DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
THE CHINESE UNIVERSITY OF HONG KONG

LYU9901

- TRAVEL NET -

FINAL YEAR PROJECT REPORT 1999 -2000

SUPERVISOR
PROFESSOR MICHAEL R. L YU

Prepared by
Ho Chi Ho Malcolm (student ID 97618593)
Lau Chi Ho Arthur (student ID 97590853)

Date of Submission: 2nd December, 1999

Table of Content

Abstract

No one can deny the rapid development of Internet. It is a trend that many kinds of business are now taking the form of operation from traditional mode to the new e-commerce model. In this report, we will summarize the research and work done in the 1st semester of our final year project – TravelNet, which is a typical e-commerce application for travel agency. We will first provide an overview of the project and a brief discussion on nowadays e-commerce applications. Then we attempt to analyze different approaches on building a Web application. Next, we will briefly describe the facilities and functions provided by TravelNet, followed by a chapter, which discusses the system design and implementation details. A chapter is also devoted to discuss the security issue of TravelNet, particularly on SSL. Last but not least, we will introduce some possible improvements on our project in the Summary and Future Work chapter.

Chapter 1: Introduction

1.1. Project objectives

In this project we will focus on application -level programming to develop a database transaction service: TravelNet. TravelNet allow s users to book for a travel itinerary over the Internet. The travel itinerary includes airplane ticket reservation and hotel reservation.

We will use Oracle and Java Servlet to develop this project. The information will be stored in different databases. The programs will try to collect information among all the databases, then search for the best item that meet clients' needs. Flexibility study on the agent technique to be used on our system will be studied in this project.

The project will include the integration of payment system, as it is an unavoidable part of an e -commerce application. Payment system in research project and real life may be integrated in the system built.

On the large collection of components (databases, payment system), it is effe ctively useful for the whole system to be distributed. Another objective of this project is to develop this application in a distributed manner. CORBA technology will be used to achieve this in the coming term.

1.2 Background

1.1.1.E-Commerce

Internet is growing every day. Not even the number of users is increasing, but also the number of ways it provides services. One of the most important issues is doing business on the net. Nowadays, more than half of the nation wide companies already had their own web si te and provided true servicing functionality. Researches showed that this type of electronic commerce on the Internet is making a great profit.

Because of the great population and popularity of Internet around the world, electronic commerce is keep growin g and developing. Together with the growth the number of users and companies involved, new Internet technology appears every day.

1.1.2.Travel Agencies

Travel agency is around the Internet today like Expedia, Travelocity, LowAirFar.com, Preview Travel, etc. This type of service provides a great convenient for individual or family travel for them to buy ticket online. It's not convenient for the travelers to check the price by consulting the airline companies and real life travel agency. The situation is similar in the case of booking hotel rooms. Online Travel agency can help them to collect and compare price instantly in order to give them a comfortable trip.

Traditional implementation of these online applications will be in CGI. CGI is very poor on accepting concurrent requests and, at the meantime, performance drop or lead to server down. New coming technologies like ASP (active server page) are good to handle simple request or generating dynamic page to user but the can't have a good integration with other system components.

What we are now going to do is to use Java Servlet to implement a travel agent. With it increasing popularity, great current performance, portability and good integration with system components, it is worthy for this project to be developed in this direction.

Besides the centralized approach in TravelNet, distributed approach will also be developed in the later version of this application. Since the project is in Java platform, this made CORBA integration possible and most of the components can be reused.

Chapter 2:
Approaches on web applications

2.1 Introduction

Internet has been a pool for people to communicate and share information and resources. It is easy to observe that many kinds of human activities have been adapted into an electronic format through the help of internet. Some significant examples include the business transaction, online chatting, etc.

Nowadays, the most popular form of information exchange and distribution is through the World Wide Web service that is built on top of the HyperText Transfer Protocol(HTTP). HTTP is responsible for exchanging text, graphic images, sound, video, and other multimedia files on the World Wide Web. The basic design of the web daemon services of HTTP is static, which means that for every request from the client, e.g. the web browser, it attempts to reference to a constant object on the server-side. The content of the object is constant. In this sense, dynamic data cannot be provided from the web service, which contradicts to the natural two-way communication between humans.

In order to accommodate the situation, the web server must be able to accept any type of request within its service context that is dynamic in nature. That requires a specialized program that can be triggered by client request to provide user-specified information. A typical example is the online query system. It is obvious that the query is mostly different each time so the web server cannot simply prepare all the result set for the request. The server-side program solves the problem in the case.

In this chapter, we will discuss on different ways of implementing web applications, which includes CGI(Common Gateway Interface), Servlet, ASP(Active Server Pages) and Java Applet. We will specifically compare CGI and Servlet in this Chapter on different properties to indicate our reasons on choosing Servlet instead of the popular CGI style of server-side programs

2.2 CGI usage for server-side program

Instead of being a programming language for writing server -side programs, it is in fact a common standard on linking the request from clients to the programs aside on the server. Basically, any program that can be executed on the server and can perform the following functions:

1. Print to the standard output ;
2. Read from the standard input; and
3. Read from environment variables

is capable of being a CGI program. However, due to the different popularity and efficiency of different programming language, only a few number of programming languages suits to be used in CGI. It includes the following:

- ✓ C/C++
- ✓ Fortran
- ✓ Perl
- ✓ TCL
- ✓ Any Unix shell (e.g. sh/csh/bash)
- ✓ Visual Basic
- ✓ AppleScript

The above languages typically can be grouped into 2 classes: the compiled language and interpreted language. For compiled language, like C/C++, it needs to be compiled first before it can be executed through CGI. For interpreted language, like Perl, it requires an installation of the interpreter such that when the program is called through CGI, the interpreter can start up to execute the CGI program . In a very broad view on the 2 classes, compiled programs are generally smaller in size and faster in execution, while the interpreted programs are generally more flexible and easier to program.

Since CGI is developed earlier than other approaches on providing server -side programming possibility, it is still a popular tool for making interactive dynamic webpages in different application in the business field and personal areas. The large amount of guestbook application is one of the examples.

2.3 Introduction to Java Servlet

A Java Servlet is a server -side component that is platform and protocol independent. Servlets can be used to extend the functionality of a Java -enabled Web server. Servlet can be imagined as a faceless applet. Servlets are load ed and invoked by the Web server in much the same way that applets are loaded and invoked by Web browsers.

Servlet can perform typical server-side processing. The Servlet can communicate with the client computer and it can also communicate with other rem ote, networked computers. In an n -tiered environment, your middle ware can be implemented as a Servlet. A three-tier architecture is illustrated figure below.

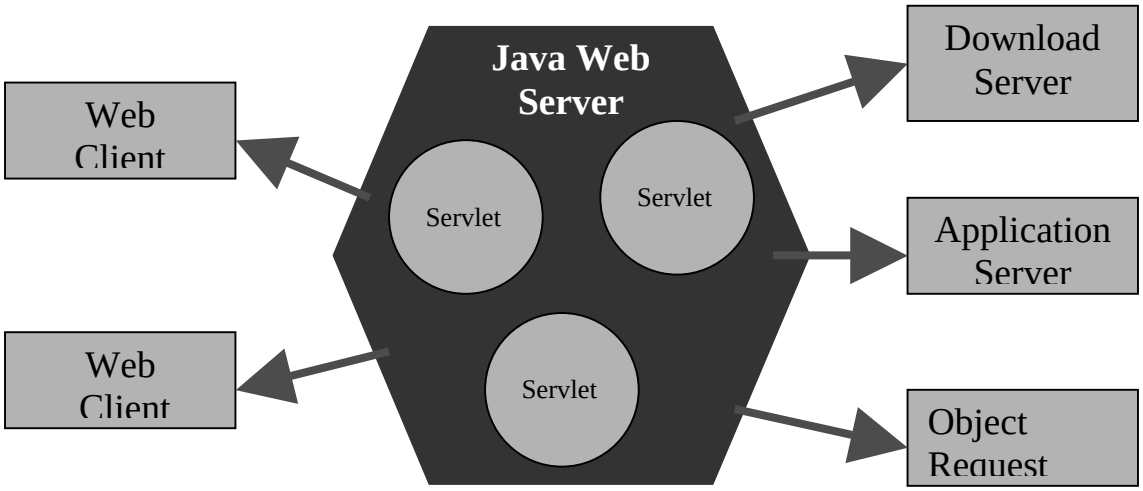


Figure 2-1:
3-tier client/server architecture using servlet

2.3.1.Servlet essential methods

Servlet has been designed in a life -cycle model. In the model, servlet is mainly executed through 3 stages. The associated methods to the different stages are init(), service() and destroy(). The following table gives a summary on the 3 methods.

Method Name	Method Description
- init()	Called only once when the Servlet is invoked for the first time. You can override this method to perform typical initialization routines such as initializing a counter or making database connections
- service()	Called by the Web server when the Servlet is request ed by the client. This is the main entry point for Servlet. You will place the bulk of your Servlet process code in this method.
- destroy()	Called when the Servlet is removed from the Web server. Destroy() is also called on each Servlet when the web serv er shuts down. You can use this method to clean up resource allocations and close any connections for sockets or databases.

2.3.2.Java enabled web server:

Servlet has to be run on top of a Java enabled web server. A Java enabled web server is a web server plus a virtual machine running in background. When request of a Servlet service is raised by client, the Servlet class will be loaded by Java virtual machine in background to the memory.

