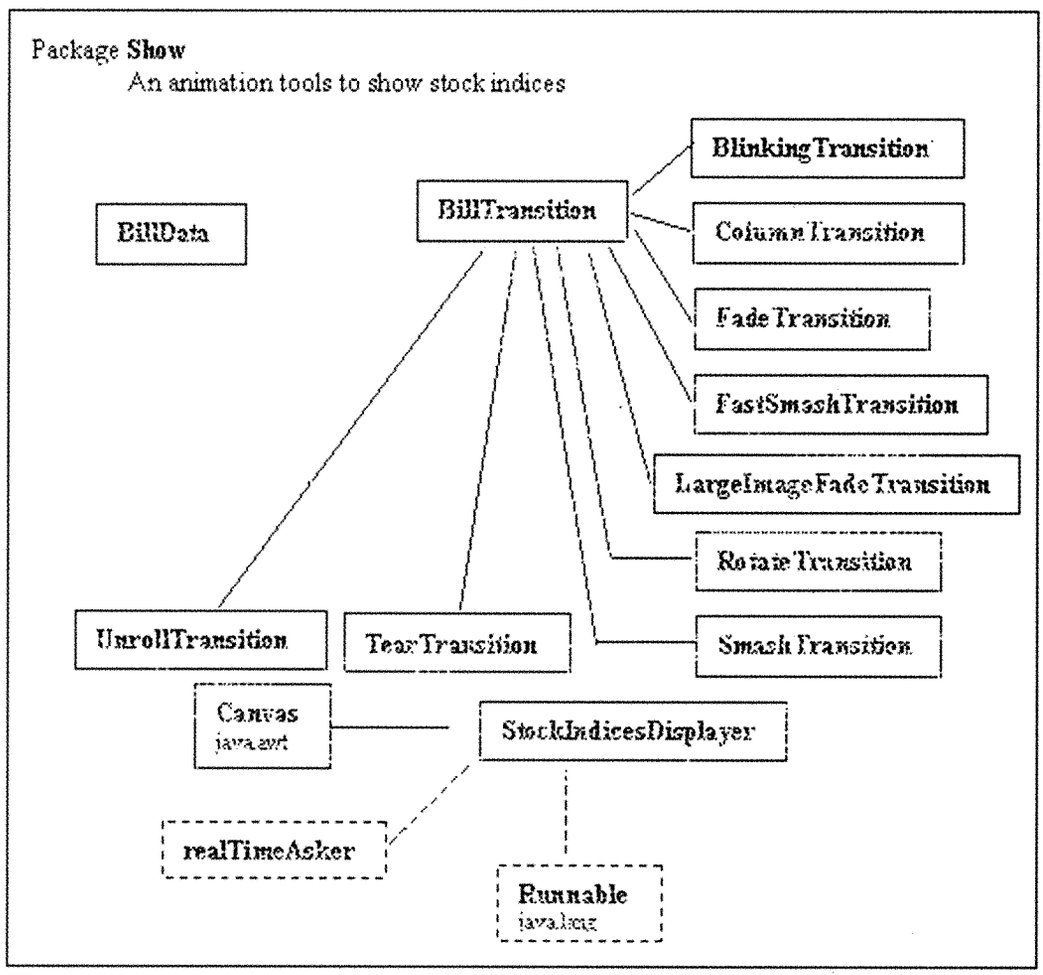


In this package, we demonstrate the reading and displaying of real time index and stock data to users.

realTimeCenter
This class acts as an intra-server within the software so that it recorded down all "realTimeAskers" and when there is real time information arriving by reading the real time stream (Communication.realTimeInputStream) in a parallel thread, it will distribute this information to all realTimeAskers in the form of a packet (Communication.packet) .

tableCanvas
It is a realTimeAsker being added to the realTimeCenter so that every time real time stock data arrives, it will display it on the screen if the users have selected the active stock to be displayed.

Apart from tableCanvas, there is also a dynamic billboard in the realTimePanel so that it is also a realTimeAsker displaying real time index information. This stocks indices displayer is defined in the package show as follows:



StockIndicesDisplayer
This is the canvas showing all the background pictures of the indices and also the real-time data of individual index. The StockIndicesDisplayer actually obtains all the background images of different indices and then store them on different BillData objects. Then, it will pass the BillData of successive images to different kinds of transition classes to generate the array of images, which will appear to have the required transition effect. Finally, the displayer will display the array of images to the screen as well as the value of the indices.

BillData
The BillData class is used to store unique data about individual BillBoards. Generally, it will grab all the pixel of the images and store them into a n x m integer array, where n and m are the width and height of the image respectively.

BillTransition
The BillTransition class is used as a base class for other classes that will create transition images between two BillData image's.

For instance, the BlinkingTransition class changes one image into another by blackening the old image completely from top and bottom to the center part of the image, just like closing the eye. Then, the completely black image is replaced by the pixels in new images, starting from the center and extending to the top and bottom just like opening the eye.

The ColumnTransition class changes one image into another by drawing increasingly larger columns of the new image onto the old image. The column size increases to the left, and the same pixels are always drawn on the left side of each column. This makes the image appear to be sliding in behind the old image.

7. Dummy Operating System

The following classes are written for the dummy operating system for this application. The dummy operating system are designed for the user to get what they want as easy as possible.

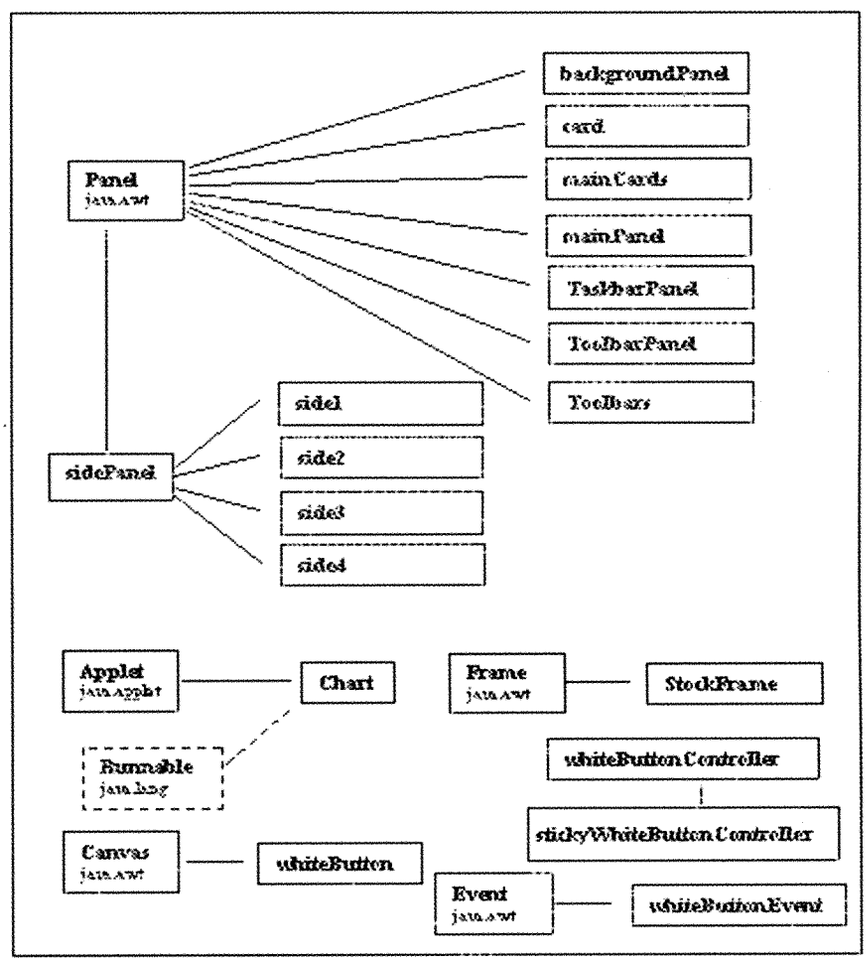


Chart
This is the staring applet of the program. Most of the initialization process will be done here. The user is prompted to wait for the initialization process to complete. This will increase the speed of the program after initialization.

StockFrame
This class defines the main working environment and will be responsible for calling all the functions of the program. The working environment consists of a toolbar to do new job, a side panel which contains all the tools for the specific job, the actual working window and a task-switching bar on the bottom of the frame.

TaskbarPanel
This class defines the task-switching bar for the system. The operating system is a "multi-tasking" system, which allows users to do several function of our program at the same time. The taskbar on the bottom have the same function as Microsoft Windows 95's Taskbar, which allows users to switch application by pressing the corresponding button for the application. For instance, one may read the real time price of the stock and at the same time analyzing several different types of stocks and search for the past data of the stocks. The users can simply switch to different windows by the taskbarbutton immediately without initializing the window again.

mainPanel, mainCards, cards
These three classes are responsible for the multi-tasking working environment for the program. The system contains four types of jobs, namely, *real-time price reporting, technical analysis, past data searching* and *help system*. Each kind of jobs will have their own cards object. In each kind of cards object, there is a mainCards object. This object contains a dynamic card array, which will expand and contract according to the user's action. Once the user prompts the program to start a new jobs, the mainCards object for that particular job will expand the array. Switching between jobs will only require to search for the required job in the array.

8. Past Data Searching

Old Data Panels
package StockPanels.DailyReport

It contains three main panels, for calendar, full report and top ten report.

1. Calendar

Class dailyReportPanel is a link between calendarPanel and main program. Class calendarPanel is used to draw the calendar which can show the year 0000 to 9999. The mouse action can choose the report of the particular date. If the data doesn't exist, an error message will come out. Otherwise, the report will be shown by calling stockTableCaller.

2. Full report

Class stockTableCaller is a link between reportScroller and main program. It can exchange the information between them. Class reportScroller draw the full report of stock of the particular date. The buttons of the fields can do sorting in ascending or descending order. The text field at the bottom can enable the search function. These functions are embedded in the class reportScroller.

There are two modes in the full report.

1. All records --- show all the records of the date
2. Filtered records ----- show the selected record by query method.

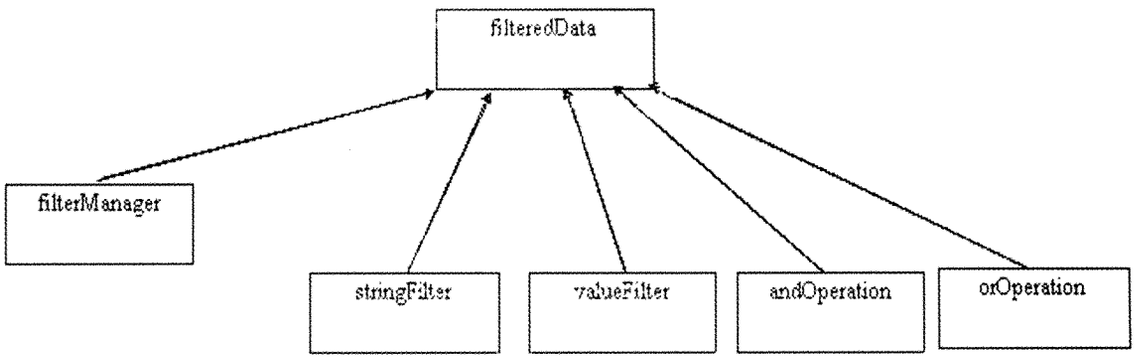
3. Top-ten report

Class statisticReportPanel is a link between statisticReport and main program. When the button in sidePanels is pressed, the tenReportDialog will show the dialog which can choose the type of top-ten report. There are total four kinds ----- *top-ten gainers, top-ten losers, top-ten active (volume)* and *top-ten active (turnover)*. After choosing, it calls statisticReportPanel and then statisticReport to display the corresponding top-ten report.

Note that there is a class reportScrollerClient. It is used to create a link between full report and top-ten report. It is necessary because we need to know some information, for example, Date, in the current selected full report for creating the top-ten report.

Query Method
package StockPanels

This package is a tool which is useful for the main panels of the programs (i.e. analysis, realtime and old data panels) . The query method is one of those which defines a query method in an object-oriented way.



The classes filterManager, stringFilter, valueFilter, andOperation and orOperation are the implementation of filteredData.

Class

filteredData: This acts as an interface in the package. There are two data/method needed to communicate between other classes. They are the resultant records and the number of selected records.

Method : stockRecord[] getFilteredRecord() to get the resultant records int getselectedRecordNumber() to get the number of selected records

stringFilter: To filter the records for the field where the type is string, that is, NAME. Only the records started with the entered string are selected.

valueFilter: To filter the records for the fields where the type is value, that is, CODE, OPEN, LOW, CLOSE, HIGH, VOLUME, TURNOVER. Only the records between the minimum and maximum are selected.

andOperation: It contains a method (called andMethod()) to operate an "AND" criteria on the input data.

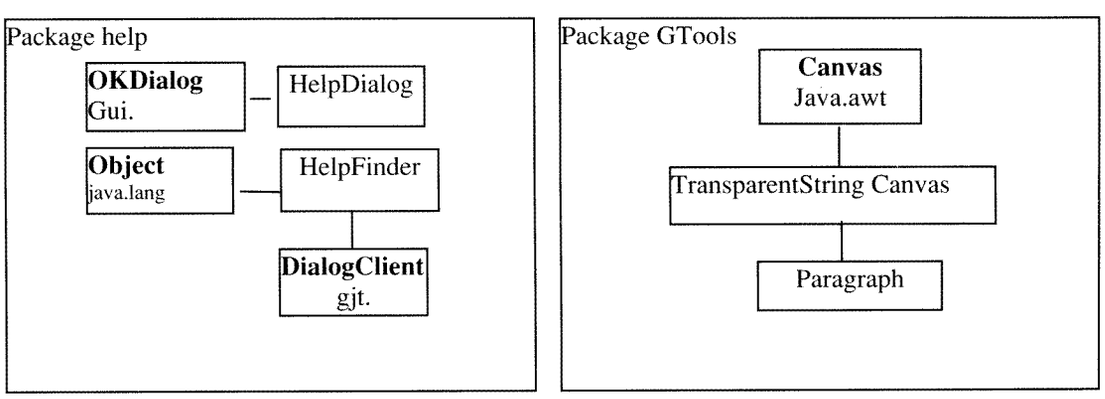
orOperation: It contains a method (called orMethod()) to operate a "OR" criteria on the input data.

filteredManager: It is used to establish a relationship between four fundamental methods, that is, stringFilter, valueFilter, andOperation and orOperation)

9. Help System

In order to make the system more user friendly, we have tried to provide the help system for the program. There are two types of help system in this program: *help dialog* and *help panel*.

The help dialog system allows the user to press F1 key to find out brief introduction about the buttons they pointed. It is packaged as Help.



HelpDialog

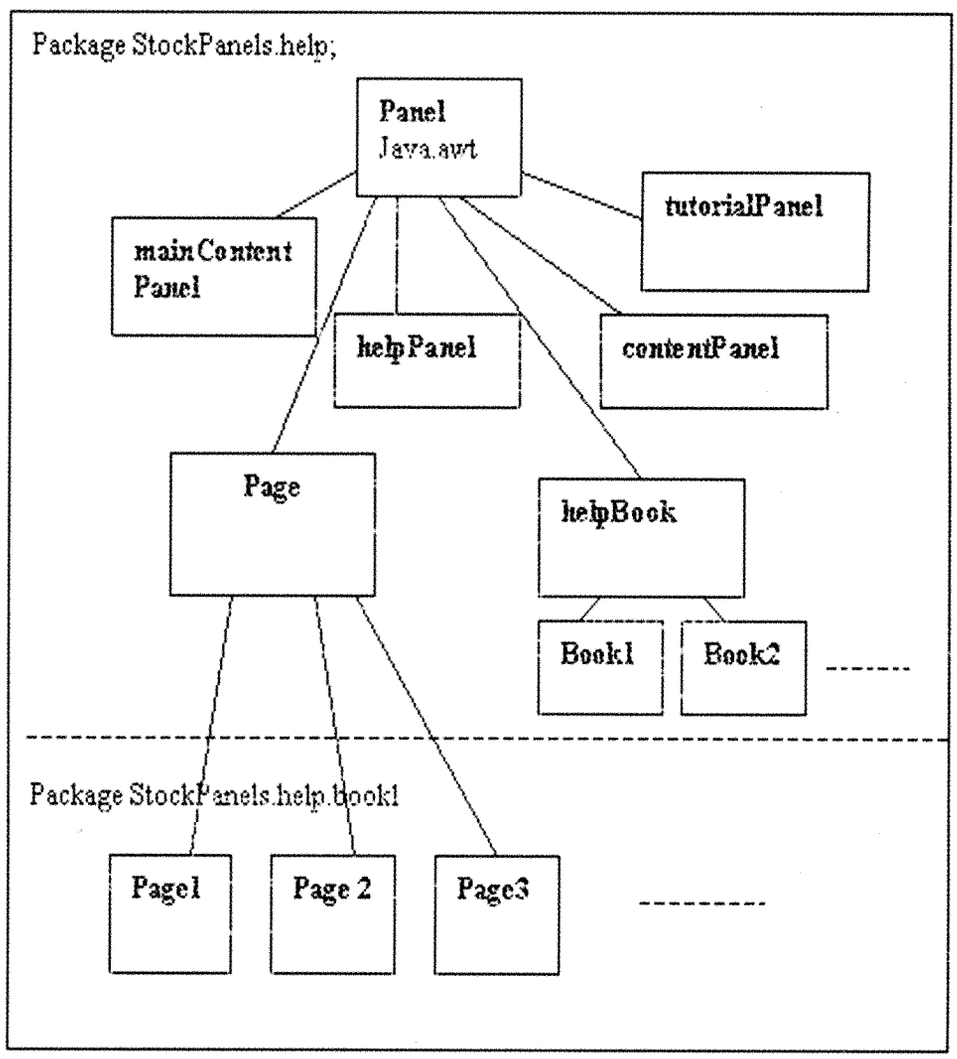
A text dialog with a OK button to close the dialog. This dialog will show the help messages in a Paragraph object when F1 key is pressed.

HelpFinder
An object which are responsible for choosing the correct help message by the position of the mouse pointer.

Paragraph
A canvas, which can automatically arrange a long sentence to a paragraph of specified width. This is very useful when the help message is very long and so there is no need to write a new specified class for the layout of the text on the help dialog.

TransparentStringCanvas
This is the parent class for the paragraph. This class allows the string to be draw on a panel with the desired size and the same background color or images.

The second type of the help system is the help panel for the program. This part of the program has not yet implemented completely. However, the main structure of the program is designed and some of the classes have been written.



The main idea for the help panel is to provide a browser for reading the technical analysis tutorial and the program function tutorial easily.

Every tutorial is considered a helpbook, say book1, it will contain several pages, say page 1, 2, 3. So there will be classes containing the content of each page like Page1, Page2 and Page3.class in the StockPanels.help.book1 package.

4. Development Support Utilities

Apart from the software mentioned above, there are also some classes which are not listed above in developing the whole system. They are mainly the data base development software including two classes for getting the daily stock information from the internet in html format, two classes for converting these html reports into stock report (.dsr) and index report (.dir) files mentioned before and one class for collecting all relevant data of a specific stock to form a history file (.tic) .

In addition, we have used the Graphical Java ToolKits from a book called graphic JAVA Mastering the AWT. This library is packaged as gjt and gjt.images in our application. The function for gjt provides some enhanced component from the Java.awt package, for example, ImageButton and Dialog , etc.

5. Future Development

Two future expansions will be possible:

Apart from viewing daily stock report, a similar function of viewing daily index report can be easily added since the file format of daily index report (.dir) has already been established.

When a real time telephone line is used instead of a simulated one, the software can connect to msql to store the intra-day stock data by a package already written, that is package StreamToMsql (not listed above).

Further expansions in the function are suggested.

Develop a user-login system so that user could have their own preferences on the system.

Calculate the profit for the user from the stocks on hand.

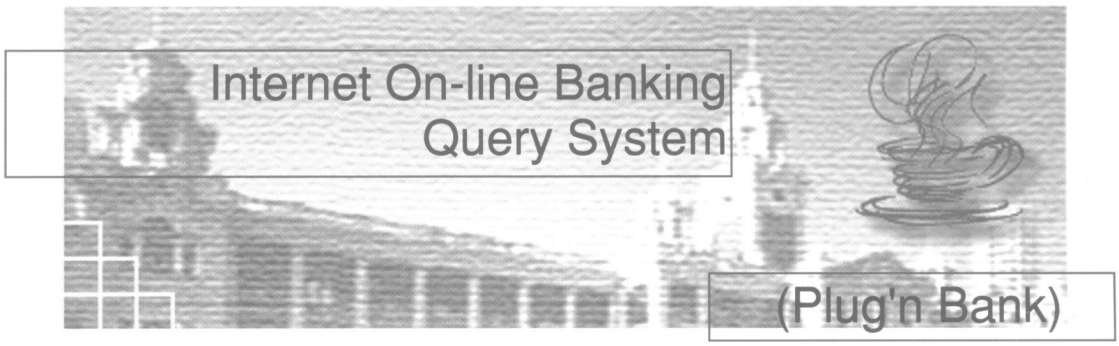
Technical Analyzing wizard

Conclusion

In this paper, we have described the implementation of a real-time price reporting and analysis system over the Internet for Hong Kong Stock Market. The system is highly flexible and extendible as the overall design is platform independent and object-oriented.

Further expansion of the program may include user's profile, profit calculation for individual users and stock trend analysis wizard.

The performance of the system is very encouraging. This shows the great potential for the Java-based application on the Internet.



Authors

Lee Tat Yi
Mok Siu Kwong
Yung Chong Wah

Abstract

We describe in this report the implementation of an internet-based banking query system. The system retrieves information from an on-line database and presents banking information in a perceptive manner. Along with the data fetching from the database, socket connection and protocol implementation are required. Some security measures are also taken in the implementation of socket object.

Introduction

An on-line banking query system allows a user to retrieve his most updated account balance and query the general banking information and client's own information on-line. Different from the traditional bank statement sent by the bank monthly, the system displays data in an easy-to-understand and easy-to-check manner without tedious and messy references. Also, in the information query, the user is able to get a mental picture perceptively without looking at lengthy descriptions. Most importantly, the system *must* take some security measures to avoid possible abuses.

With the explosive growth of the Internet technologies, shopping through the Internet is already nothing new at all in many developed countries. Certainly, banking on the net will be the next in the near future. Since the Java language is becoming more mature, and along with its attractive *portable* and *object-oriented* nature, implementing the system by Java can highly ease the jobs of maintenance and system-upgrade. Also, the system can be made more interactive to meet clients' requirements.

The purpose of this project is to develop a client-server banking query system in Java language. The client data is stored in a Mini SQL database. The Mini SQL is a shareware and is a subset of SQL. That means it works in about the same way as SQL except not that powerful. The system retrieves data from the database when necessary in a safety manner and displays them to the user. Also, we have demonstrated what the information query will look like.

1. The Balance Checker

For users who have successfully registered can check their *most updated* account(s) balance, credit card/ATM card debit status and purchased services (for example, travel insurance, life insurance) status. It should be noted that the system is essentially placed at a Java-enabled web page.

Whenever a user goes into the system, username and password will have to be keyed in first as in Figure 1.

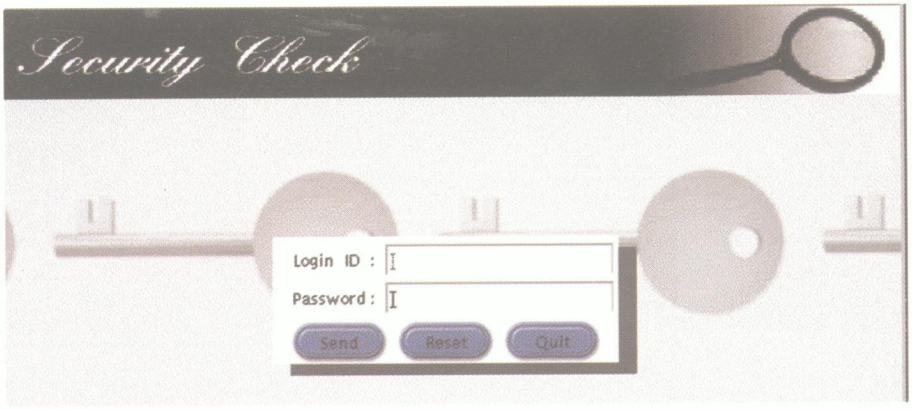


Figure 1. The password key in layout

When the user successfully get in, the window as shown in Figure 2 will appear. In this window, there are three *boxes* showing the *user last login date and time*, the *last unsuccessful login date and time* and the *number of trials of unsuccessful login* at the top. Obviously, the user can look at the *last login date and time* to find out if there have been any unauthorised person breaking into his account. That is, the user can check if his password is safe enough. The *last unsuccessful login date and time* and the *number of trials of unsuccessful login* can also have the same function.

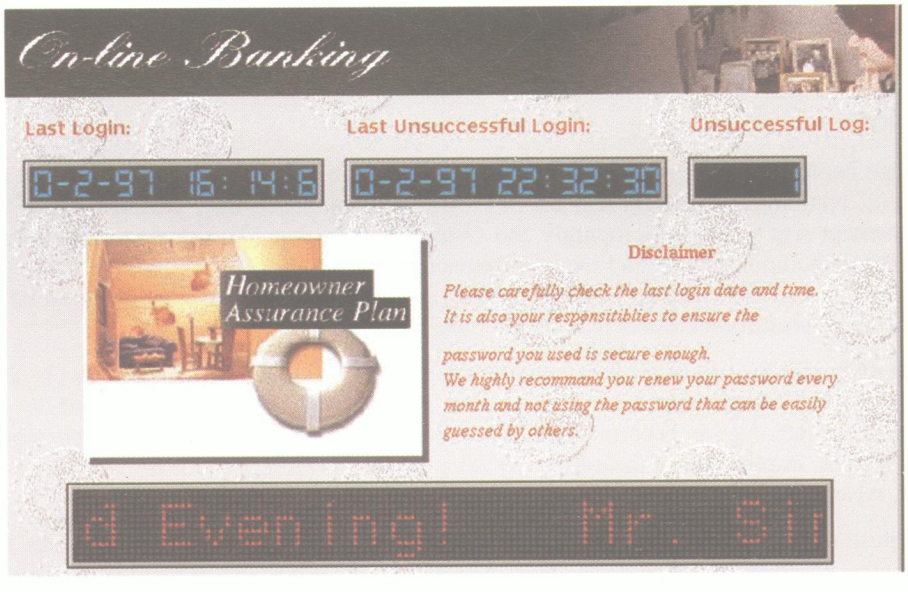


Figure 2. The information page layout

Moreover, there is an advertisement slide-show in the middle of the page. The pictures, which are in bitmap, can be easily renewed by the bank. The system administrator only has to put the new pictures at the assigned directories *(the image directory where java programs are placed)* with the assigned names *(name started with adv, for example, adv1.jpg)*.

