

Chapter 1

Overview

1.1 Introduction

This section is about the overall view of the project that represents the brief history of the genesis of the civil aviation in Nepal and present scenario of the system that are implemented in the head offices and the operating branches of the different airlines companies in operation within the nation. The problems associated with the implementation and the problems faced during the preparation of this project that helps to eliminate the problem to certain extent. The integration of the airlines companies and agents dispersed in different areas will be operated through this system into a single reliable platform is the major objective of the project.

1.2 Preliminary investigation

At the genesis it was important to study the existing system and its performance, which made different field, visits. The information taken from the visit in several airline companies and ticket-booking agents have been the milestone of this project. Information regarding the process of booking ticket from booking agents was studied. Booking agents takes the booking rights of the certain number of ticket of specific airlines for specified time. The tickets have to be booked within that period. Booking agents do all these bookings through telephone in a single call. They have to pen down PNR numbers of each customer while booking tickets. The information management and data flow process was also studied during the visits. Many other international airlines were visited frequently how they are operating. They told that they are recently using Gallery software for the ticketing where it has too many cumbersome and problem while installing and buying. From these, it has been felt that this project would put the one more milestone in the aviation industry of Nepal, which will remove all the problems in this industry. From the investigation the following existing system block was concluded:

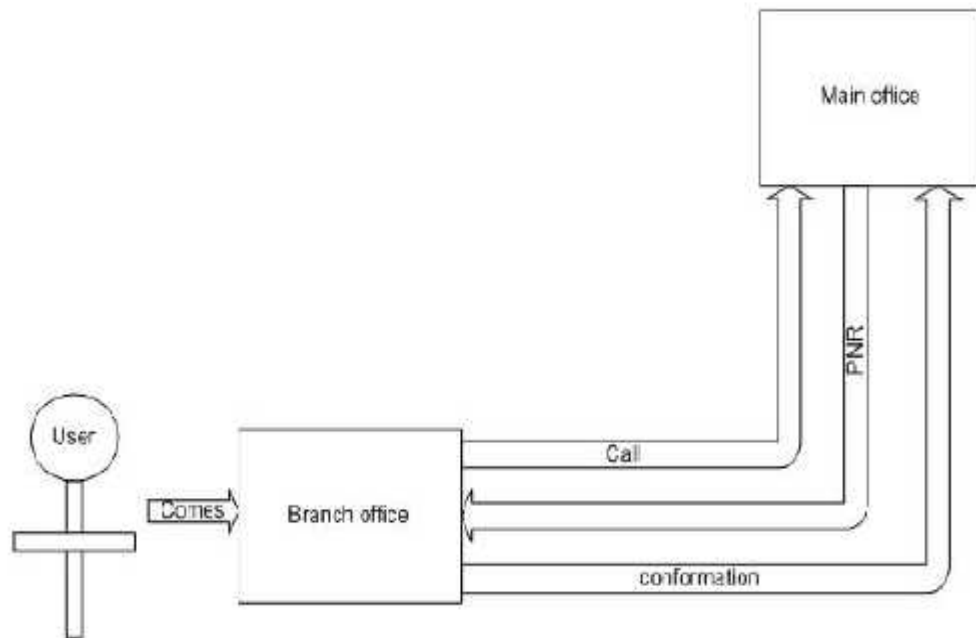


Fig 1: Existing system

1.3 Problem analysis

Problems are always the path in the development of new concept in the field of computer science that breeds a new product. Problem in ticket reservation system in context of Nepal can easily be tracked through the system. In the current situation the ticketing agents book tickets in haphazard way. The owner gives the authority to book some number of tickets to their booking agents. The booking agents have to book or sell the ticket in specified time. They all do these activities through telephone calls. They all rely on telephone system, the customer has to ask to the booking agents for the tickets through phone. Further most of the jobs are manual that results in inappropriate and ineffective decision-making, which push far away from competition and information technology. Most of the activities on most organization are gradually being replaced by automated systems, where very less human efforts play a role and provide a very high throughput. To automate the airlines, ticket reservation system will be major effort during the project lifeline where additional accessories related will also be taken under consideration. The airlines companies provide PNR number to each ticket. When the customer buys the tickets, company gives these PNR

numbers to the booking agents at the time of ticket conformation. The booking agent asks the PNR number through telephone. If the agents provide incorrect PNR number to the passenger will have the problem to find his flight as well as his seats.

The phone charge to be paid by the booking agent as well as Airlines Company is high. Customer gets difficulty in buying tickets through booking agent, as he has to ask for the desired ticket to the airlines company and then have the result. It is also time consuming task. The booking agents are not transparent to the airlines company as well as the customer. The booking agents in the pick time hide the ticket and sell in his desired price to the customer. Competitive markets among the different airlines companies have been lacking. The following drawbacks were observed on the existing system:

- 1) Delay on calling to main office.
- 2) Inconvenient to the customer
- 3) Fine on mistakes of passenger name record
- 4) Don't have the facility of integrated system.
- 5) Heavy charge on telephone call.
- 6) Indulge in paper work
- 7) No transparency

1.4 Project work Structure

Chapter 1 covers introduction about the whole project and all background of aeroweb.

Chapter 2 covers about the specific objective of the project.

Chapter 3 covers the overall system planning and necessary requirements for the project.

Chapter 4 covers the brief description about the design process of the project.

Chapter 5 covers about the technology used to change the project work in the real system.

Chapter 6 covers the implementation and testing phases of the software.

Chapter 7 covers the pictorial overview of the project components.

Chapter 8 covers the conclusion and further extension of the project in future.

Chapter 2

System Development Concept

2.1 Objective Definition

The specific objective of the project can be divided as the following:

a) Airlines integration in a single system

The system is an integration of all the airlines company. All airlines companies operate and manage their organizations through this common platform. For this it requires to manage the relationship between the airlines companies so that private information does not flow outside their system and the information to be issued to public can be facilitated to enhance their efficiency by providing their advertisements and schemes.

b) Storage of branches and agents of the concerned airlines companies

Different branches and the representative agents located on the different parts of the nation can be stored and maintained under the associated airlines company with the sole control of their activities from the main office. The branches and the agents access the same database as the main office but depending on the degree of their information need as categorized by the head office.

c) Data storage in a single database

The storage tank or the data repository is designed with a concept of common requirements and fields analyzed from the visits of various airlines companies so that it fulfills the data requirements of all airlines companies.

d) Addition and removal of airlines companies, agents and branches

The system will provide a full flex facility for the addition of new airlines companies, agents and their branches including their respective removal. However the addition /removal of the airlines companies will be done by the system administrator and the addition/removal of the agents and branches will be done their respective airlines companies.

e) Report generation

This is one of the important parts in the project. Generated report will be helpful in the decision making process. The reports are generated in various formats depending on the degree of their information need.

f) Price comparison and route determination

The system facilitates the customers to compare the prices and facilities among the various airlines and select their desirable routes to their destination. This part is supposed to enhance competition among the airlines companies.

2.2 Task and Activities

System development life cycle includes the various steps towards the development of any system and it has various steps that have to be followed from its start to its end. Similarly this project also used systematic development life cycle with the concept of software engineering as its backbone, which can be summarized, in the following steps:

2.2.1 Development Phase Definition

Software development is carried out systematically in various phases and the phases are broadly categorized into system planning, analysis, design and implementation. Every phase has its own significance so that all the steps should be carried out in a proper way to get the good outcome and these steps are thoroughly followed during the software development process.

2.3 Problem Definition

This step is the initiation of the project where the concept was developed along with the identification of the problem. The concept that was developed required the real system. For that purpose the field visit was made to various airlines companies, their branches and their agents in order to know the existing systems and the problem that they are facing in the existing system. The concept that was generated before the field visit and after visit was the base of the problem definition so that the system could be developed according to the requirements. The data that are required to solve the problem is also collected.

2.4 System Analysis

This phase is the identification of various tasks and activities that has to be performed. It is the process of developing a detailed analysis of the problem so that the developers can better understand the problem. During this phase, the system to be developed was analyzed in the scientific way for the betterment of the system development and also various tools to be used was considered in the analysis phase. This step helps for the better understanding the scope, feasibility and the requirement of the system.

The steps can be summarized as the following.

a) Understanding the Problem

This is similar to the problem definition but this is much in depth so that the problem can be understood in a proper way.

b) Feasibility Analysis

This step deals with the various types of the feasibilities that are needed during the system development. Mainly there are different types of feasibility that are needed to be considered and they are

- i) Technical Feasibility
- ii) Economic Feasibility
- iii) Operational Feasibility
- iv) Schedule Feasibility
- v) Legal Feasibility
- vi) Strategy Feasibility

c) Establishing the System Requirements

This includes the description of the requirements of the user and the groups to whom the product has been targeted.

2.4.1 System Design

This step gives the rough idea of the system to be developed. Various modules were identified for the system such as the input output modules and various intermediate modules that are required for the system development. It is the creation of the roadmap that shows developers how to convert the system requirements into a workable, operational

system by exploring different design and identifying the best design for the project. It is basically seen in the following two steps:

a) Logical Design

It identifies the records and relationships to be handled by a system. It focuses on the logic or the reasoning behind the system by breaking down into subsystem and the relationship among them.

b) Physical Design

It addresses the physical aspect of the system i.e., input and output devices, hardware configuration of memory networks and so on. It also defines data structure, access methods and file organizations

2.4.2 System Development Coding

Once the design process has been accomplished, the system development phase should be started with coding phase along with the coding and testing. Coding is the phase where the pure coding of the project was done according the part that was distinguished in the design process.