Since the Servlet stays resident in memory, it's very fast. Sharing static or persistent information across multiple invocations of the Servlet allows you to share information between multiple users.

2.4 Advantages of Java Servlet

2.4.1. Performance

The problem with traditional CGI applications is performance. Each time a CGI application is requested by the client, a new process is spawned. This is expensive when the Perl interpreter is loaded and executed for each client request. This could easily lead to performance problems at popular Web sites that handle requests from mu ltiple users. One solution to this problem is addressed in the Java Servlet Architecture. The first time a Java servlet is requested, it is loaded into the Web server's memory space. Subsequent client requests for the Servlet result in calls to the Servle t instance in memory. This process is more efficient than the traditional CGI implementation. As a result, the performance of server-side applications increases.

Database connection is a great overhead in a process. In the case of CGI, every process (eve ry client) makes a new connection to a database. This increases the resource occupied and the workload of the server. For Servlet, database connection is established in the life cycle method (init) which is only run once. Every new thread (client requests), Need not make a connection for their own.

Even Java is not as fast as native programming languages, but it is not the most important factor on network application. The most important factor is network traffic.

2.4.2. Portability

You can develop a complex ser ver-side application without restricting it to a particular hardware platform. Client-side Java applets introduced the notion of platform independence for the client. Java Servlets take this idea to another level: the server. Today your application server can reside on a Windows NT platform and then you can later move it to the UNIX platform. This migration can take place without the headaches associated with porting code and without the need to recompile your Java Servlets.

PERL scripts can usually be moved from platform to platform, but CGI and server extensions written in high-level languages such as C are not as portable. Meanwhile, the server side scripting is also lack of portability even the performance is fairly good.

2.4.3. Extensibility

One shortfall of server-side programming in scripting languages such as Perl and VBScript is that of reuse. If you have to create another server-side module based on existing code then the only reuse you have with scripting languages is to reuse part of the code.

Since Servlets are written in Java, you gain all the object-oriented features of Java such as reuse. You can create an object framework of common Servlets and reuse them in future applications. For example, you can create a simple Servlet for processing of HTML form data. Later, another developer can use this Servlet as is or extend it to add custom functionality. Supporting the idea of modularity, Servlets can communicate with other Servlets on the Web server. This mechanism, known as Servlet chaining, allows the output of one Servlet to be passed as input to another Servlet. As an example, a database query Servlet can retrieve sales data and pass this data to a charting Servlet. The charting Servlet simply prepares a graphical representation of the data and returns it to the client.

Java is a robust, well-designed and fully object-oriented language. Specialized Java libraries, development tools and database drivers are becoming available all the time, and Servlets can utilize Java code from any source.

2.4.4. Security

Many CGIs written in Perl are vulnerable to attacks where the end user tricks the CGI into executing a command on the server. Servlets aren't at risk of running unintended shell commands.

Servlets are compiled class files while CGI/Perl is delivered in its original source form. Depending on who has access to your Web server, you may prefer not to install source code.

A Comparison table for server side program for web application:

Extension Method		Performance	Portability	Capability	Safety	Security	Development
CGI	Scripting Languages	very poor	excellent	depends	excellent	poor	depends
	C *C++*	poor	poor	good			poor
FastCGI	*C* *C++*	very good	not yet				
	Scripting Languages	ok		depends			good
Server API	*C/C++ Web Server API*	excellent	very poor	excellent	very poor	poor	poor
	Java Servlet API	very good	excellent		excellent	excellent	good
Server-side scripting		good	poor	depends	excellent	excellent	good

2.5 Other Alternatives

Besides CGI and Servlet, there are still other alternatives on implementing web-based application. The most common one is the Microsoft Active Server Pages (ASP) and Java Applet.

2.5.1 ASP

An ASP is an HTML page that contains one or more scripts that are processed on a web server before the page is sent to the user. It is developed by Microsoft. In terms of functionality, an ASP is similar to a CGI application, which involves programs that run on the server, for providing dynamic tailored page to the user. Typically, the script in the web page at the server uses input received as the result of the user's request for the page to access data from a database and then builds or customizes the page on the fly before sending it to the requestor.

An ASP file mainly includes a script written in VBScript or JScript in an HTML file or by using ActiveX Data Objects (ADO) program statements in the HTML file. The

output of the HTML file to the user is just the same as those files without ASP. The whole process is transparent to the user.

ASP provides a fast and efficient method on generating content specific pages, especially for database transactions. However, it has the disadvantages that it only runs on MS-Windows platform and it mainly relies on Microsoft products.

2.5.2 Java Applet

Java Applet is yet another popular method of implementing web application. It is easy to find the existence of Java Applet in the world of internet. In terms of architecture, it is different from those mentioned before. Java Applet is considered to be client-side program while the others are mainly executed on server. In other words, users have to download the applet through the web browser before it can be run. Once the applet is loaded, users can operate the applet in the browser. Usually, the applet will be run using the internal Java Virtual Machine (JVM) provided by the web browser.

Java Applet is a complete program that is designed specifically to be running in a web browser. To achieve the web application, users should make use of it to send information or request to the server. There are 2 modes for the server to facilitate the service to the applet. The first one is that there is a background server process or daemon to listen for any request from the applet and carry out further operation like database retrieval and send back the data to the applet later. Another one is that the applet directly connects to the database or other information providing servers for the request.

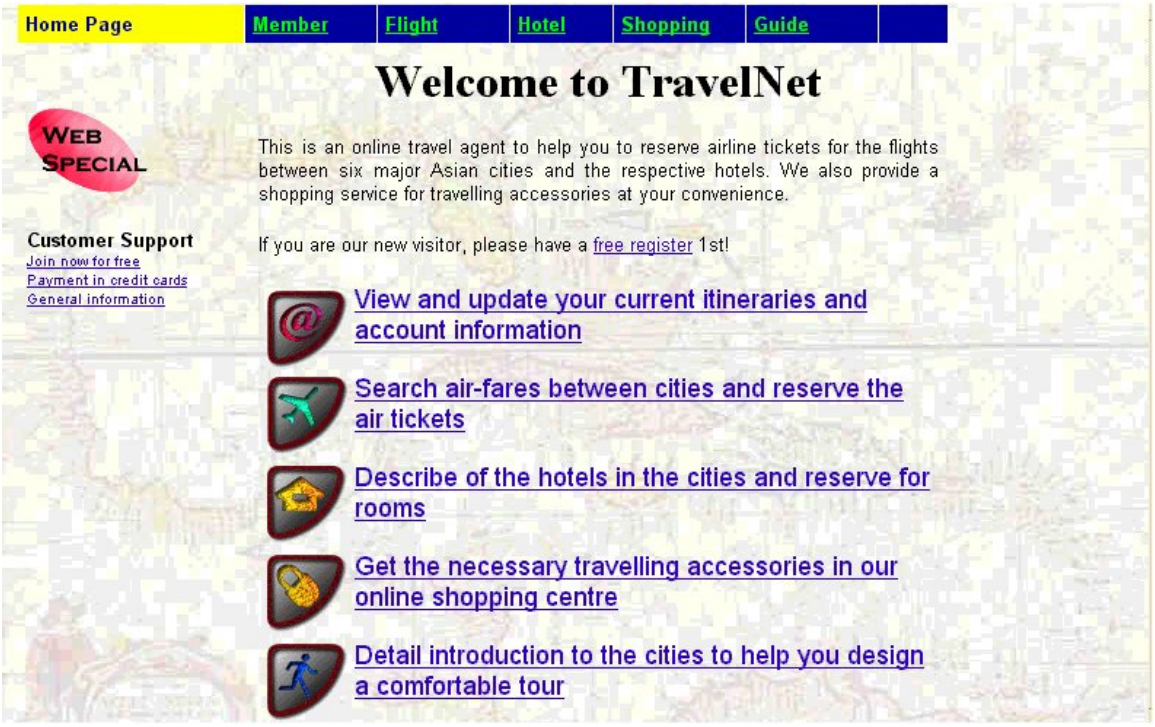
The advantage of using applet is that it tends to lower the traffic of web servers as it divides some computation of server-side to the client-side. However, since Java Applet may include some methods that are not supported by the JVM of the web browser, it may require users to download a Java plug-in, which is inconvenient. Despite the incompatibility problem, the security restriction of Java Applet makes it difficult to provide large-scale applications. Basically, Java Applet is not allowed to set up connection to hosts that are not the one in which the Java Applet downloads

from. Although, it is possible to break the restriction by using signed Applet, which means the Applet is trustworthy to connect, it increases the security threat which is not suitable for application involving payment transaction or others which require transmission of confidential data over the net.

Chapter 3:
Facilities of TravelNet

3.1 Introduction

TravelNet is an online travelling agency. It is necessary to provide enough facilities and function such that it makes no difference from other existing on-line agencies. In this chapter, we will describe the facilities and functions provided in TravelNet, which includes User registration, User profile management, Itinerary management, Flight reservation, Travel accessories shop, Travel Guides and Payment . The picture below is a screen-shot from the main page of TravelNet.



All the service of TravelNet are listed in this page for users to choose and use.