At the bottom, the LED panel will display information, such as the weather and the Heng Seng index. Actually, the information displaying is not from data feed on-line. However, it is possible for our project to link with the *data feed* in this part.

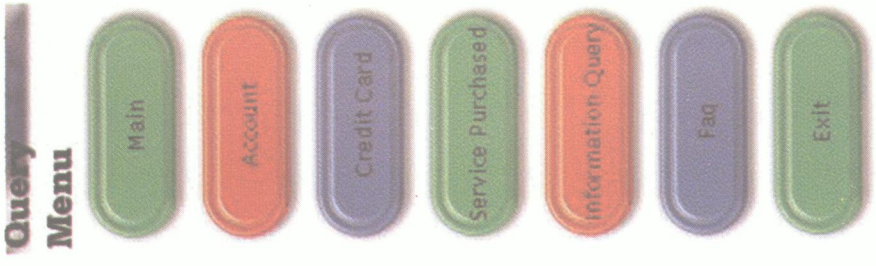


Figure 3. The button panel

A button panel will also be shown up as in Figure 3. The user can check whatever information just by clicking the corresponding button. If the user clicks the *Account* button, the *account balance checking* page will be brought up.

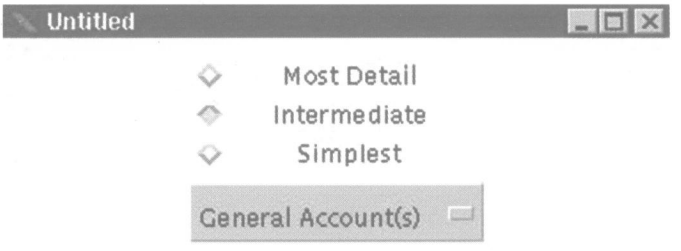


Figure 4. The panel box

In the *account balance checking* page, the table in the middle will first shows the general information of *all* account(s) of the user. This table only includes the *account id*, *account name*, and the most updated *balance* of every account. If the user wants to look at the detailed information of one of the accounts, he will have to click at the panel box as shown in Figure 4. Noted that the checkbox of this panel box will only have the name(s) of the account that the user possessed. The user can also choose *the level of detail* from the checkbox : *Simplest*, *Intermediate*, and the *Most Detail*. Figure 5 depicts the *Current Account* in *Intermediate* level of detail. At the bottom of the page, there are three boxes, which show the *balance*, *interest rate* and the *Account number* in a perceptive manner. The purpose of these boxes is to show the information that the user usually look for so that the user need not do any tedious information searching. Of course, the information shown in these boxes can be changed easily by the bank so as to meet the clients' requirements.

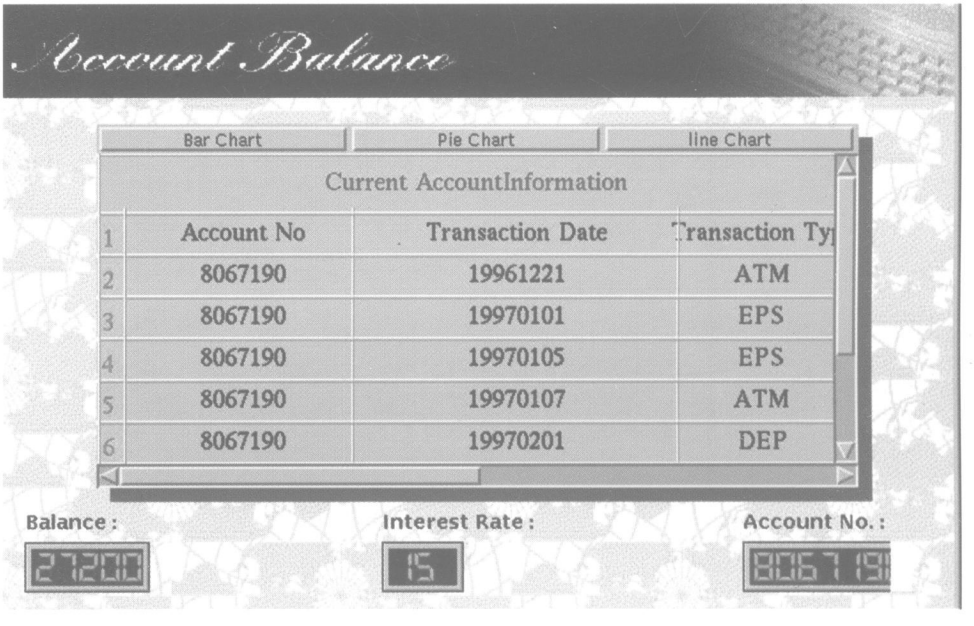


Figure 5. The current account checking page

In the table, there are three buttons: *Bar Chart, Pie Chart and Line Chart*. The user can select a column from the table and simply click on the button to have the data plotted on whatever chart he likes. It should be noted that the clicking on any cell of a column can already make the whole column selected. The selected column will be highlighted.

The user can also check his card(s) balance and service purchased just by clicking the *Card* and *Service Purchased* buttons respectively. Figure 6 shows the layout of the *card balance checking* page. This is the same as the *account checking page* except the three bottom boxes instead display *debit amount*, *maximum possible debit* and *card number*.

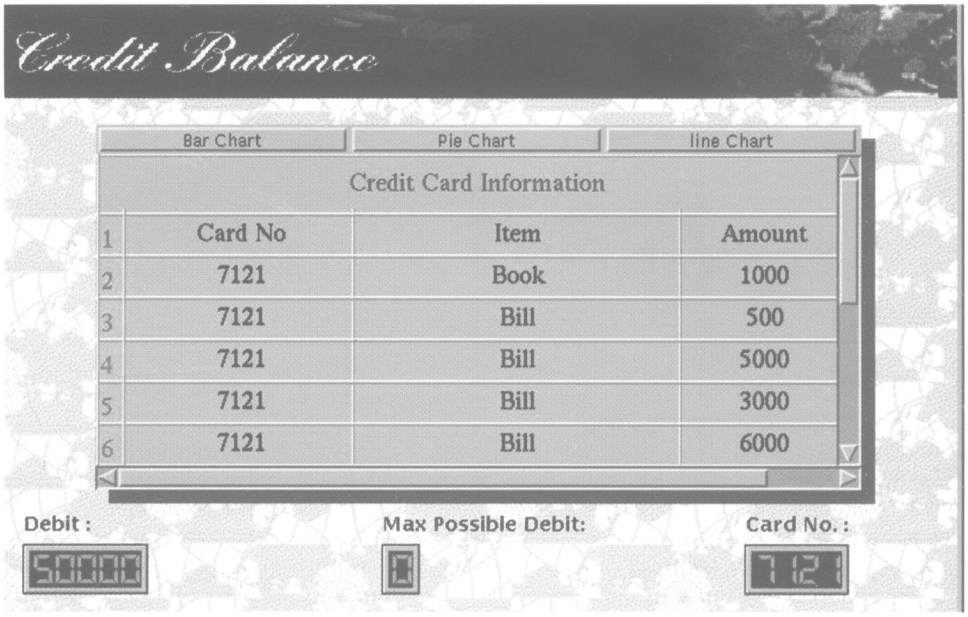


Figure 6. The credit card checking page

2. The Client/Server Communication

The Establishment of Socket Connection

In the communication between the client and the server, ***socket*** is used. It is chosen for its reliability in data stream network connection compared with datagram protocol *(datagrams are not guaranteed reaching their destination).*

The server object, codename *BankServer*, is a multithreaded object running on the server *(indy14.eee.hku.hk)* where the homepage is placed *(otherwise, there will be a security error).* Whenever it runs, the port 8888 is listening. When a client connects to the BankServer, the ServerSocket class will allocate a new Socket object for the client. Then, the server goes back to the listening on the ServerSocket for additional client connections.

For the ease of system administration, a window will be shown in the server side whenever the system is running.

Refer to Figure 7, the number of clients in connection, the client IP address, their connection dates and time, and their requests are listed.

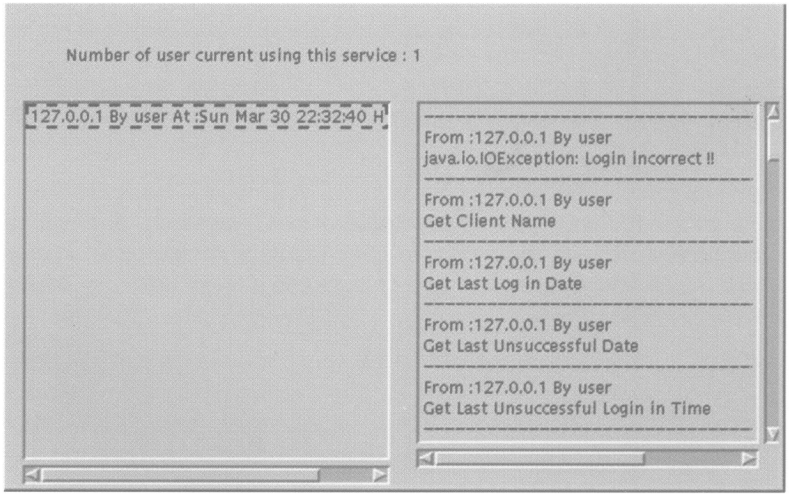


Figure 7 The Server Monitoring Window

The Protocol Implemented

For communicating between the server-sided programs and the client-sided programs, **data packets** and **protocol command words** (Control Words) have to be designed.

There are four types of data packets implemented:

(i) The login packet

Control	Login name length	Login name	Password length	Password

The Control refers to the Control words (Protocol command words) and is shown in table below.

(ii) The information packet

Data

(iii) The validation packet

Control

(iv) The data request packet

Control	Account ID

The Account ID is fetched from the database at the beginning. In some cases (for example, get all the account information), the Account ID field can be omitted.

We have also implemented a set of protocol command words (Control Words) as shown in the following table.

Physical Meaning		Byte Pattern
From Server to Client	From Client to Server	
	Sending username and password	000
Password Validation Pass		100
Password Validation Fail		200
	Get all account information	010
	Get specific account information	011
	Get specific account interest rate	012
	Get specific account balance	013
	Get all account number(s)	015
	Get all card information	020
	Get specific card information	021
	Get specific card debit	022
	Get specific card max debit limit	023
	Get all card number(s)	025
	Get all purchased services information	030

Table 1 The protocol command words

Noted that all of the command words are three bytes long. The actual command reference can be found in the program comments.

Actually, the above *Control Words* are only useful at the beginning stage. When the user comes to the system homepage, having entered his login name and password, the *login packet* is used to send the login name and password to the server program for checking. After checking the password, the server program will send back a *validation packet* to the client program. In the meantime, if the password validation passes, the socket remains opened and the server program will update the *latest successful login date and time* in the database; otherwise, the socket closes and the server program will update the *latest unsuccessful login date and time* and the *number of unsuccessful login trials* in the database.

In the event that the user successfully gets into the system and requests information, for example, the balance of his current saving account, a *data request packet* will be sent from the client machine to the server. Then, the data required will be fetched from the database and transmitted to the client machine by using the *information packet*.

The Security Measures Implementation

In order to ensure the system is running smoothly even under malicious operation, several necessary measures has been taken. These include:

- time-out checking;
- username and password length checking;
- unexpected data retrieval checking; and
- maximum client access limitation

The **time-out checking** refers to the checking of the idling time of every socket by the *server-side programs* from time to time. When a client who has left the socket unattended for 900 seconds, that is, no request from the client, the socket will close. This can avoid misuse by others due to the previous user leaving without logout.

Since the system has to create memory block to hold the username and password, the **username and password length checking** becomes vital to prevent crackers sending *very lengthy* username and password to crash the system.

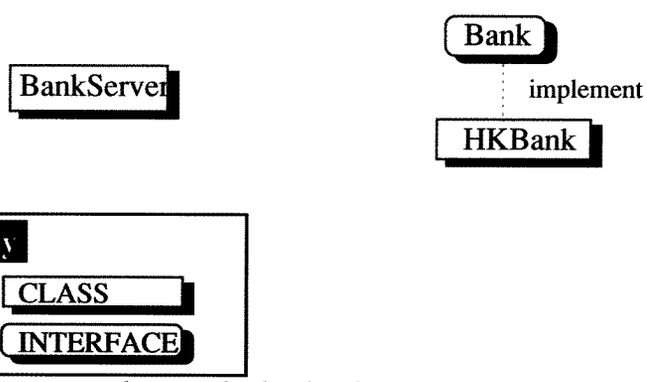
Moreover, as the client program may be modified by hackers to **retrieve unauthorised data**, the server-side program has to check if the data request from the client is unexpected. If so, the server disconnects as soon as possible. Also, the system checks whether the account number retrieved is owned by the user before sending data from the server.

In order to ensure the system is running at its optimum point, the **number of clients accessing simultaneously** has to be limited. The system administrator can decide the maximum number of clients access based on the system resources and the available network ports. The default limit is set to 30 clients and can be easily changed if necessary.

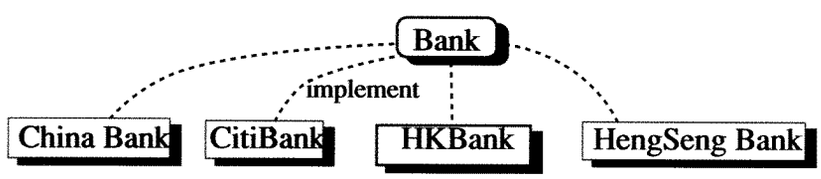
The Object-oriented Architecture

Because of the object-oriented nature of java programming, it is very easy for this system to implement new connection protocol without modifying any code in the server programs. Even though a new bank using totally different database comes, there will be no problem at all.

Basically, in the server side, there are two *java objects* and one *java interface*. The *BankServer* object is only for the socket connection. *Bank* is a java interface and *HKBank* is for the connection to database and retrieval of data from the database. The logical diagram of the server side programs are shown below:



In the case there are more than one bank using the system, say HongKong Bank, CitiBank, Heng Seng Bank and the Bank of China. The object diagram will become:



3. The Database Organisation

The database architecture can be hierarchically represented by a level logical view as in Figure 8.

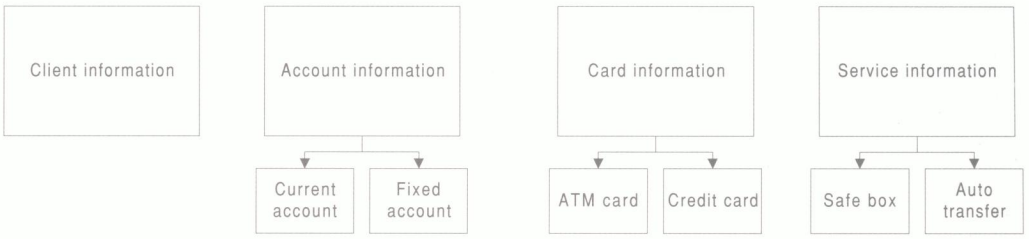


Figure 8. The logical view of database

Obviously, it is a two-level hierarchy. The first level tables contain a list of msql statements for fetching data from the second level tables. The physical view of the tables is shown in Appendix I.

4. The On-line Mortgage Loan Calculator

The Functions of the calculator

Since today's mortgage loan market shares a significant proportion in a bank profit earning, and clients always want to know the period and amount he has to pay without waiting for tedious computation, the *on-line mortgage loan calculator* is designed to meet the needs of both parties.

From this *on-line mortgage loan calculator*, the user only have to enter his *loan amount*, *salary*, *marginal saving* and *autonomous expenditure*, then the **monthly mortgage repayment** will be computed at once. Besides, the user can choose different mortgage loan scheme from the checkbox *Repayment Scheme* as shown in Figure 9 in order to decide which scheme is better.

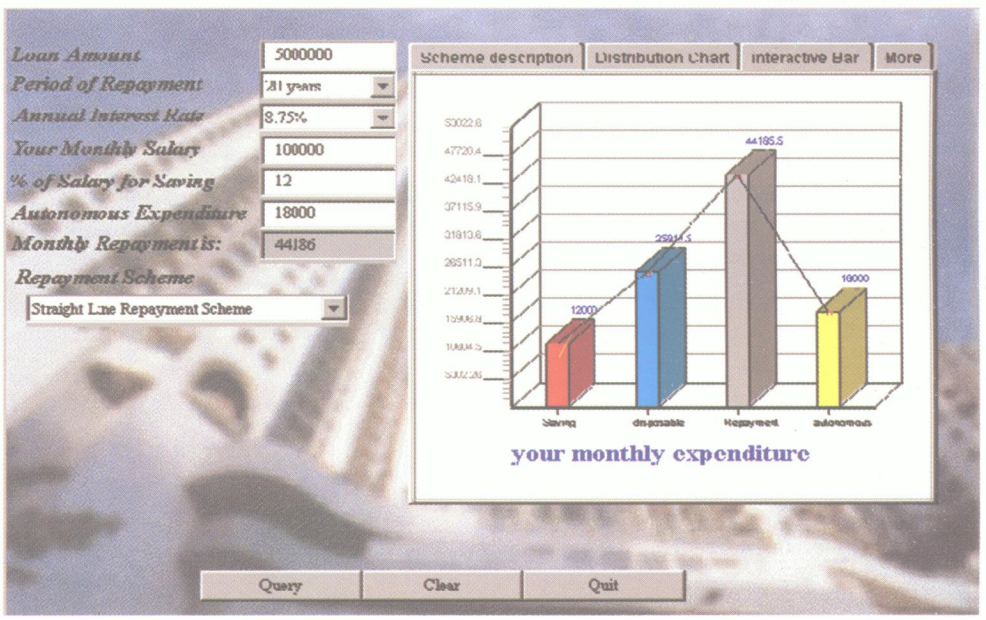


Figure 9. The mortgage loan calculator with bar chart

Having computed the monthly repayment, the user can choose to show the bar chart from the panel. The bar chart is for comparing the amount between the user's items of expenditures and saving. Actually, the bars can be dragged upward and downward for the user to see the effects if one of the amount of the items changes.

For example, if a user finds the period of repayment being too long, and he wants to see what will be the result if he saves less and pays more monthly, he can drag the bar entitled *saving* downward. Then, the bars entitled *autonomous expenditure, disposable expenditure* and *repayment* will change accordingly.

In this case, the user can see the changes immediately. This can help the user in planning what he should do. The user can also find out the maximum amount of repayment and loan he can afford.

Noted that whenever the salary of a user is too low to afford the loan, the calculator will notify him.

Furthermore, the user can get the idea of the relative composition of his *expenditures*, *savings* and *repayment* easily by clicking the *Distributed Chart* from the panel to show the bar chart as in Figure 10.

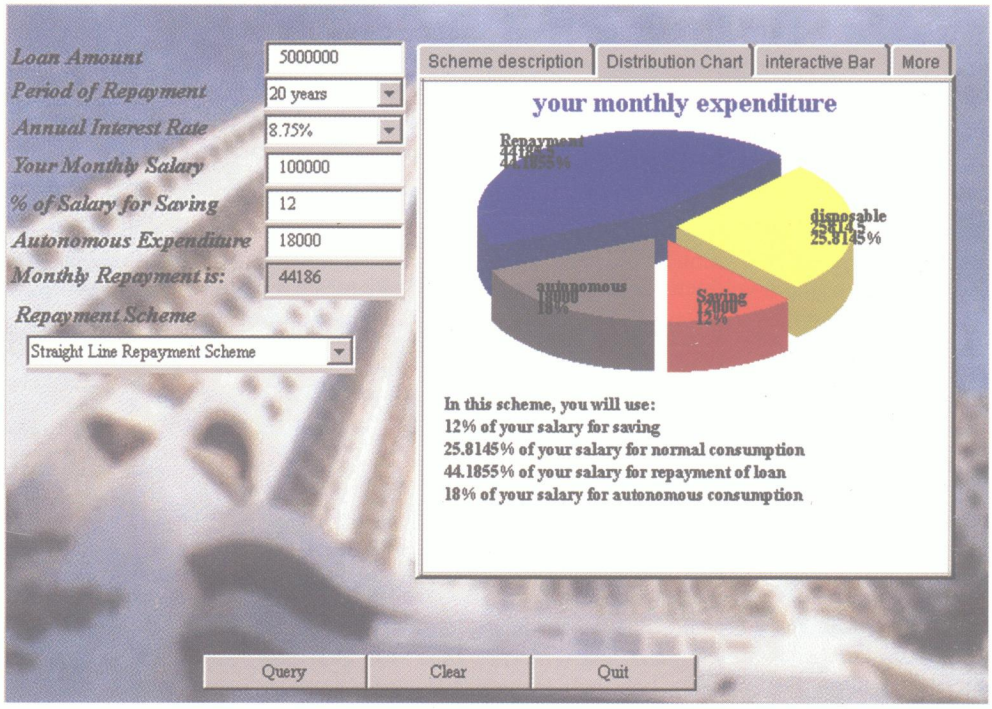


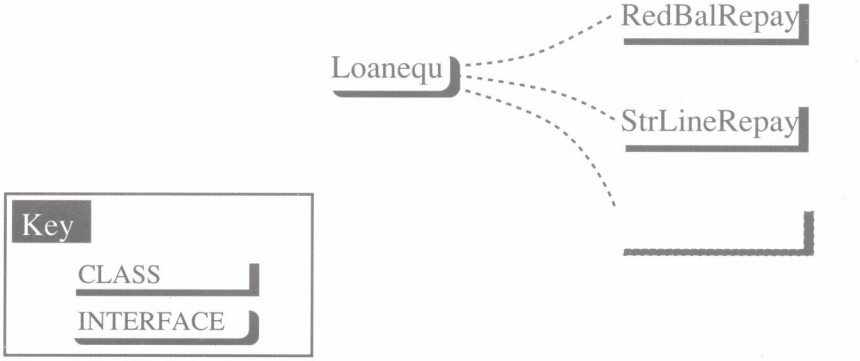
Figure 10. The mortgage loan calculator with bar chart

Further Implementation

Since today's banking industry is so competitive, banking services has become so wide ranging that are sometimes even beyond imagination. Calculators for implementing other banking services similar to the *on-line mortgage loan calculator* in nature may be needed.

Moreover, new schemes of mortgage repayment may come to the market from time to time. That means the way of calculation may have to change quite often. In traditional programming architecture, such kind of changes may cause serious problems in coding.

However, as this *calculator* is implemented by Java, its *object-oriented* nature actually makes changes easy. This can be found out by taking a look at the object diagram of this *calculator*.



Here, the *reduced balance repayment scheme (RedBalRepay)* and the *straight line repayment scheme (StrLineRepay)* have been implemented under the *interface* of *Loanequ*. Whenever there is a new scheme, we only have to implement it under the *interface* of *Loanequ*. The existing java programs can be left unchanged.

Conclusion

In this project, we have implemented a banking query system on a Javatized web-page. The design is actually very flexible due to the *object-oriented* nature of Java. We have changed the graphical interface twice and there is *no* code-amendment requirement at all. What we have to do is only to write a new graphical interfacing object.

For the using of socket connection for accessing database, although it is a little bit slow, it is more reliable. Also, besides the security means that shipped with Java 1.1, some supplementary security measures can be implemented easily when socket is used.

The mortgage loan calculator demonstrates the difference between the traditional tedious information query and the Javatized *interactive* and *perceptive* query. We intend to construct more objects like this. However, owing to limited time, we only implemented the *mortgage loan calculator* completely.

Appendix I

The physical view of the tables in database are shown as follows:

User table

Field name	Value
Surname	Chan
Sex_status	M
Given_name	Siu Ming
Uid	95177
Passwd	12345678
Tel	25469089
Email	smchan@eee.hku.hk
Address	5th Floor, Hong Kong Road
Login_name	User
Unsuccess_login	
Unsuccess_date	

Account Method table

Field name	Value
Acct_n	Current Account
GetAllAcctInfoF	select acct_name, ac_no, balance, interest, imethod from cacct_fn
GetAllAcctInfo	select cacct_ai.acct_n, cacct_uif.ac_no, cacct_uif.cb, cacct_ai.interest, cacct_ai.imethod from cacct_uif,cacct_ai where cacct_uif.uid=
GetAcctInfoMaxF	select ac_no, trans_id, trans_date, trans_type, amount, balance from cacct_fn
GetAcctInfoMax	select ac_no, trans_id, trans_date, trans_type, amount, balance from cacct_pastr where ac_no=
GetAcctInfoNormF	select ac_no, trans_date, trans_type, amount, balance from cacct_fn
GetAcctInfoNorm	select ac_no, trans_date, trans_type, amount, balance from cacct_pastr where ac_no=
GetAcctInfoMinf	select ac_no, trans_date, amount, balance from cacct_fn
GetAcctInfoMin	select ac_no, trans_date, amount, balance from cacct_pastr where ac_no=
GetAcctInterest	select interest from cacct_ai
GetAcctBalance	select cb from cacct_uif where ac_no=

Card Method table

Field Name	Value
Card_n	ATM Card
GetAllCardInfoF	select card_no, pri_ac from atm_user_fn
GetAllCardInfo	select card_no, pri_ac from atm_user where uid=
GetCardInfoMaxF	select card_no, pri_ac, s_ac1, s_ac2, o_date from atm_user_fn
GetCardInfoMax	select card_no, pri_ac, subsidiary_ac1, subsidiary_ac2, o_date from atm_user where card_no=
GetCardInfoNormF	select card_no, pri_ac, o_date from atm_user_fn
GetCardInfoNorm	select card_no, pri_ac, o_date from atm_user where card_no=
GetCardInfoMinF	select card_no, pri_ac from atm_user_fn
GetCardInfoMin	select card_no, pri_ac from atm_user where card_no=
GetCardDebt	
GetCArdMaxDebt	

Service Method Table

Field Name	Value
Serv_n	Safe Deposit
GetAllServInfoF	select box_no, o_date, box_type, branch, charge from sd_user_fn
GetAllServInfo	select sd_user.box_no, sd_user.o_date, sd_user.box_type, sd_user.branch, sd_box_info.charge from sd_user,sd_box_info where sd_user.box_type=sd_box_info.box_type and sd_user.uid=

Account tables

-user table

Field Name	Value
Uid	95177
Open_date	19970303
Cb	50000
Ac_no	423752189
Summary	

-pass records

Field Name	Value
Uid	95177
Ac_no	423752189
Trans_id	8273
Trans_date	19970304
Trans_type	ATM
Amount	1000
Balance	50000

-interest rate and methods

Field Name	Value
Acct_n	95177
Interest	0.15
Imethod	Simple interest

-field names

Field Name	Value
Open_date	Account Open Date
C_balance	Current Balance
Ac_no	Account no.
Trans_id	Transaction id
Trans_date	Transaction date
Trans_type	Transaction type
Amount	Amount
Balance	Balance
Interest	Interest Rate
Acct_name	Account Name
Imethod	Interest Method

Authors

Lai Ho Chong
Tsang Siu Kin
Wong Hoi Cheung
Wu Yik Chung

Abstract

Hong Kong is an international famous metropolis. It is not surprising to see that the city is planned considerately and thoroughly. One good example is the public transportation network in Hong Kong. In only a tiny area of land, the transportation system is so complex and dense that it not only allows us to travel around Hong Kong easily and conveniently but also confusing us. The purpose of our program aims at advising the end user to travel around Hong Kong Island by bus with the shortest distance and cheapest fare when connecting to our homepage through the world wide web at any time in any place.