2.4.3 Integrating and Testing

This phase runs parallel to the coding process where the code developed is tested and the new codes are integrated to get the system requirement. The testing is of the type as unit testing, system testing and acceptance testing. The testing runs parallel to the coding process so that this is considered as the integrated process.

2.5 Refining the System Needs

Once the specified steps of the software engineering have been carried out in an orderly process then the refinement in the system development can be obtained. This helps in the decision making process for the refinement of the project. The system to be designed requires the fine use of the database as well as the decision-making methods should be incorporated to generate the report. Since the system involves the monetary transactions and other sensitive elements such as tickets, agents, companies and their branches data's, securities. The security has been provided depending upon the information need for the

system user. Account lockup system has been implemented for the agents, companies and their branches to avoid unauthorized accesses to their accounts. Other security factors have been considered accordingly.

2.6 Cost Benefit Analysis

This is another important phase that needs attention for the consideration whether the system provides benefit upon the cost in the existing system or not. The system does not require any special type of hardware and software to be installed on the client system since the access to the system is provided through the browsers using the world wide web, so that the cost will not go much higher than the cost of the computers, internet and the cost of the development charge. The changing scenario that has to be considered during the design also influences the cost.

Chapter 3

System planning and Requirement Analysis

This chapter is the brief description of the design processes that have been incorporated during the development of the system. Since the planning and the requirement specifications are set towards the various designs that are compatible with the requirements. The requirements must be focused out from the end user who is the main source of the requirement

3.1 Schedule Planning

Any project that has to be developed should be done under certain time duration. Similarly, fixed time schedule has been prepared during the system development.

3.2 Requirement Analysis

This is the main thing that needs to be considered for the system development process. Without the system requirement specification, the system development is not feasible. Number of visits to various airlines company, their branches and their agents that help and support in project development at the time of requirement analysis.

3.2.1 Purpose

The purpose of this requirement is to list out the basic information requirements for the users and information providers i.e. the airlines companies as the project is oriented for all airlines companies operating within the nation. The common requirements have been selected according to the information collected from field visit.

3.2.2 Scope:

The “a E r O W E B “is collective module software rather than individual components. The system is full-fledged web based where the users perform every activity using their web browsers. It is an integrated approach of software engineering. The basic components of the system are: -

a) Customer information system

This part of the system is intended for the local users or the information consumers. Information regarding the airlines companies, their branches and their authorized agents from where the deeper level of information regarding the flights are provided in this section.

b) Login section

The system allows the registered users to login to their respective accounts and perform their assigned task for which they are responsible. The registered users may be the companies, branches and their agents with companies having the total control on the operation of their branches and their respective agents.

c) Security section

This component is responsible for maintaining user access security. Any system user whether it is the company, branches or the agents blocked from accessing their accounts for three consecutive wrong accesses to their accounts. The locked users are supposed to contact to the administrator for unlocking their accounts. The section helps to prevent the intruders to login as system users.

d) Ticket system

This part of the system allocates books, transfers or cancels tickets for a particular company, branches or their agents. However the tasks of issuing the tickets to a particular branches or agents are confined to the company only.

e) Storage of branches and agents of the concerned airlines companies

Different branches and the representative agents located on the different parts of the nation can be stored and maintained under the associated airlines company with the sole control of their activities from the main office. The branches and the agents access the same database as the main office but depending on the degree of their information need as categorized by the head office.

f) Data storage in a single database

The storage tank or the data repository is designed with a concept of common requirements analyzed from the field visits of various airlines companies so that it fulfills the data requirements of all airline companies.

g) Addition and removal of airlines companies, agents and branches

The system will provide a full flex facility for the addition of new airlines companies, agents and their branches including their respective removal. However the addition /removal of the airlines companies will be done by the system administrator and the addition/removal of the agents and branches will be done their respective airlines companies.

h) Report generation

This is one of the important part in the project where generated report will be helpful in the decision making process. The reports are generated in various formats depending on the degree of their information needed.

i) Price comparison and route determination

The system facilitates the customers to compare the prices and facilities among the various airlines and select their desirable routes to their destination. This part is supposed to enhance competition among the airlines companies.

j) Weather forecasting

This part of the system forecasts the present and future weather of different locations accessible through airways for flights to and from the destinations. This section includes the minimum and maximum temperatures, the rainfall, clouds and the wind condition of the respective places where the flights are supposed to be held.

3.3 List of Documents

The list of project-related references and applicable documents that bear on this specification has been enclosed with this document.

- i. Field Visit Application Form
- ii. Ticket reservation desk
- iii Administrator desk
- iv. Branch working desk
- v. Agent's working desk
- vi. Customer desk

3.4 Assumptions and Dependencies

No assumptions have been raised for the development of the proposed system as the organizations are using their own software, which differs from the software of other airlines companies. However it is assumed that if the airlines companies agree to apply and work under this developed system, they will be more successful to facilitate their customer and get work under the competitive environment.

The system has been observed to be technically feasible as all the airlines companies have the technical infrastructures needed to adopt the proposed system, coz no any specialized hardware and software are required for them to install in their existing system. Simply their present computers with internet connections will do for this system application. The system can be expanded as per their further requirement along with providing the technical guarantees of accuracy, reliability, easy access and data security.

The system developed is an independent product not relying in any of the third party components and is developed with the most common requirements of most airlines companies for functioning.

3.5 Requirement Attributes

3.5.1 Client

The software is targeted to all the airlines companies, their branches and their agents.

3.5.2 Target Version

This is intended to integrate all existing airlines companies, their branches and their agents operating on different parts of the globe on a single platform and access allowed through web to all the interacting parties and will be upgraded according to their requirements.

3.6 Requirements

3.6.1 Functional Requirements

The software is targeted to all the concerned airlines providing the flight services for the proper management of the whole system of operation. The functional requirements include the following portions.

a) Computers:

Since the system is completely software based, computers at every operating section are the essential parts. This is for the access of the system through the browsers. Any users whether it is the companies or their branches, agents or the customers requires the computer to access the system.

b) Internet connection:

Since the system is web based, the user requires an internet connection to their computers to access the system running on the remote server. They do so with the help of the web browsers installed on their system.

c) The user, administrator, companies, branches, and agents section:

These parts of the system are the internal subsystem that defines the overall system's components. The users are the customers with the minimal supply of information from the system. They can only view the existing airlines, available seats, choose routes and select the best airlines based on comparison provided by the system.

The administrator is the system handler. The administrator can handle the overall system performance and have the total control of the whole system. They can add /remove airlines companies, branches and agents from the system and can add/remove any additional system components as per the operators demand.

The companies, branches and agents are the other system users with the defined level of hierarchy respectively. The company standing at the top level of hierarchy controls their branches and their agents. The branches and the agents work under the company's rules and regulation. The company issues tickets to their branches and agents with proper rules and regulations for them. The agents and branches sell the tickets and can transfer the tickets to their excessive tickets to others who are lacking.

d) The report generation system:

This part of the system is responsible for the generation of reports for the different levels of operating entities except the customers. The category of report generated depends upon the information requirements of the entities.

e) Route and price comparison:

This part of the system compares the shortest route and the lowest price for the customers querying. This function can be implemented by using the Dijkstra's algorithm or other simple comparison algorithms.

3.6.2 Nonfunctional Requirements

a. Usability

This software has a GUI interface that gives the access to the system through the web and is easier for the user to learn since the complexity can be reduced by the modern tools of the OOP. This requires a person who is efficient in the computer, however not a computer professional is required and who understands the airlines transaction system well can be an efficient user.

b. Reliability

The software is a prototype for the requirement and it cannot fulfill the requirements that are not dealt with but can be further extended according to the user's requirement.

c. Performance

The performance of the software is based on how much the airlines companies want to rely on the software for their system management.

d. Supportability

Since the entire OOP is done in the JAVA technology, which has the global market and the functions and classes, are readily available as per requirement, which makes it a reliable platform to use.

e. Scalability

Since it is a kind of MIS this is a large-scale program and the weight of the requirements can affect the performance.

f. Extensibility

Since this is a MIS system it can be further extended to solve the problems that can occur or the problems that has not been identified currently.

3.7 Design Constraints

The software tools used are:

1. JDK 1.5.0 or latest version of JAVA
2. Macromedia Dream weaver
3. Java server pages (JSP)
4. SQL server 2000
5. XML Parser for Java
6. Hypertext markup language (HTML)
7. Bluej

3.8 Interfaces

This section defines the interfaces that must be supported by the application. For simplicity, the interface requirements have been categorized into five sections as follows. The following description for the interface requirements conforms to the mechanism through which the user interacts with the system or software-sections with each other.

3.8.1 User Interfaces

User interfaces will be implemented as Graphical User Interface (GUI). A set of customizable graphical components, whose similar look and feel emphasizes the system consistency. The GUI for desktop and distributed applications must assure the convenience to their users.

3.8.2 Company Interfaces

This is also the GUI but it provides the working platform for the company transactions. Various facilities with much power have been added to this interface. The features like adding/removing agent's branches, generating reports, news etc. relay to concerned authorities have been added here.

3.8.3 Agents/ branches interface:

These GU interface provides the services the ticketing procedures for the branches and agents. The interface provides booking, conformations, billing, transferring tickets plus the invoice and minor report generation to the agents and branches.