Figure 3-1: Main page of TravelNet

3.2 User Registration

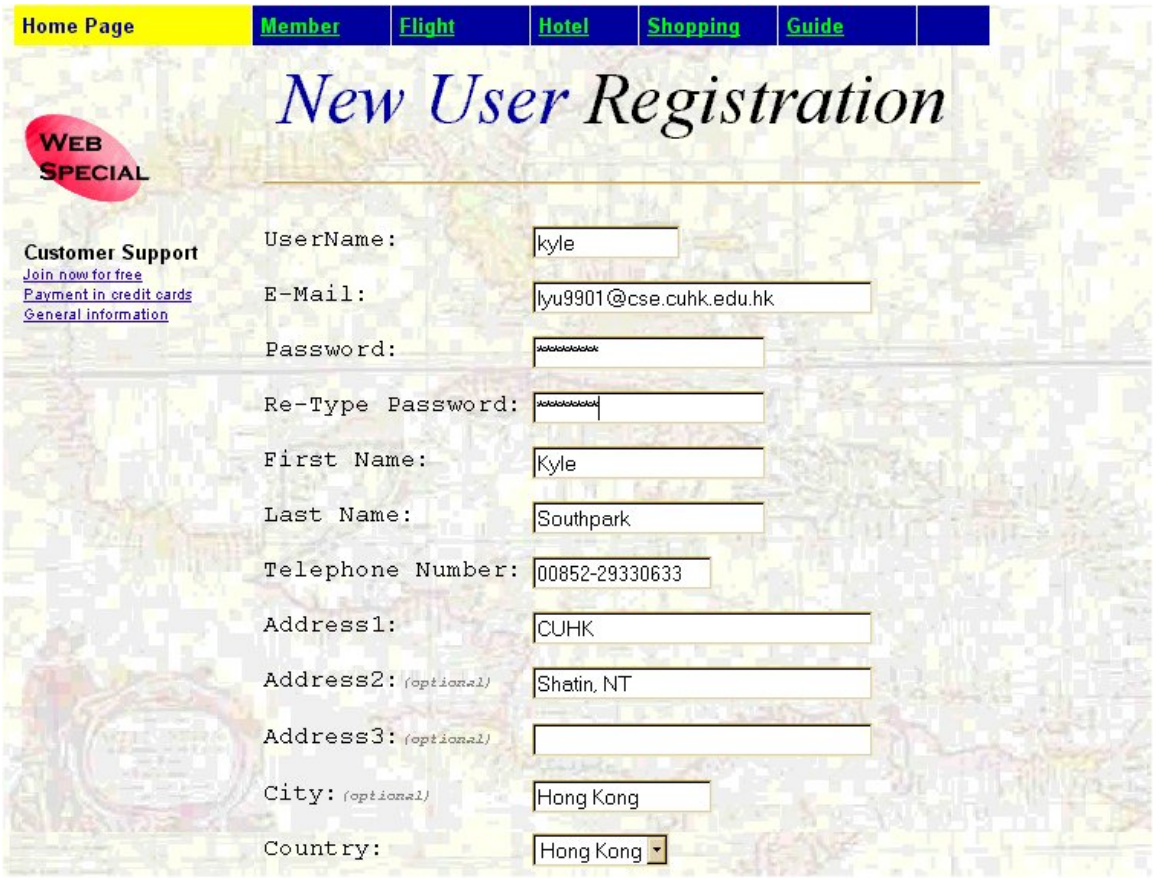


Figure 3-2: User registration page

In order to use the service of TravelNet, users are required to hav e a user account in our system. New users that haven't got a user account can apply for a free user account from us. Once the application is successful, they can use our system as soon as possible. The registration for a user account is simple and strai ghtforward. Users are required to input username, e -mail address, password, their real name, telephone number, and address. Since the username should be unique in our system, checking will be carried out to ensure the uniqueness. If the username which is stored in our database already exists in the system, warning will be given out and user should re -enter the username that match the requirement. Any successful registration will be confirmed to users by e -mail sending confirmation. The picture is a scr een-shot of user registration page.

Once users get their user account, they can login to our system to enjoy all services provided. In order to provide enough security to transmitting user password over the network, security feature has been implemented for such purpose. The detail of the security feature will be discussed in chapter 6.

The following picture shows the login screen of TravelNet.

Figure 3-3: User login page

3.3 User Profile Management

Users can change their registered information in TravelNet anytime after th ey have login. Except that they cannot change the username , which shows the identity of them in the system , all other information can be changed. These include names, e - mail address, address, phone number, location, etc. To provide higher level of security, changing of user account password requires the input of the old password to verify that it is the user to change it.

The following picture is the page for viewing and changing user account information.

Figure 3-4: User profile management page

3.4 Itinerary Management

Each user is ass ociated with an itinerary to their account. It stores the items that the reservations are going to be made or it has been made. Items that will appear in the itinerary include the reserved flights and those flights that are going to be reserved. Users can edit their itinerary by adding or deleting items. Also, they will mainly carry out the reservation process in this page. A detail list on the reservation status will be shown such that users can view and make any modification conveniently.

3.5 Flight Search and Reservation

Flight search is a key element in the TravelNet. With this feature, users are allowed to consult the airlines' databases with users requirement and make reservation on the search result. The system requires users to input some basic elements on the search. The basic elements of queries includes the departure and arrival cities, the departure date, the types of flight (one way/round trip), the class of service(first class/ business class/ economy class), the age category of the ticket(below 12/adult/above 65). Possible additional requirement includes the exact range for departure time, the choice on fare(e.g. is there any penalties for refund of tickets), the airline company, etc. Usually, the optional requirement helps to lower the size of the search result while the basic method is also provided to enhance the flexibility of the search.

There are 3 types of search for different uses. They are the one way search, the round trip search and the multiple destination search. One way search is a simple search on the availability and the fare of the single flight. Round trip search is a search that query on round-trip tour. Usually, a round-trip ticket is cheaper than 2 one-way flight. It is useful and money saving if the users have a definite plan on their trip. Multiple destination search provides the function which users can make search for multiple cities in a period of time of visit so that the result is generated once. This eases the search option for users who choose this type of travel.

There are 2 types of search result available for queries. The first one is the normal search result, which display all the available matched flights. The second one is limit the result output to show only the flights, which are the lowest fare. It will be a useful function particularly to users who Once the result is generated to users, it allows them to put it in the itinerary for further reservation.

The following picture is the page for one -way search. For convenience purpose, the design of the interface is made such that most of the search options are selected through simple selection of pre -defined values. This lowers the risks of for user to have typo that makes a wrong search.

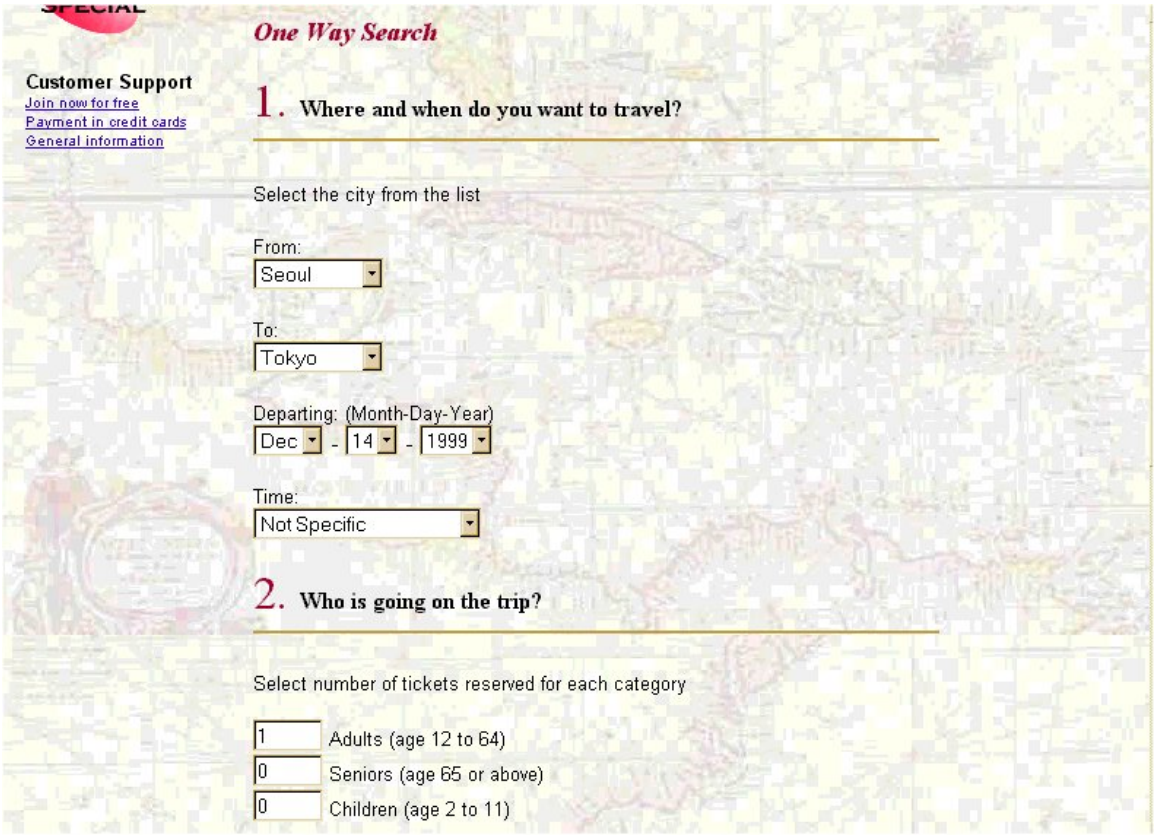


Figure 3-5: One way flight search page

3.6 Travel Accessories Shop

In real life , travelers must have some travelling accessories that bring with them during the trip. Luggages, maps and travel guides are examples of those necessary accessories. To provide a full integrated service to our users, TravelNet also includes an online travelling accessories shop for travelers to buy the accessories with ease.