Introduction

The program is mainly written in a form of Java Applet and also with the accompany of HTML, so it can be accessed by using any web browser which supports Java, like Internet Explorer version 3.0 or Netscape Navigator version 3.0. Our program is mainly divided into 3 modules (as shown in Figure 1). One is the *user interface module* and it provides a user friendly graphical environment for enquiry. The second is the *MSQL database module* and it is for the transfer of data from the MSQL database within the program. The final and the most important one is the *algorithm module*. It is this module that calculate the shortest and cheapest route for you. This algorithm can support the case of changing buses between the source and destination in the intermediate point and theoretically it can support infinite changing of bus.

As there are so many bus routes and bus stops in just a small Hong Kong Island, it is nearly impossible and also impractical to include all the bus stops. As a result we introduced the concept of transit points. A transit point is actually a main place or landmark on Hong Kong Island and all the bus around that region will be considered to have included in that transit point. As a result, a transit point contains a large collection of buses. All the operations of the program are based on these transit points.

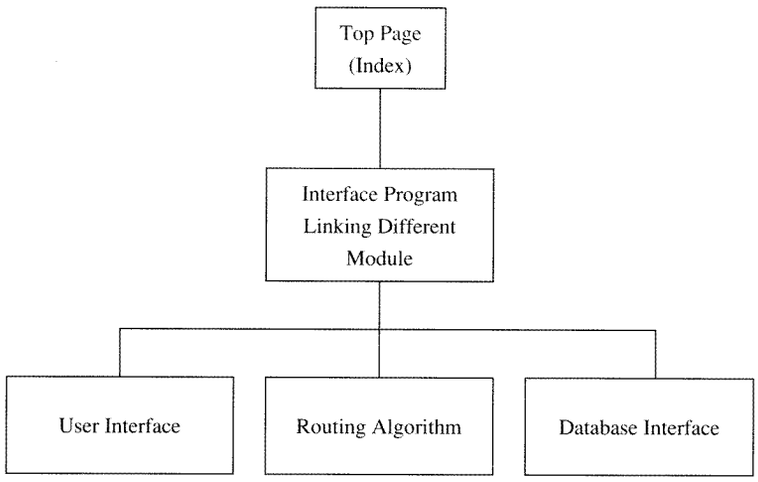


Figure 1. Main modules of the program

1. Philosophy of the Routing Algorithm

In the routing Algorithm, the concept of Transit point was employed. A transit point is a small area of location in which it contains one or more bus stops which are close to each other.

In each transit point, all the buses which passes through that transit point are associated. The bus numbers together with other information of that buses (for example, fares, direction, air-conditioning, etc.) are stored in the database. Detailed description of the database will be included in the database section.

Having selected the source and the destination points, the Routing Algorithm will find the shortest and the cheapest bus route for the users. But what is meant by the shortest and the cheapest? The following has the explanations.

If the user clicks on points that allow buses going from the source to the destination directly, and if more than one bus route can do the same job, then the algorithm will find the bus with the lowest fare. The bus number, the fare and whether the bus is air-conditioned will be shown as the result. If the bus number selected have both air-conditioned and without air-conditioned bus running together, the algorithm will show both type of fares.

If the user clicks on points that do not have direct bus going from the source to the destination, then the algorithm will find a transit point that can act as an intermediate point with buses going from the source point to the intermediate point and from the intermediate point to the destination point.

However, there may be more than one transit point which can act as intermediate points. The algorithm will calculate the total distance between the source point and the intermediate point and between the intermediate point and the destination point for each potential intermediate point. Then the intermediate point with the shortest total distance will be selected.

From the source point to the intermediate point, there may be more than one bus linking two points, the algorithm will find the bus with cheapest fare and the same will be done for the intermediate point to the destination point such that the total fare of the whole route is the lowest.

By Selecting the bus routes, the bus numbers, fares and whether the bus is air-conditioned will be shown as the result. As in the first case, if the bus number selected have both air-conditioned and without air-conditioned bus running together, the algorithm will show both types of fares.

Here, we assume that one intermediate point is good enough since almost any two points in Hong Kong Island can be reached by a bus route involving only one intermediate point.

Actually the algorithm can be extended to three, four or even infinite intermediate points, but for the determination of a intermediate point, the process involves a large volume of data to be transferred from the database to the user machine. So for routes involving two intermediate points, the time needed for calculation will be unacceptably long and therefore the algorithm is only written for one intermediate point.

2. Structure of the Database

The database consists of two tables, *Node* and *NODE_INFO*.

For the table Node, as illustrated in Table 1 is a small part of the table, the first column is the name of a transit point, the second the bus number, the third the bus fare, the fourth the direction of the bus in the East or West direction and the fifth the direction of the bus in the North or South direction, the sixth column the state of air-conditioning of the bus and the last the bus terminal.

Location	Bus_No	Fare	Direction1	Direction2	Air_conditioned	Terminal
HKU	3	3.8	E		No	Central
HKU	3	3.8	W		No	Pokfield Road
HKU	4	3.5	E	N	No	Central
HKU	4	3.8	W	S	No	Wah Fu
HKU	23	4.3	E		No	North Point
HKU	23	3.8	W		No	Pokfield Road
HKU	23	5.8	E		Yes	North Point
HKU	23	5	W		Yes	Pokfield Road
HKU	43	3.5	E	N	No	Admiralty-MTR
HKU	43	3.5	W	S	No	Wah Kwai
HKU	91	3.5	E	N	No	Central
HKU	91	3.8	W	S	No	Ap Lei Chau Est
HKU	91	5	E	N	Yes	Central
HKU	91	5.8	W	S	Yes	Ap Lei Chau Est
HKU	94	3.5	E	N	No	Central
HKU	94	3.8	W	S	No	Lei Tung Est
HKU	504	5.8	W	S	Yes	Wah Fu

Table 1. The table Node of the database.

The table NODE_INFO, as illustrated in Table 2, records the information of the predefined transit points. The first column is the number assigned to the predefined transit point, the second the name of the predefined transit point, the third the street of the transit point location, the fourth the x-coordinate of the transit point, the fifth the y-coordinate of the transit point and the last column the description of the transit point.

Node_Number	Location	Street	XPos	YPos	Description
0	HKU	Boham Road	150	180	outside the University of Hong Kong
1	Central-MTR	ade Road	385	160	outside World Wide House
2	Wan Chai	Hennessy Road	560	180	near Southorn Playground
3	Admiralty-MTR	Queenway	460	190	Outside Taikoo Plaza
4	ShunTakCentre	abc Street	300	100	outside Shun Tak Center
5	Causeway Bay-SOGO	Hennessy Road	690	180	outside the SOGO store
6	ABERDEEN	ABERDEEN ROAD	380	445	
7	WONG CHUK HANG	WONG CHUK HANG ROAD	500	460	
8	AP LEI CHAU	AP LI CHAU BRIDGE	355	535	
9	WAH FU	WAH FU ESTATE	230	400	
10	CHI FU	CHI FU ROAD	240	330	
11	PEAK	Peak Mansion	350	270	
12	Repulse Bay	Repulse Bay Road	745	550	
13	Stanley	Stanley market	910	750	

Table 2. The table NODE INFO of the database.

Figure 2. The control flow of the routing algorithm.

4. Features of the Interface

The features of the main program are shown in Figure 3. Part of the map of the Hong Kong Island is shown and by using the scroll bar, other parts of the Hong Kong Island can also be seen. Those transit points are represented by the logo within the map. On entering the transit point, the logo will become larger and the name of that transit point will also be shown. When clicking on the transit point, a window will appear and all the available bus numbers at that transit point will be shown. Two buttons will also appear, one is the SELECT button and the other ZOOM. The ZOOM button can show the real map of that region when pressing on it. The SELECT button is for the selection of the source and destination. After choosing the source and destination, a word START and STOP will appear on the transit points which have been chosen to indicate that it is a source and destination respectively. Also the name of the transit points that have been chosen will shown in the textfield. Selection of transit points can also be done by choosing different choices in the choice menu. The RESET button is for the purpose of cancelling all the transit points that have been chosen and the SUBMIT button is for the submission of the transit points and the start of the calculation of the routes and fare.

After calculation, the result will be shown in the "resultframe". Figure 4 and Figure 5 show two "resultframes". Figure 4 is the result when there is no changing of bus. The table will list the source and destination that have been chosen for calculation and the bus number shown is the shortest and cheapest route. Figure 5 is the result when there is changing of bus once. The first row is for the route from the source point to the intermediate point where one should change bus at this intermediate point. The second row is from that intermediate point to the destination point. With this table list, the user can easily find out which bus number, where he should take the bus and also the required bus fare.

Figure 3. Interface of the program.

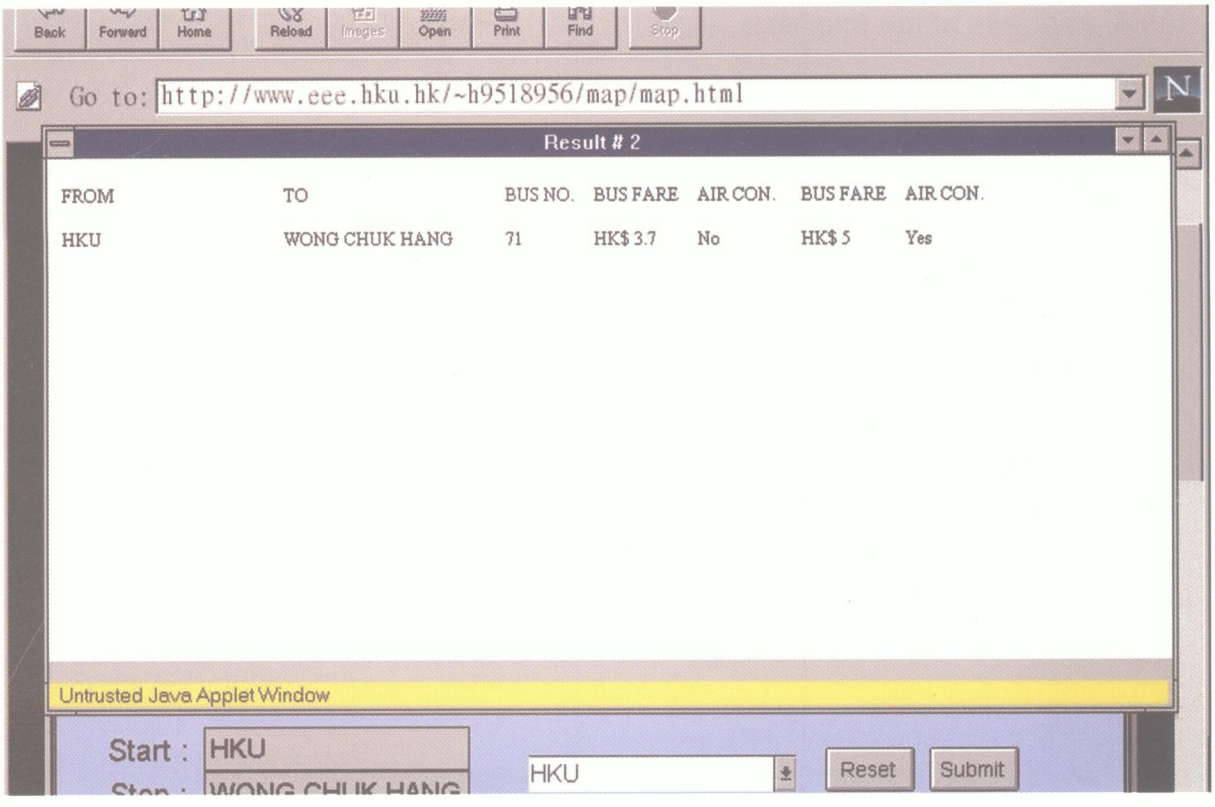


Figure 4. New frame to show result without changing of bus.

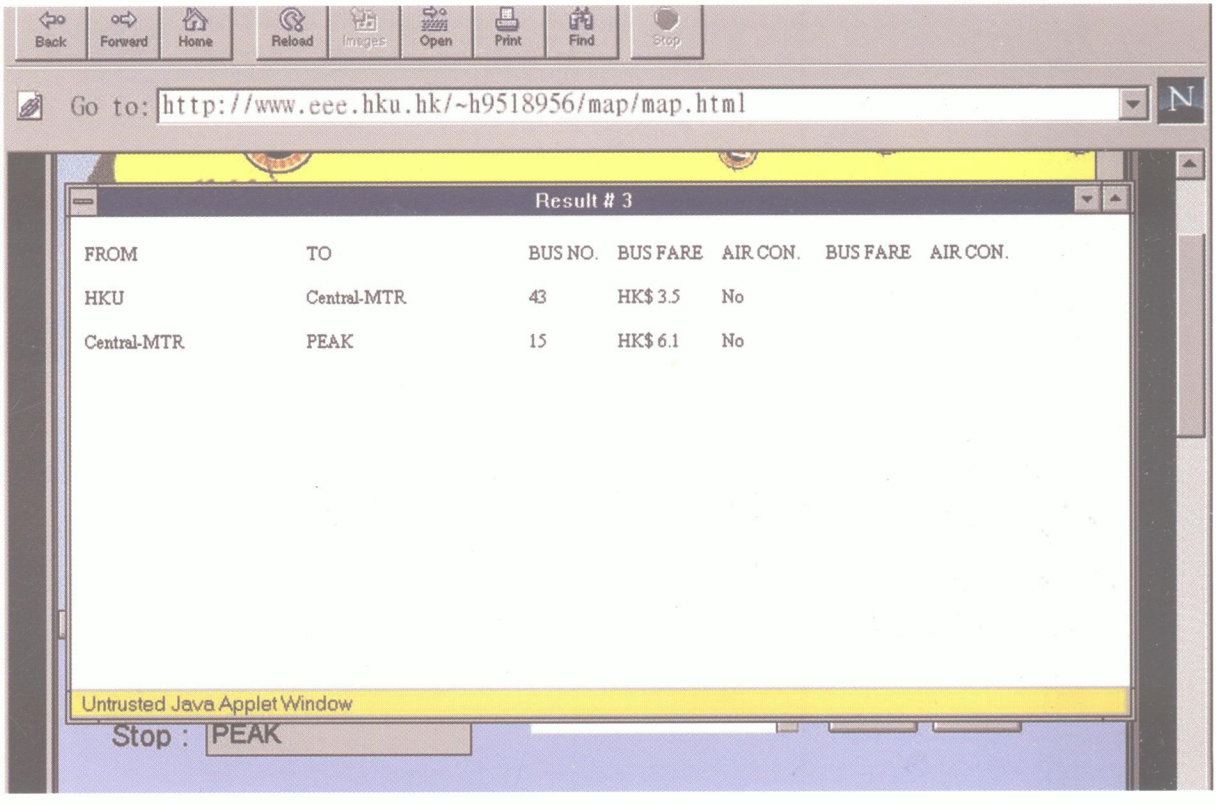


Figure 5. New frame to show result with changing of bus.

5. Weakness and Improvement

This program is quite good in the sense that it can suggest the user to travel with the shortest path and the cheapest bus. However, there are still grounds for improvement.

The main concern is the speed of the program. If the user chooses two points which have no direct bus travel between them, the user should change bus at some other places. During this calculation process, the program will query into the database many times. This means that the local computer should go to the remote database server many times. Therefore, the calculation speed strongly depends on the data transmission speed between the database server and local computer. If there is a direct linkage between the server and local computer, the overall speed is acceptable. However, if the local computer is connected to the server through a modem, the speed will be very low due to the limitation of the telephone line.

In order to improve the calculation speed, we can obtain all the data from the database during the applet initialization process and store it in memory. If we want data from the database, we can get it from the memory. Since this process is only done by the CPU and memory, the calculation time will be decreased by a large extent. However, there is one drawback. This method will increase the time for initialization. Comparing the two methods mentioned above, the latter is better since it only slows down the speed during initialization. It will give the user a better feeling if it has a faster calculation speed.

The second weakness of our program is the limitation of choice. The user only can select a small number of points in Hong Kong Island. The improvement is to update our database so that we can include all the data in Hong Kong. On the other hand, we shall have a whole Hong Kong picture for the user to choose his/her source and destination. For the applet, we are not able to change anything. Since all the points in the applet only depend on the data in the database. Therefore, if the database is updated, there is no need to change the applet.

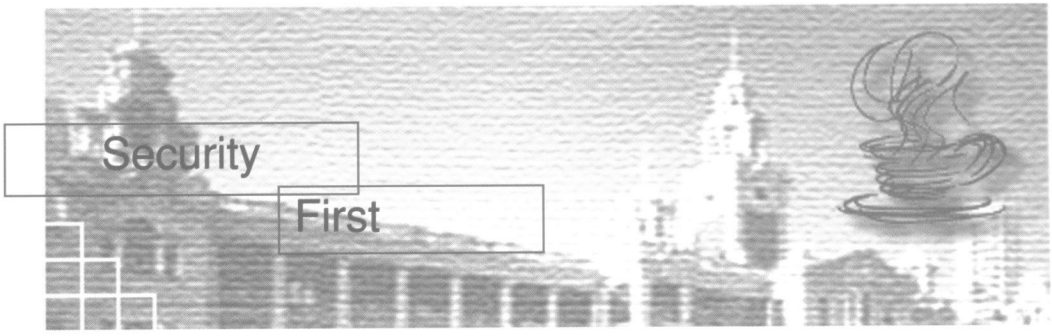
The third weakness of our program is that it do not allow the user to input the selection criteria. By default, the system will suggest the user to take the shortest path and the cheapest bus. However, the user may want to know all the possible path or/and all the possible buses. This criteria setting is not allowed in the program. To improve this, two check boxes in the homepage should be displayed so that user can change the selection criteria. Of course, it needs to have small changes in the calculation.

Although some improvements are needed, it is still a very useful program, not only for the foreigner but also for the people of Hong Kong. It can give the user the most valuable suggestion: the shortest and cheapest path.

Conclusion

With the dramatic development of the Internet and world wide web, Java Applet programming will obviously be very popular and widely used in the near future, due to its highly object oriented programming and operation platform independence. After this project, we have widely explored into the programming techniques and also the Java methods. We have also learnt how to co-operate between group-mates and search for information. The whole course is highly worthwhile since with the knowledge of Java programming, we have learnt one more practical skill which will be very useful in the future.

Nowadays, many companies have set up their own servers and homepages to promote their companies and products worldwide. Java Applet will be an essential part in the homepages because it provides a more interactive environment. With the skill of Java programming, it will make us more competitive. In conclusion, we are satisfactory with the whole course and also the program designed.



Authors

Au Kin Chung, Alex
Cheung Ho Yam, Allen
Cheung Kin Lung, Brian
Chung Ka Leung, Charlie
Ho Hin Wing, Terry

Part 1: Encryption

Introduction

As the exponential growth of the Internet demonstrates, TCP/IP solves many problems in a remarkably scalable way. However, TCP/IP was not designed to offer secure communication services. On the other hand, the implementation of Java on the World Wide Web has become one of the hottest topics on the Internet. Therefore, we have to consider to put two things together, that is, using Java to provide a secure way for Internet communication and transaction.

Objectives

After an in-depth study of the existing security measures used in the Internet, we recognized that the goal of our Java application or package should be able to handle or be further developed to solve the following problems.

- To authenticate users to make sure they are who they claim to be. Standard Web protocols such as TCP/IP and HTTP make impersonating a person or an organization relatively simple. For example, if Alice connects to http://www.well-known-retailer.com, how does she know that site is actually operated by the well-known retailer.
- To protect the privacy of our communications, both those in real time (such as the data flowing between a Web client and a Web server) and those with store-and-forward applications such as email.
- To ensure that messages have not been tampered with between the sender and the recipient.
- To safeguard confidential documents to ensure that only authorized individuals have access to them.

Indeed, the main concept of security measures used in the Internet nowadays is mainly key encryption, which is a sophisticated matter. It is nearly an unreachable task for our group to design our new security method which is better than the existing one. Therefore, the main task of our project is to study the existing security measures in the Internet and make use of those existing security measures to design and to provide some methods to design a security scheme by using Java language for the Java applets or applications.

In this report, it will be divided into two main parts, one is used to illustrate those security measures we have learnt; second part is used to show the effort we did in achieving aforementioned objectives.

Core Knowledge of Internet Security

What Is Cryptography?

Cryptography comprises a family of technologies that include the followings:

Encryption transforms data into some unreadable form to ensure privacy. Internet communication is like sending postcards in that anyone who is interested can read a particular message; encryption offers the digital equivalent of a sealed envelope.

Decryption is the reverse of encryption; it transforms encrypted data back into the original, intelligible form.

Authentication identifies an entity such as an individual, a machine on the network, or an organization.

Digital signatures bind a document to the possessor of a particular key and are the digital equivalent of paper signatures.

Signature verification is the inverse of a digital signature; it verifies that a particular signature is valid.

All of these technologies make use of sophisticated mathematical techniques.

Content of Project

Among different encryption methods available in the market, they have different properties in security levels and speed. Therefore, in order to choose a suitable encryption scheme for specific application, we have to consider several factors:

- Speed of the encryption – Is the encryption method fast enough for our application? If we use larger bit size key pair to encrypt the data, the encryption time will properly be longer.
- License fee of the algorithm – Is the algorithm using requires a large amount of license fee? It is because many of the encryption algorithms available are developed by private commercial companies, which require the company users a certain amount of payment.
- Origin Country of the algorithm – Countries like the USA has laws to protect the computer security encryption method from being exported to the other countries. Therefore, before using the algorithm, you should take care of the origin of that algorithm so as to avoid offending the law.

In the following paragraphs, you will be introduced into different encryption methods and their properties

Symmetric Cipher

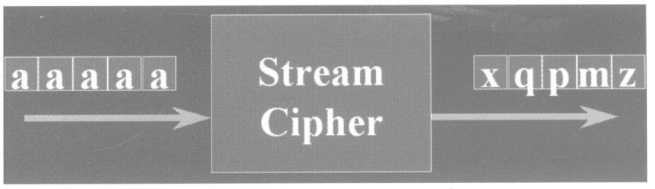
The word "Symmetric" indicates that only ONE session key is used to encrypt (encode) the data on both sides. "Cipher" is a data encryption algorithm. And, this can be further divided into two kinds, stream cipher and block cipher. The data we are encrypting is called plaintext, whereas the encrypted version of the data is called ciphertext.

Stream Cipher

It is a simple single-character-in, single-character-out cipher which does the encryption one character at a time.

Each time a stream cipher reads a character, it uses the key and accumulated data from the other characters it has processed to figure out how to scramble the next byte of data.

In the cryptiX package, there are several popular ciphers in cryptiX.crypto, for example, Rot13Cipher is a rather primitive cipher which is used in newsgroup posting, which simply maps a character to another, for this reason, this algorithm does not require a secret key between the sender and receiver.



Block Cipher

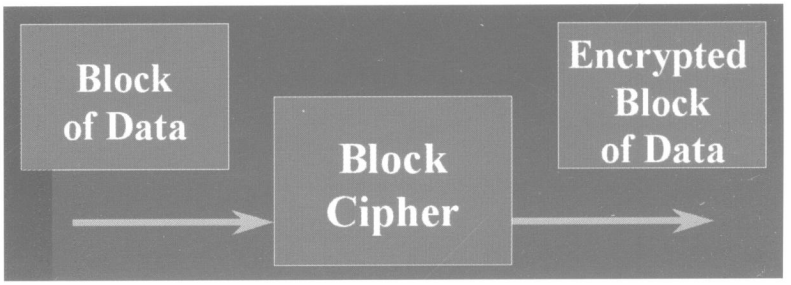
It, unlike stream cipher, encrypts whole blocks of data at a time. The block cipher can scramble all the bits in a block so that the bits for the first byte of the block can be scrambled and placed in strange places.

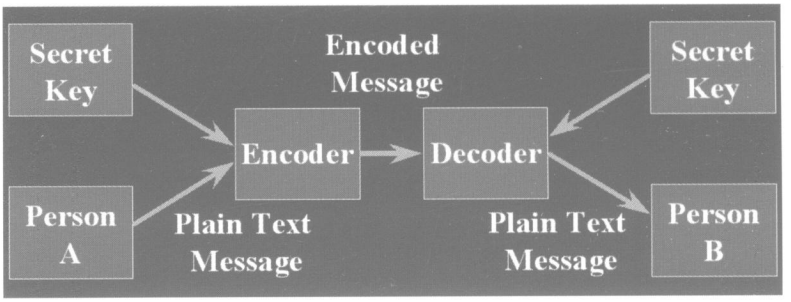
In the cryptiX package, there are also several block ciphers provided in cryptiX.crypto:

For examples:

- DesCipher – The Data Encryption Standard (DES) is one of the most popular encryption algorithm used in both commercial and government sectors. The DES consists of two components: an *algorithm* and a *key*. The DES key is a sequence of eight bytes, each contains eight bits. During encryption, the DES algorithm divides a message into blocks of 64 bits. It operates on a single block at a time, dividing the block in half and encrypting the characters one after another.
- Des3Cipher – As the user want to further increase the security level for the DesCipher, another variant of DES algorithm comes out which is called DES3. It will encrypt the data by DES algorithm for three times with 2 different session keys.
- IdeaCipher
- NullCipher

How cipher works?





Level of security

One of the problems with cipher encryption is that it requires the agreements of both sender and receiver on a secret key or algorithm, therefore, if anyone finds out the secret key, one can spy the data being exchanged.