3.8.4 Software Interfaces

The system requires drivers for communicating with the adopted database but the drivers need to be installed only at the servers.

3.8.5 Communications Interfaces:

The web browsers such as Microsoft Internet explorer, Netscape browsers, Mozilla Firefox or others provide freely available browsers to communicate between the users and the system.

Chapter 4

System Design

This chapter is the brief description of the design processes that have been incorporated during the development of the system. Since the planning and the requirement specifications are set towards the various designs that are compatible with the requirements.

4.1 Application Environment Establishment

The development of the new system always requires the establishment of the environment where the system has an efficient working condition. This step always takes the operating system into considerations as well as the software package that are used for the software development process. Since JAVA is used as the main programming tool, which is, platform independent so there is not the specific operating system for running the program but following can be recommended for the program.

1. Windows 98
2. Windows XP
3. Windows 2000
4. Other Operating System

Also for the efficient performance of the software the various software tools needed are.

- 1.JSP
- 2.Tomcat
- 3.HTML
- 4.Database SQL server 2000
- 5.XML
- 6.Ireport
- 7.BlueJ

A brief review of these software tools are presented as under:

4.1.1 JAVA SERVER PAGES (JSP):

Java Server Page (JSP) is a technology defined by Sun Microsystems to create dynamic content on the web. Unlike the static HTML page the JSP makes the server side application

more flexible. They are HTML documents that are interleaved with Java, which provides the dynamic content. JSP is a server side application that accepts a request and generates the response. Generally, the request is made from web client, and the response is the generated HTML document that gets sent back to the web client. JSP is a server side application that has access to the resources on the server, such as servlets, Java Beans and Databases.

Java Server Pages technology allows web developers and designers to easily develop and maintain dynamic web pages. JSP enables rapid development of web-based applications that are platform-independent. JSP separates user interfaces from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content. A JSP page is simply an HTML web page that contains additional bits of code that executes application logic to generate dynamic content. This application logic may involve JavaBeans, JDBC objects, Enterprise Java Beans (EJB), and Remote Method Invocation (RMI) objects, all of which can easily be accessed from a JSP page. For example, a JSP page may contain HTML code that displays static text and graphics, as well as a method call to a JDBC object that accesses a database; when the page is displayed in a user's browser, it will contain both the static HTML content and dynamic information retrieved from the database.

The separation of user interface and program logic in a JSP page allows for a very convenient delegation of tasks between web content authors and developers. It also allows developers to create flexible code that can easily be updated and reused. JSP pages are automatically compiled as needed; web authors can make changes to presentation code without recompiling application logic. This makes JSP a more flexible method of generating dynamic web content than Java servlets, whose functionality Java Server Pages extend.



Fig2: Web server Implementation in JSP

Since the system is a web based application and the contents on the web needs to be changed frequently. The application of JSP has been the important factor to be considered during the development and the coding phase.

4.1.2 HOW JSP'S BECOMES SERVLETS:

Initially a JSP or servlet class is started. The server is responsible for instantiating the JSP/Servlet and it uses the new() method to accomplish this .This new() method is the Java method for creating space in memory for an object. After the JSP/Servlet is instantiated, the init(...) method is invoked for initialization purposes. Then the JSP/Servlet moves into the ready state and is prepared to handle client requests. Service (..) method can be written to handle business or doGet(..) or doPost(..)method for business logic. The JSP/Servlet is then destroyed when the server invokes the destroy() method. The garbage collector cleans up the memory with the finalize () method.

The JSP/servlet lifecycle is shown below:

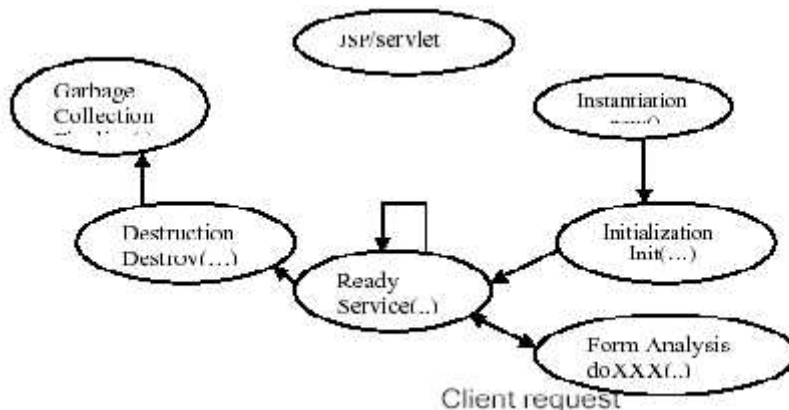


Fig3: The lifecycle of JSP/Servlets

4.1.3 Blue J

The bluej software is found to a very beneficial tool developed for java platform. The tool was used for testing the classes created and their connections to the databases. Relations between the classes were easily analyzed which helped to construct the class diagrams more easily and with accuracy.

4.2 Application Design

This is the design of the input and output sections, various process involved and the interfaces that are required for the interaction of the program with the user.

4.2.1 Input Design

The input to the system can be of many forms such as ticket information, agents' information, companies' information, and customers' information.

4.2.2 Output Design

The output of the program is the computer monitor that gives the output in the form of softcopy to the user also, the hard copy output can be obtained by publishing the weekly annual, monthly or yearly reports. The output of the program always needs to be stored into the data warehouse for the future use.

4.2.3 Process Design

Process model shows the overall process and the processes that are supported by the system. Data flow models may be used to show the processes and the flow of information from one process to another. The figures that show the process in this project shown in the Data Flow Diagram that incorporates the process along with it. DFD graphically illustrates movement of data between external entities, the process and data stores within the system.

a) Object Oriented Analysis (OOA)

It is the technique to study existing objects and to see if they can be reused or adapted for the new users and define new and modified objects that will be combined with existing objects into useful business computing applications. This analysis is extensively used in the development of the project which defines one of the model or the paradigms of software development called as 'system assembly from reusable

Components' assumed to be the part of the system and the system development process focus on integrating these parts rather than developing them from the scratch. This approach is centered on a technique referred to as object modeling.

b) Object Modeling

It is a technique for identifying objects within the system environment and the relationship between those objects. The object modeling approach prescribes the use of methodologies and diagrammatic notations that are completely similar to the ones used for data modeling and process modeling. Unified Modeling Language is a set of modeling conventions that is used to specify or describe a software system in terms of objects.

c) UML

The UML does not prescribe a method for developing systems-only annotation, which is now widely accepted as a standard for object modeling. UML provides a wide array of diagrams that can be used for analysis and design at both the system and the software level. The UML notation is similar to the flowchart and is used to represent what happens as the system performs its function. The following nine types of diagrams have been defined in the UML:

- 1) Use case diagrams
- 2) Class diagrams
- 3) Object diagrams
- 4) Sequence diagrams
- 5) Collaboration diagrams
- 6) State diagrams
- 7) Activity diagrams
- 8) Deployment diagrams
- 9) Component diagrams

All of these diagrams are not required for the purpose of development, so only some of them have been used in this software design process. Brief descriptions of these diagrams are presented as under:

1) Use Case Diagrams

It graphically depicts the interactions between the system, external systems and users. It graphically describes who will use the system and in what ways the user expects to interact with the system. The use case narrative is used in addition to textually describe the sequences of steps of interactions.

2) Class Diagrams

Class diagrams depict the system's object structure. It shows object classes that the system is composed of as well as the relationships between those object classes.

3) Object Diagrams

These are similar to class diagrams, but instead of depicting object classes, that model actual object instances-showing the current values of the instance's attributes. The object diagram provides the user with the snapshots of the system's objects at one point in time. This diagram is not used as often as a class diagram, but when used, it can help a developer better understanding the structure of the system.

4) Sequence Diagrams

It graphically depicts how objects interact with each other via messages in the execution of a use case or operation. It illustrates how messages are sent and received between objects and in what sequence.

5) Collaboration Diagrams

These diagrams are similar to sequence diagram but do not focus on the timing or sequence of the message. Instead it represents the interaction between objects in a network format.

6) State Diagrams

State diagrams are used to model the dynamic behavior of a particular object. It illustrates an object's life cycle-the various states that an object can assume and the events that cause the objects to transition from one state to another.

7) Activity Diagrams

Activity diagrams are used to graphically depict the sequential flow of activities of either a business process or a use case. They also can be used to model actions that will be performed when an operation is executing as well as the results of those actions.