In our travel accessories shop, users can buy luggages, maps, guides and other travel related stuffs. Users first select the product that they have interest to purchase of appropriate amount. Then they can add the item into the shopping basket. After they have shopped around, they can checkout the items by paying. The currently supported payment method is by credit card. Users need to enter the name of the cardholder, the expire -date of the card and the corresponding card number for payment. At present, the payment method is simple. We will move it to a more sophisticated and secured one for later improvement.

The picture below shows the shopping picture for luggages. Users ca n easily add the item by selecting the appropriate quantity of the chosen products and click the "Add to Basket" label.

Figure 3-6: The snap-shot of part of the travel acessories shop(luggage).

The following picture is the page for showing the content of the shopping basket. Each items are listed with its price. Users can drop any undesirable products in this page.

Figure 3-7: page for viewing shopping basket

3.7 Travel Guides

TravelNet also provides the online travel guide on different cities. Informations like basic description of the cities, map of the cities, introduction of some famous spot and the currency. More useful information may be added for improvement.

3.8 Payment

As TravelNet provides online transaction for products like airline reservation and the travel accessories shop, payment consideration is needed. Our current approach is quite simple, which makes use of the credit-card payment method. Users are required to provide information of the card for the payment transaction. Users should supply the name of card holder, the credit card number and the expire-date of the card. Since

the current approach has less concern on the security of the payment method, a more sophisticated payment method will be implemented as the future work.

The following picture is a snapshot on the payment page for travel accessories.

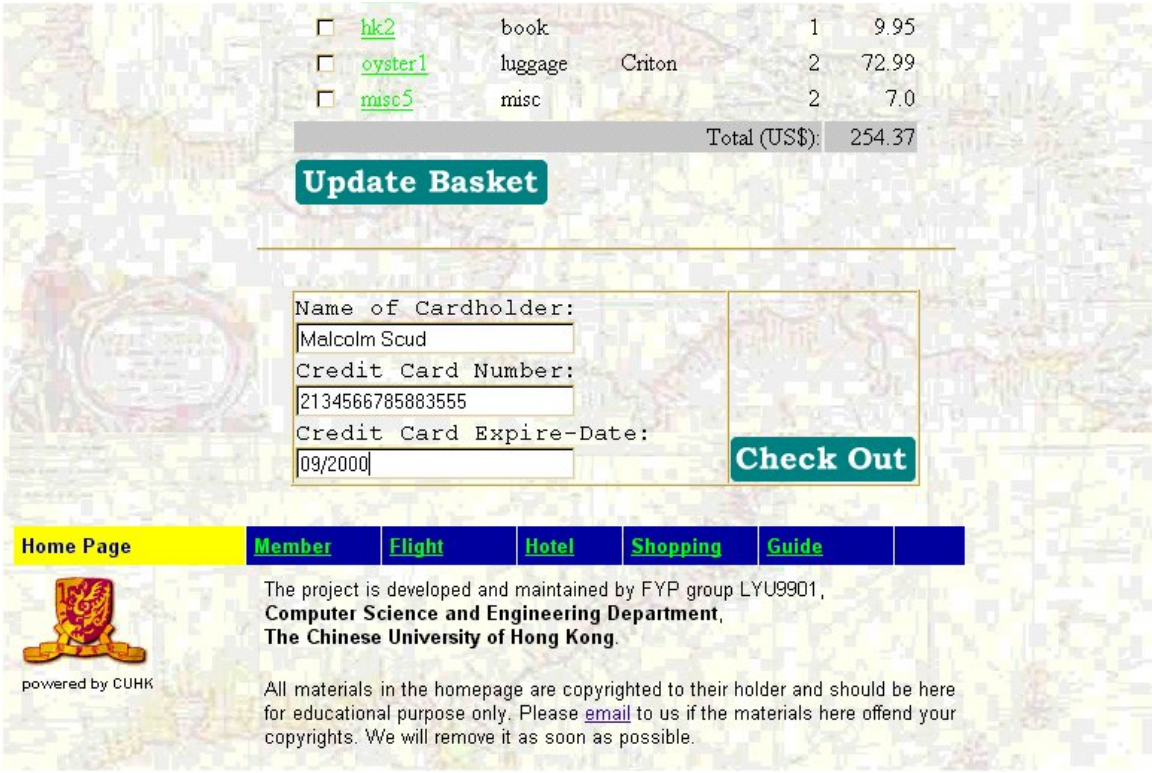


Figure 3-8: A snap-shot showing the payment page

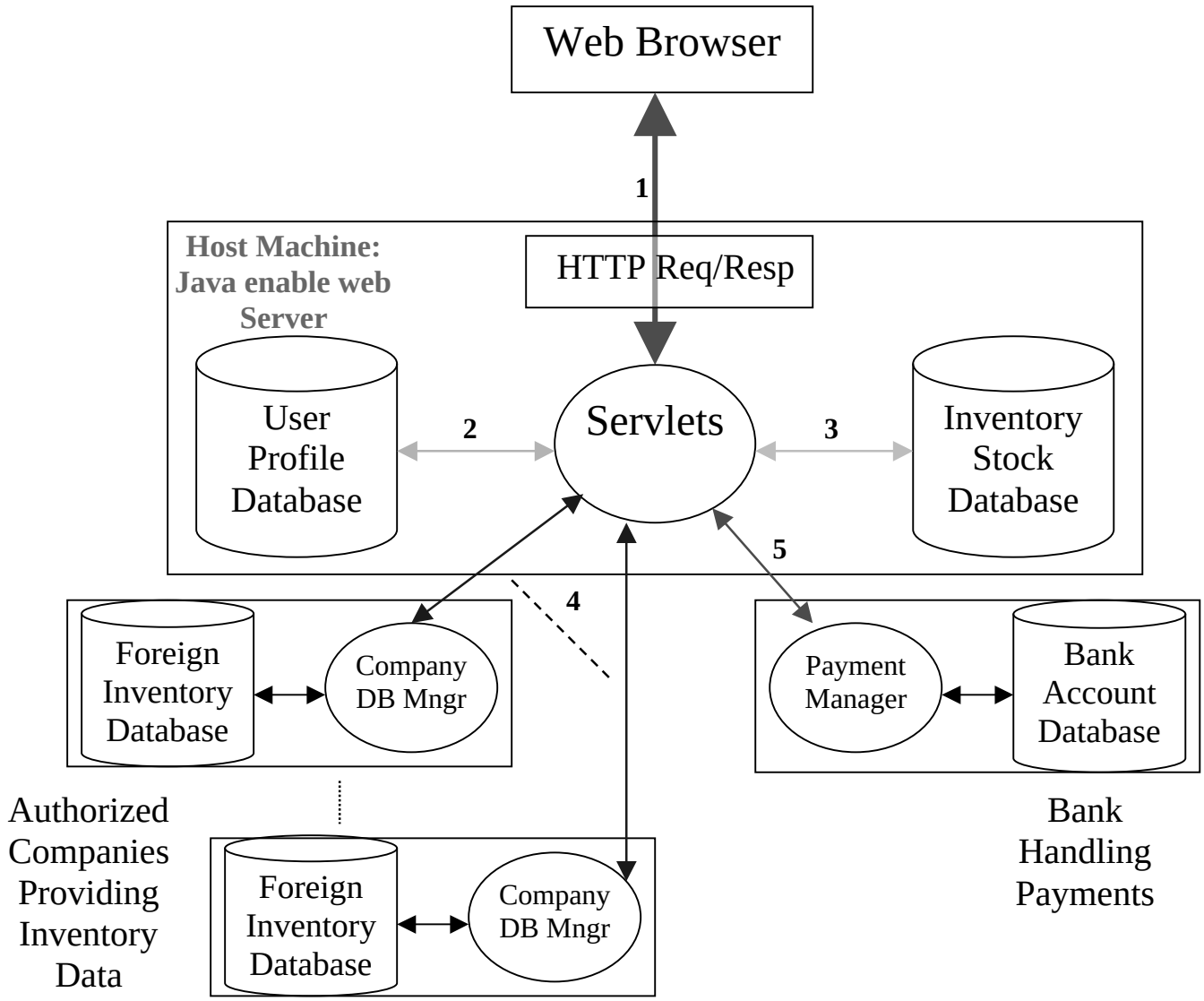
Chapter 4:
System design

4.1 Introduction

In the following section, it will cover the system design issue of TravelNet. These include:

- Architecture: The system infrastructure and data flow between system components
- Communication Interface: Interfaces for different components to communicate with each other.
- Database Structure: Database structure involved in TravelNet System will be stated.
- Web Site map: The hierarchy of the web TravelNet web site.
- Shopping basket: Brief description of the Travel shop Basket design.

4.2 System Architecture



1) Client communication with Web server through HTTP

2) Servlets access local user profile database using JDBC

3) Servlets access local inventory stock database using JDBC

4) Consulting Flight Companies for flight query and booking.

5) Servlets contact the payment system for transaction/validation through Bank Interface

6) Internal communication.