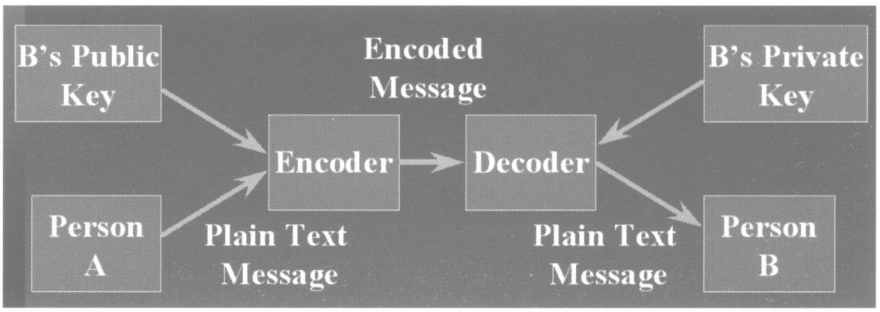
From this point of view, it should be avoided to hardcode the secret key into the applet or application because the key may be revealed easily by the reverse engineering of Java class.

In fact, if the key is kept secretly, the encrypted data can hardly be decoded. Moreover, the cipher can provide a faster way to encrypt the data than the asymmetric key encryption method which will be discussed later.

Public Key Cryptography

Why we have to use asymmetric key?
As aforementioned, symmetric-key or secret-key cryptography uses the same key to encrypt and decrypt messages. This is a familiar real-world phenomenon: We use the same key to unlock and lock our car doors, for instance. The problem with symmetric-key cryptography is to have the sender and receiver agree on a secret key without anyone else knowing it. How can they do this? Over the phone or on a floppy disk? All of these are cumbersome, slow, and error-prone techniques. In addition, the number of Keys tends to be much larger than the number of nodes; that is, people may have multiple keys for different purposes. In order to avoid the situations depicted above, we can make use of asymmetric key.

What's Public-Key Cryptography?
Public-key cryptography was invented in 1976 by Whitfield Diffie and Martin Hellman to solve precisely the above problem. With public-key cryptography, each person gets a pair of keys, a public key and a private key. Each person's public key is published, while the private key is kept secret. When Alice wants to send Bob a secure message, she encrypts it using Bob's public key. When Bob gets the message, he decrypts it using his private key. The sender and receiver no longer have to share secret information before they can communicate securely.



In the CryptiX package, you can use the classes inside CryptiX.rsa to generate RSA public key and private key. With this package, you can specify the key size and output format, the generation process is simple enough.

Practical Cryptography

In practice, both symmetric-key and public-key techniques are used in popular security protocols such as SSL and S/MIME because symmetric-key algorithms tend to be much faster than public-key algorithms. Let's visit Alice and Bob again. They want to communicate securely, but they also want to communicate quickly.

Here's what they can do:

1. Alice generates a random number (key) that will be used for actually encrypting her message to Bob.
2. Alice encrypts the random number with Bob's public key.
3. Bob decrypts the random number with his private key. Now Bob has a secret shared between him and Alice only. They can use the corresponding keys to encrypt and decrypt messages to each other.

In reality, most security protocols are much more complicated than this, but the three-step process above gives the idea of the fundamentals. SSL is one of the examples of a security protocol that uses these techniques to safeguard communications.

Digital Signature

Why use digital signature?
In normal situation, the data may be for public access, for example, indexes of stock market. Therefore, this kind of information does not need encryption. However, how can we ensure that this information is what we want to get from the true provider, that is, Bank? The solution to this question is digital signature which can ensure that the data are coming from the trusted sender and have not been modified by third party.

What is it?

The idea behind a digital signature is using a special form of encryption to create a much smaller version of the data. This smaller version of the data is the signature of the data which will be appended to the original data. The encryption used is public key cryptography and the method to obtain the smaller version is called "message digest".

It is because the encrypted information in a digital signature is not a complete representation of the data – that is, one could not decode it and get the original information back. In other words, a digital signature is a kind of one-way encoding. We cannot get the original information back, but given the original information, we can verify that it was signed with a particular key.

Moreover, since the signature is generated using the original information, even if a tiny part of it is changed, the digital signature would be completely different.

When to use?

It is better to use an example to demonstrate the need of using digital signature. Suppose Alice wants to send Bob a contract or credit card number electronically and Bob needs an electronic signature to verify authenticity. First Alice sends the document. She uses a hash algorithm to generate a fingerprint for the document, encrypts the hash value with her private key, and sends the encrypted hash value to Bob. This is Alice's digital signature. Bob uses the same hash algorithm to fingerprint the document he received and then unencrypts the hash value he received from Alice using Alice's public key. If the two hash values match, then Bob not only knows that the document he received is authentic, but he also knows that Alice's signature is real. Conducting commercial transactions in this way is undoubtedly more secure than using paper signatures, because paper signatures can be forged. And if the information that Alice sent to Bob is sensitive (for example, if it contains a credit card number), then it, too, can be encrypted so that only Bob can read it.

This model--or one similar to it--is probably the one we'll use to conduct business on the Internet and elsewhere. It is the basis for the U.S. government's proposed Digital Signature Standard (DSS), which relies on the Secure Hash Algorithm to produce hash values and a public-key cryptosystem known as the Digital Signature Algorithm (DSA) to produce signatures from hash values. The DSS has been criticized for various reasons, but much of the criticism comes from parties who have a financial interest in seeing that it's not adopted.

Time will tell which, if any, method for creating digital signatures will become the standard. Regardless of the outcome, what is of greater importance is that it is possible to conduct electronic commerce in a secure fashion.

If the bank has embedded the time-stamp as well to the digital signature, we can use this time-stamp to reveal the exact time when the bank signed this statement.

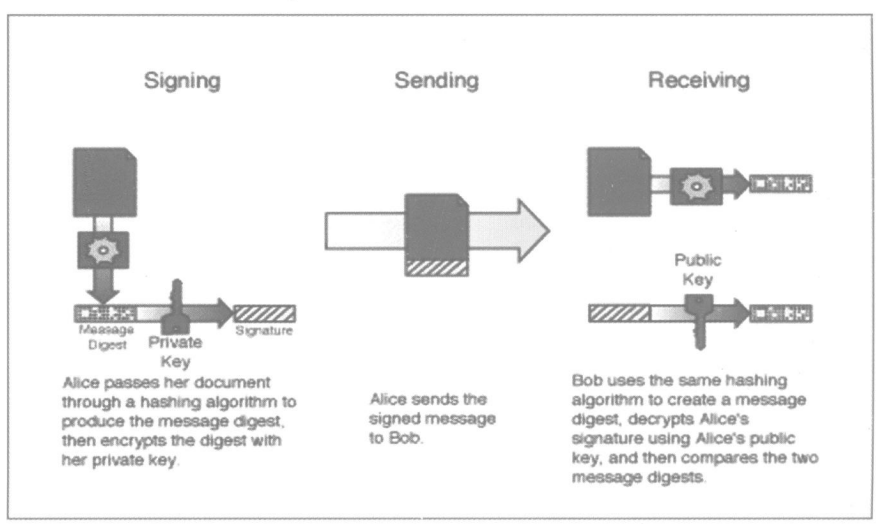


Diagram shows the process of using digital signature

In CryptiX package, we provide Signature Stream to users so that they can check with the digital signature append to the byte stream and ensure the data stream has not been modified and the identity of the sender.

Certificate

Since we can found that managing different public keys from different bodies is a difficult task; moreover, we want to ensure the public key is exactly the key we want of that body. The solution is using certificate, also called digital ID or digital passport, which is generated by Certification Authority. The certificate generally contains public key and identity, and basic information of the user to whom the public key belongs. This certificate is defined by an ITU standard called X.509 and is the digital equivalent of an employee badge, passport, or driver's license.

Table shows the content of a X.509 certificate

Field	Description	Examples
Subject's distinguished name (DN)	A name uniquely identifying the owner of the certificate	C=HK, O=Caba Security Group, OU=Technology, CN=Marc Andreessen
Issuer's distinguished name (DN)	A name uniquely identifying the certificate authority that signed the certificate	C=US, O=VeriSign, CN=VeriSign Class 1 root
Subject's public key	The owner's public key	512-bit RSA key
Issuer's signature	The certificate authority's digital signature from which the certificate derives its authenticity	RSA encryption with MD5 hash (signature itself is not human readable)
Validity period	Dates between which the certificate is valid	Not before Wed, Apr 9, 1997, 15:54:17 Not after Fri, Apr 31, 1998, 15:54:17
Serial number	A unique number generated by the certificate authority for administrative purposes	02:41:00:00:01

Certification Authority

Certification Authority (CA) is a trusted signing authority. When we create our own private/public key pair, we have to send the public key to the CA along with information verifying who we are. The CA then verifies our information and digitally signs our public key with the CA's digital signature.

What's CA doing?
When we download digitally signed codes, we shall also receive a certificate containing the server's digital signature, signed by the CA. We then use this certificate to verify the signature on the code. In other words, the server is saying to us "Here is some code that I have signed, and here is a copy of my signature that has been notarized by the CA".

Other Security Measures

Anti-Playback Strategy
Playback attack is often useful when encryption is used to set up a login connection or similar session, but the rest of the session is unencrypted. An eavesdropper could play back the login sequence and make a login connection some time after the original login, without ever knowing the secret key.
How to avoid this kind of attack?

- The receiver generates a random number and a session number with each session incremented by 1 and sends it to the sender
- The sender must use the randomly generated number inside the message it sends to the receiver.
- The receiver ignores any messages with an invalid number.

In our CryptiX package, there is a SecureSessionOutputStream.class which do the exact thing described above, each byte stream sending out will be appended with a session key and will be incremented. On the receiver side, we can use SecureSessionInputStream.class to verify the stream automatically which is transparent to the users.

Part 2: Security Level And Access Control

Introduction

As the modern world has been pushed to be so advanced and efficient, processes like information delivery and source distribution become so sophisticated that they should be computerized.

For these computerized systems to perform reliably and logical in order to serve the fast-changing human society, object-oriented and flexible security policies are required.

Objectives

In this part of our Java security project, an Access Control package is to be worked out in Java which can be used to implement security policies in Java applets in the Internet Browser as well as Java applications.

A demonstration application is to be developed using the package in order to implement a basic Access Control Policy in banking system.

Access Control and Security Level

The term *Access Control* describes the manipulation of the relations of access among objects. The objects can be users to users , users to data (programs) and programs to programs, etc.

There exists many policies in Access Control. The policies employed in our Access Level package consist of Discretionary Access Control and Mandatory Access Control.

Discretionary Access Control is defined as "A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) onto any other subject."

It is different from the *Mandatory Access Control* which are driven by the results of a comparison between the user's trust level or clearance and the sensitivity designation of the information. Mandatory Access Control is applied in permission inheritance among objects in our package.

Security Level (trust level) is a relative classification of objects which describes the range of information that can be accessed. Usually, more information is exposed to an object with a higher security level.

Basic Criteria for Access Control

We have the following considerations in developing the Access Control package (called the SecLev Package) :

- Flexible data management which can provide an efficient way to change the data structure or access permissions.
- Sufficient information hiding in which modification to the control should not touch the content of the data.
- Secure restriction on permissions to modify the Access Control.

SecLev Package - Tools to implement Access Control

Class Hierarchy of the SecLev Package

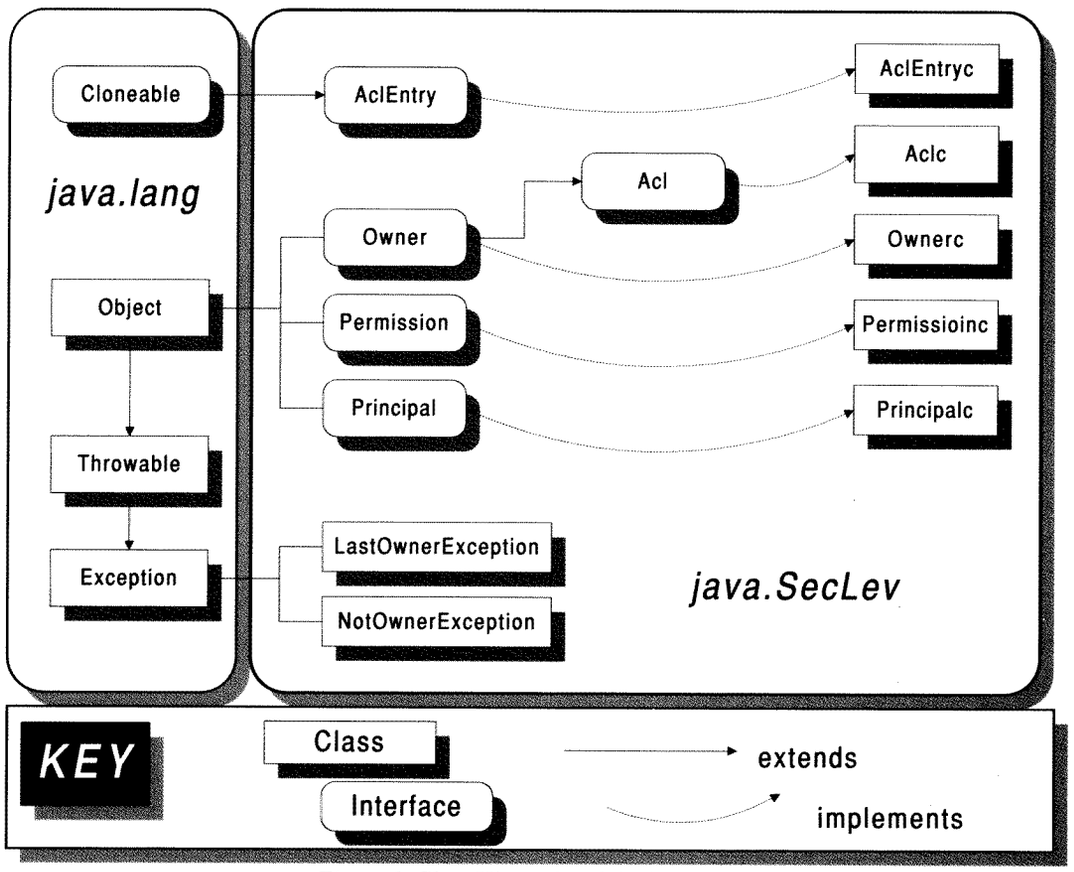


Figure 1. Class Hierarchy of the SecLev Package

Points to Note:

1. There are five interfaces in the package, namely *Acl, AclEntry, Owner, Principal* and *Permission.* Each interface has a class with an extension 'c' in their names to implement the interface.
2. The Acl interface extends the Owner interface because the Owner interface is used to maintain a list of owners for each Acl. Only owners are allowed to modify an Acl.
3. There are two kinds of Exceptions: namely *NotOwnerException* and *LastOwnerException.* NotOwnerException is an exception that is thrown whenever the modification of an object(such as an Acl) is only allowed to be done by an owner of the object, but the Principal attempting the modification is not an owner. LastOwnerException is an exception that is thrown whenever an attempt is made to delete the last owner of an Acl.

Mechanism of the SecLev Package

Points to Note:

1. An Acl is a data structure that guards access to resources. An Acl can be thought as a data with multiple Acl entries.
2. Each Acl entry, of interface type AclEntry, contains a set of permissions associated with a particular principal. A principal represents an entity such as an individual or a group.
3. Each Acl entry is specified as being either positive or negative. If positive, the permissions are to be granted to the associated principal. If negative, the permissions are to be denied.

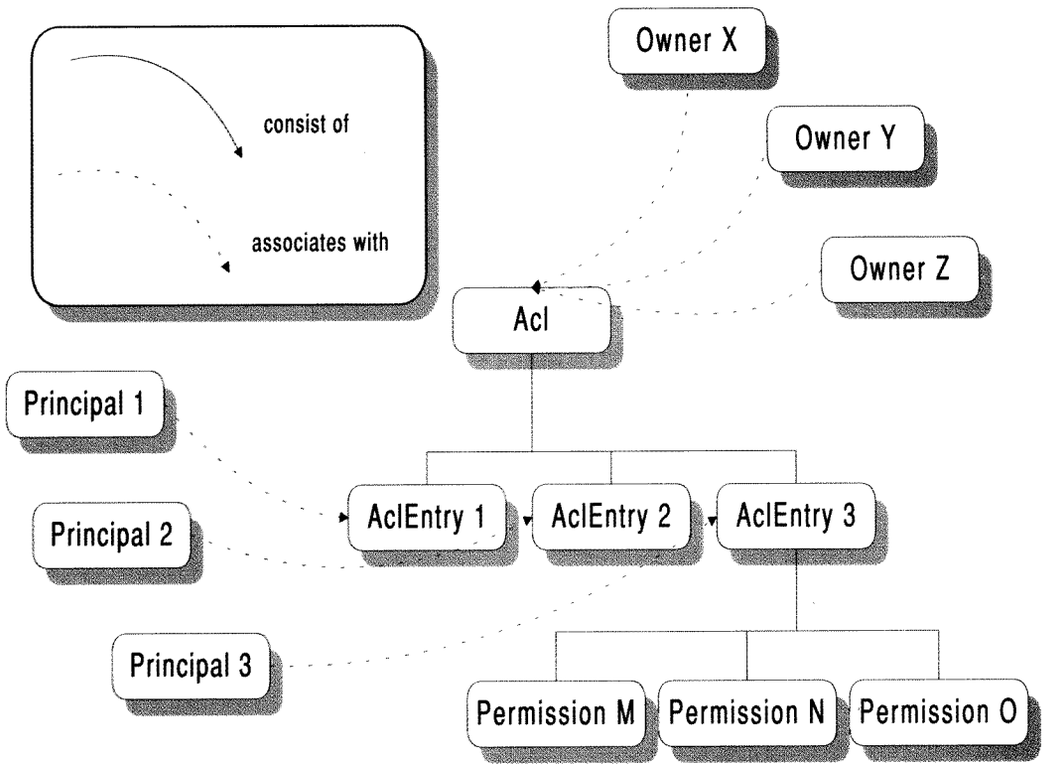


Figure 2. Mechanism of the SecLev Package

Calculation of granted permissions

Inheritance of group permissions with individual permissions override groups

1. In the above example, F is a member of D which further belongs to B, so as B and D have permissions P2 and P4 respectively, F inherits such permissions in its resulting permission list.
2. In practice, this can make the creation of permission for certain individual much easier. For example, when a thousand clients open savings accounts in a bank, we do not need to key in the same number of times their attributes inherited from savings account. Instead, any opening of savings account automatically uses the relationship to inherit them.
3. What about individual changes? The principle of individual permissions overriding groups based on this. Using the above example again, though A is the 'grandparent' of D and A permits P1, D prohibits P1(contains N1), so P1 is NOT permitted and hence when inheriting to F, the latter does not get a P1 permission.
4. In practice, this can make the system more flexible and we need not care what the 'parent' inherits to the son. That is, what the parents inherit is just for convenience in preventing entering a long list of permissions x times for x clients. For example, if we normally do not let account clients see the password but those having amount greater than a million is an exception, then we simply let them say 'super accounts' have the permissions without worrying about prohibition from the parent 'general account'.

Same Positive and Negative permissions cancel each other

The relationship between higher and lower groups are explained above. What about parallel groups? That is, between parent and parent, between grandparent and grandparent, etc.. There are two kinds of treatment. The first case is like that in B. With unknown of reasons there are both P7 and N7, that is, both permission and prohibition on P7. In such cases, they cancel each others as if they never exists, so you cannot see permission or prohibition on P7 in the son F. In practice, it has two meanings: First, it prevents accidental data-entering errors, which are rare. The more common case is in large companies, the data-entries of permissions and prohibits, due to maybe security reasons, are done by different people. Then the prohibition overrides permission for the same principle.

Cancellation of permissions for groups in parallel paths is excluded

What about between parent and parent in the same level like C and D(counted from the lowest level, in this case starting from F)? This is exactly opposite to that in same parent. That is, permissions override prohibitions. Hence P4 in D overrides N4 in C, so P4 can be still inherited to F.

In practice, this can easily be understood. For example, if savings account prohibits clients from accessing certain data whereas credit-card account permits that, it is odd that you cannot access the data, treated differently as other credit-card account clients because you have a savings account! In other words, this principle ensures that one can enjoy the uttermost set of access rights.

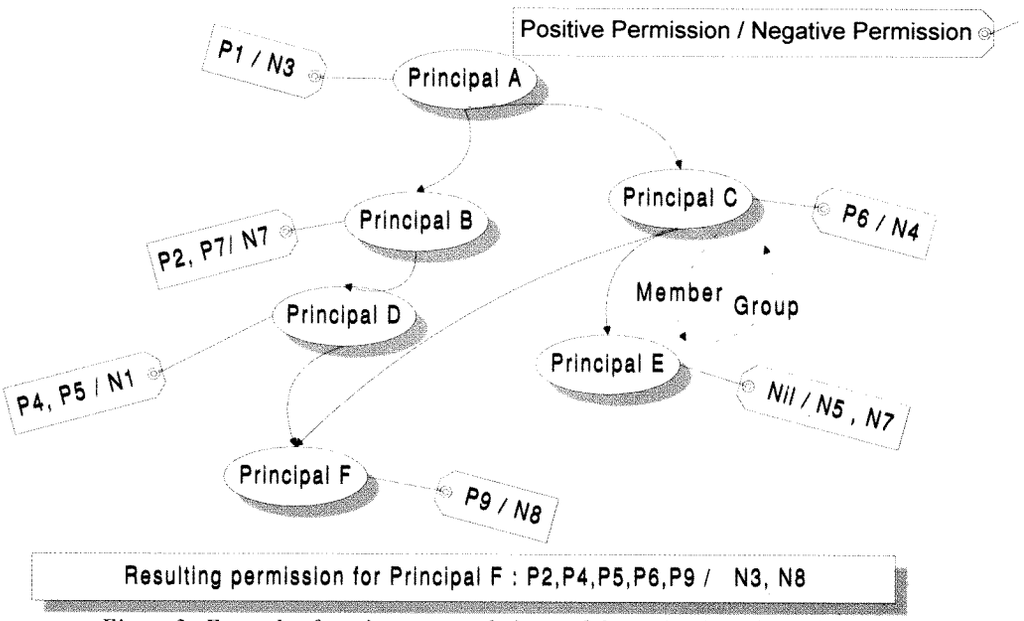


Figure 3. Example of nesting group relation and determination of member permissions

Demonstration Program: A Simple Banking System

Performance

This simple demonstration program tests the basic function of the SecLev package - to calculate the resulting permission set using the above principles. Basically it is an interface which allows users to do the following things:

1. New Account
2. Modify Account
3. Show Account Permissions
4. New Client
5. Modify Client
6. Show Client Permissions
7. Exit
8. Help

One can use 1, 2, 4 and 5 to initiate new accounts(or clients) or modify existing accounts. Each time the program will first show a list of existing accounts and you can select an existing one or type in a new one. Also shown is a list of existing parent accounts of the current account concerned. You can also use the buttons to modify them.

When the above data are entered, a new window replaces the present one. This shows a list of available permission choices with an existing permission and prohibit list. Again buttons are available to modify them.

3 and 6 uses the Acl principles to calculate the permission result and display it(with the data shown in clients' case, just as a demonstration of skills in msql).

Hierarchy

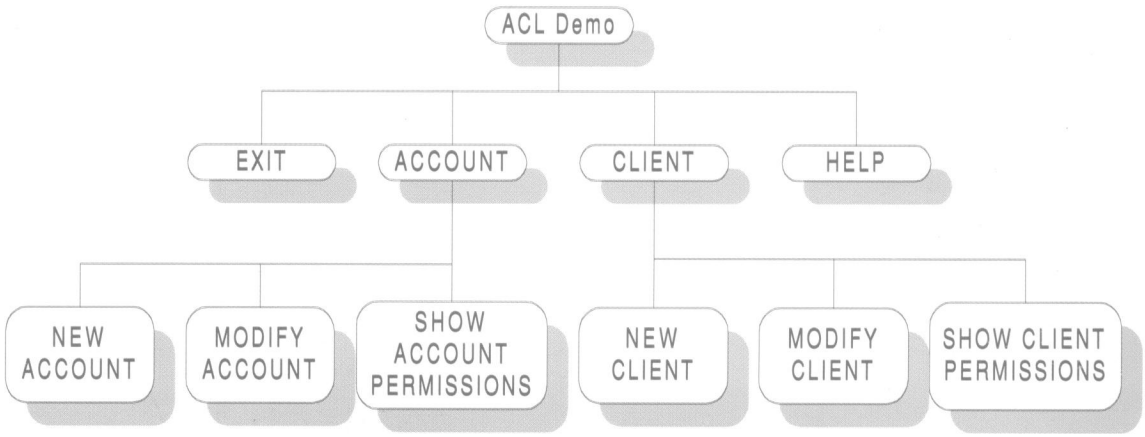


Figure 4. Block diagram of the demonstration program

The demonstration program consists of an AclFrame, which is the major Dialog to be loaded when the program is called through a html by the appletviewer. Here are the GUI classes called corresponding to the 8 functions discussed above:

1. AccountDial and then PermDial
2. AccountDial and then PermDial
3. ShowAccPermDial
4. ClientDial and then PermDial
5. ClientDial and then PermDial
6. ShowClientPermDial
7. N/A
8. HelpDial

Demonstration Program GUI

Here are some dialogs in the demonstration program corresponding to the Account handling. The GUI interfaces and the performance of the Client handling are very similar.