8) Deployment Diagrams

Deployment diagrams describe the physical architecture of the hardware and software in the system. It depicts software components, processors, and devices that make up the system's architecture

9) Component Diagrams

Components diagrams are used to graphically depict the physical architecture of the system. They can be used to show how programming code is divided into modules.

a) Object

Objects are the basic runtime entities in an object-oriented system. An object is something that is or is capable of being seen, touched, or otherwise sensed, and about which users store data and associated behavior each of which represents an instance of some class. These classes may be in the form of hierarchy. The major elements of the objects are data abstraction and encapsulation, modularity and hierarchy whereas its minor elements are persistence, typing and concurrency.

b) Attributes

Attributes are the data that represents characteristics of interest about an object. An instance of an object consists of the values for the attributes that describe a specific person, place, thing or event.

c) Behavior

Behavior refers to those things that the object can do and corresponds to function that act on the object's data. In object oriented circles, an object's behavior is commonly referred to as a method, operation or service.

d) Data abstraction and Encapsulation

Encapsulation is the process of wrapping up of data and functions into a single unit called class or in another way it can be defined as the approach of keeping set of data and methods together inside one container (the capsule or the class). Abstraction refers to the act of representing essential features without including the background details or explanations; it also refers to as information hiding.

d) Class

A class is a set of objects that share the same attributes and behavior. A class is sometimes referred to as an object class. In fact objects are the variables of the type class.

e) Inheritance

Inheritance is the process by which objects of one class acquire the properties of objects of another class. It supports the concept of hierarchical classification. Inheritance means that methods/attributes defined in an object class can be inherited or reused by another object class. The concept of inheritance provides the idea of reusability.

f) Generalization/specialization

It is a technique wherein the attributes and behaviors that are common to several types of object classes are grouped into their own class, called super type. That object class then inherits the attributes and methods of super type object class.

g) Message

A message is passed when one object invokes one or more of another object's methods to request information or some action. When a program is executed, objects interact by sending messages to one another.

h) Polymorphism

The term polymorphism means the ability to take more than one form. It means "one name multiple forms". Applied to object oriented techniques, an operation may exhibit different behaviors in different instances it means that the same named behavior may be completed differently for different object/class

4.2.4 Interface Design

The interface design is another important aspect that needs to be considered during the program development because the interface is the main user interaction section that the user of the program uses for their convenience purpose of the data entry and the handling of the software.

Chapter 5

System Development

Introduction

This chapter deals with the technologies and the techniques that are used in transforming the design into the real system. It also describes briefly how the system has been developed by the use of tools such as the JAVA, (JSP) Macromedia dream weaver. During the development phase the spiral model of the software development is used. Here the risk factors are considered and resolved throughout the process. The product that has been developed once is passed through the spiral cycle of planning, risk analysis, engineering, construction and customer evaluation.

5.1 Database Management System (DBMS)

A database is a collection of related data necessary to manage an organization. It includes transient data such as input documents, reports and intermediate results obtained during processing.

A database models the data resource of an organization using the relationship between different data items. The data model is independent of any application programs. Parts of one database may be used by different application programs.

A DBMS is a set of procedures that manage the database and provide the access to the database in a form required by any application program. It effectively ensures that necessary data in the desired form is available for diverse applications of different organizations. Since this project is a database oriented system for which the SQL server 2000 has been used for the storage of the data in the systematic way in the tables. In fact database is a collection of interrelated files or tables in which records of one file/table are physically related to the records in another file/table. A MIS is a type of Information System that requires the help of the DBMS for the storage of the data and without the DBMS MIS won't be meaningful. The database and the DBMS are closely related to each other; however one cannot be imagined without the other. The relationship is shown as under:

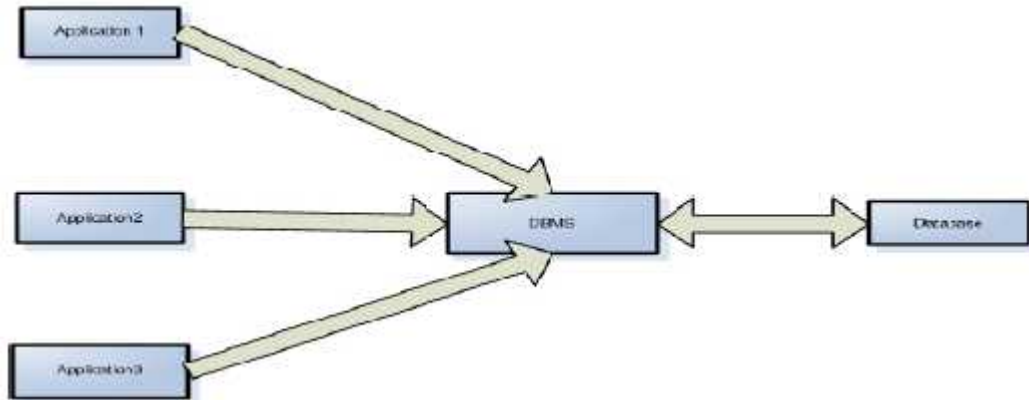


Fig 4: Database and DBMS relationship

5.1.1 Objectives of Database Management

1. A database should provide for efficient storage, update, and retrieval of data.
2. A database should be reliable, high integrity and promote user trust in that data.
3. A database should be adaptable and scalable to new and unforeseen requirements and applications.
4. A database should identify the existence of common data and avoid duplicate Recording. Selective redundancy is sometimes allowed to improve performance.

5.1.2 A good database system should:

1. Satisfy the current and future application need of an organization.
2. Cater to unanticipated user requirements in the best possible way.
3. Be expandable with the growth and changes in the organization.
4. Be easy to modify with changes in hardware and software environments.
5. Validate the data before storage.

5.2 Management Information System:

MIS is a set of organized procedures which when executed provides information from raw data to support decision making. It is an integrated user machine system that provides information to support operations, management analysis and decision-making function in an organization

The Management Information System (MIS) is a system to convert data from internal and external sources into information, in an appropriate form, to manager at all levels in all functions to enable them to make timely and effective decisions for planning, directing and controlling the activities for which they are responsible.

The **A E R O W E B** MIS system is used to convert all the input and the output transaction that occurs within the airlines system with a period of time and provide the required information to the managers at all the levels so that good decision can be made. MIS is defined as systems, which provide tools to enable the administrative tasks of the organization to be carried out, provide information to managers to enable them to plan and control their operational area and provide data to inform strategic planning. They include both business systems available to all and office systems used by the central administration. To be successful, MIS must be designed and operated with due regard to organization and behavioral principles as well as technical factors. Management must be informed enough to make an effective contribution to system design and information specialists (System analysts, accountants, operations researchers and others) must become more aware of managerial functions and needs so that jointly, more effective MIS are developed. Management do not always know what information they need and information specialists often do not know enough about management to be able to produce relevant information for the managers they serve.

5.2.1 Design:

MIS design can be divided into conceptual and detailed system design.

a) Conceptual System Design

In conceptual system design, the activities like definition of problems, setting system objectives, establishing system constraints, determination of information needs and information sources, developing the alternative conceptual design and selecting and documenting the best design of the system concept are performed. The process of developing the conceptual design is shown in the diagram below:

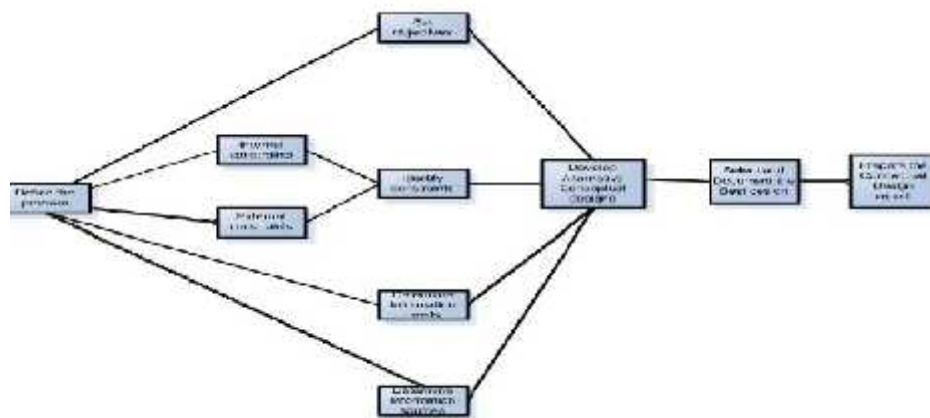


Fig 5: conceptual system design

b). Detailed System Design

Detailed system design concerned with system in detail. It defines subsystems like detailed operating subsystems and information flows, degree of automation of each operation. The other procedures like designing forms, developing the database, system testing modeling, and simulation and planning and documenting the detailed design also falls under the detailed system design. It informs and involves the input, output and processing of organization as well as software, hardware tools and documentation of the detailed design.

The process of detailed system design is shown in the diagram below:

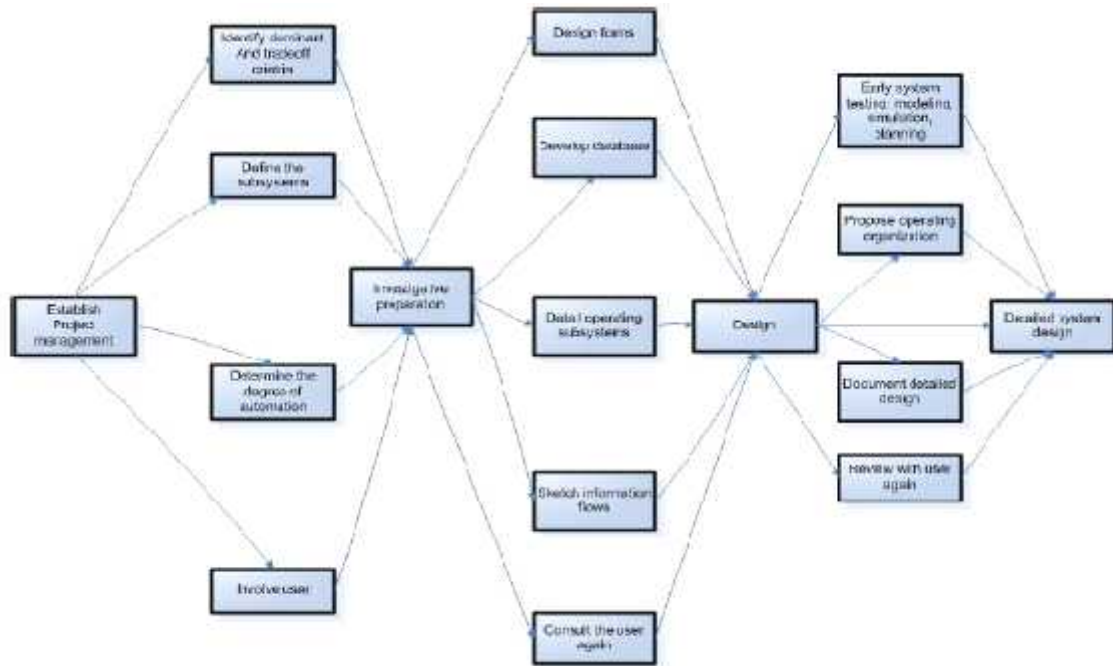


Fig 6: Detailed system design

5.2.2 Implementation:

Organized and developed procedures are applied for implementing MIS like training and operating personnel, computer related acquisition, developing form for data collection and information dissemination developing files, test the system, evaluation of MIS, control and maintenance of the system etc

5.3 Automation Process

Automation process is a part of an information system and this is also incorporated in our project to provide efficient output. It is an assembly of computer hardware, software, firmware or any combination of these configured to accomplish specific information-handling operations, such as communication, computation, dissemination, processing, and storage of information.