Figure 4-1:
System architecture of TravelNet

Description on data flows:

1) The Client will generate a request from the web browser to TravelNet w eb server. The request can be 2 type:
   a) Normal access: Server will return the requested file (like HTML web pages, zip files) to the client through the same communication channel in HTTP.
   b) Servlet invocation: Server notified that it's a Servlet request. The corresponding Servlet will be executed. According to the type of operation the Servlet carried out, output may or may not be generated but normally the Servlets will create a response in HTML to the user. This HTML mostly generated dynamically according to the result of Servlet executions.

   In this communication channel, user's private information will be passed like visa card number, address, telephone… Etc, so this channel will be on SSL according to the need of privacy of the transferring information.
2) The connection to the profile database will carry may be due to the requests of *registering new user, logging in process, update profile, retrieval of user details* . Since the database is local to the Servlet (or connected in Intranet), these kinds of database access can be done directly by Servlet using JDBC.
3) Inventory Stock of the shop in TravelNet is again stored locally so this connection is made be JDBC. Checking the availability of certain product and updating of stock will involve this channel.
4) This com munication is a foreign connection connected to different airline companies' database manager. Query and result will be in this channel. Between the TravelNet Servlet and Airline's database manager, there is an agreed interface for them to communication in stead of using JDBC. Process of consulting flight prices and making a booking of specific tickets will be requested through this channel. The result and status will be returned through the same channel.
5) Payment is again a foreign request. The bank will pro vide a suitable interface for our server to finish a payment transaction. In reality, this channel must be secure but payment system is not a main concern in this stage, so we didn't do encryption here. In the future, our system will connect to a secure pa yment system through a given socket provided by the bank and encryption will be done before transmission. Current system implementation of the bank is a bit simplified but the structure still valid in secure situation.

6) Internal connection path. Just an abstract path of the data flow inside.

4.3. Communication Interfaces

- **Airline Database Manager**

Flight information query

```
FLIGHT_ID FLIGHT_QUERY
   (DEPARTURE_DATE, DEPARTURE_TIME SOURCE, DESTINATION,
    TYPE_OF_FLIGHT, CLASS_OF_SEAT, AGE_GROUP,
   USER_REQUIREMENT)
THROWS  (NO_FLIGHT_MATCH)
```

This interface allows our travel agent to query the database of a specified flight company.

Inputs:
DEPARTURE_DATE = the desired departure date of the flight
DEPARTURE_TIME = the desired departure time of the flight (Optional)
SOURCE = the source city for the customer to take off
DESTINATION = the destination city for the customer
TYPE_OF_FLIGHT = one-way and round trip
CLASS_OF_SEAT = Economy, Business, 1st Class
USER_REQUIREMENT = terms of tickets
AGE_GROUP = age group of the customer

Output:
FLIGHT_ID = the flight ID of the specific flight in the airline company

Exception:
NO_FLIGHT_MATCH = This airline doesn't provide the tickets match the specified requirement.

Flight booking request

```
FLIGHT_BOOK (DEPARTURE_DATE, FLIGHT_ID TYPE_OF_FLIGHT,
      CLASS_OF_SEAT, AGE_GROUP,  USER_REQUIREMENT,
      USER_INFORMATION)
THROWS
     (NO_FLIGHT_MATCH, BOOKING_FULL)
```

This interface allows our travel agent to book a specified in a flight company.

Inputs:
DEPARTURE_DATE = the desired departure date of the flight
FLIGHT_ID = the flight ID of a specific flight
TYPE_OF_FLIGHT = one-way and round trip
CLASS_OF_SEAT = Economy, Business, 1st Class
USER_REQUIREMENT = terms of tickets
AGE_GROUP = age group of the customer

USER_INFORMATION = the information of the customer who book the ticket.

Exceptions:
NO_FLIGHT_MATCH = This airline doesn't provide the tickets match the specified requirement.
BOOKING_FULL = the specified booking is already full

Flight price search

FLOAT GET_FARE(FLIGHT_ID)
THROWS (NO_FLIGHT_MATCH)

Input:
FLIGHT_ID = the flight ID of a specific flight for price query

Output:
FARE = the fare for the specific flight of given class of seat and type of flight

Exception:
NO_FLIGHT_MATCH = This airline doesn't provide the tickets match the specified requirement.

- **Payment manager**

  Visa card validation interface

  VALIDATE_VISA
  (VISA_NUMBER, CARD_HOLDER_NAME, EXPIRE_DATE)
  THROWS (INVALID_VISA)

  This interface allows client (TravelNet) to check whether the corresponding visa card information is valid according to the bank database.

  Inputs:
  - VISA_NUMBER = the visa card number to be checked
  - CARD_HOLDER_NAME = the name written on the visa card
  - EXPIRE_DATE = the expire date of the visa card

  Exception:
  - INVALID_VISA = Invalid visa card information. It may be card number integrity error or expire date / holder name not match the specific card.

  Visa card debit credit interface

  DEDUCT_CREDIT_FROM_VISA_CARD
  (VISA_NUMBER, CARD_HOLDER_NAME, EXPIRE_DATE,
  DEBIT_AMOUNT, CREDIT_ACCOUNT)
  THROWS
  (INVALID_VISA, NOT_ENOUGH_CREDIT,
  CREDIT_ACCOUNT_NOT_EXIST)

  Inputs:
  - VISA_NUMBER = the visa card number to be checked.
  - CARD_HOLDER_NAME = the name written on the visa card.
  - EXPIRE_DATE = the expire date of the visa card.
  - DEBIT_AMOUNT = the amount to be debit from the visa card.
  - CREDIT_ACCOUNT = the bank saving account the amount to be credited to.

  Exceptions:
  - INVALID_VISA = Invalid visa card information. It may be card number integrity error or expire date / holder name not match the specific card.
  - NOT_ENOUGH_CREDIT = the credit for this credit card is not enough for this amount of payment.
  - CREDIT_ACCOUNT_NOT_EXIST = the credit saving account did not exist at all.

4.4 Database Structure

4.4.1.TravelNet Local DataBases

- **USER_PROFILE:**
  This database stores all necessary information of TravelNet users. Credit card number is not a compulsory field because it is not secure to store the credit card number in the database.

Name	Type	Nullity	Integrity
USERNAME	VARCHAR2(12)	NOT NULL	PRIMARY KEY
EMAIL	VARCHAR2(30)	NOT NULL	
PASSWORD	VARCHAR2(20)	NOT NULL	
FIRSTNAME	VARCHAR2(20)	NOT NULL	
LASTNAME	VARCHAR2(20)	NOT NULL	
TELENUM	VARCHAR2(15)	NOT NULL	
ADDRESS	VARCHAR2(90)	NOT NULL	
CITY	VARCHAR2(15)		
COUNTRY	VARCHAR2(5)		
CREDITNO	VARCHAR2(16)		

- **STOCK:**
  Inventory stock will be stored in this database. It reveals the actual stock of TravelShop.

Name	Type	Nullity	Integrity
PRODUCT_ID	VARCHAR2(10)	NOT NULL	PRIMARY KEY
PRICE	FLOAT(126)	NOT NULL	>0
STOCK	NUMBER(38)	NOT NULL	>0

- **TRANSCATION_RECORD:**
  Payment transactions will be recorded in here. For later reference or complain from users.

Name	Type	Nullity	Integrity
TRANS_NO	NUMBER(38)	NOT NULL	PRIMARY KEY
CARD_NO	VARCHAR2(16)	NOT NULL	
AMOUNT	FLOAT(126)	NOT NULL	>0
TRANS_TIME	DATE	NOT NULL	

4.4.2.Simple Bank Databases

- **BANK_VISA:**
  A database for all the credit cards information that will be used in our community. This database can't be accessed directly by TravelNet. All the accesses of this database are through the Payment manager.

Name	Type	Nullity	Integrity
NAME	VARCHAR2(30)	NOT NULL	
VISANUM	VARCHAR2(16)	NOT NULL	PRIMARY KEY
CREDIT	FLOAT(126)	NOT NULL	
EXPIRE	DATE	NOT NULL	

- **BANK_SAVING**

This database stored saving accounts of the bank.

Name	Type	Nullity	Integrity
ACC_NUM	VARCHAR2(20)	NOT NULL	PRIMARY KEY
NAME	VARCHAR2(40)	NOT NULL	
AMOUNT	FLOAT(126)	NOT NULL	>0

4.4.3.Airline Companies Databases

- **FLIGHT_INFO**

A database stores all the flights operated by the airline company.

Name	Type	Nullity	Integrity
FLIGHT_NUM	VARCHAR2(6)	NOT NULL	PRIMARY KEY
SRC_PLACE	VARCHAR2(3)	NOT NULL	
DEST_PLACE	VARCHAR2(3)	NOT NULL	
DDATE	DATE	NOT NULL	
DTIME	TIME	NOT NULL	
ATIME	TIME	NOT NULL	
AIRCRAFT	VARCHAR2(4)	NOT NULL	

- **FLIGHT_SCHEDULE**

A database for weekly schedule of specific flights

Name	Type	Nullity	Integrity
FLIGHT_NUM	VARCHAR2(6)	NOT NULL	PRIMARY KEY
SUN	VARCHAR2(1)	NOT NULL	
MON	VARCHAR2(1)	NOT NULL	
TUE	VARCHAR2(1)	NOT NULL	
WED	VARCHAR2(1)	NOT NULL	
THU	VARCHAR2(1)	NOT NULL	
FRI	VARCHAR2(1)	NOT NULL	
SAT	VARCHAR2(1)	NOT NULL	

- **FARE_INFO**

A database stores the fare list of each class of tickets in terms of one-way flights and round-trip flights.