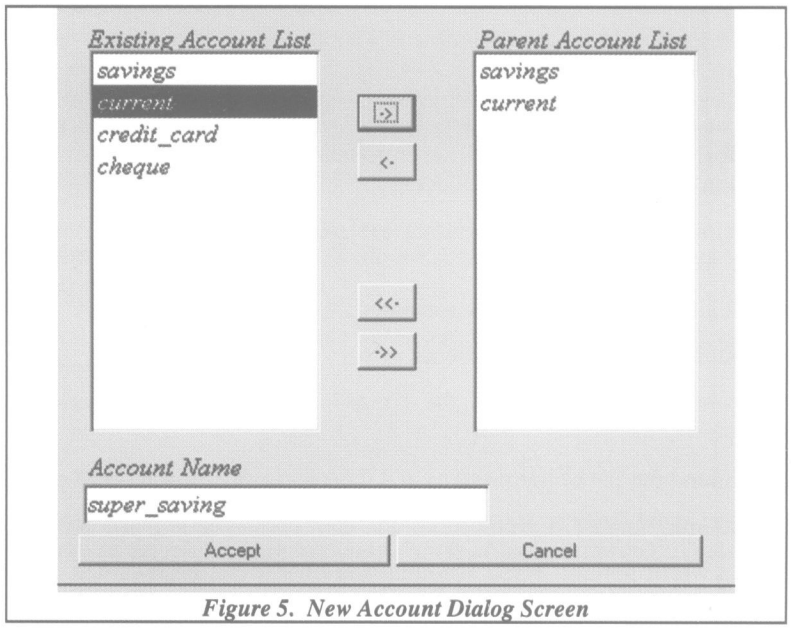


Figure 5. New Account Dialog Screen

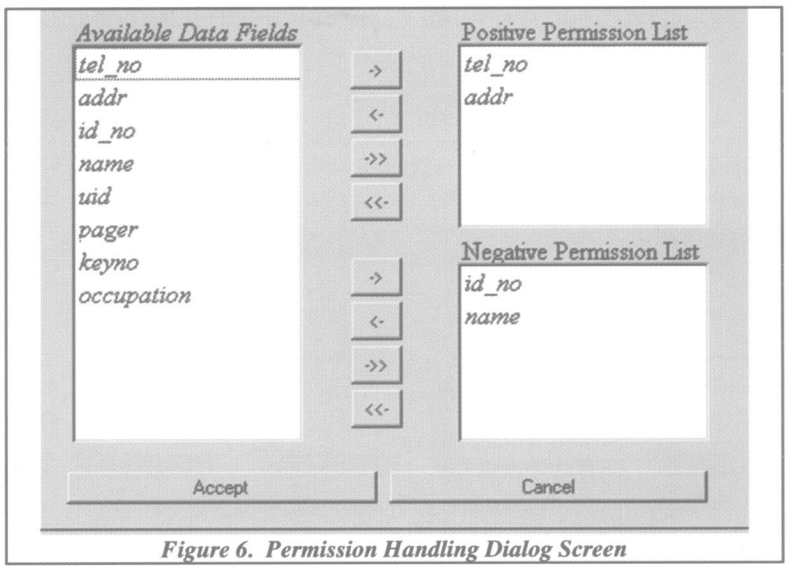


Figure 6. Permission Handling Dialog Screen

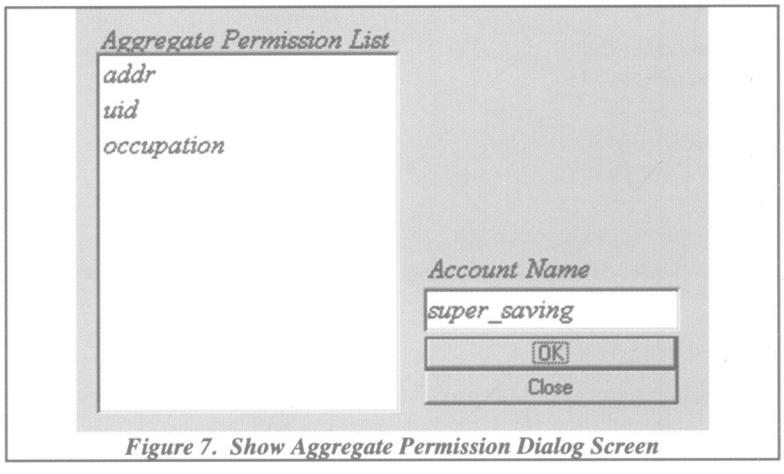


Figure 7. Show Aggregate Permission Dialog Screen

Database Structure

The database uses the msql. The technical points are not to be discussed here as details can also be referred to the classes described above. Instead, we introduce the database formats. Here are the tables used in this demonstration program:

1. user - table as a simulation to where the data are stored and access rights to them are to be calculated from the Acl principles.
2. Account - table containing the details of the accounts.
3. Client - table containing the details of the clients.
4. AccRelat - table containing the details of the group-member relationship between accounts and clients.
5. Permit - table containing the list of permissions and their corresponding account or client.
6. Prohibit - table containing the list of prohibitions and their corresponding account or client.

There are several points worth noting: First, accounts and clients can be regarded as the same object from the programmer's point of view, but from the bank's view, they are obviously two different things. So we place them in two tables. This is more advantageous than we place them in the same table and put one more flag in each entry to classify whether this is an account or a client. The condition is similar to the cases of Permit and Prohibit.

Further Development

There is of course imperfection in the demonstration program. For example, the program does not test the functions 'delete Accounts', 'remove Permits', etc.. Also, we can add a tree diagram on top of the existing one-level parent list to enhance better view of current relationship, but these are less important as they are quite 'doable'. Instead, we discuss the following areas which worth taking further exploration:

1. What is actually a permission?
    - At present, permission only means 1 or 0 in the access right to certain data.
    - The real situation is more complicated. First, you may be permitted to access part of the data. For example, if an entry is a combination of several parts in the database, then simply 1 or 0 logic does not work.
    - Second, there may not be only one table as in the demonstration. Then again the simple logic does not work.
    - Third, which is the most common case is the right may not be accessing data solely. Instead, rights to enter a certain homepage, or to modify certain data will also be allowed.
2. Would the data be accessed by the operator of the system?
    - This is a question most concerned by the user of this package. In the demonstration, normal operators of the system cannot access the data directly. But still if they can change the permissions and prohibitions, etc.. Then the problem remains to be indirectly changing permissions so they may still access the data after some work.
    - Then the focus will be on the identity of owners and there should be levels deciding some owners are more powerful in the sense that they control the rights to change the rights of normal operators or owners.
    - This area also concerns if the system is independently workable if the crew who knows the system leaves the company.
3. What can we do to the database if the company is to merge with another ones?
    - This is a increasingly common trend in the society. Many companies merge and associates with other ones. Thus the database should be merge-able.
    - That is what should be done in the function Clone in the class Acl, which is not done in the demo program. As a later view, this function may be developed into a new interface as many points should be noted in this, like comparison between the two Acls to be added or how to reduce possible actions in merging the databases.
4. Is the package flexible, that is, usable for a large range of areas?
    - This package is now inclined to be done in banking systems. For example, in parallel groups the permissions and prohibitions do not cancel as regards to banks, but this may not be true when considered in other areas.
    - As a result, we have to introduce some options and corresponding criteria to make the package more flexible.

Conclusion

In the demonstration program, some basic Access Control polices were implemented. Generally, the package can serve as a useful tool in implementation of basic security policies.

The next step for the development of this package may be to test its efficiency in handling a larger and more complicated data structure. Great room for development still exists in achieving goals as specified in the further development.

On the other hand, together with the Encryption works in previous part of this project, we hope that a comprehensive security policy can be worked out from these tools.

Part 3: Client-Server Login and Query

Introduction

This is a java applet which allows an end-user to login a bank and make queries into his own account. With the explosive growth of the internet and world wide web technologies, it is feasible to allow online banking services through platform independent web-browsers. Java is a suitable language to implement this application.

To make the application more secure, the Client-Server approach is adopted (Cryptography and Access Control are done by other group-mates). The Client Side applet provides interface for users to interact and request information it wants from the Server. The Server Side application approves identity and pass information from another database server to the Client. These two sides can be run on two different machines. They communicate with each other through a protocol. The request the Client can issue to the Server up to now is 'login' and 'getRecord'. The Server packs this request into a sequence of mSQL statement and forwards it to the database concerned. Thus the advantage is that the underlying database structure is totally transparent to the Client. This avoids the applet or application from connecting to the database directly and making intentional or unintentional damage to the database.

Mechanism Involved during Client-Server Connection

For more reliable communication, the Socket class has to be used. It implements a reliable stream network connection. A common model for network communication is for one or more Clients to send requests to a single Server program.

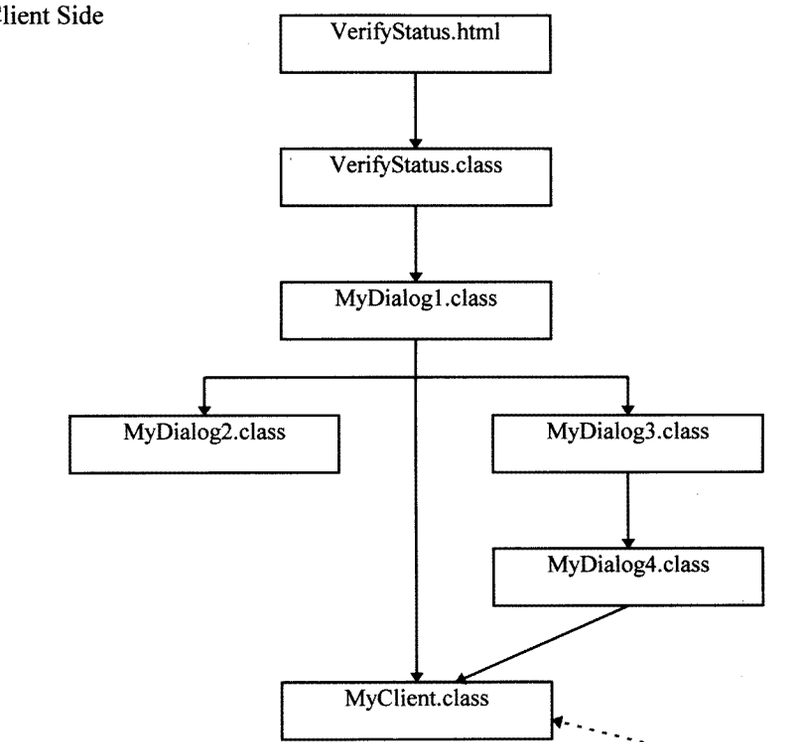
To establish a connection to the Server, the client creates a Socket object. Then it creates a DataInputStream to read lines of text from the socket InputStream and a PrintStream to write lines of text to the socket OutputStream. Finally, it writes request to the Server, reads the reply from the Server, and displays the results.

The Server uses the ServerSocket class to accept connections from Clients. When a Client connects to the port that a ServerSocket is listening on, the ServerSocket allocates a new Socket object (connected to some new port) for the Client to communicate through. The Server can then go back to listening on the ServerSocket for additional Client connections.

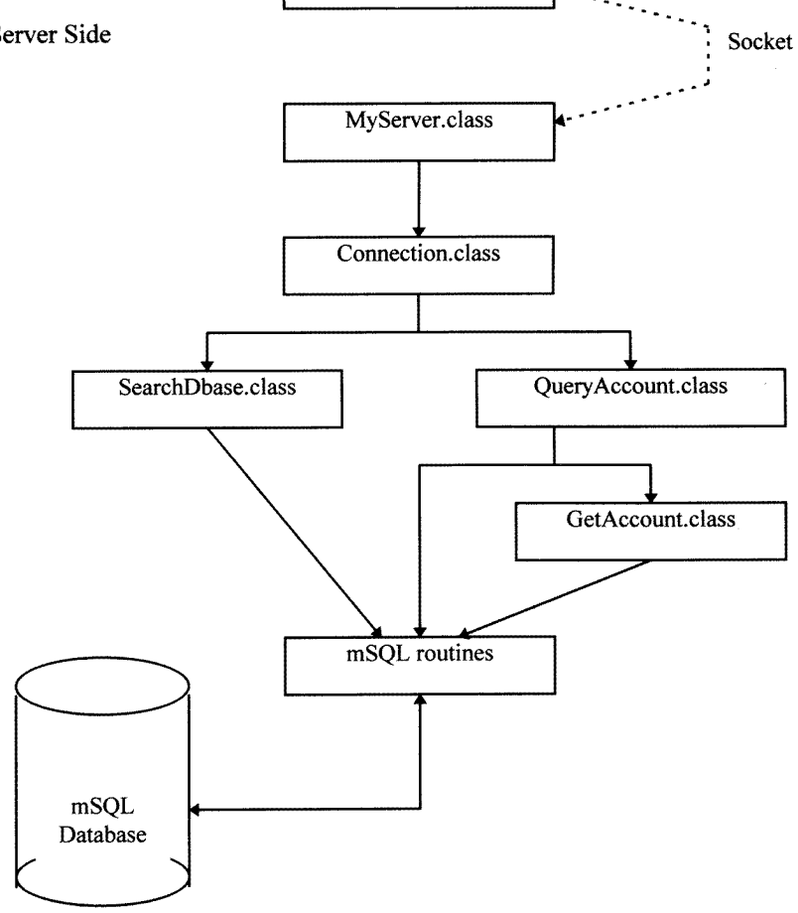
The Server is multithreaded. The Server object itself is a thread. Each time a Client connects, the ServerSocket creates a new Socket and the Server creates a new thread (a new Connection object) to handle communication over the Socket. This new Connection object handles all communication with the Client. The constructor method for the Connection object initializes streams for communication through the Socket object and starts the thread running. The run() method of the Connection object does all the communication with the Client, and performs the ervice' that this Server provides. In this case, it simply reads the request from the Client, then performs mSQL query, and sends the results back to the Client.

Sample Program Organization and Demonstration

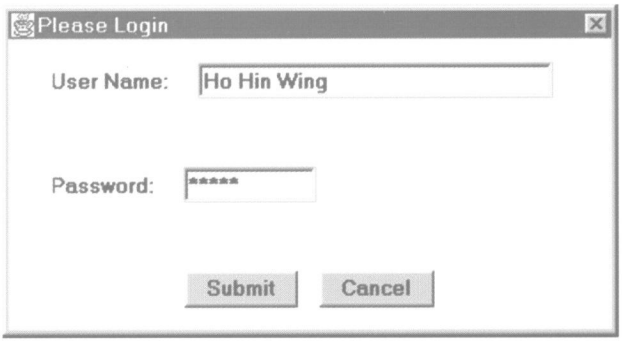
Organization of Client Side



Organization of Server Side



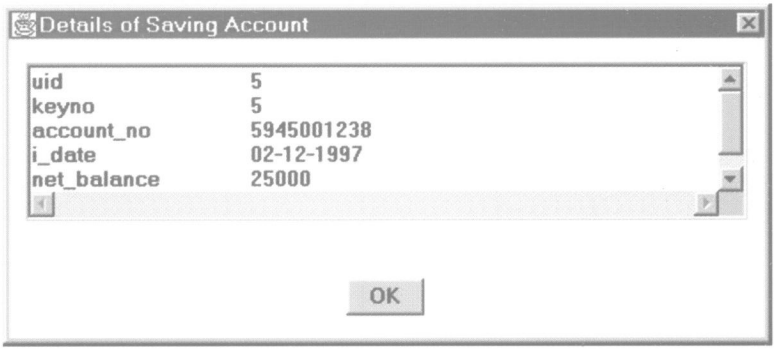
When the program starts, a dialog box is popped up which prompts for the user login user name and password.



The Client Side sends the 'login' request to the Server Side. The Server Side then verifies the status of the user from the database and replies the Client. If there is no such a user, the Login Failed dialog box is displayed.

If there is such a user, the following dialog box is displayed, from which the user may select his own accounts to view.

For example, if the user clicks the 'saving Account' button, the Client Side sends the 'getRecord' request to the Server Side. The Server Side, from the account table of the database, finds the appropriate record of the user and sends it back to the Client. The record is then displayed in the user machine.



It should be noted that the program can be modified to cater for more services for the Client applet, for example, transaction, investment, access control. Moreover, the Socket InputStream and OutputStream can be added with DataEncryptionStream and DataDecryptionStream to allow secure data transfer between bank and user.

Authors

Lau Hing Chun , Lobo
Lei Sai Weng
Wong Kwok Lung , Alex
Wong Pui Fan
Yeung Wing Yan

Abstract

We describe in this report the implementation of an internet-based shopping system for a supermarket. Products sold at the supermarket are displayed. Along with each product, a set of related data, for example, name, brand, size, price, discount, etc., is attached. Flexible sales techniques can be applied to each product. In addition, a search application can be used by specifying any combination of related data of a product.

Introduction

With the explosive growth of the world-wide web technologies, it is feasible to have on-line communication between customers and a supermarket. It allows a user to buy goods from a supermarket at home which is a new way of shopping is convenient for the customers.

On the other hand, such approach of managing a supermarket has advantages over the conventional one. It saves money and manpower in operating a supermarket.

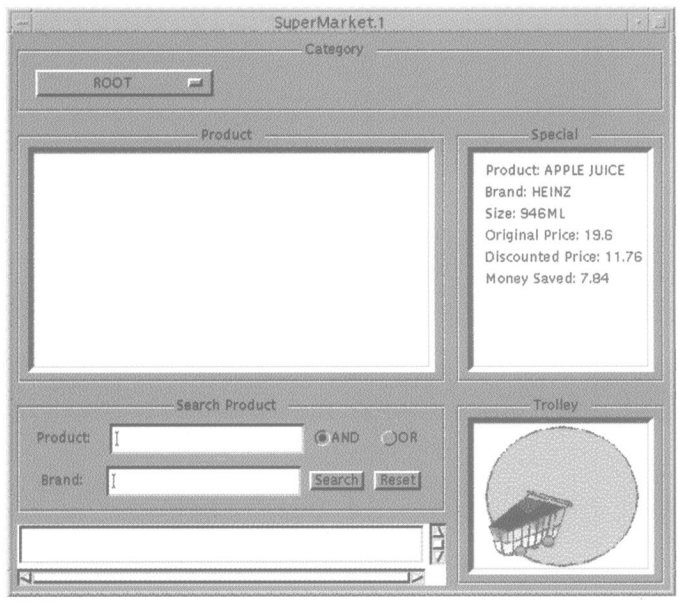
The purpose of this research project is to develop a Java-enabled client-server on-line shopping system. The system makes interfaces with databases which are storing the information of each product sold in a supermarket, for example, name, brand, size, price, etc. So, it has the ability to display the products with their corresponding information. Moreover, it also allows a user to find a product by merely entering a related string. When a user finishes his/her shopping, it records the goods chosen by the user and the personal information of the user which will be useful for delivery. It also has the ability to calculate the amount that a user has to pay with flexible sales techniques in mind.

To illustrate the usefulness of the system, we have implemented a site that displays and sells some products that are usually found in a supermarket. The properties we associated to the products displayed in our site are based on information from the Wellcome Supermarket. Moreover, in this site, the search function for a product and the advertisement section are also implemented.

1. A Sample Client Application

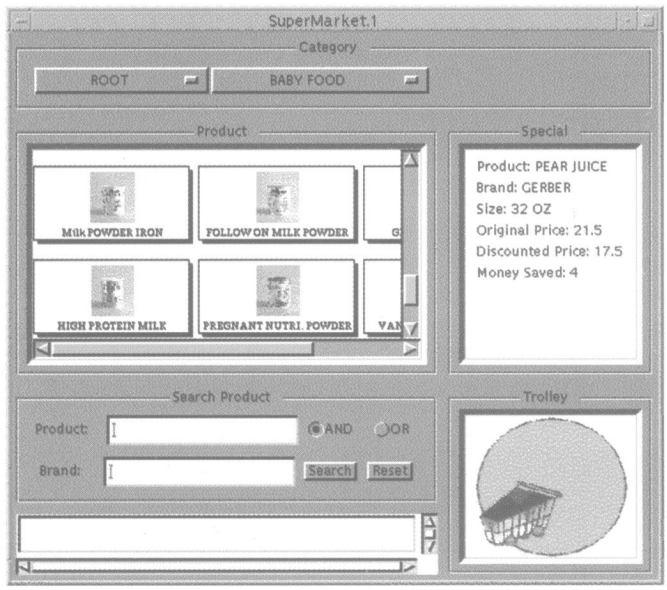
Before discussing the design of our on-line shopping system, it would be better to introduce a sample client application. The application allows an end user to search and buy products in a supermarket.

The Java-enabled web page developed for the on-line supermarket shopping system is shown below.



To find a kind of products from the virtual supermarket for example, juice for baby, a user can click the 'ROOT' button in the 'Category' section and select the category for which the products belong to from the list, that is, BABY FOOD. Then, the selected category will become the name of the adjacent button. Also, all the products belong to the selected category will be displayed in the 'Product' section.

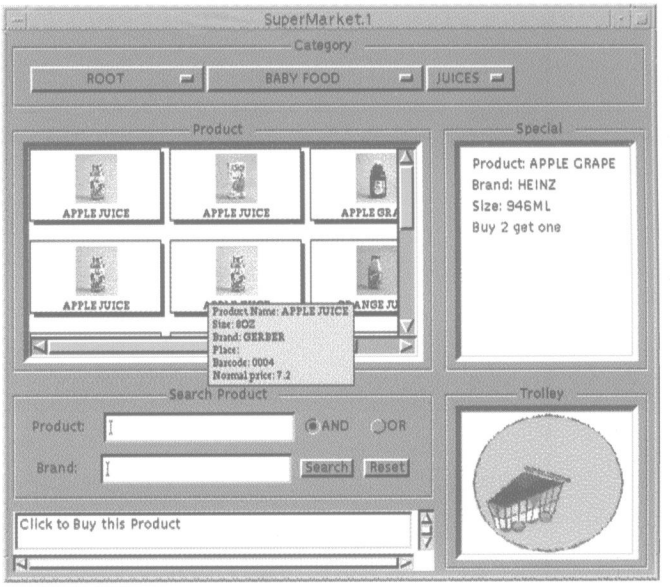
The above events are shown below:



When the user clicks the button adjacent to the 'ROOT' button to select a sub category, that is, JUICES, to describe more clearly of what he/she wants, the selected sub category will become the name of its adjacent button. Also, all the products belonging to this specific sub category will be displayed in the 'Product' section .

When the mouse points to one of the products displayed in the 'Product' section, the related information of the product will be displayed in a new pop-up window.

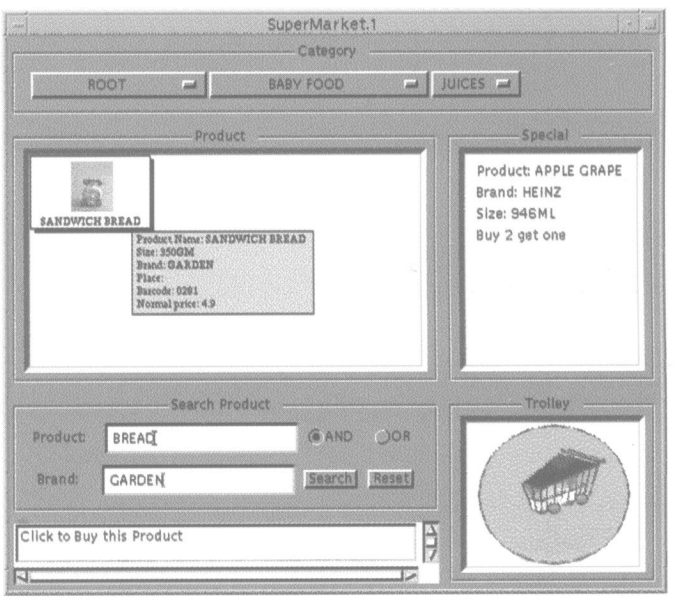
The above events are shown below:



All the above designs help customers more concentrated on the kinds of goods they intend to buy. This will make shopping much easier for customers.

To find a specific product, a user can enter the strings in the 'Product' blank and/or 'Brand' blank with the logic 'AND' and 'OR' chosen in the 'Search Product' section. For example, bread is entered in the 'Product' blank, Garden is entered in the 'Brand' blank and logic 'AND' is chosen in the 'Search Product' section. If the 'Search' button is clicked, the search results will be displayed in the 'Product' section.

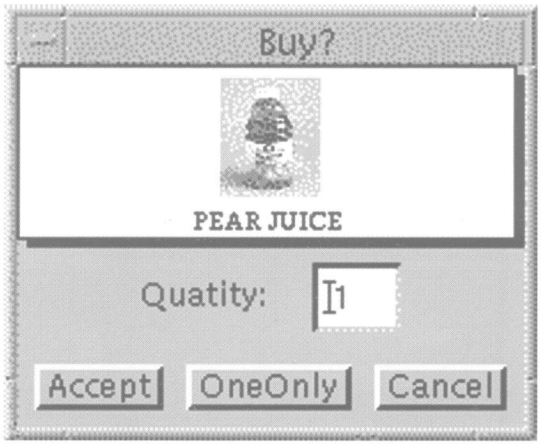
The above events are shown below:



This function allows customers to find a specific product they want to buy at an extremely fast speed instead of wasting their time in searching it from its category group .

To buy a product, a user can click the product displayed in the 'Special' section which shows all the products or in the 'Product' section. Then, a pop-up window appears to ask the user to enter the amount he/she wants.

The following shows the pop-up window described above.



After shopping, the user can check the kind and amount of what goods he/she has bought by clicking the 'Trolley' section. Whenever the 'Trolley' section is clicked, a new section 'My Trolley' will be popped up to display the user's checklist. At this point, the user can cancel any of the checkouts.

The following shows the pop-up section 'My Trolley'.

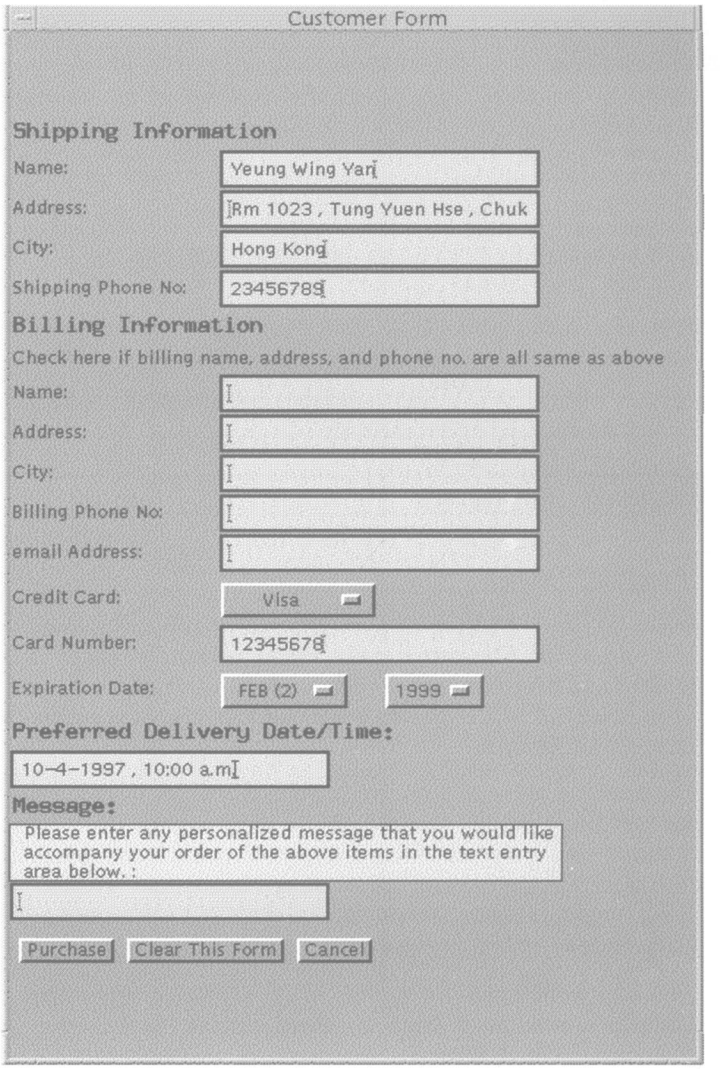
When the user confirms the checklist , he/she can press the 'Purchase' button. A form will be shown to ask the user to fill in some personal information for delivery purpose.