5.3.1 Benefits

1. Faster cycle times: more new features added and processed in short time
2. Reduced costs: reduce extensive human labor
3. A smart scalable, knowledge based automation infrastructure: give you fast controllable, repeatable, predictable and automated results.
4. Increased competitiveness: this gives high quality product in reasonable price and in time.

5.3.2 Automated a E R O W E B system

1. For the airlines system, the computer information system can do routine supervision and Continuous monitor, so that extensive checking by the employee is not required.
2. So that all level managers communicate with the other levels through web with out much effort.
3. Computer based information system monitors every financial and other major transaction, So there is just no need for extensive checking, there by eliminating unnecessary, non- Value adding positions for middle management.
4. Work flow automation captures the formal organization structure faithfully and incorporates them in to the design of IS that eventually helps in the development of a good system.

5.4 Visual Programming API

JAVA:

It is an Object Oriented Programming Language that was brought into the market by the SUN Micro system. Object oriented technique is the new method of programming that brought about a great change in the software development. It is away of developing software by describing problems through the use of elements or objects from the problem space rather than a sequential set of steps to execute in the computer. The core JAVA Application Programming Interfaces are a collection of the pre-built components called class libraries. Instead of reinventing the wheel with Java one can start with the standard

libraries. These standard libraries have evolved over time, gaining more pre-built components with each Java release.

5.5 Java server pages

With the release of the JavaServer Pages 2.0 specification, a new milestone has been reached. Looking backward, we see that using JavaServer Pages has become a time-tested way to develop Web pages with dynamic content. JSPs are the technology of choice in many Web-based applications. Developers know that, because JavaServer Pages have Java at their foundation, they can build upon all that the platform provides, including a robust and rich set of technologies.. Beginning with JavaServer Pages 2.0, it's become easier to create Web applications. In fact, , it is easier to build to build sophisticated sites that use databases or work with XML documents—things that used to require an intermediate knowledge of the Java programming language and its platforms

When the browser asks the Web server for a JSP, the Web server passes control to a JSP container. A container works with the Web server to provide the runtime environment and other services a JSP needs. It knows how to understand the special elements that are part of JSPs. Because this is the first time this JSP has been invoked, the JSP container converts it into an executable unit called a servlet. The entire page, including the parts that are in HTML, is translated into source code. After the code has been translated, the JSP container compiles the servlet, loads it automatically, and then executes it. Typically, the JSP container checks to see whether a servlet for a JSP file already exists and whether the modification date on the JSP is older than the servlet. If the JSP is older than its generated servlet, the JSP container assumes that the JSP hasn't changed and that the generated servlet still matches the JSP's contents. Because it takes some time to generate a servlet and compile it, the JSP container wants to minimize the number of compiles it has to perform, so it tries to avoid unnecessary compiles.

Servlets:

Servlets are an underlying technology behind JavaServer Pages. Servlets allow the programmer to work directly with requests made to a Web server and to form responses that include content that is returned to a client. Unlike JSPs, which are oriented toward presentation, with their combination of HTML and scriptlets or other elements, servlets are composed entirely with Java. As you might expect, servlets are

great for performing lower-level functions that work with data or implement business models. Just as an applet is a small application that extends the functionality of a web browser a servlet is a small piece of code that extends the server functionality.

Description of java servlet class

Unlike JSPs, servlets are pure Java classes. The *ServletContext* interface provides a means for the servlers to communicate with the surrounding web or application server. This communication can take the form of requests for the system resources, reports written by servlets to a log file, and so on. Indirectly *ServletContext* also allows servlets to communicate with each other, primarily by sending requests to other pages. The servlet classes are defined in the *javax.servlet* package . The javax.Servlet package defines three classes. Two of these, ServletInputStream, read and write data respectively. The third *GenericServlet*, implements boththe servlet and the *Servlet Config* interfaces and forms the basis for all real servlets. The web specific versions of the servlet classes are included in a package called *javax.servlet.http*, the heart of which is the *HttpServlet* class, which extends the *GenericServlet*. This class's *service()* method takes *HttpServletRequest* and *HttpServletResponseobjects* instead of the generic versions from *javax.Servlet* package.

Compiling the Servlet

Before you compile the servlet, make sure the servlet classes are in your class path. The location of the servlet classes varies depending on which servlet container you are using. For example, if you are using the Apache Tomcat server, the servlet-api.jar file is located in the common\lib directory underneath your main Tomcat installation

Where Are the Classes in Tomcat 5?

In past versions of Tomcat, the classes that support servlet development have been located in `servlet.jar`, located in the `common\lib` directory. For Tomcat 5.0, which includes support for the Servlet 2.4 and JSP 2.0 specifications, the class is named `servlet-api.jar`—at least for the moment.

Where Are the Classes in the J2EE SDK?

If you are using the J2EE SDK, the JAR file that includes support for servlets and JSPs is named `J2EE.jar` and is located in the `lib` directory off of the J2EE home directory. Since it has been used the Tomcat, and J2EE SDK, we may have to adjust the procedures slightly for compiling and running the Servlets.

Tomcat Is Part of the Standard

Tomcat is the official playground for servlet and JSP reference implementation development. Once the specifications start to firm up, development on a new version of Tomcat begins right away. A release of Tomcat is ultimately bundled with Sun's J2EE SDK when the SDK is finalized. Tomcat itself is a production quality implementation and is continually refined. Tomcat 5.0 supports the Java Servlet 2.4 and JavaServer Pages 2.0 specifications.

5.6 Tools Used For the Developments

5.6.1 Macromedia Dreamweaver

Macromedia Dreamweaver MX is a professional HTML editor for designing, coding, and developing websites, web pages, and web applications. Whether one enjoys the control of hand-coding HTML or prefers to work in a visual editing environment, Dreamweaver provides them with helpful tools to enhance their web creation experience.

The visual editing features in Dreamweaver allow to quickly creating pages without writing a line of code. We can view all our site elements or assets and drag them from an easy-to-use panel directly into a document. We can streamline our development workflow by creating and editing images in Macromedia Fireworks, then importing

Dreamweaver is fully customizable. We can create our own objects and commands, modify keyboard shortcuts, and even write JavaScript code to extend Dreamweaver capabilities with new behaviors, Property inspectors, and site reports.

Since the Macromedia Dreamweaver allows to create the dynamic web pages. A personal web server is most important for testing the pages. For this purpose we have choose the Apache's tomcat server.

5.6.2 Tomcat 5

Tomcat 5 implements the Servlet 2.4 and Java Server Pages 2.0 specifications from the Java Community Process, and includes many additional features that make it a useful platform for developing and deploying web applications and web services. The Tomcat runs on a localhost on port 8080.

SETTING UP APACHE'S TOMCAT SERVER:

A servlet is a java class that implements the servlet interface and accepts requests and generates responses. The requests can come from Java classes, Web clients or other Servlets.

Servlets perform a wide range of functions. A servlet can:

- a) Dynamically build and return an HTML file.
- b) Process user input.
- c) Interact with server resources such as databases, other application.
- d) Process input from many clients.
- e) Allow the server to communicate with a client applet.
- f) Forward requests from server to another for load balancing purposes.
- g) Create and return an entire HTML web page containing dynamic content based on the nature of the client request.
- h) Communicate with other server resources, including databases and Java-based application. The apache's Tomcat server is local web hosting server. Since the application was a web based application, the server was used to test the pages on the local machines.

5.6.3 Microsoft SQL Server 2000

Microsoft SQL Server2000 extends the performance, reliability, quality, and ease-of-use of Microsoft SQL Server version 7.0. Microsoft SQL Server 2000 includes several new

features that make it an excellent database platform for large-scale online transactional processing (OLTP), data warehousing, and e-commerce applications. The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services.

This tool has been used for the purpose of the database maintenance system where various sequential queries languages were implemented in order to view the data in the tables or to store the data in the tables. Java Data Base Connector (JDBC) has to be incorporated in order to implement the database in Java programming. This is much more secure database service provider and the handling is very easy with the help of the sequential query language

Features of SQL Server 2000

Microsoft SQL Server 2000 features include:

a) Internet Integration.

The SQL Server 2000 database engine includes integrated XML support. It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web applications.

b) Scalability and Availability.