Name	Type	Nullity	Integrity
FLIGHT_NUM	VARCHAR2(6)	NOT NULL	PRIMARY KEY
OW_FCLASS	FLOAT(10)	NOT NULL	>0
OW_BCLASS	FLOAT(10)	NOT NULL	>0
OW_ECLASS	FLOAT(10)	NOT NULL	>0
RT_FCLASS	FLOAT(10)	NOT NULL	>0
RT_BCLASS	FLOAT(10)	NOT NULL	>0
RT_ECLASS	FLOAT(10)	NOT NULL	>0

- **PLANE_SIZE**
  A database stores the capacity of each plane of 3 classes of service(first class/business class/economy class).

Name	Type	Nullity	Integrity
AIRCRAFT	VARCHAR2(4)	NOT NULL	PRIMARY KEY
FCLASS	NUMBER(3)	NOT NULL	
BCLASS	NUMBER(3)	NOT NULL	
ECLASS	NUMBER(3)	NOT NULL	

- **TICKET**
  A database stores the capacity of each plane of 3 classes of service(first class/business class/economy class).

Name	Type	Nullity	Integrity
FLIGHT_ID	VARCHAR2(6)	NOT NULL	PRIMARY KEY
DDATE	DATE	NOT NULL	PRIMARY KEY
FCLASS	NUMBER(3)	NOT NULL	
BCLASS	NUMBER(3)	NOT NULL	
ECLASS	NUMBER(3)	NOT NULL	

- **USER_ITINERARY**
  A database which stores the sold ticket for internal usage.

Name	Type	Nullity	Integrity
TICKET_NUM	VARCHAR2(12)	NOT NULL	PRIMARY KEY
FLIGHT_ID	VARCHAR2(6)	NOT NULL	
NAME	VARCHAR2(40)	NOT NULL	

***Note:** The above is the database schema for each airline company. Since it is not available to have multiple database for us to use, we simply simulate the situation by append a code as a prefix to the database table to represent the ownership of the table. For example, the code for Cathay Pacific Airways is CX, so all the tables that belongs to the company are started with CX_, like CX_TICKET and so on.

4.5. Web Site Map

The web site is well structured using the functions provided in TravelNet. Each branch corresponds to a module of TravelNet system

The figure followed shows the hierarchy of TravelNet

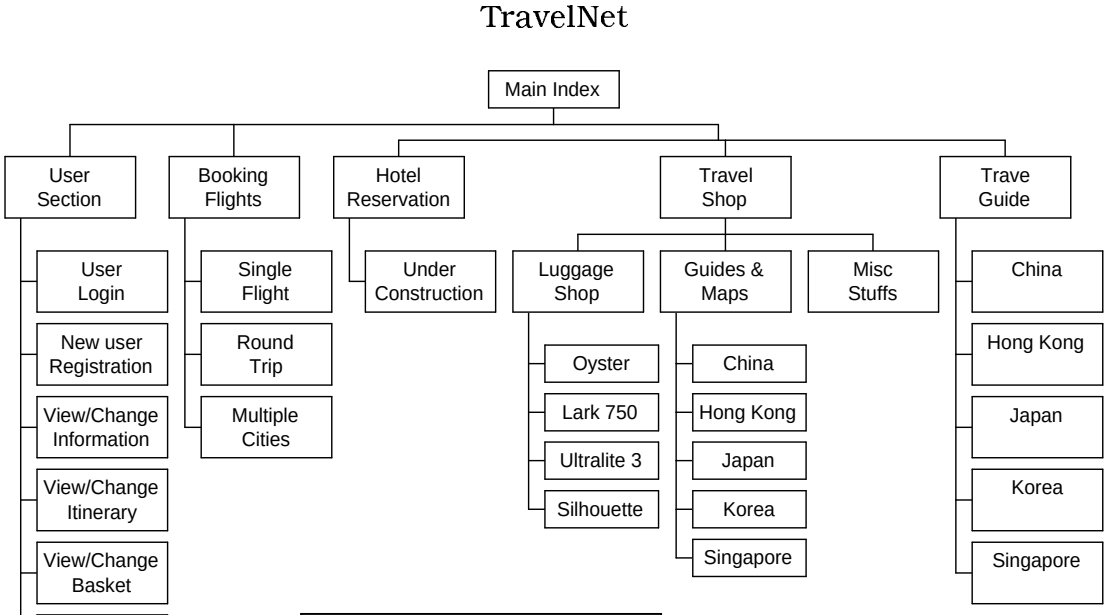


Figure 4-2:
Site hierarchy of Travel Net

4.6 Shopping Basket

4.6.1 Introduction

Shopping basket store the goods a user picked up during his/her current login session. User can add any shop items into it, view it and update it any time. When a user want to checkout and pay, he/sh e just have to input the correct credit card information. The following figure is an instance of a shopping cart of Malcolm Scud.

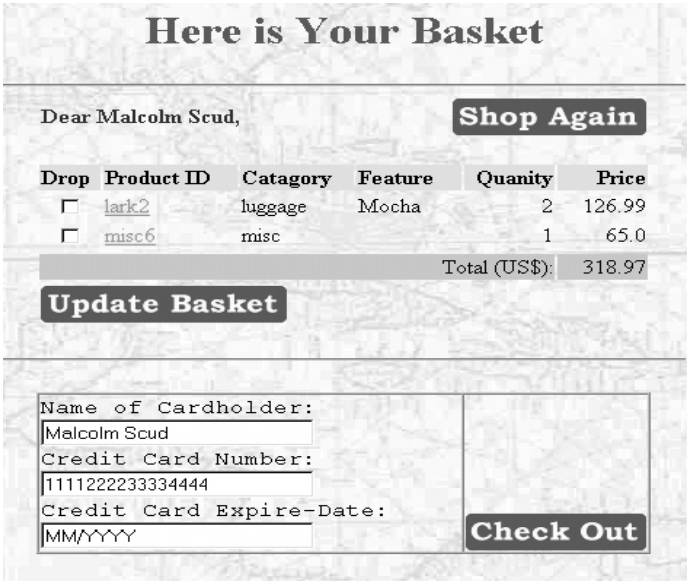


Figure 5-3:
A screen shot for basket page

4.6.2 Basket Design

The basket contains a list of shopping items. It provides operations to add, remove and get related information of the basket. Operation will be listed below:

Put a shop item into basket:
VOID PUT_SHOP_ITEM (PRODUCT_ID, PRICE, QUANITY, PRODUCT_TYPE, OTHER_DETAIL)

Remove an item from basket:
ITEM REMOVE (PRODUCT_ID)

Get the price of an item in the basket:
FLOAT GET_PRICE(PRODUCT_ID)

Get the quantity of an item in the basket:
INT GET_QUAN(PRODUCT_ID)

Get the other detail of an item in the basket:
STRING GET_DETAIL(PRODUCT_ID)

Get the total amount of all items in the basket:
FLOAT GET_TOTAL()

Chapter 5: Security

5.1 Introduction

Security is a major concern of all online transaction. It is because for most transactions, confidential data are involved in the transmission over the public network. Confidential data includes user account password, credit card information are always subject to be exposed and stolen in internet. Therefore, a good policy of providing secure channel for transmitting those confidential is highly demanded. A matured security implementation is often a component for the success on e -business by increase customers' confidence on accepting and using the service.

TravelNet is also an online business provider. Therefore undoubtedly, we have to implement a secure channel for the payment process during the airline ti cket reservation and the travel accessories shop. In current days, there are a number of ways to provide security features for transaction. After doing an analysis of the methods, we have chosen the SSL(Secure Socket Layer) approach for the security between client and server.

In this chapter, we will introduce the background of SSL and how it works. Moreover we will discuss the choice of SSL for our system and some of the implementation details of SSL into our design.

5.2 Background of SSL

SSL, an open, non-proprietary protocol designed by Netscape, is perhaps the most common way of providing encrypted transmission of data between web browsers and web servers. SSL is in use (65,407 sites) chiefly in the US (70%) and gives users the assurance that the information transmitted from their machine to the merchant is secure. Netscape has offered SSL as a proposed standard protocol to the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) as a standard security approach for Web browsers and servers.

It is the Transmission Control Protocol/Internet Protocol (TCP/IP) that governs the transport and routing of data over the Internet. Other protocols, such as the HyperText Transport Protocol (HTTP), Lightweight Directory Access Protocol (LDAP), or Internet Messaging Access Protocol (IMAP), run "on top of" TCP/IP in the sense that they all use TCP/IP to support typical application tasks such as displaying web pages or running email servers.

The basic idea of Netscape on security is that the programming for keeping your messages confidential ought to be contained in a program layer between an application (such as web browser or HTTP) and the Internet's TCP/IP layers. The SSL protocol runs above TCP/IP and below higher-level protocols such as HTTP or IMAP. It uses TCP/IP on behalf of the higher-level protocols, and in the process allows an SSL-enabled server to authenticate itself to an SSL-enabled client, allows the client to authenticate itself to the server, and allows both machines to establish an encrypted connection.