2. Online Supermarket Shopping System

The online supermarket shopping system is convenient for those who runs a supermarket. With 'Object Oriented' idea in mind and using the Java DataBase Connectivity interface in designing our system, a manager of a supermarket can easily add new products to the stock by adding the information of the new products to the database. To change the price of a product, one can simply change the price specified in the database. The changes in the information stored in the database will automatically update the information shown in the user interface of our system.

From time to time, supermarkets will put some products on sale to attract customers. So, the system should respond to the fast changing sales strategies effectively. If a supermarket is going to put a product on sale, the manager can simply change the index representing the method of sale (for example, fix discount, discount by percentage) and the amount of discount given to the product in the database. When a customer buys the product through internet, the new price will be used immediately .

When a new sales strategy is introduced , the manager of a supermarket can upgrade the system by just implementing the new sales method in Java individually and creating a new index to represent the new method, and put it in the database. This will not require to change any source code of other programs of the system.



3. Data Organisation

In our program, there are 6 tables in our MSQL database "SuperMarket.1". They are 'Category', 'Index', 'Product' , 'Sale' , 'Package' and 'CustomerInfo' .

'Category' Table

In the 'Category' table, it contains two fields: *category* and *parent*. The main categories or sub-categories of our products are stored in the field 'category'. The corresponding parent category of the products are stored in the field 'parent'.

For example , there is a main category called babyfood which is under the root. Then babyfood and root are stored in the fields 'category' and 'parent' respectively. Another example is a sub-category called juice which is under the babyfood. Then juice and babyfood are stored in fields 'category' and 'parent' respectively .

This table is designed for easy listing of the main categories which are under the root and the sub-categories which belongs to their corresponding parent categories.

The structure of the database 'Category' table is shown below .

category	parent
babyfood	root
juice	babyfood
...	...

'Index' Table

In the 'Index' table, it contains two fields : *code* and *category*. The barcode of each product will be stored in the field 'code'. And the category of corresponding product will be stored in the field 'category'.

For example, a product with a barcode 1123 is under the category juice. Then 1123 and juice are stored in fields 'code' and 'category' respectively.

This table is designed for easy searching of products which belong to certain categories. Also, it links with the 'Product' and 'Category' tables.

The structure of the database 'Index' table is shown below.

code	category
1123	juice
2345	breakfast
...	...

'Product' Table

In the 'Product' table, it contains nine fields: *pdt_name, code, brand, place, size, price, picture, method* and *parameter*. The name, code, brand name, country of origin, volume, normal price, picture, sale method and the parameter of each product which will be stored in the corresponding fields . In the field of 'method' , 1 to 4 represent the product to be sold at normal price, priceoff, percentoff and buy X get one free respectively.

For example, there is a product named Apple juice with a barcode 1123 and its brand name is Heinz. All the details of this product will be put into the table 'Product'. If 2 and 1.5 are stored in the field 'method' and 'parameter' respectively, it means that this product is put on sale with $1.5 deducted from the original price. If 3 and 0.8 are stored in the field 'method' and 'parameter' are stored respectively, it means that this product is put on sale with 20% deducted from the original price. Also, if 4 and 3 are stored in the field of 'method' and 'parameter' respectively, it means buy three get one free.

This table is designed for storing all the important information of the products. Hence, we can know the products' brand name, place of origin, barcode, name, price....etc. Also, we can know whether it is on sale.

The structure of the database 'Product' table is shown below .

pdt_name	code	brand	place	size	price	picture	method	parameter
Apple juice	1123	Heinz	USA.	10 oz	12.5	1123b.gif	2	1.5
...	...	...	...	...	...	...	...	...

'Sale' Table

In the 'Sale' table, it contains one field only: *code*. The barcodes of the products which are put on sale are stored in the this table .

This table is designed for listing all the products and packages which are on sale on the Sale Panel for easy buying !

The structure of the database 'Sale' table is shown below.

Code
1123
3456
...

'Package' Table

In the 'Package' table, it contains seven fields: *pkg_name, price, picture, code, pdt_list, method* and *parameter*. The name of the package is stored in the field 'pkg_name' and the fields 'method' and 'parameter' have the same meanings as those in 'Product' table. In the field 'pdt_list', all the barcodes of the products belonging to this package are put here and the total price of all these products is put in the field 'price' .

For example , there is a package named Package 1 which contains three products and the barcodes of these products are 1123, 1245, and 1547. These barcodes will be stored in the field 'product_list' in the format 1123&1245&1547. If the total price of these products is $120(summing the normal price), then 120 will be stored in the field 'price' . For the 'method' and 'parameter', they are just the same as mentioned above.

This table is designed for storing all the information of the packages.

The structure of the database 'Package' table is shown below .

pkg_name	price	picture	code	pdt_name	method	parameter
Package 1	120	2713b.gif	p0003	*1123&1245 &1547*	2	10.5
...	...	...	...	...	...	...

'CustomerInfo' Table

In the 'CustomerInfo' table, it contains twelve fields: *sname, saddress, scity, sphoneno, bname, baddress, bcity, bphoneno, email, cardno , preferred* and *message* . All the shipping and billing information will be stored in the corresponding fields at the instant of purchasing .

Initially , this table is created without data . When a customer finishes his/her shopping and fills in the customer form, all the information filled in the form will then be put into this table. Then the purchased products can be delivered to the customer according to the stored information .

The structure of the database table 'CustomerInfo' is shown below

(1: initial state 2:after someone send the form) .

sname	saddress	scity	sphoneno	bname	baddress	...
Li Tai Ming	18A, Wing Wah Bldg.	Kowloon	25686849	Ho Ping	12F, Sa Po Bldg.	...
...	...	...	...	...	...	...

Conclusion

In this paper, we have proposed and implemented a supermarket shopping system over the internet. We have also demonstrated the system through a Javatized web-page application. This system is flexible and extendible as the overall design is platform independent and object oriented.

The online supermarket shopping system is a new way of shopping. It makes supermarket shopping very convenient for customers. Moreover, it makes the administration of a supermarket easier.

Authors

Cheung Chi Hung
Wong Ngai
Poon Pong King

Abstract

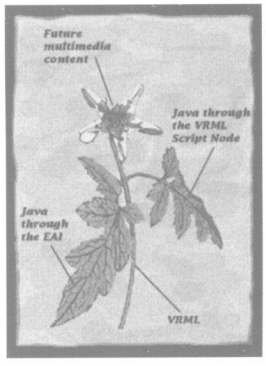
We describe in this paper the evaluation of the capability of VRML 2.0 (Virtual Reality Modelling Language Version 2.0) in communicating with JAVA. The already implemented examples feature the communication between a java applet (xxx.class file) and an embedded virtual world (xxx.wrl file) inside the same web page. Existing examples include a VRML showcase working with database, a simple VRML authoring tool, and several animated worlds interacting with java applets. There are also two static worlds encompassing the landmarks of Hong Kong, namely the *Lo Pan temple* and the *new airport main building*. A Netscape browser (version 3.0 or later) is needed in viewing the above sites and a Netscape plugin named Cosmoplayer (version 1.0 beta 3a or later) is required to load these virtual worlds.

Introduction

The release of the VRML 2.0 standard on August 4, 1996 made interatctions with the virtual worlds possible which was an impossibility with the VRML 1.0 standard.

The tools that are used extensively in creating the interactive virtual worlds include the Live Connect technology of Netscape and the External Authoring Interface (EAI) which is a set of java classes accompanying Cosmoplayer.

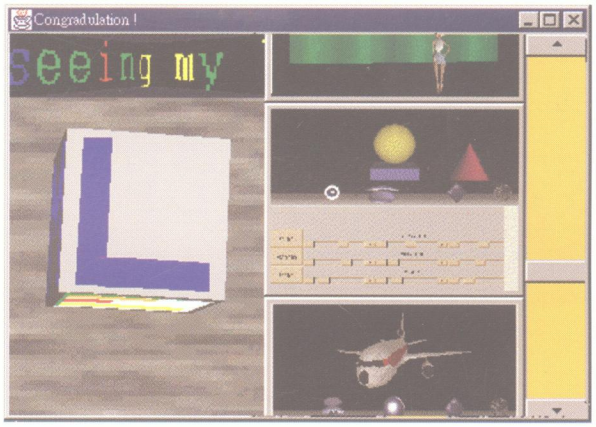
The blending of Java and VRML enables true multimedia, experiential, and dynamic content on the Web. Since both are cross platform open standards, anyone can wield the mighty sword of development power. More importantly, any end user on any system will be able to view this content. The most powerful feature of the EAI is that it allows VRML to co-exist and interact with ALL the features of a Web browser - HTML, Shockwave, JavaScript, etc. In short, Java and VRML is a powerful combination that results in an executable content, optimized for today's low bandwidth environments.



1. A Virtual Tour

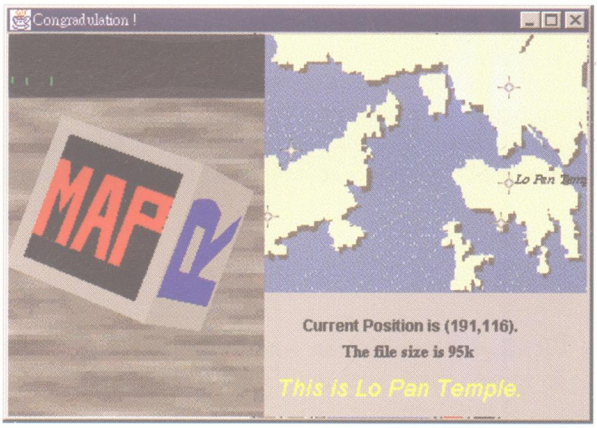
1. The front end user interface

In order to be consistent with the style of our project, this interface discards the usage of common image buttons but uses a draggable cube with each face corresponding to an equivalence of a button. Users may double click on each face and its corresponding content will be displayed on its right. Currently three faces are implemented:



Scrollable image button ('L' for link)

This face is a set of scrollable image buttons that are linked to various VRML worlds.



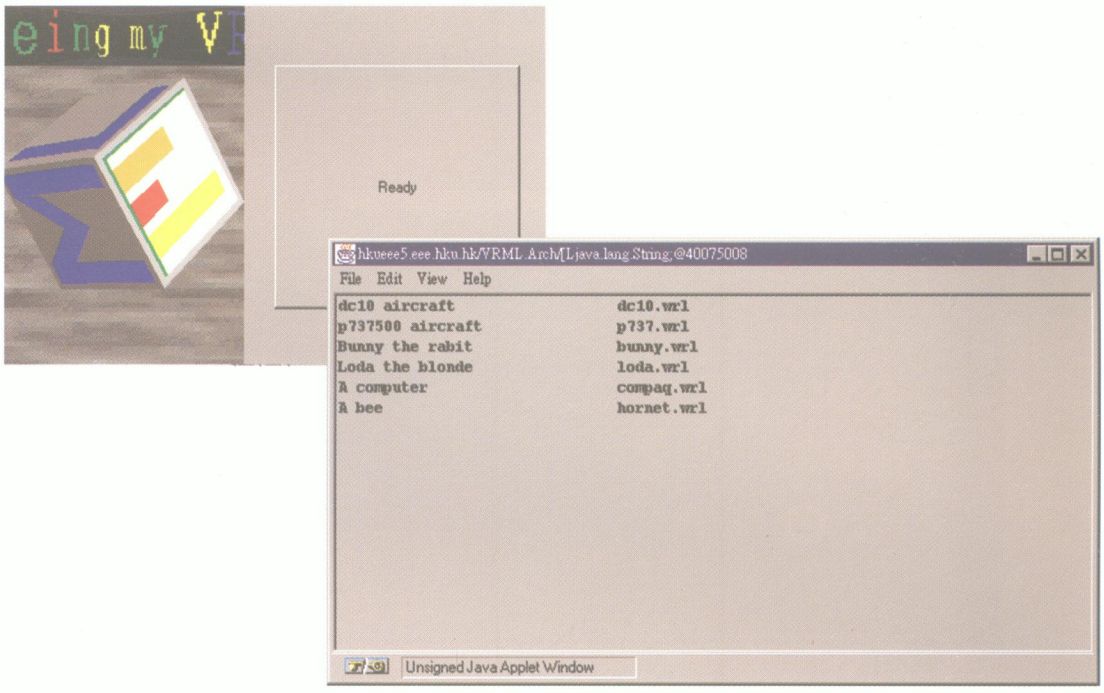
Map ('Map' face)

The image on the frame contains a 2D map of Hong Kong showing the different scenic spots. It is an user-friendly map that requires no input of a word. The user does not even need to know the name of the place. Having a little idea of the desired location, the user can just point the mouse around the location, a message with the name of the location will be popped out. Beneath the map is a display area showing the further information of the active site including the VRML file size of that spot, if available. Such message is important as the file of a virtual 3D world is usually quite large and as we all know, the download time is a critical factor on the network.

If the user decides to enter the 3D world of the specific location, he/she may simply click on the mouse and he/she will be placed in the virtual world in the browser window..

The significance of this 2D to 3D mapping provides not only a user-friendly link to the 3D world by its physical location, but also an updated map because the database is accessed every time the map is accessed. The information shown in the map can then be changed implicitly in the database without changing a word in the JAVA code.

mSQL database viewer ('M' face)



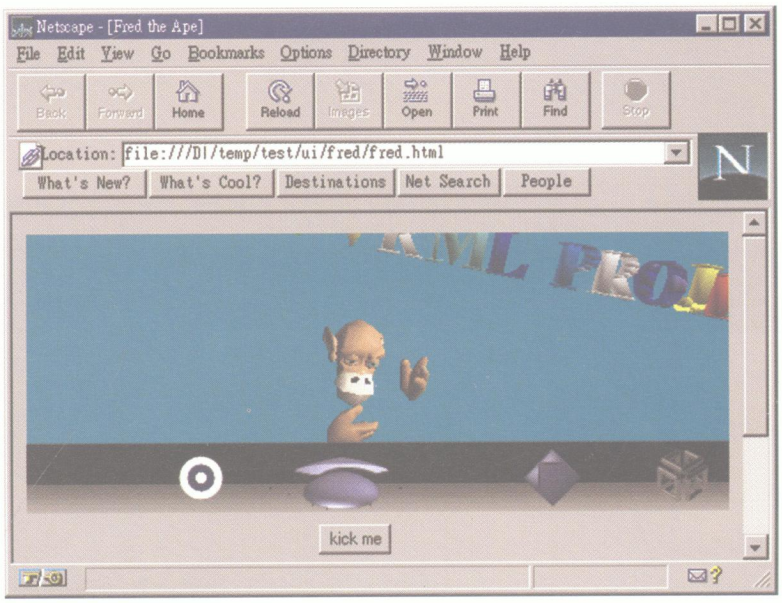
This viewer is an example applet provided by the mSQL author. By transforming it into a JAVA panel, it can be invoked by the "Ready" button. Users could then browse through all the existing tables in the database server. In this example, the relative URL's of the VRML showcase is listed in the table viewer.

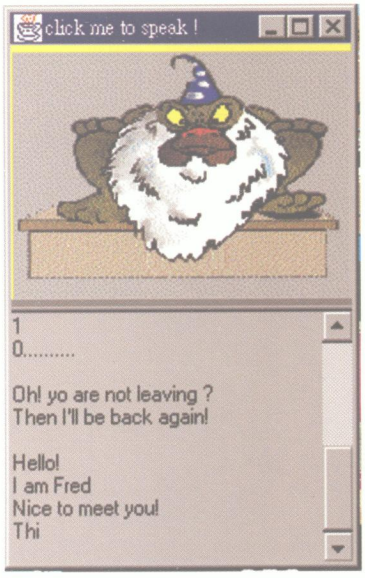
2. Loda the Pretty Woman

This world introduces the user a completely alive 3D agent. The animation followed by clicking on the woman is especially charming. When the green curtain behind senses a mouse click, it slides open and a 3D hypertext link equivalence will bring the user to a new horizon with JAVA mixed with VRML.

3. Fred the Ape

This virtual world contains a lovely character which, when clicked by the mouse, will become alive

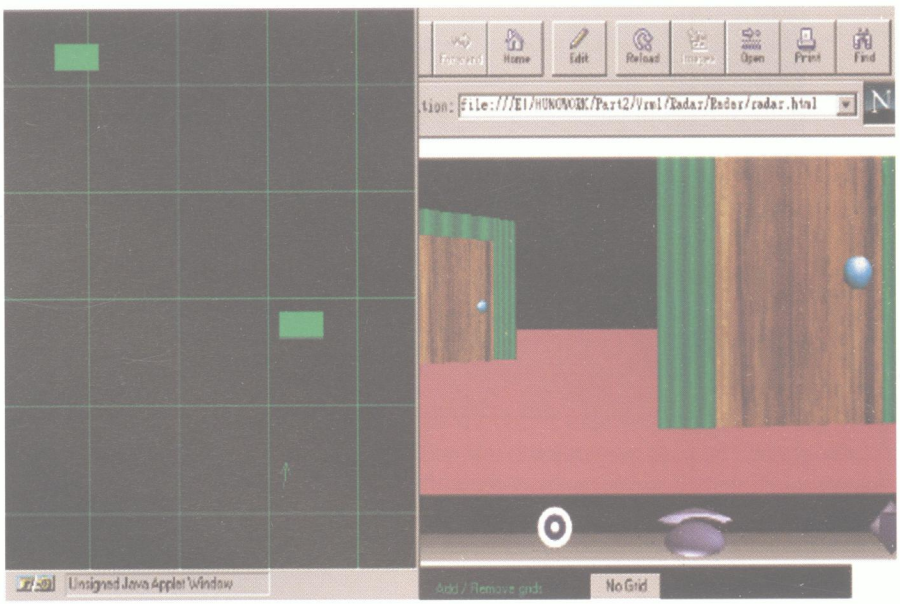




while invoking a JAVA window capable of doing anything JAVA can do. In our case, this window contains an animation and a dialogue box that introduces the VRML world.

In fact, this world can be transformed to any world, say, a virtual bank with a teller behind the counter, so that when customers click on that teller, a corresponding JAVA window is popped up to handle the customers' queries, the advantages being very obvious as everyone will be attracted by a 3D world which is the natural way of perception.

4. The Radar Room

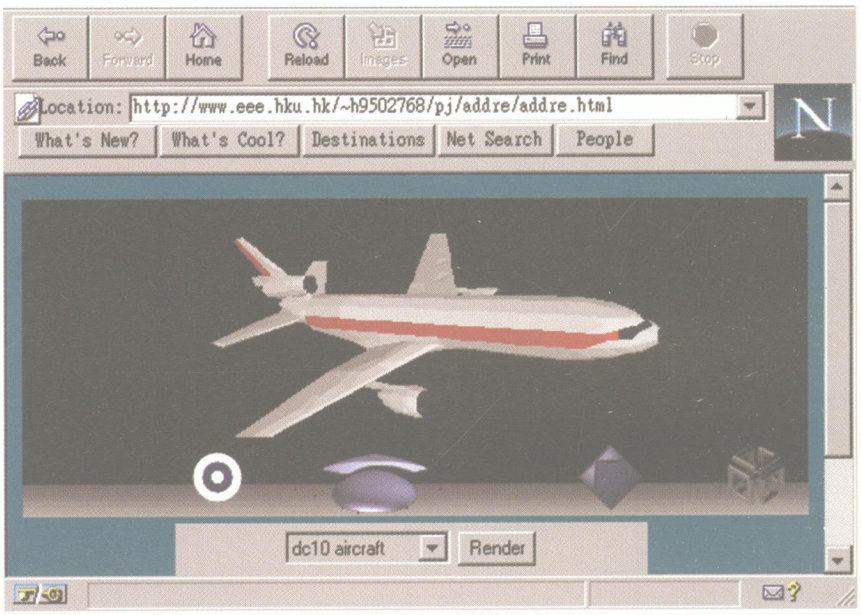


One having the experience of traversing a virtual world may have the feeling of losing oneself. After several translations and rotations in all 3 dimensions, the position or the orientation of the viewer is usually left undetermined. This web page comes out with the idea of solving this problem. It contains a JAVA radar window showing the current coordinates of the viewer. As the viewer wonders, the arrow will keep track of the displacement accordingly. For convenience, it will also change the scale as the viewer moves in and out. Inside the applet, there is a button named "Grid" or "No grid". As the button is clicked, there will be grids protruding from both the x- and y-direction as viewer's visual aid. The grid will also rescale if the user walks in or out of prescribed region.

In this radar window, JAVA is used to manipulate the coordinates of the users to perform the displacing, rescaling and the griding effects. The x-, y- and z-coordinates of the viewer is passed from the world to JAVA every time there is a change in the position. Alternatively, one can further perform a sequence of task if the viewer enters some predefined coordinates like invoking pre-defined animation, adding ground-and-sky backdrops to scenes etc.

5. The VRML Showcase

This site is reached by one of the scrollable image buttons on the first user interface.

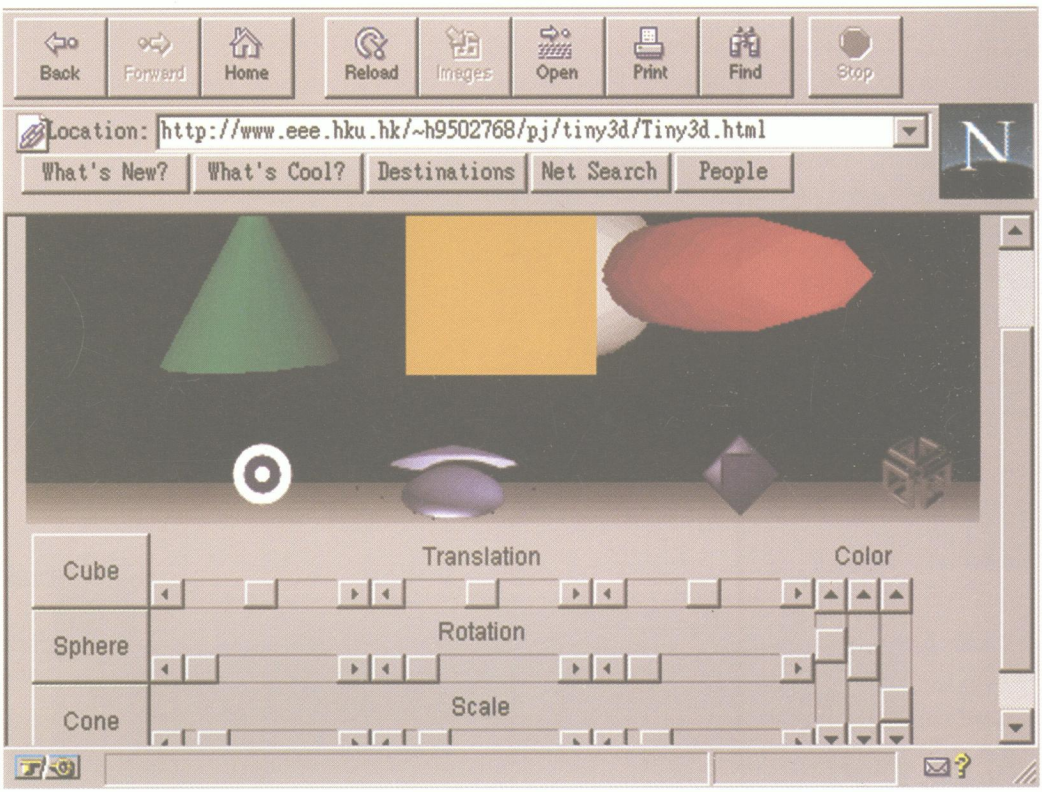


This world together with the JAVA stuff below is given the name showcase because the JAVA applet does the job of searching through the mSQL database and retrieving the corresponding URLs (universal resource locator) of the virtual worlds available in the database table. As referenced in the database viewer before, the existing models are added to the JAVA selector item and shown to the user upon his/her hitting the 'render' button. An aircraft model is displayed here as a result of such action.

Though the principle is simple, this function is notably useful when an institution or a company has various models of a particular product that it wants to show to the public or customers. Everyone all over the world with the appropriate plugin can then access the model and examine it three dimensionally. This can substantially save the cost of printing tons of catalogues or brochures and also help protect the environment. Another possibility of its usage is a 24-hour, easily accessible electronic online museum.

6. The Simple VRML Authoring tool

In order to further arouse users' interest in VRML, here is a very simple but powerful virtual world editing tool which utilizes the communication capability of the virtual world with EAI.

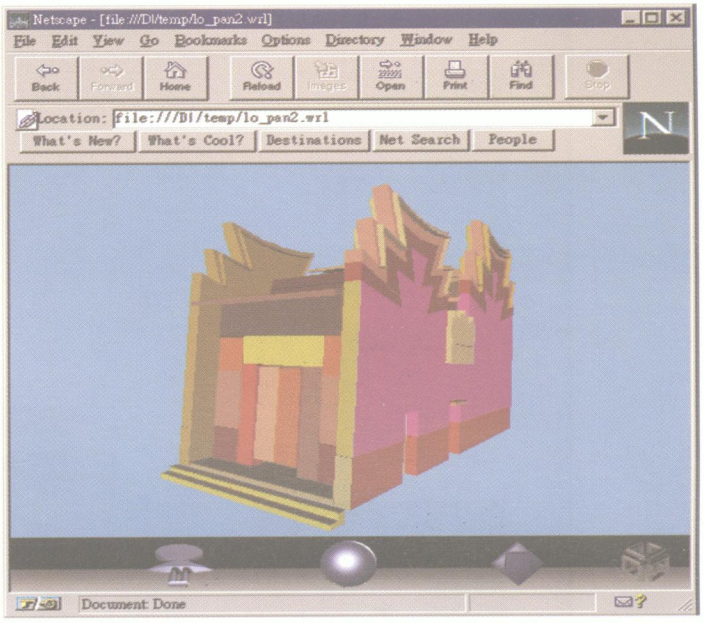


As its name implies, this site provides the users with an opportunity to experience the joy of building VRML worlds using three basic constructs, namely cube, sphere and cone. Users can dynamically add in any numbers of the basic constructions into the Cosmoplayer window, followed by controlling the position, rotation, scale and color of every construct. With a little imagination, an unlimited number of original virtual worlds can be built by anyone.