The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows 98 through large, multiprocessor servers running Microsoft Windows 2000 Data Center Edition. SQL Server 2000 Enterprise Edition supports features

such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.

c) Enterprise-Level Database Features.

The SQL Server 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database. SQL Server 2000 distributed queries allow you to reference data from multiple sources as if it were a part of a SQL Server 2000 database, while at the same time, the distributed transaction support protects the integrity of any updates of the distributed data. Replication allows you to also maintain multiple copies of data, while ensuring that the separate copies remain synchronized. You can replicate a set of data to multiple, mobile, disconnected users, have them work autonomously, and then merge their modifications back to the publisher.

d) Ease of installation, deployment, and use.

SQL Server 2000 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites. SQL Server 2000 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems. These features allow you to rapidly deliver SQL Server applications that customers can implement with a minimum of installation and administrative overhead.

e) Data warehousing.

SQL Server 2000 includes tools for extracting and analyzing summary data for online analytical processing. SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

Database Architecture

Microsoft SQL Server 2000 data is stored in databases. The data in a database is organized into the logical components visible to users. A database is also physically implemented as

two or more files on disk. When using a database, you work primarily with the logical components such as tables, views, procedures, and users. The physical implementation of files is largely transparent. Typically, only the database administrator needs to work with the physical implementation.

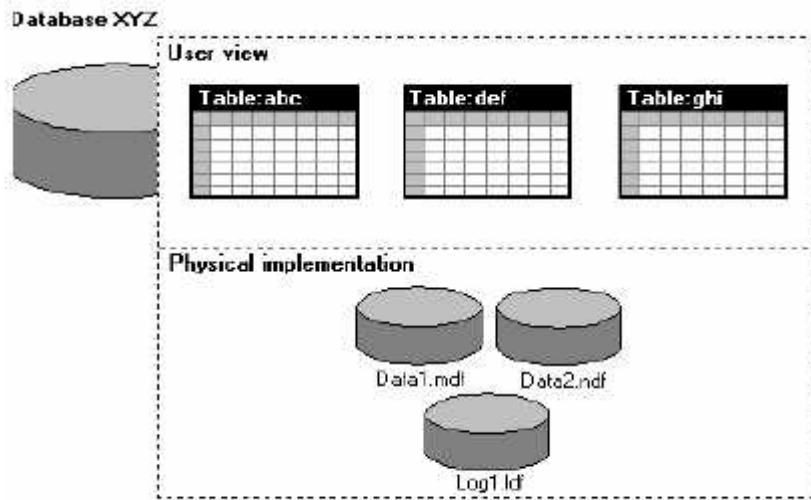


Fig7: Database implementation

Each instance of SQL Server has four system databases (**master**, **model**, **tempdb**, and **msdb**) and one or more user databases. Some organizations have only one user database, containing all the data for their organization. Some organizations have different databases for each group in their organization, and sometimes a database used by a single application. For example, an organization could have one database for sales, one for payroll, one for a document management application, and so on. Sometimes an application uses only one database; other applications may access several databases.

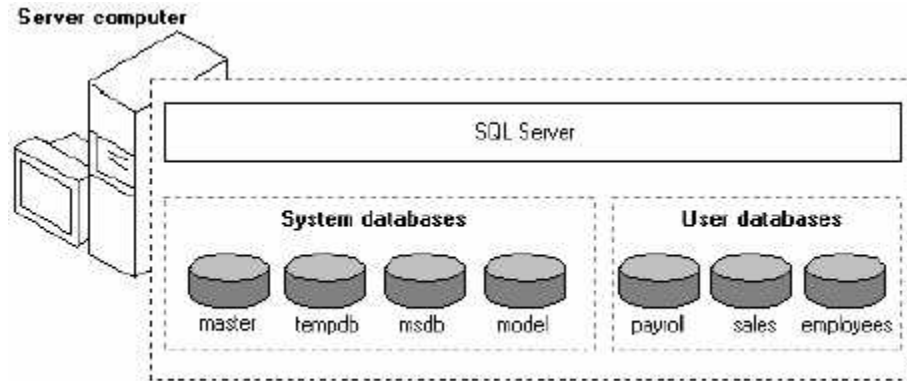


Fig 8: Server and database interaction

It is not necessary to run multiple copies of the SQL Server database engine to allow multiple users to access the databases on a server. An instance of the SQL Server Standard or Enterprise Edition is capable of handling thousands of users working in multiple databases at the same time. Each instance of SQL Server makes all databases in the instance available to all users that connect to the instance, subject to the defined security permissions.

When connecting to an instance of SQL Server, your connection is associated with a particular database on the server. This database is called the current database. You are usually connected to a database defined as your default database by the system administrator, although you can use connection options in the database APIs to specify another database. You can switch from one database to another using either the Transact-SQL `USE database name` statement, or an API function that changes your current database context.

SQL Server 2000 allows you to detach databases from an instance of SQL Server, then reattach them to another instance, or even attach the database back to the same instance. If you have a SQL Server database file, you can tell SQL Server when you connect to attach that database file with a specific database name.

Data Warehousing and Online Analytical Processing

Microsoft SQL Server 2000 provides components that can be used to build data warehouses or data marts. The data warehouses or data marts can be used for sophisticated

enterprise intelligence systems that process queries required to discover trends and analyze critical factors. These systems are called online analytical processing (OLAP) systems. The data in data warehouses and data marts is organized differently than in traditional transaction processing databases. Enterprise-level relational database management software, such as SQL Server 2000, was designed originally to centrally store the data generated by the daily transactions of large companies or government organizations. Over the decades, these databases have grown to be highly efficient systems for recording the data required to perform the daily operations of the enterprise. Because the system is based on computers and records the business transactions of the enterprise, these systems are known as online transaction processing (OLTP) systems.

The data in OLTP systems is organized primarily to support transactions, such as:

- a) Recording an order from a point-of-sale terminal or entered through a Web site.
- b) Placing an order for more supplies when inventory levels drop to a defined level.
- c) Tracking components as they are assembled into a final product in a manufacturing facility.
- d) Recording employee data.
- e) Recording holders of licenses, such as restaurant or driver licenses.

5.6.4 OLAP Systems

Systems designed to handle the queries required to discover trends and critical factors are called online analytical processing (OLAP) systems. OLAP queries typically require large amounts of data. OLAP data is organized into multidimensional cubes. The structure of data in multidimensional cubes gives better performance for OLAP queries than data organized in relational tables. The basic unit of a multidimensional cube is called a measure. Measures are the units of data that are being analyzed. The specific dimensions and measures defined for the cubes in any particular OLAP system depend on the kinds of analysis important to the enterprise. Transforming OLTP data from relational tables into OLAP cubes, and the design of the cubes, is a complex area that is the subject of many third-party books. OLAP systems operate on OLAP data in data warehouses or data marts. A data warehouse stores enterprise-level OLAP data, while a data mart is smaller and

typically covers a single function in an organization. Bluej The BlueJ environment was developed as part of a university research project about teaching object-orientation to beginners. The system is being developed and maintained by a joint research group. The aim of BlueJ is to provide an easy-to-use teaching environment for the Java language. Special emphasis has been placed on visualization and interaction techniques to create a highly interactive environment that encourages experimentation and exploration. BlueJ is implemented in Java, and regularly being tested on Solaris, Linux, Macintosh, and various Windows versions. It should run on all platforms supporting a recent Java virtual machine.

BlueJ supports:

-) Fully integrated environment
-) Graphical class structure display
-) Graphical and textual editing
-) Built-in editor, compiler, virtual machine, debugger, etc.
-) easy-to-use interface, ideal for beginners
-) Interactive object creation
-) Interactive object calls
-) Interactive testing
-) Incremental application development

In particular, Bluej has the environment that is really object-oriented. An environment that emphasizes classes and objects as its basic units of interaction, we are naturally led into a way of thinking in terms of classes and objects. Blue j provides an environment that supports interaction and experimentation. We want an environment that uses visualization to display class structure. After all structuring the problem is one of the most important aspects of object-oriented programming.

Chapter 6

System Implementation and Testing

6.1 Motivation

Since this is a distributed system and web based software the implementation of this software will be very much helpful to the targeted group. This will surely be the product that the dispersed airlines companies are looking forward to implement so that they can achieve the goals that they wanted. New technology and changes are always good and appreciated, but it is not always easy. Converting to a new system necessitates that system users be trained and provided with documentation. Training can be performed one by one or in a group. Since this is a web based application, no specialized type of training is required a simple software training can be held for the first users

6.2 Installation

The import phase of the software development can be considered as the installation and the implementation of the software that has been developed for a particular purpose. This step specifies specification of the various software required to be installed. As well as the definition of the hardware and that is required for the fulfillment of the proper working of the software that has been developed.