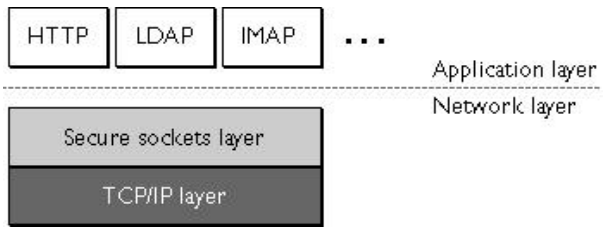


Figure 5-1:
An indication of position in the layers of TCP/IP protocol

Netscape's SSL uses the public -and-private key encryption system from RSA, which also includes the use of a digital certificate.

These capabilities address fundamental concerns about communication over the Internet and other TCP/IP networks:

- ✓ *SSL server authentication* allows a user to confirm a server's identity. SSL -enabled client software can use standard techniques of public -key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs. This confirmation might be important if the user, for example, is sending a credit card number over the network and wants to check the receiving server's identity.

- ✓ *SSL client authentication* allows a server to confirm a user's identity. Using the same techniques as those used for server authentication, SSL -enabled server software can check that a client's certificate and public ID are valid and have been issued by a cert ificate authority (CA) listed in the server's list of trusted CAs. This confirmation might be important if the server, for example, is a bank sending confidential financial information to a customer and wants to check the recipient's identity. However, th is function is not used as it is not a common practice for every user to apply for a client certificate before using our service. We just use our user account system for this purpose.

- ✓ *Encrypted SSL connection* requires all information sent between a clien t and a server to be encrypted by the sending software and decrypted by the receiving software, thus providing a high degree of confidentiality. Confidentiality is important for both parties to any private transaction. In addition, all data sent over an en crypted SSL connection is protected with a mechanism for detecting tampering--that is, for automatically determining whether the data has been altered in transit.

The SSL protocol includes two sub -protocols: the SSL record protocol and the SSL handshake protocol. The SSL record protocol defines the format used to transmit data.

The SSL handshake protocol involves using the SSL record protocol to exchange a series of messages between an SSL -enabled server and an SSL -enabled client when they first establish an SSL connection. This exchange of messages is designed to facilitate the following actions:

1. Authenticate the server to the client.
2. Allow the client and server to select the cryptographic algorithms, or ciphers, that they both support.
3. Authenticate the client to the server(optional).
4. Use public-key encryption techniques to generate shared secrets.
5. Establish an encrypted SSL connection.

SSL comes in two strengths, 40 -bit and 128 -bit, which refer to the length of the "session key" generated by every en crypted transaction. The longer the key, the more difficult it is to break the encryption code.

5.3 Procedures on SSL Connection

This session will show the detailed procedures on establishing an SSL connection through server authentication approach.

The SSL protocol uses a combination of public -key and symmetric key encryption. Symmetric key encryption is much faster than public -key encryption, but public -key encryption provides better authentication techniques. An SSL session always begins with an excha nge of messages called the SSL handshake. The handshake allows the server to authenticate itself to the client using public -key techniques, then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows.

1. The client sends the server the client's SSL version number, cipher settings, randomly generated data, and other information the server needs to communicate with the client using SSL.

2. The server sends the client the server's SSL version number, cipher settings, randomly generated data, and other information the client needs to communicate with the server over SSL. The server also sends its own certificate.
3. The client uses the certificate sent by the server, which contains validity period, the issuer (CA), and the domain name of the server. If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client goes on to Step 4.

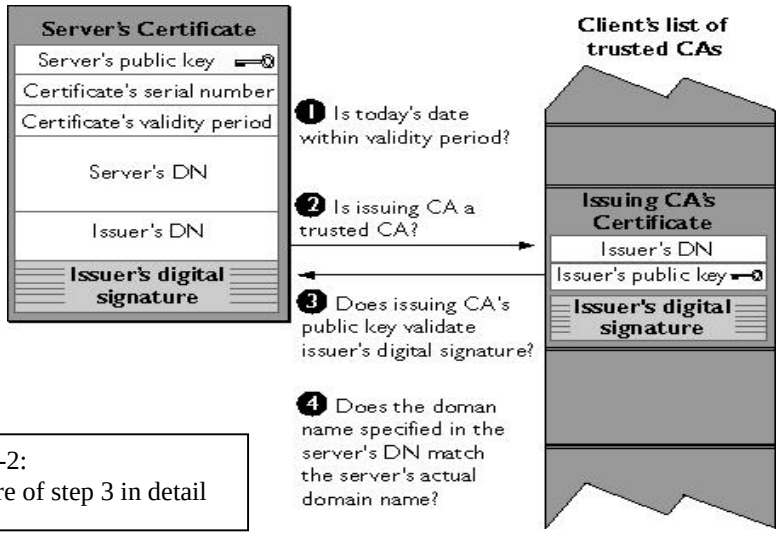


Figure 5-2:
Procedure of step 3 in detail

4. Using all data generated in the handshake so far, the client (with the cooperation of the server, depending on the cipher being used) creates the premaster secret for the session, encrypts it with the server's public key (obtained from the server's certificate, sent in Step 2), and sends the encrypted premaster secret to the server.
5. The server uses its private key to decrypt the premaster secret, then performs a series of steps to generate the master secret. Client is also responsible for generating the master secret using the same premaster secret.
6. Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL session and to verify its integrity.
7. The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends an encrypted message indicating that the client portion of the handshake is finished.

8. The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends an encrypted message indicating that the server portion of the handshake is finished.
9. The SSL handshake is now complete, and the SSL session has begun. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity.

The key point of server authentication is that the client encrypts the premaster secret with the server's public key. Only the corresponding private key can correctly decrypt the secret, so the client has some assurance that the identity asso ciated with the public key is in fact the server with which the client is connected. Otherwise, the server cannot decrypt the premaster secret and cannot generate the symmetric keys required for the session, and the session will be terminated.

5.4 Implementation of SSL in TravelNet

There are several reasons for us to choose SSL as our security feature.

1. SSL is a matured product and it is free to use.
2. There is a wide range of products we can used to implement SSL into our design.
3. Our approach on directly using web browser as client agent, although favours flexibility and allow it to be common to public, it limits our choice on security features.

Since a number of web servers and the major web browsers(e.g. Netscape and Internet Explorer) have already supported SSL, the major thing for us to implementing SSL is to get a server certificate and a fixed IP machine for the web server such that we can use it to apply for a digital certificate for the web server. Once the machine is settled, we have applied a tri al certificate from Entrust Technologies, which is an international CA. Trial version of the certificate works just the same as the commercial one except its valid period is shorter. As CUHK has its own CA now, we may get a certificate issued by CUHK CA which it validity period is longer and free of charge.

After installation the certificate into the web server, the SSL connection is ready to use. In our system, we just need to refer our code (html) for form submission by https, which is a syntax of calling SSL through URL. An indication of the SSL enabled connection is by a small lock icon in the browser.

Figure 5-3: Apperance of lock icon without SSL enabled connection

Figure 5-4: Apperance of lock icon with SSL enabled connection

Chapter 6:
Summary and Future Work

6.1 Summary

In these few months, we have successfully completed a certain amount of work .

After a detailed analysis on the web application model and some implementation concerns, we have built an online travel agency, TravelNet, which is a real -life application and practical to provide service. In order to provide the facilities and functions of TravelNet, we have chosen Java Servlet, a comparatively new technology for web programming, in our design. Since Servlet can outperform the traditional CGI-style web application, our experience on building the system becomes invaluable for meeting the trend of using Servlet.

Also, with the help of Java, it is easy for us to make a convenient further development of the system into a CORBA distributed system, which is more fault -tolerant and interoperable. To be a real -life e-commerce application, we have als o handled the security issue between client and server by implementing SSL in our system.. The current schedule only allows us to implement a simple payment method that is less secured. However, it will be one of the major targets to convert the existing one into a more secure payment method and possibly include other payment scheme like smart card as well.

Development is a continuous process. We will keep on our development to make it best.

6.2 Future Work

In the session, we will discuss the future w ork to be done on our project for enhancement. It includes integration of CORBA, secure payment method, micro payment using Mondex and hotel reservation.

6.2.1 Integration of CORBA

Introduction to CORBA

Simply stated, CORBA allows applications to commun icate with one another no matter where they are located or who has designed them. CORBA was introduced in 1991 by Object Management Group (OMG) and defined the Interface Definition Language (IDL) and the Application Programming Interfaces (API) that enable client/server object interaction within a specific implementation of an Object Request Broker (ORB).