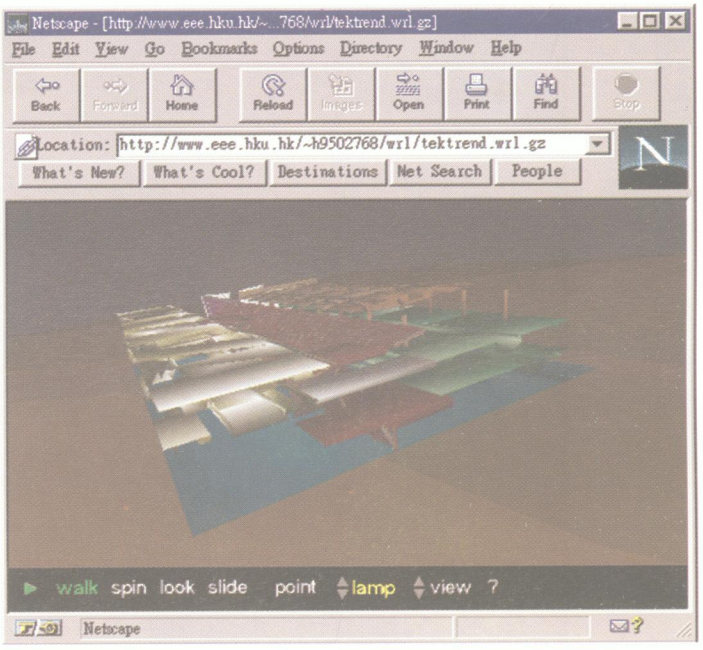
A direct implication of this applet is that one can extend it to hold more constructs like the roofs and walls of a building so that users may even construct a building out of their own design.

7. The Lo Pan Temple and the new Airport Building

To preserve buildings of historical value and to foresee some remarkable buildings in the future, one can rely on VRML. In our case, the Autocad files of the famous Lo Pan Temple located in Causeway Bay, Hong Kong, and the future new airport main building at Chep Lap Kok are transformed into VRML worlds so that everyone over the world can take a tour inside them, without going to the physical location. If he/she would like to get some background knowledge about the place before the visit, such function is extremely useful for the Tourist Association in promoting the landmarks of local country.



The Lo Pan Temple

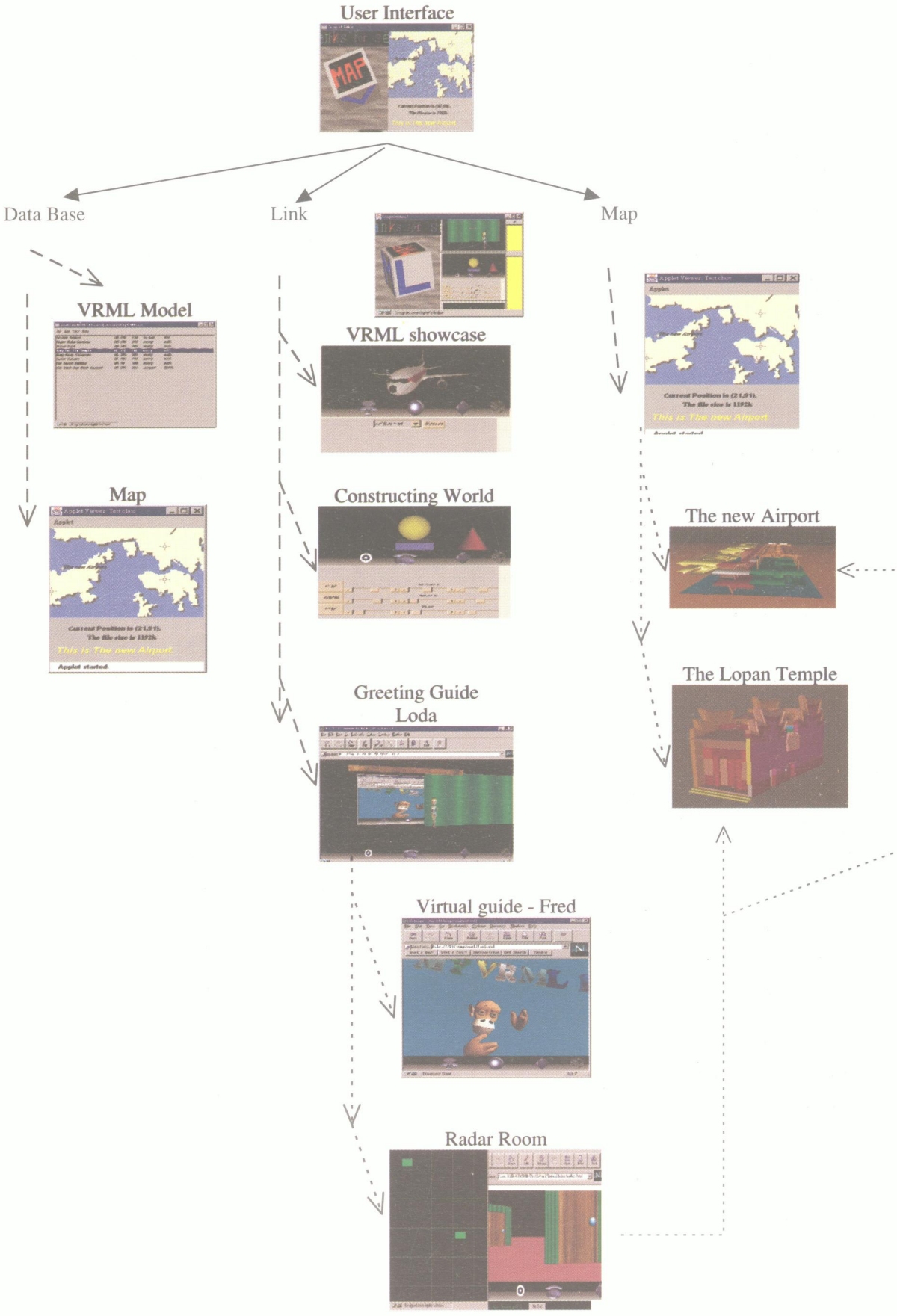


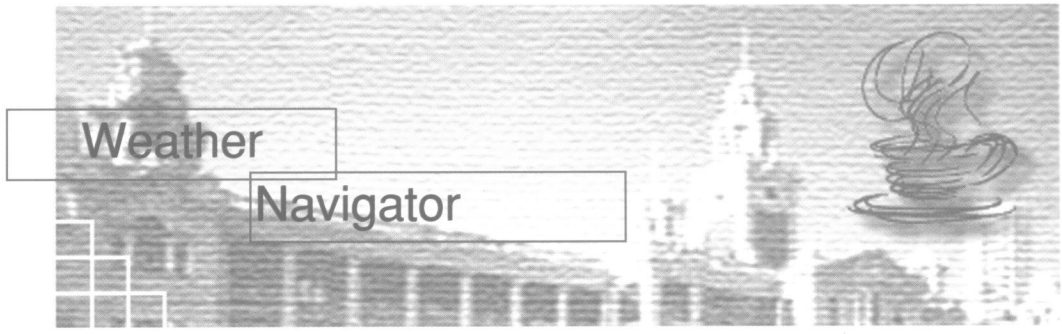
The New Airport Main Building

Conclusion

As an elaboration, this project can actually be extended to the development of a virtual world in which users from all over the world can interact with each other. Such an idea is completely feasible because JAVA provides the full capability of socketing and networking, along with the newly born EAI which eliminates any remaining impossibilities. A tentative idea can be different users controlling different virtual beings inside a world and do all sorts of interactions or transactions just like everyday life. Such system will also be found useful in the commercial sector because purchasing is made a great fun with the addition of a new dimension.

Appendix


User Interface
Data Base
Link
Map
VRML Model
VRML showcase
Map
Constructing World
The new Airport
The Lopan Temple
Greeting Guide
Loda
Virtual guide - Fred
Radar Room



Authors

Lau Chun Kwok
Sin Tung Yiu
Sum Lai Tak
Sze Tse Leung

Abstract

This paper describes the implementation of an internet based information system for the presentation and enquiry of the weather conditions all over the world.

The whole system consists of two main parts: the *Hong Kong Map* and the World Map. Under the Hong Kong Map, the weather conditions of many locations in Hong Kong can be obtained by just moving the cursor to that location. Under the World Map, the weather conditions of many major cities in the world can also be obtained.

The weather information can be forecasted and presented in tabular and graphical forms.

Introduction

With the rapid development of the world wide web and the fast growth of internet users, it will be a good idea to provide online information for different users to access through different platforms.

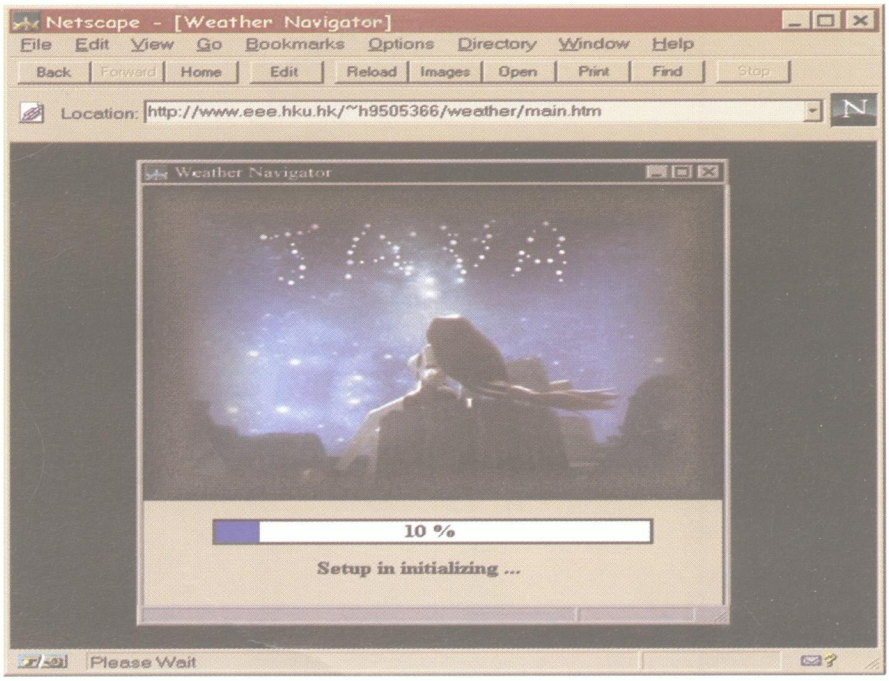
The Javatized Weather Navigator is an information system for enquiring the weather conditions all over the world. It provides real-time weather information which ordinary html homepages cannot do.

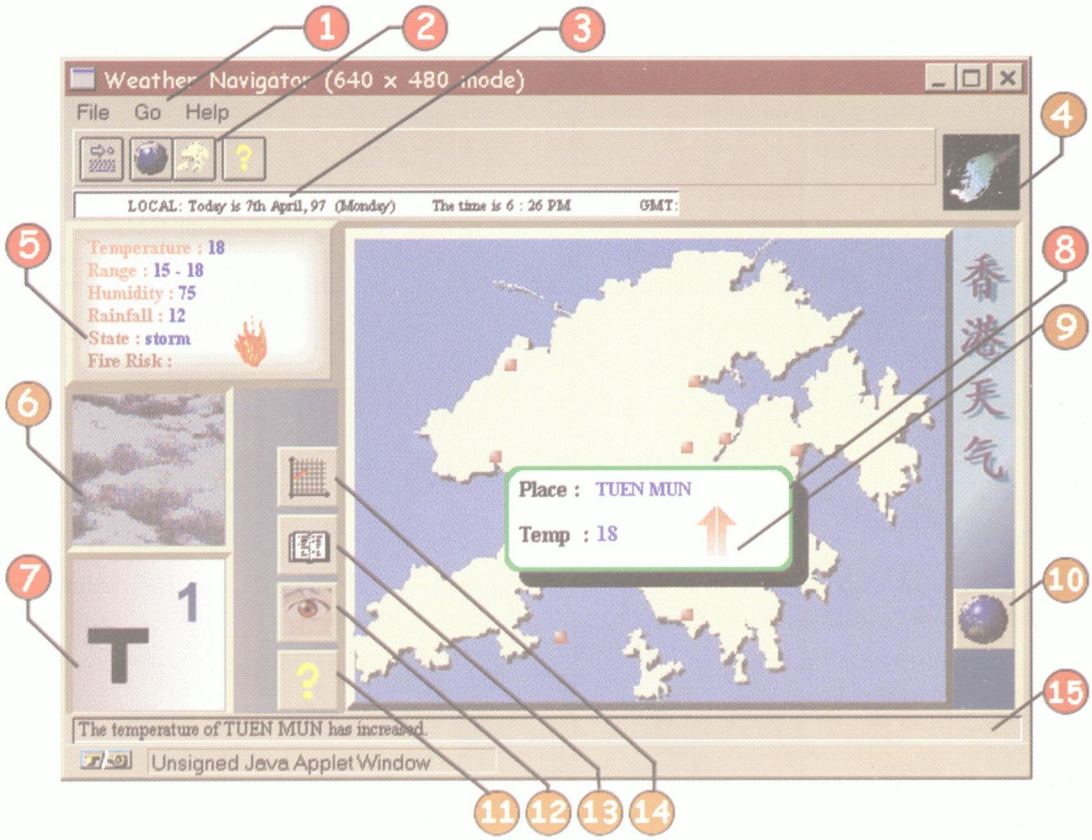
By using the Weather Navigator, one can get updated weather information in Hong Kong or in any major cities all over the world.

1. Overview

When the Weather Navigator (Navigator.java) is about to be loaded (called from main.java), the following display will be seen, showing the initialization is in process.

After the initialization process, the Hong Kong map (Hkmap.java) will first be displayed which is shown on the next page.





Hong Kong map (Hkmap.java)

No.	Remarks
1	Menu bar
2	Image button bar (Toolbar.java)
3	Ticker tape showing local time and GMT (Ticker.java)
4	Animation that becomes active when an event is happening
5	Display of current weather condition
6	Animation showing the general status of Hong Kong
7	Display to show any special remarks
8	Dialogue box displayed (box.java) when the mouse is point to one of the location on the map (Local.java). The dialogue box shows the place and the current temperature
9	Animation showing that whether the temperature has increased, decreased or remained unchanged when compared with previous hour.
10	Animated image button (imgButton.java) for going to the World map
11	Animated image button --- Graphical representation of previous records
12	Animated image button --- Searching past records
13	Animated image button --- Three day's forecast of Hong Kong weather
14	Animated image button --- Help
15	Status bar showing any additional information

World map (Worldmap.java)

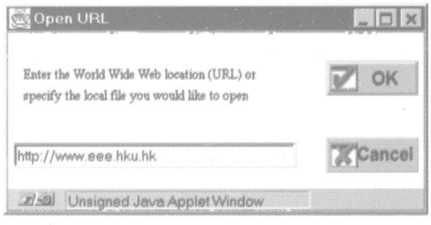
No.	Remarks
1	Menu bar
2	Image button bar
3	Ticker tape showing local time and GMT
4	Animation that becomes active when an event is happening
5	Display of current weather condition of a particular city (board.java) when the mouse is point to that particular city (City.java)
6	Animation showing the general status of that particular city
7	Dark region showing that it is at night time (implement RGB ImageFilter)
8	Bright region showing that it is at day time
9	Animated image button (imgButton.java) for going to the Hong Kong map
10	Animated image button --- Graphical representation of previous records
11	Animated image button --- Searching past records
12	Animated image button --- One day forecast of all cities
13	Animated image button --- Help
14	Status bar showing any additional information

2. How to use

The Weather Navigator is a user friendly tool because many of the operations are similar to the basic window application, which takes little time to learn the operation.

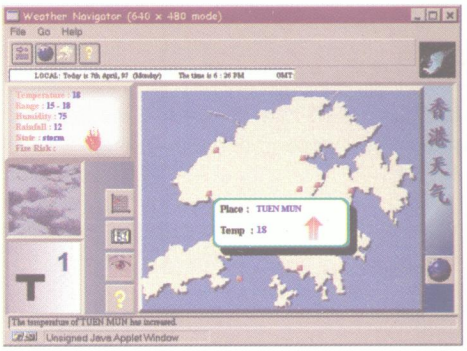
The Weather Navigator consists of two sections, *Hong Kong section* and *World section*. We can switch between these sections by the HK map button, world map button in the toolbar(2) or Animated image button (9) or (10).

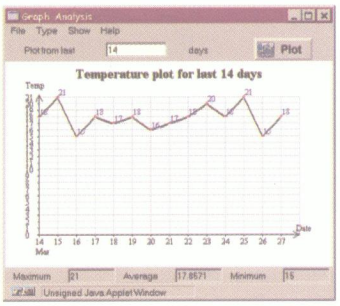
As the Weather Navigator can work independently with the browser, the URL of the web page can be changed using the first image button on the toolbar. When the button is pressed, the following dialogue box appears. Then the URL can be entered.



Hong Kong Section

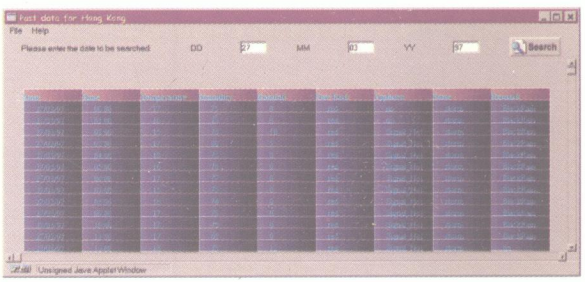
After initializaton, the general weather condition in Hong Kong will be displayed. If the weather information of a particular location is required, we can move the cursor to the spot of that location, then the temperature values and changes in temperature values will be displayed near that location.

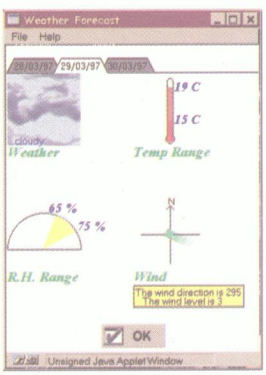




When the graph button (14) is pressed, the following new window appears. This window plots the past temperature data against date. The range of date can be changed from a week to a month. The menu bar in the top can be used to change the preferences in the graph. For example, the temperature values can be displayed or not by using the menu bar.

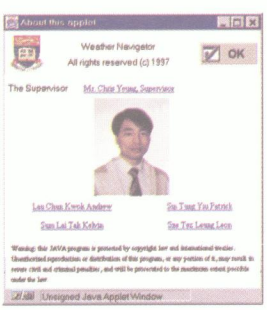
When table button (13) is pressed, the past temperature data will be shown. The table displays weather information such as temperature, humidity, rainfall, fire risk, etc.





When the forecast button (12) is pressed, the forecast weather information will be shown. This window shows the weather condition, temperature, humidity and wind direction.

The help button in the toolbar or button (11) gives a brief help information in a new window.



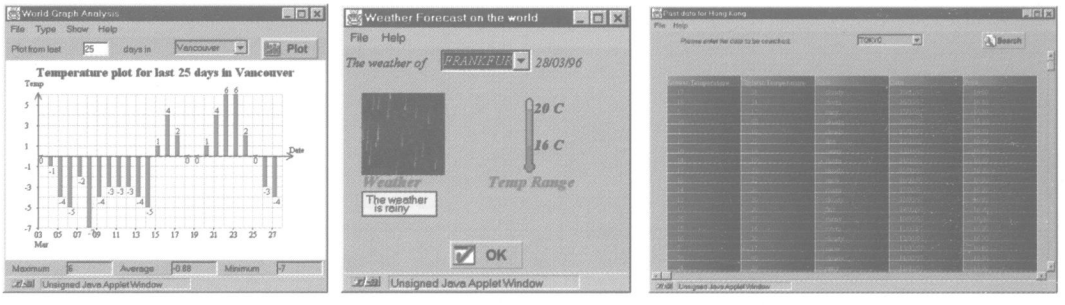
The about window can be called from the help window which shows the project group members and the supervisor. There are links which bring to the corresponding homepages of that person.

World section

This is similar to the Hong Kong section, therefore only different features are described.

In world map, we can see which cities are at day time or night time by the brightness of that cities.

In the plot graph button (13), we can choose the different cities by the choice menu at the top of the window. The same operation can also be applied to the table button (12) and the forecast button (11).



3. Program Structure

The Navigator object can be regarded as the kernel of the Java program. It communicates and controls all other peripheral components such as HKmap and Worldmap. The hierarchical diagram of the program is illustrated in the figure below.

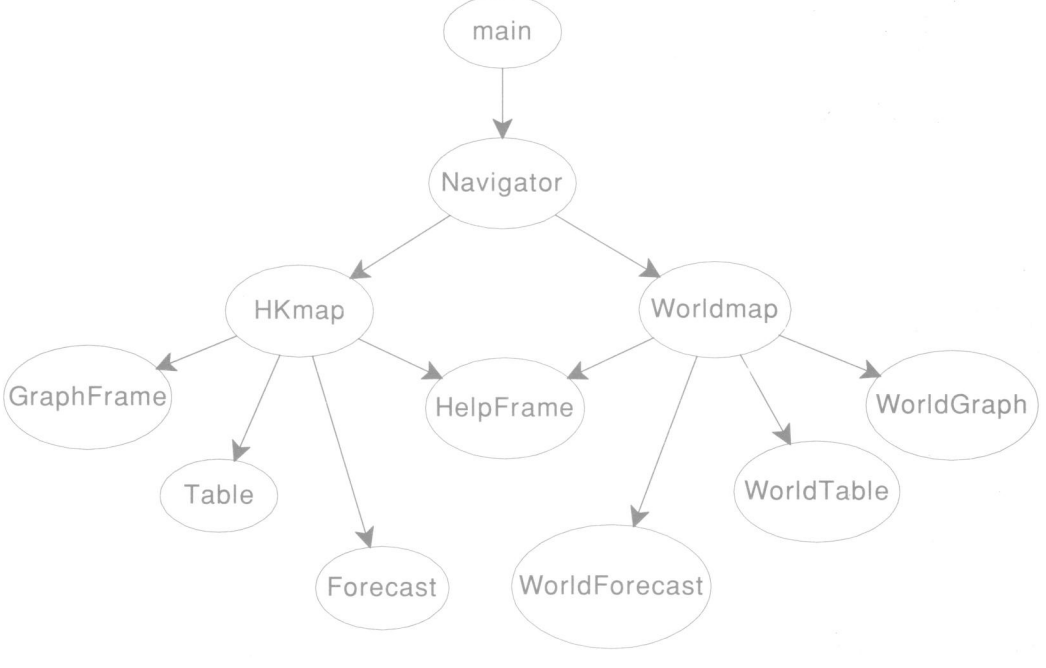


Figure Basic program structure of the Weather Navigator

The HKmap and Worldmap objects are inheriting the Navigator and they can be switched between the browsing window when a button is clicked. Under these two maps, there are other three components opening a new window for graphical analysis, tabular form viewing and weather forecasting. In addition, a common object for online help can be called by both maps. The HKmap and Worldmap

actually pass the request for opening a new window to Navigator so that the calling algorithm is simpler for all peripheral components and all windows will be destroyed once the Navigator is closing.

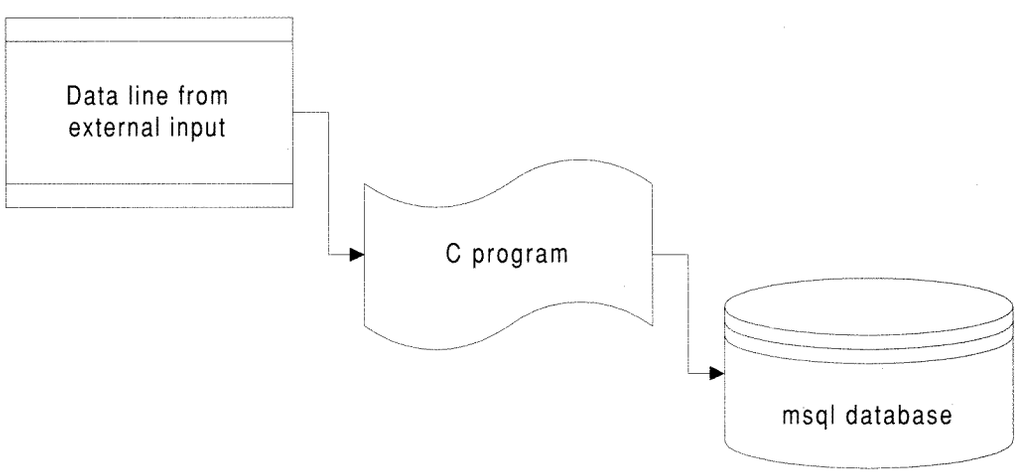
Java language uses the concept of object oriented design so that each module serves a particular function and the underlying principle is totally transparent to other modules using it. One of the advantages of object oriented programming is that the program is reusable. For instance, the imgButton object creates a clickable canvas which serves as a button to accomplish a special task. This object is reusable because it can be called by other objects for many times. Furthermore, this is flexible since some of the buttons can have animations on the buttons while others may not. This flexibility of the module can be done by the method of constructor overloading. In fact, all the Java programs in Weather Navigator are object oriented and most of them have been reused for many times.

For the sake of saving time in running the program, the initialization process takes up the time for accessing the database and loading animations. The main program therefore divides the loading into a sequence: It first loads the Navigator object to open a new window and then starts to get most updated weather information from database. Eleven districts in Hong Kong and eighteen major cities in the world will be loaded onto HKmap object and Worldmap object respectively. Next, the world map is filtered to shade for night time area and all animations for weather conditions are loaded simultaneously. Finally, the HKmap object is loaded onto Navigator window and the initialization is complete.

Each object inheriting HKmap or Worldmap will fetch information from database again when clicked by user. For example, the graphcial temperature analysis and weather forecast. This can ensure accessing the most up-to-date data for weather analysis and the user can then know the information in real-time.

4. Future Development

The Weather Navigator need to get data from the msql database. As the weather data will change in some time interval, the data in the database do not change at the present state. Therefore, a program is needed to update the msql database. For example, a C program is used.