The following Hardware and additional software are the main requirement of that Aero Web system:

-) Operating system Windows or Linux System
-) CPU 1.6 GHz and above
-) System Memory(RAM) 256MB or above
-) Hard drive 100GB or as per required
-) Network Cables
-) Database Language MS SQL Server 2000
-) Program Platform JVM (Java Virtual Machine)
-) Web server

6.3 Technician Interface

- a) Updated design and layout. All buttons on the ticket form, including the save and print buttons, are now available on a toolbar the form.
- b) Settings dialog has been updated to include a tabbed interface to easily categorize the settings types.
- c) Technicians can define Live Support Chat auto text. This is the text that is automatically sent when the tech joins a chat session. Example: “Hello. How can I help you today?”
- d) Technicians can now add customers from the technician interface. If the customer ID on the new ticket is not found in the database, the tech will be prompted to create it. This feature can be turned on and off in the admin console, and can be setup for use with our without admin approval for added customers.
- e) Technicians now have a calendar where they can enter upcoming tasks and view existing ones, including due dates and completion times. Techs can also add their tasks to the calendars of other techs and other groups, for group projects. The technician can be alerted to pending tasks for the day when logging in.
- f) Billing functionality is now available in the technician interface. Here technicians can enter time and descriptions of billable jobs, easily tracking time and cost. Full invoicing is available in the administration console.

6.4 Customer Interface

- a) Completely redesigned layout, which is more visually appealing and easier for your customers to use.
- b) The administrator can more easily add your own logo to the customer interface, including a web based upload feature.
- c) Customers can now view their chat session history for a quick reference to previous conversations with your technicians.
- d) There is a new News section. Here you can display important news to your customers, which they will see right when they log in. Customers can also view previous news articles you have posted.

- e) The “Open Tickets”, “Closed Tickets” and “Search Tickets” pages have been consolidated into a single “My Tickets” page, where customers can easily filter and search their existing tickets.
- f) Customers now have the ability to re-open closed tickets, if you allow it in the administration console’s settings. This can cut down on customers having to create multiple tickets for the same issue.
- g) If billing functionality has been enabled and customer invoice viewing has been turned on in the administration console, customers can view and print their complete invoices.
- h) The redemption form reward the customer a certain weight age for each of their flight from a particular airlines company, which goes on accumulating in each of their successive journey from that airline. A free ticket is awarded after the weight age reaches a certain limit.

6.5 Administrative Console

- a) New database functionality, such as SQL Server connection testing.
- b) Improved knowledge base with a better, faster rich-text editor that allows attaching of images and files. Articles can also be put into categories you create, and can also be set to “Internal Only” for articles that should only be available to technicians.
- c) News section for posting important news announcements to the customers interface.
- d) Complete billing functionality to create and view invoices. You can choose to track costs based on a fixed cost per ticket, a fixed cost or jobs or variable costs per job. You can define all categories, costs, taxes and billing cycles.
- e) Administrator has the full control over its branches and agents. Administrator can add/remove and control all the operation of its agents/branches.
- .f) Improved reporting – You can now use a report builder to create, save and run your own reports. There are many different options to choose from to report the data you need to see. This will show you which customer’s tickets get resolved the fastest/slowest and the speed at which technicians and groups resolve tickets.

6.5.1 Configuration and Administration > Reports > Build Reports

System can use this page to build, run and save your own custom reports. The Title and Description fields are required. Given the report a short Title that reflects its function. You can enter a more detailed description of what the report does in the Description field. Select which tickets you want to include on the report. You can choose from “All Tickets”, “Tickets Created” and “Tickets Closed” for the time frame you want to specify.

You can enter an exact date range for the report or select a time Period. Selecting an exact date range will only display tickets matching your criteria between the given dates. If you select a period, such as “This Month”, it will display the tickets matching your criteria and that were created, closed or both (depending on the value you have selected) for the current month. For example, if you want to see all tickets that were created this month, you would select “Tickets Created” from the first dropdown, and select “This Month” as the Period.

If you want to display tickets that contain a certain keyword, enter it in the Keyword field.

Select the customers whose tickets you want displayed on the report.

“Tech Ownership” defines the relationship between the ticket and the Technician ID you select in the next field. Example: If you want to see tickets with the technician “guest” as the assigned technicians, select “Assigned To” from the “Tech Ownership” list and the technician’s ID from the “Technician ID” field. If you would rather view tickets by group rather than technician ID, select the group name instead.

Select the Ticket Statuses you wish to include in the report. For example, you may only want to see tickets with a status of “Closed”.

Select the Ticket Sources you wish to include in the report. For example, you may only want to see tickets that were created from incoming e-mails.

Select the Ticket Categories you wish to include in the report. For example, you may only want to see tickets that were placed in the category of “Error”, infant or adults. (Your actual Ticket Categories may vary)

Under “Sort Order”, select the field you wish to sort by and the order in which to search. Example: If you want the report to be sorted by the date tickets were opened from most recent to oldest, you would select “Date Opened” and “Descending”. You cannot sort by Added Fields.

Select the columns you wish to display on the report. You can add or remove columns by selecting them and using the arrows between the “Selected Columns” and the “Available Columns” fields, or you can double click on the column names. Columns will be displayed in the order that they are listed in the “Selected Columns” field.

Once you have entered and selected all of your desired values, click the “Run Report” button. The results of the report will be displayed. If no tickets are found, you will see the message “No tickets found matching your search criteria”. If tickets were found, you will see the list of tickets it found. In either case, you can click on the “< Back” button to refine your criteria, or click the “Save” button to save the report to the database.

Saving the report to the database will save the compiled SQL Statement for quickly running the report in the future.

Once the report is saved, you will be redirected to the Saved Reports page.

Snapshots of the various interfaces are provided below for the general user understanding:

6.6 System Testing

System testing is the process of testing the software that has been developed by the process that gone through analysis, design and development. It includes well defined steps that results in the successful construction of the software. A software testing strategy should be flexible enough to promote a customized approach. At the same time, it must be rigid enough to promote reasonable planning and management tracking as the project progresses. A number of software testing strategies have been proposed in the literature. All provide the software with a template for testing and all have the following characteristics.

- a) Testing begins at the component level and works “outward” toward the integration of the entire computer based system.
- b) Different testing techniques are appropriate at different points of time.
- c) Testing is conducted by the developer of the software and an independent test group.
- d) Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

6.6.1 Verification and Validation

Software test is one element of the broader topic that is often referred to as verification and validation (V&V). Verification refers to the set of activities that ensure that software completely implements a specific function. Validation refers to the set of activities that ensure that the software that has been developed is traceable to the customer requirements. Verification and validation encompasses a wide array of Software Quality Assurance (SQA) that include formal technical review, quality and configuration audits, performance monitoring, simulation, feasibility study, documentation review, database review, algorithm analysis, development testing, qualification testing and installation testing. V&V is the name given to the checking and analysis process that ensures that software conforms to its specification and meets the need of the customers who are paying for it. V&V is a whole lifecycle process. It starts with the requirements reviews and code inspections to product testing. The V&V activities are followed at each stage of the software process. These activities check that the results of process activities are as specified. Although verification and validation seems the similar terms they are used to refer the two different activities as follows:

Validation: are we building the right product?

Verification: are we building the product right?

6.6.2 Unit Testing

Unit testing focuses verification effort on smaller unit of software design- the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. It is white box oriented, and the step can be conducted in parallel for multiple components

In this project development unit testing of the different modules were done separately such as the company's module, the customer module, branches and agent's module and administrative module. Each of these modules contains various different sub modules that are tested during the design and development process. This is just the first step in the

testing process where the modules that are developed are tested whether they can perform correct work or not.

6.6.3 Integration testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and builds program structure that has been dictated by design.

In the project development process the integration testing was done once the various components of the definite module was developed so that after integrating the various components in the module the errors generated were cored and integrated into a single module. These modules are capable of working independently as directed by the program structure. The integration testing in our case has been done in two stages one along with the different components of the same module to check whether they are performing correctly or not, the next one to integrate the various models into the single system.

The testing phases and the software processes are carried out simultaneously as shown in the diagram below:

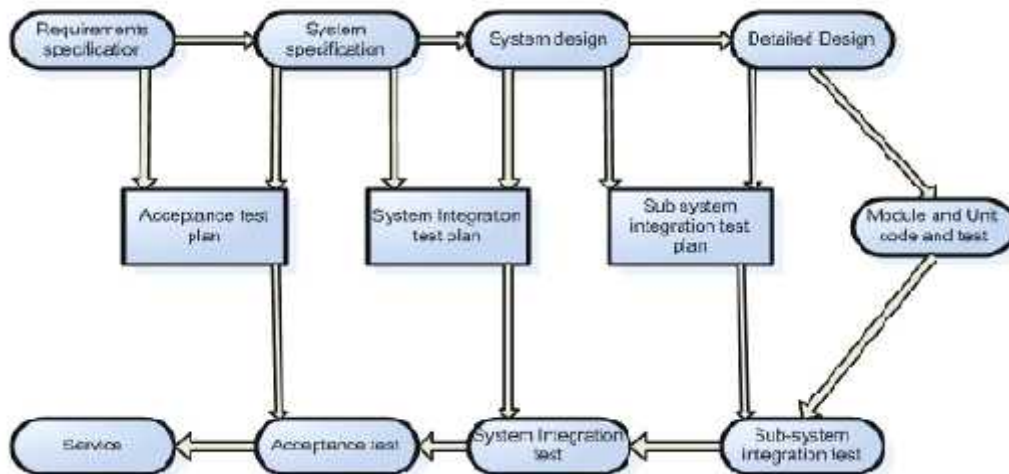


Fig 9: Testing phases in the software process

6.6.4 White Box Testing

The purpose of the white box testing is to check whether the program is working according to the program structure developed or not. The white box testing include the process of providing some known and desired data to the program and to check whether the programs functions according to the logic or not, also known as the Glass-box testing. The white box testing of the software that was developed was also successful. All the logical conditions and the queries that were used for the database programming were tested and they were found to be correct so there was no errors the errors that were discovered were dealt with during the time of the programming stage thus removing any logical and structural errors. This testing was almost repeated and done during the code development process where the codes, SQL that were written were continuously checked by compiling the program as well as by running the program so that there might not be errors during the program generation process.