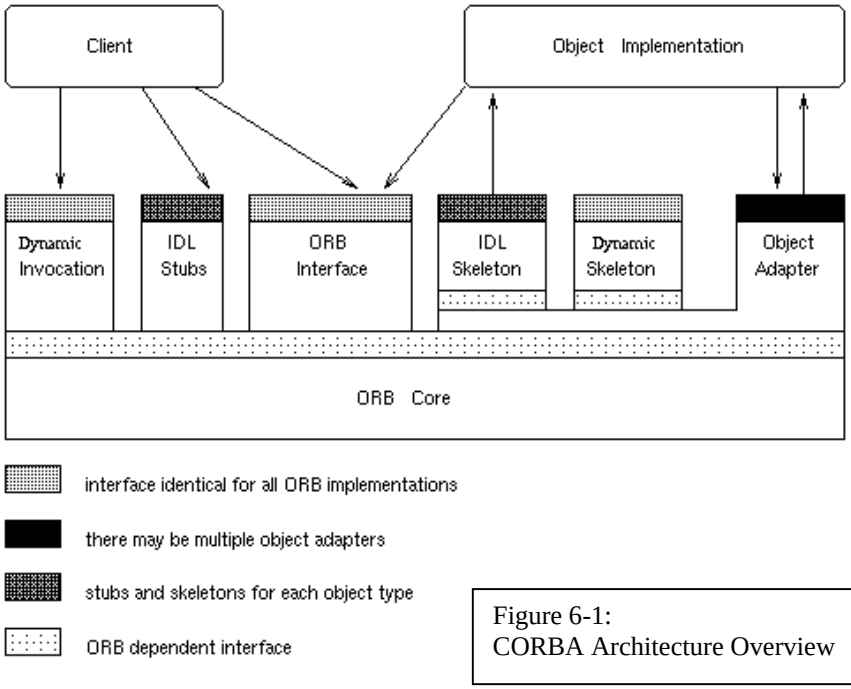


Figure 6-1:
CORBA Architecture Overview

The (ORB) is the middle ware that establishes the client -server relationships between objects. Using an ORB, a client can transparently invoke a metho d on a server object, which can be on the same machine or across a network. The ORB intercepts the call and is responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results. The client d oes not have to be aware of where the object is located, its programming language, its operating system, or any other system aspects that are not part of an object's interface. In so doing, the ORB provides interoperability between applications on differen t machines in

heterogeneous distributed environments and seamlessly interconnects multiple object systems.

Integration

TravelNet is suitable to be implemented in a distributed manner. Foreign components like flight company manager and hotel reservation manager can be implemented in different platform or different programming language. In order for them to communicate with TravelNet components, CORBA is a suitable choice for the middle ware in between them.

In the near future, the current Java version will be extended to CORBA version. Distribution of the system can increase the autonomous of each component and replication and load balancing can be achieved. Which the help of naming services provided by CORBA, location of each component will be transparent to the users. Distributed TravelNet can be a high portability, compatibility, efficiency and fault tolerance system.

6.2.2 Secure Payment Method

Payment method is an essential issue of any e-commerce application. Although we are not going to study it and develop it deeply in this project, effort should be made for the integration of some payment system. We are going to co-operate with a postgraduate student's (Steve Chong) Secure Payment System to demonstration the ability of our TravelNet to integrate with an existing payment system.

Unfortunately, due to the time limitation of our group and the postgraduate student to do this integration, the process of combination of these two systems will be done later in this winter.

The brief communication interface and channel is being drafted. Sockets connection will be our communication channel; the payment system will provide a socket listener for any payment request to be raised. Asymmetric encryption will be used in between the authorized merchants and payment system. The encrypted message can be describe as follow:

A = {CUST_NAME, CUST_NO, EXP_DATE, CARD_TYPE,
AMOUNT, TRANS_ID, MERC_NAME} *(Encrypted by merchant's private key)*
B = {A, MESSAGE_DIG, MERC_ID}
(Encrypted by acquirer's public key then merchant will send message B to acquirer.)

Parameters:
CUST_NAME = customer name or exactly the owner of the card
CUST_NO = card number
EXP_DATE = card expiry date
CARD_TYPE = VISA, MASTER or AE
AMOUNT = total amount of the transaction
TRANS_ID = Transaction id of this payment (unique)
MERC_NAME = merchant's name
MESSAGE_DIG = message digest of message A
MERC_ID = merchant ID (unique)

Issues on the distribution of the key are still in negotiation. Agreement will be made shortly for the systems incorporation. A joint paper with Steve will be written for this integration in the mean time.

6.2.3 Micropayment in Mondex

Micropayment is the payment that only involves a small amount of transfer of money from customers to merchants. It provides an alternative revenue source for content and service providers. It is a more efficient method that credit card for transaction, which the values of the service and products involved are low.

Mondex is one of the advanced electronic cash micropayment systems over the Internet. Its unique security architecture enables a range of functionality not offered by any other electronic cash scheme.

Mondex is preferable to be used for simple, everyday cash transactions. In TravelNet, the travelling accessories shop offers a good chance to adopt Mondex as one of the payment method for buying and selling goods. Also, new kinds of service may be also added into the existing design.

Due to the potential cooperation of a commercial firm on Mondex products and CUHK, we have the chance to try out the device in near future. If the hardware

device is available to us in the next few months, it will be a good experience to adding Mondex as one of the payment method in our system. From the view of the user, it is a flexible design of payment that all ows other method instead of the traditional credit card approach.

6.2.4 Hotel Reservation

Besides the existing airline ticket reservation, hotel reservation is also considered as an important element of any travel service provider. With its existence, it is possible to offer complete tour package to users and full travel service can be provided. Although the complexity of hotel database is no smaller than the airline database, we should be able to handle and implement it for a longer period of time in collecting data and make a real-life compatible design on our TravelNet.

Chapter 7: References

[1] B.Eckel. *Thinking in Java*, Prentice Hall Inc. 1998.

[2] "JDK™ 1.1.8 Documentation".
http://java.sun.com/products/jdk/1.1/docs/index.html

[3] "The Java Tutorial".
http://java.sun.com/docs/books/tutorial/

[4] Victor Wolters. *Introducing Internet Information Server*, Que. Oct 14, 1996

[5] "Security in Internet Transaction".
http://www.holt.ie/text/security.html

[6] "Web Application Development".
http://www.winwinsoft.com/articles/wad.html

[7] "Expedia.com".
http://expedia.msn.com

[8] "Travelocity"
http://www.travelocity.com

[9] "Introduction to SSL"
http://developer.netscape.com/docs/manuals/security/sslin/index.htm

[10] "How SSL works"
http://developer.netscape.com/tech/security/ssl/howitworks.html

[11] C. Darby , "Developing 3 -Tier Database Apps w/ Java Servlets" , *Java Developers Journal*, Feb 1998

[12] IBM Corporation. "The Web Application Programming Model" . *IBM Application Framework for e-business*. IBM Corporation.

[13] Z.Yang, K.Duddy. "CORBA: A Platform for Distr ibuted Object Computing ". *Operating Systems Review, 30(2):4-31*. ACM SIGOPS, Apr. 1996.

Appendix

A. Software

- Java API 1.1.8.

  Java is an object -oriented language, which is poplar all around the world today. Because of its portability, it grows along with th e Internet related technologies. Its complete and robust API brings programmer and software developer a convenient developing environment. Since it is slower than native programming language, Java is not suitable for low level programming or real time proc essing. On the other hand, it is perfect for net working application programming. One of the most critical factors determining the performance of network application is the connection speed. So it compromise slow execution speed of Java.

- Java Servlet API 2.1

  Servlets are the Java platform technology of choice for extending and enhancing Web servers. Servlets provide a component -based, platform -independent method for building web -based applications, without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), Servlets are server- and platform-independent.

  Written in Java, Servlets have access to the entire family of Java APIs, including the JDBC API to access e nterprise databases. Servlets also access library of HTTP -specific calls, and all the benefits of the mature Java language, including portability, performance, reusability, and crash protection.

- Windows NT Server 4.0 with IIS 4.0
  Windows NT Server is a qu ite common commercial product Microsoft Windows NT Server 4.0 is a multipurpose operating system specialized on Server operations . A multipurpose operating system does more for less

because it integrates a variety of network services that you need to run your business. The services it provides are designed to address customer requirements in every category.

The Internet Information Server is a popular web server providing internet services like web, mail and news. Its functionality can be extended by ins tall suitable ISAPI.

- Servlet Exec 2.2

ServletExec is a Servlet engine. It is a high -performance, reliable, inexpensive web application server and Servlet engine that implements the Java Servlet API and JavaServer Pages (JSP) standards, components of the Java 2 Platform, Enterprise Edition (J2EE) suite of standards defined by Sun Microsystems. ServletExec runs on all major web servers and operating systems.

B. Hardware

- Host machine
  - ❖ Pentium II 300MHz, 96 MB memory
    A mid-end machine is needed for a web server to handle requests concurrently especially our system request handler is Java Servlet. A Pentium 2 300MHz is just meet our demand. It is a server with a static Internet address. The internet name is ntsvr4.cse.cuhk.edu.hk.

C. Client-side Requirement

- Netscape 3.0+ or Internet Explorer 4.0+

Travel Net client only needs a simple web browser. It is recommended that client browser is SSL enable because the client will submit critical information through the Internet. This unprotected transmission is very insecure. If

information is being hacked, hacker may use this information for illegal shopping.

D. Program Listing

This is a table showing the statistics of the modules in our system.

Module	Operation	Number of Lines	Number of Characters
User Management	Register	237	8981
	Login	163	5316
	Logout	19	444
	ViewUserInfo	168	6817
	UpdateUnfo	150	5553
	CheckLogin	61	2285
	UserSession	34	947
Shopping	ViewBasket	35	1024
	UpdateBasket	44	1346
	AddToBasket	107	3597
	ShopBasket	1364	42
Flight Booking	QueryFlight	204	7591
	BookFlight	173	4562
	FlightItinerary	153	5120
Payment	CheckOut	175	6728
Bank	BankOperations	116	4171
Supplemetary	Database	44	1373
	Mail	38	1471
	Html	19	523
	BasketTemplate	96	4549
	Total:	3400	72440