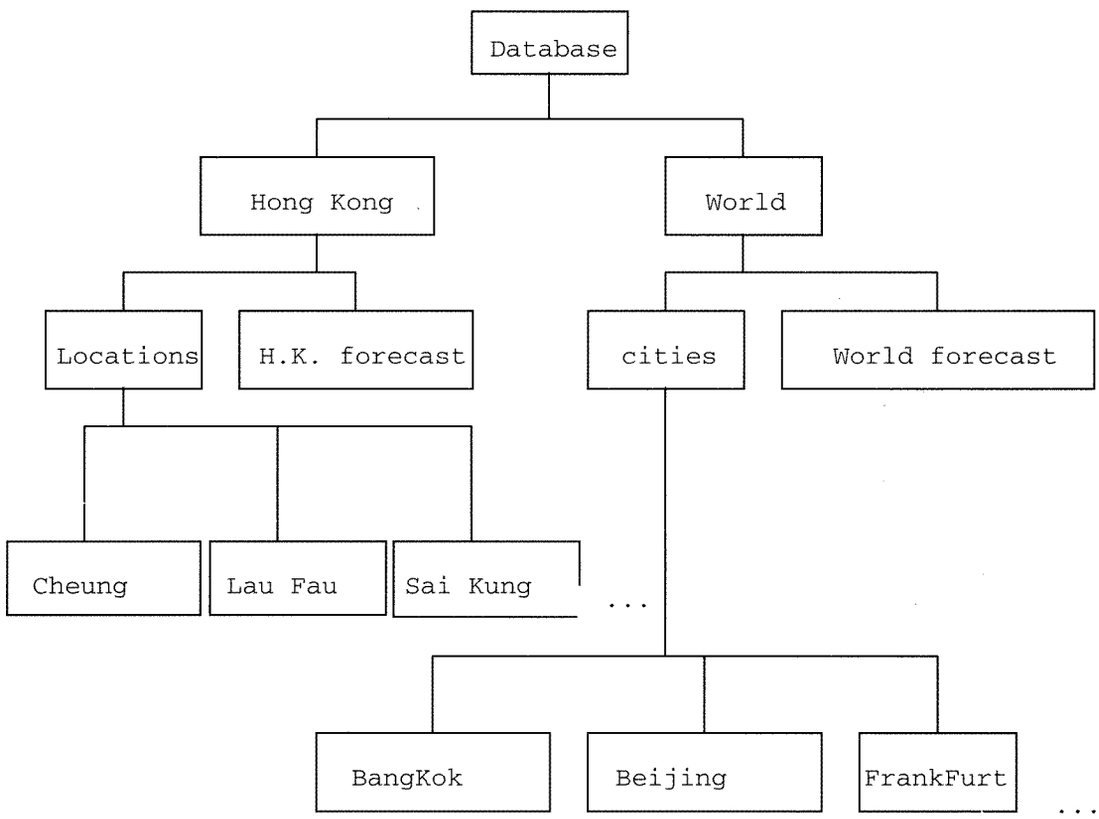


Other features such as Tidal information, Satellite information diagram, Thunder Storm warning, Landslide and flooding warning services, Air pollution index firm, Data search ... are trying to be added so as to make it more interesting and useful.

As multimedia are becoming and more popular nowadays, audio output plug-in will also be added into the Weather Navigator later. For example, the temperature value will be pronounced upon request.

5. Database Organization

The following graph shows the database structure for the information system design. All the data the system can access are put in a database called Weather. Inside the database, there are a number of tables arranged in the order as depicted by the following graph. The total data are divided into two parts: the data about Hong Kong weather conditions and the data about the world weather conditions. Under each part, the data are subdivided into two parts: data used to make weather forecast and data showing the real-time weather conditions.

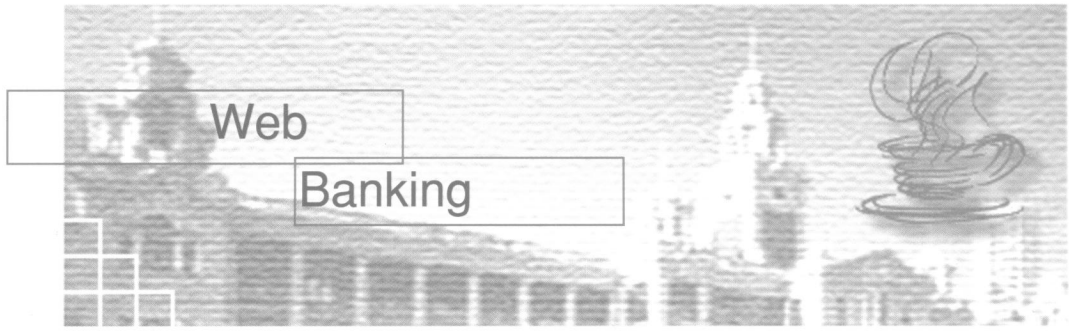


Conclusion

In this paper, we have introduced and explained an enquiry system, the Weather Navigator. Its program structure and database structure have been thoroughly discussed.

This enquiry system is highly flexible as the overall design is object-oriented and the Javatized program can support different platforms with the msql database support.

It also provides an interactive way for the users to enquire the weather conditions through a user-friendly graphical interface.



Authors

Ko Ming Him
Yung Chor Ho
Lam King Fai
Wong Sze Hang

Abstract.

We discuss in this paper on what we have achieved in the project of Internet Banking, which are the transaction service, mortgage query and credit card application.

Introduction

With the blooming Internet business services, it is inevitable that Banking service is about to stand a chance to show off in one of those. Generally speaking, it includes all sorts of banking services such as accounts creating, the common accounts balance checking and transaction, buying and selling of foreign currencies, credit card application, home mortgage query, and many other services that one can enjoy in a real bank.

To fulfil the promising development of 'Internet Banking', Java, being an evolution from C++ and now the leading and searing language, seems to be the most suitable to bear such a mission.

What turns up in the project will be major on four areas. They are the *user's identity checking system*, the *credit card application, hone mortgage query services*, and the *transaction service*.

1. User's Identity Checking System

Before a user can use our service, his/her identity must be checked by our system for security purpose. In the user's identity checking system, client and server approach is used to check the user's identity.

A Sample Client Applet

The graphical user interface of the system is like this:

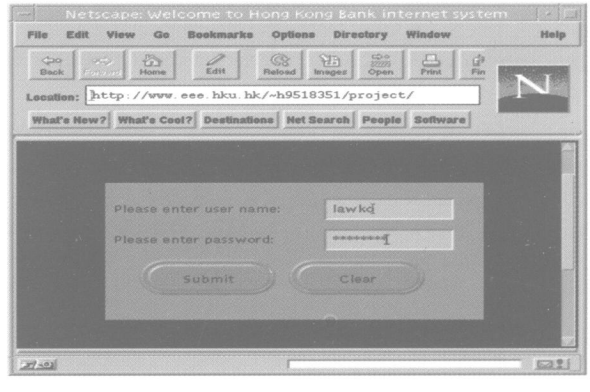


Figure1-1. GUI of the login system

The application allows a user to enter his/her username and password. This application will send all the information to a server (to be discussed later in this paper) through socket by building data stream. When a user click the submit button, he/she will expect a result coming from the server. If his/her identity is invalid, a message will be given out on status bar of web browser. If the network is busy and a message from the server does not come in a reasonable time, time-out error will be shown on Java Console too.

When a permission is granted with an appropriate message from the server, a main menu will come out to have a large Hong Kong Bank logo. There are three buttons in the main menu for the user to choose, namely *Credit Card Application, Home Mortgage* or *Transaction Services.*

Suppose the user presses one of the buttons, the username of that user will be passed to the homepage of one of the three applications by JavaScript Object. This uses the LiveConnect technique for Java applets to communicate with Java Script functions. The username is entered into the Java applets of one of the three applications in the form of a parameter. However, there is a major limitation of LiveConnect, the JavaScript Object which has not become a standard at present, can only be supported by the Netscape but not the Internet Explorer..

netServer - The user's identity checking server

The user's identity checking server, codename 'netServer', is a hidden process running in one or more web sites. In our project, the server is running in a web site called indy14.eee.hku.hk. The system is illustrated by Figure1-2 as shown below:

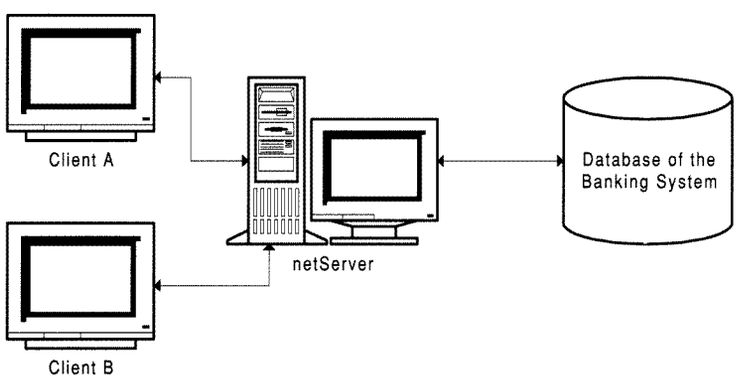


Figure1-2. An overview of the user's identity checking system

When a user enters our web site, the client applet is loaded into his/her local computer. The user will be connected to the server by clicking a submit button in the client applet and a new thread will be created in the host machine. All the information transmissions are initialized and the communication between client and server is built up. Then, the password and username is sent to the server by data stream. The server receives the signal from the client and searches the database for identity verification. Finally, the server responds to the client by sending an appropriate permission message.

In the server side, an interface, codename 'Dbaccess', is set up for searching the database. It passes only one boolean return date type method called Connect. This interface can allow many different user's identity checking object to be implemented as shown in Figure 1-3.

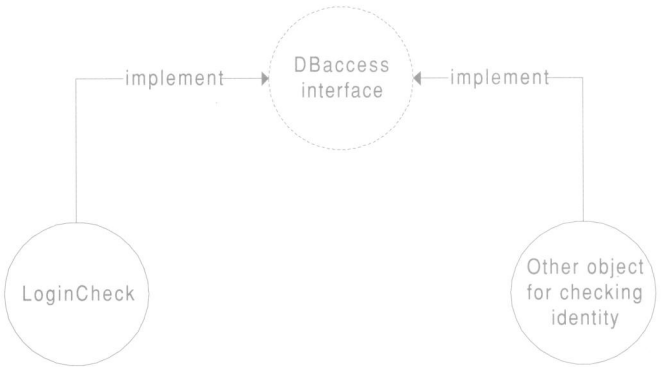


Fig.1-3 Object Diagram for searching database for user's identity checking

2. Credit Card Application

In the early days, a paper credit card application form must be filled before a new credit card can be issued to a person, which is very inconvenient as well as not environmental friendly. Therefore, we describe in this paper how the credit card application can be implemented by Java so that people can apply his/her card in his/her home.

Graphical User Interface

Figure 2-1 shows the applet of the credit card application form.

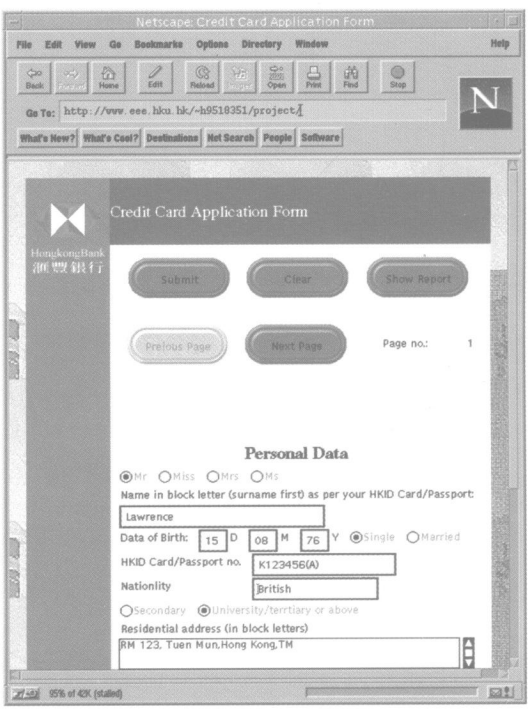


Figure 2-1. A sample of an applet of the credit card application form

There are five main panels in this applet. Two of them are button panels which hold buttons (called ImageTextButton). One of them is used as a back which hold everything. The other, FormMain, is a eight-page core part of the application form. Many objects can be extended to achieve different purposes such as different credit amount limit or services.

A report panel and an object name FormFront, will show the credit card information of a user, which can be seen when the applet is first loaded or the "show report" button is clicked.

A special button, object name ImageTextButton, is designed. This button is a sub class of Canvas. It has disable function and focus function which is similar to Java built-in button.

Core Part of credit card application applet

When a username parameter is passed from main menu to this applet, all the user information is loaded to the form so that the user does not need to type the information again. After filling in some necessary information, the user can click "submit" button to issue his new card. All the information of the user in the database will be updated and a new card will be mailed to him/her within 2 weeks or even earlier according to the bank's services. The user can read the report to ensure that his/her card is issued.

Another, object name generalTableAccess, is designed to handle all the data manipulations between users and the database. Many objects can be extended to access different tables.

Database Design

Two tables are used in the credit card application applet. They are *PersonInfo* and *creditAcc*. PersonInfo holds users' information with creditAcc holds users' credit card information. HKID card number is used as a primary key field to search through the two tables. Once the applet is loaded, a HKID card number is selected by using the username parameter and all the remaining information related to that user will be found by using that HKID card number as a search key.

Figure2-2 shows the overall structure of the credit card application system.

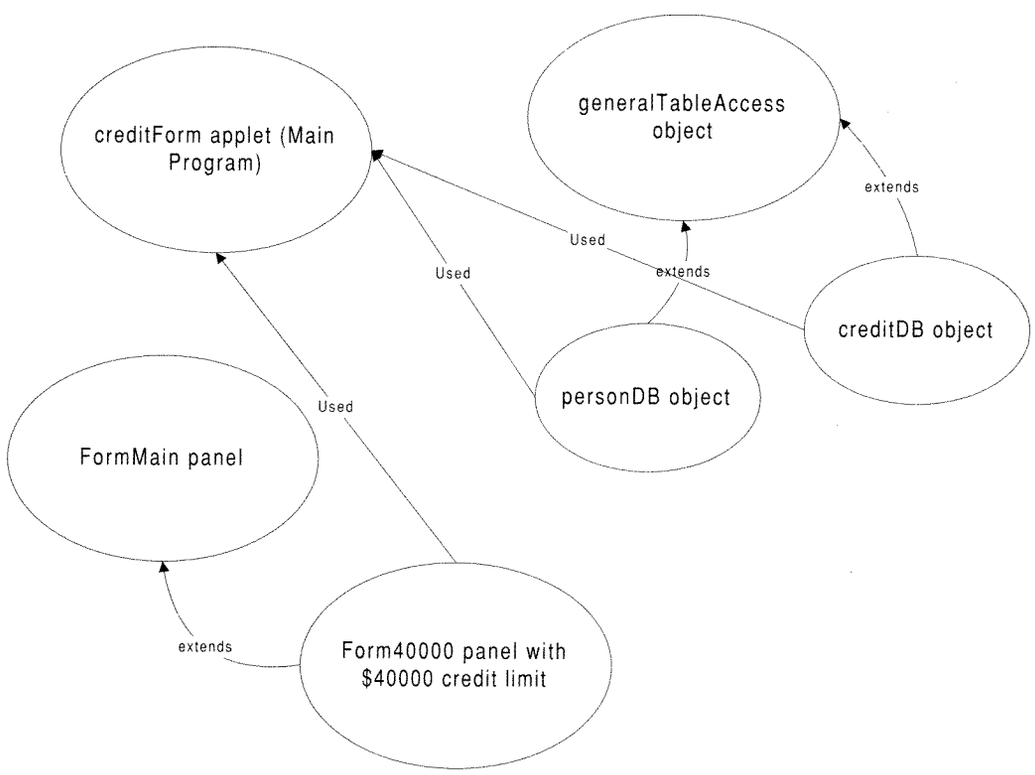


Figure2-2. Structure of credit card application system

3. Home Mortgage

Before one buys a new house, it is very important to check the amount of mortgage he can afford. Therefore, the purpose of mortgage query section is to create a comprehensive schedule showing the amount of mortgage he can afford according to his monthly income and expense.

User entering the Web Banking can learn the amount he will have to pay in this query section.

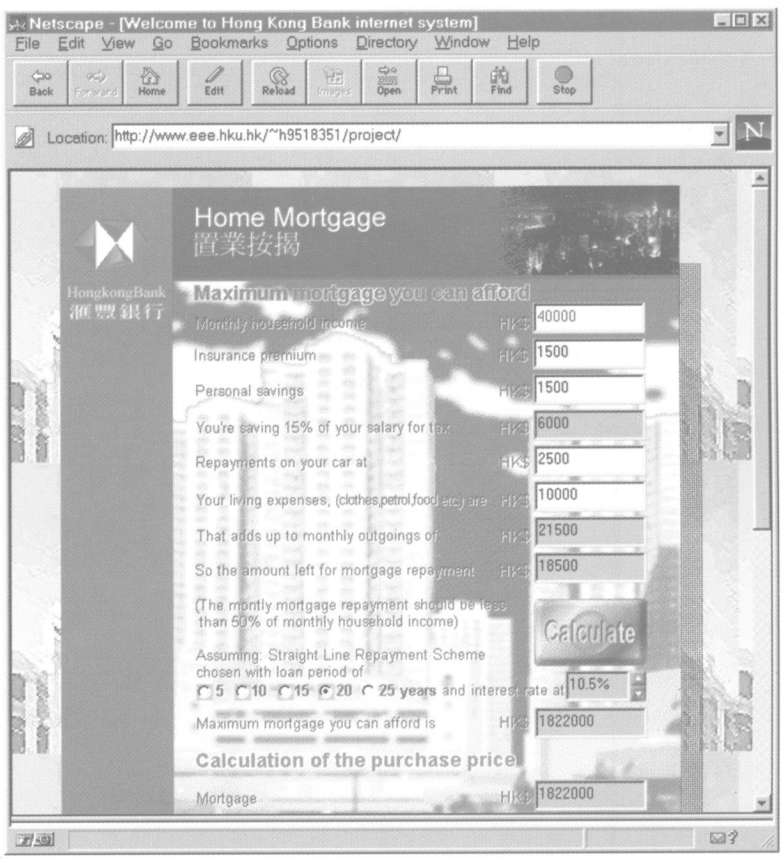


Figure3-1. Sample of the Home Mortgage Applet

The first item is the income fetched from the database containing all personal information of clients, for the sake of convenience such that the user do not have to type it again.

The following are the default values of the various monthly expenses, such as the insurance premium, repayment on the user's car and living expense.

Different combinations of loan period and interest rate can be made according to individual users.

Once all these values are found satisfactory, and with a single small click on the "calculate" button, a bunch of figures results. They are down payment, assignment, agreement, stamp duty and agency fee, all of which are real-time process based on the up-to-date algorithm.

Hierarchy of Home Mortgage Computation

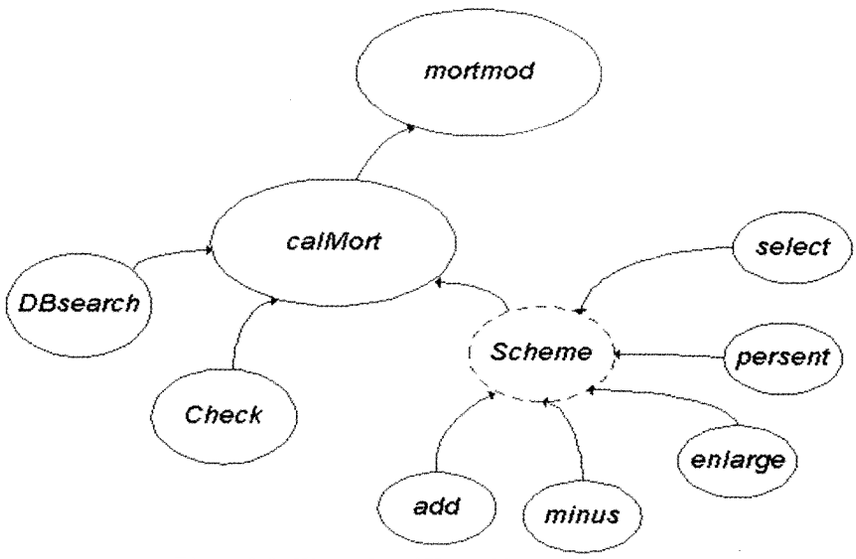


Figure 2-2. Hierarchy of Home Mortgage Computation

CalMort is the main class in calculation of mortgage. For completeness and perfection, functions like stamp duty, assignment, agreement, agency fee, etc. are created, which complicated the program structure as object classes like Dbsearch dealing with the database and Check verifying the validity of the user input numbers are also embedded.

Database Design

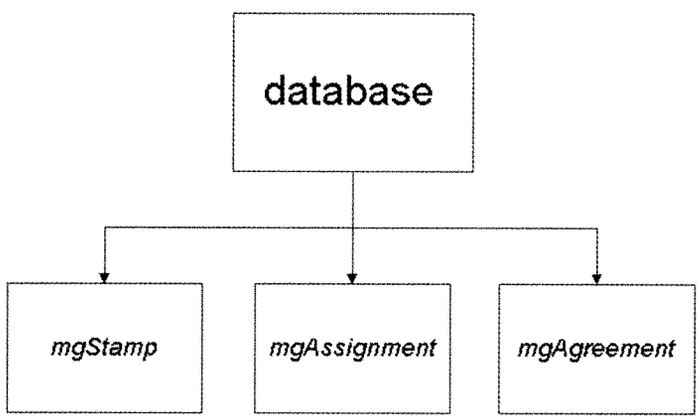


Figure 2-3. Database Organization

Tables of legal fee, stamp duty, assignment, agreement were necessary for online computation.

4. The Transaction Service

The transaction service is one that transfers money between accounts of a particular client, this function makes money transaction in the bank easy.

The main features of this transaction service is its user interface, the user interface is designed to be user friendly. Also, it contains graphics and animation so as to attract the clients use this service. To make a successful transaction, a user only have to select two accounts from selection buttons, and then select the direction of transfer and type in the amount that he wants to transfer. When everything is ready, all he has to do is to press the confirmation button to confirm the transfer. The whole procedure is simple!

Figure 4-1. Sample of the Transaction service applet

To use this program is simple, once the user has logged in, choosing the transaction service button, all his saving accounts in the bank will be displayed automatically onto the buttons on the top of the applet. If the user wants to see the account information, he just has to press the button of the account number, then the information will be displayed on the panel just under the account buttons. Two account information can be viewed at a time.

If the user wants to transfer money between the two selected account, first, select the direction of transfer by pushing the arrow button on the transaction service panel just under the information viewing panel. Then, type in the amount to be transferred, make sure that there are at least ten dollars in the account after. If the amount typed is out of range, the transaction will not take place and an error message will be displayed at the dialogue at the bottom of the applet. When everything is ready, just push the confirmation button and the transfer is completed.

For the program of transaction service, we divided the program into two main parts :

1. Database Access
2. Graphical interface

1. Database Access

There are three main modules for accessing the database. The main module is called GenealTableAccess, it is used to perform basic read/write of the general relational-database structure. And the main module is extended by another two modules, *PersonDB* and *SavingDB*, the two modules perform no operation but only contain the information and definition of a particular table in the database structure. Thus the two modules will be called upon when a particular table is access.

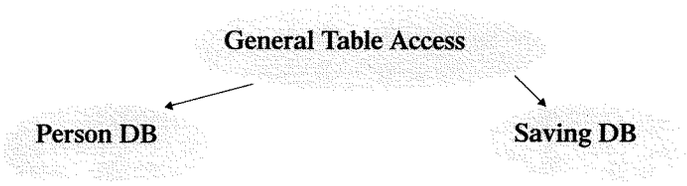


Figure 4-2. DataBase access structure

2. Graphical Interface

The graphical interface is the main part of this program, it mainly contains five main modules, they are **ColorPanel, MovePanel, TouchButton, WordPad** and **InfoProvider**. Other modules are just extension of these modules.

ColorPanel : The panel that will automatically be pressed when the cursor enters, whereas, it will pop up while cursor exits. The color of the panel will also be specified in the calling interface.

Modules that extend ColorPanel is - InfoPanel (display information of a account)

TransPanel (Panel for money transfer)

MovePanel : The panel with the background is a moving picture.

Module that extends the MovePanel is : SelectPanel (Used to contain the account button)

TouchBotton : As its name implies, when the cursor is moved to the button, it is pressed automatically.

When the mouse button is pressed, the white floating spot will be pressed and turns red, to restore, just press the mouse button again.

WordPad : The moving LED Display, which is used to display the pattern stored in module Pattern.

InfoProvider : The panel that is used to display the necessary information. It acts as a communication channel.

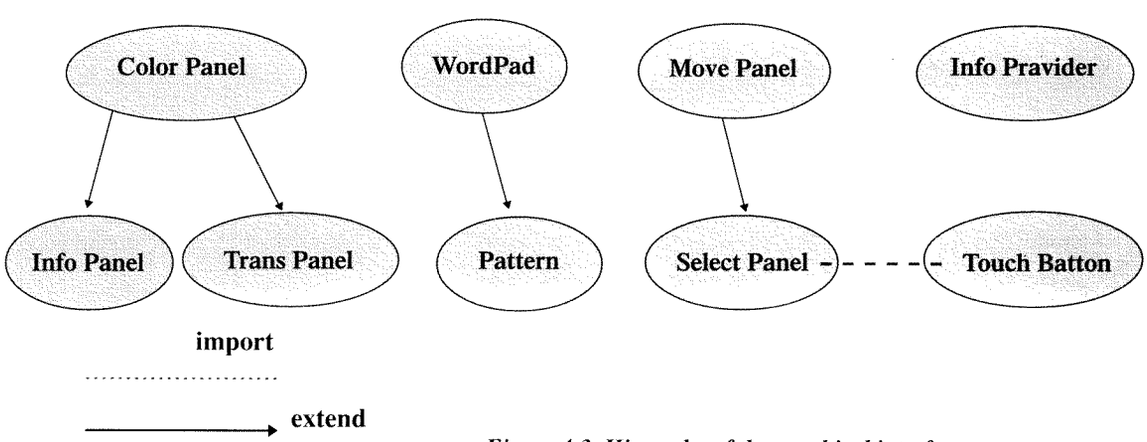


Figure 4-3. Hierarchy of the graphical interface