The white box test is shown in the diagram as under:

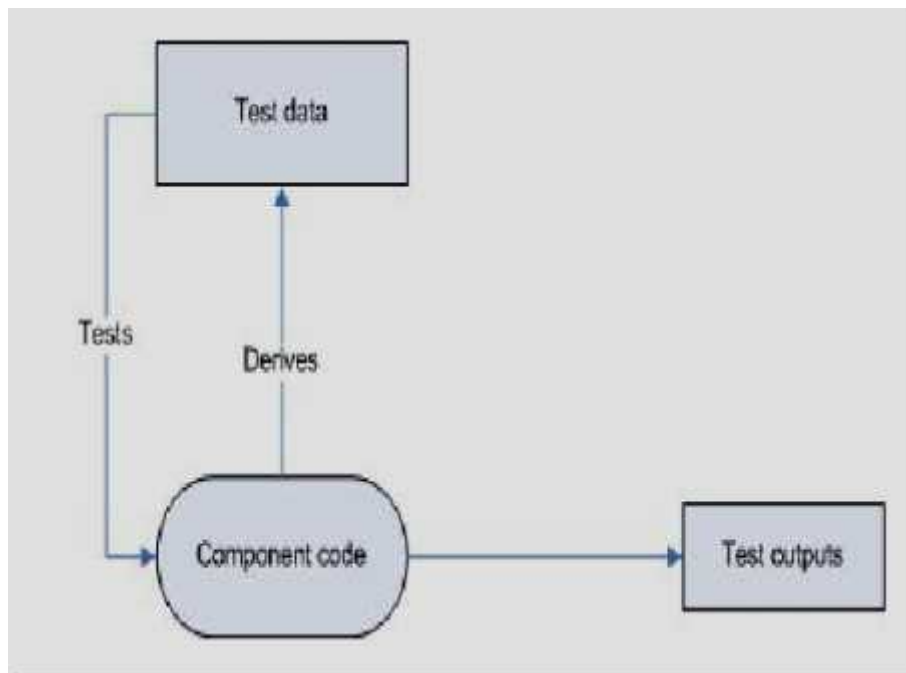


Fig 10: White box testing

Chapter 7

Diagrammatic Overview of the Project

7.1 System Diagram

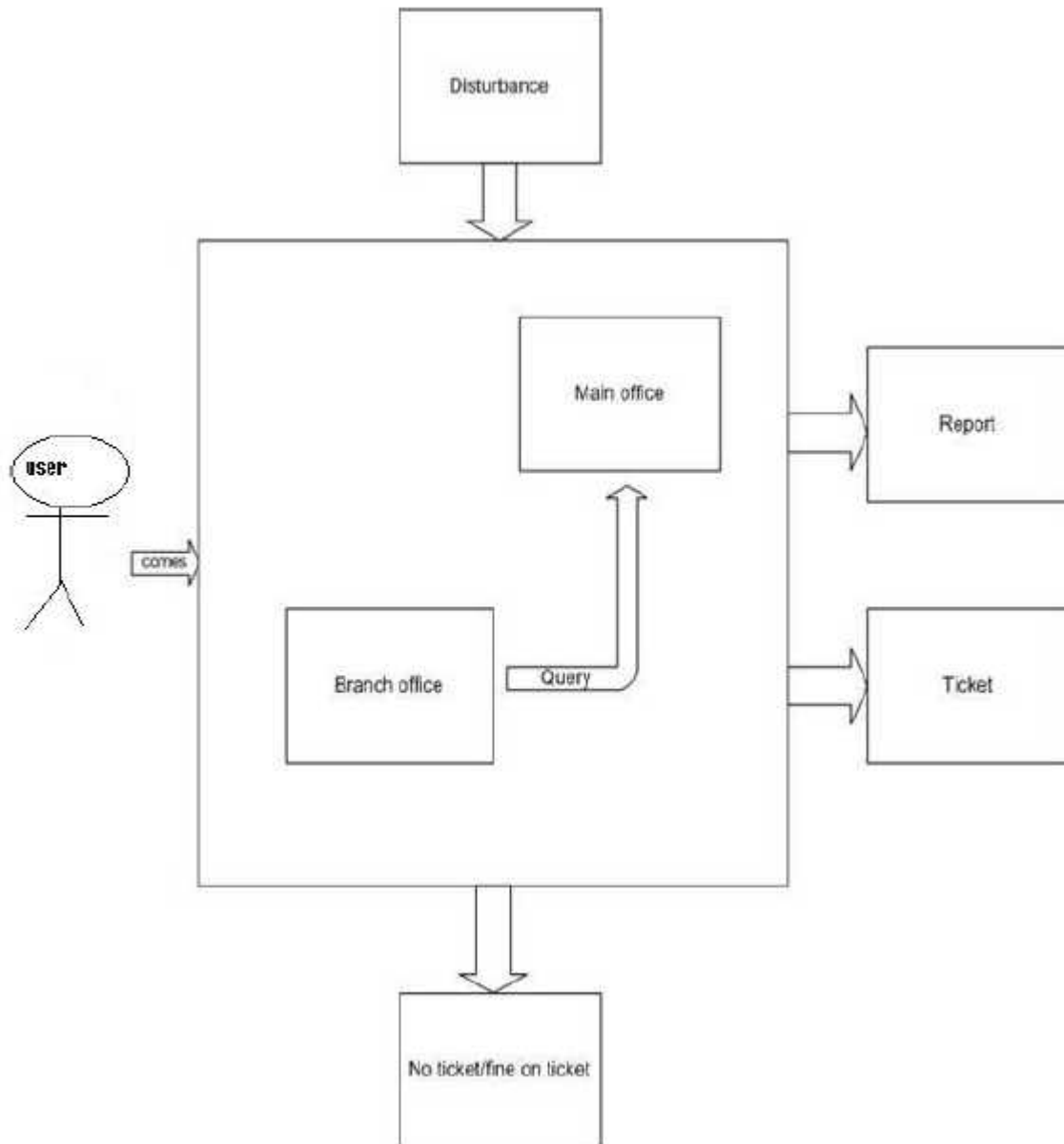


Fig 11: System Diagram

7.2 Use Case Diagram

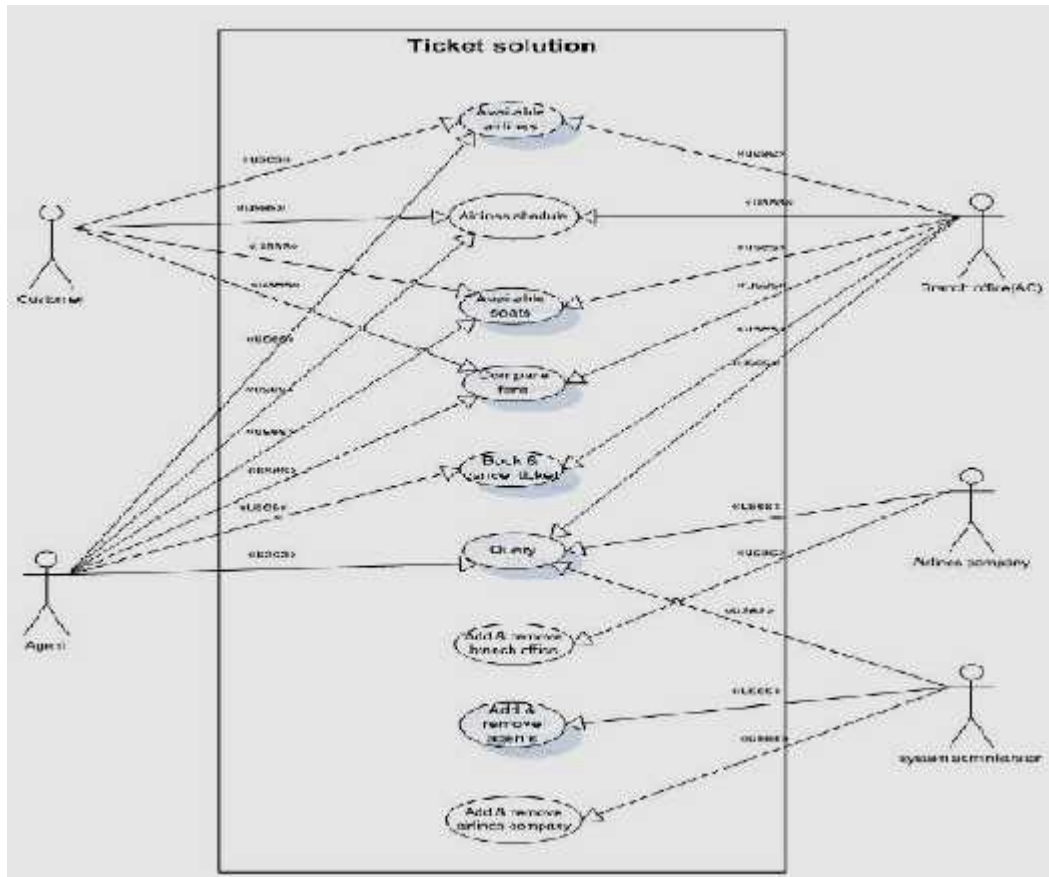


Fig 12: Use Case Diagram

7.3 Schema Diagram

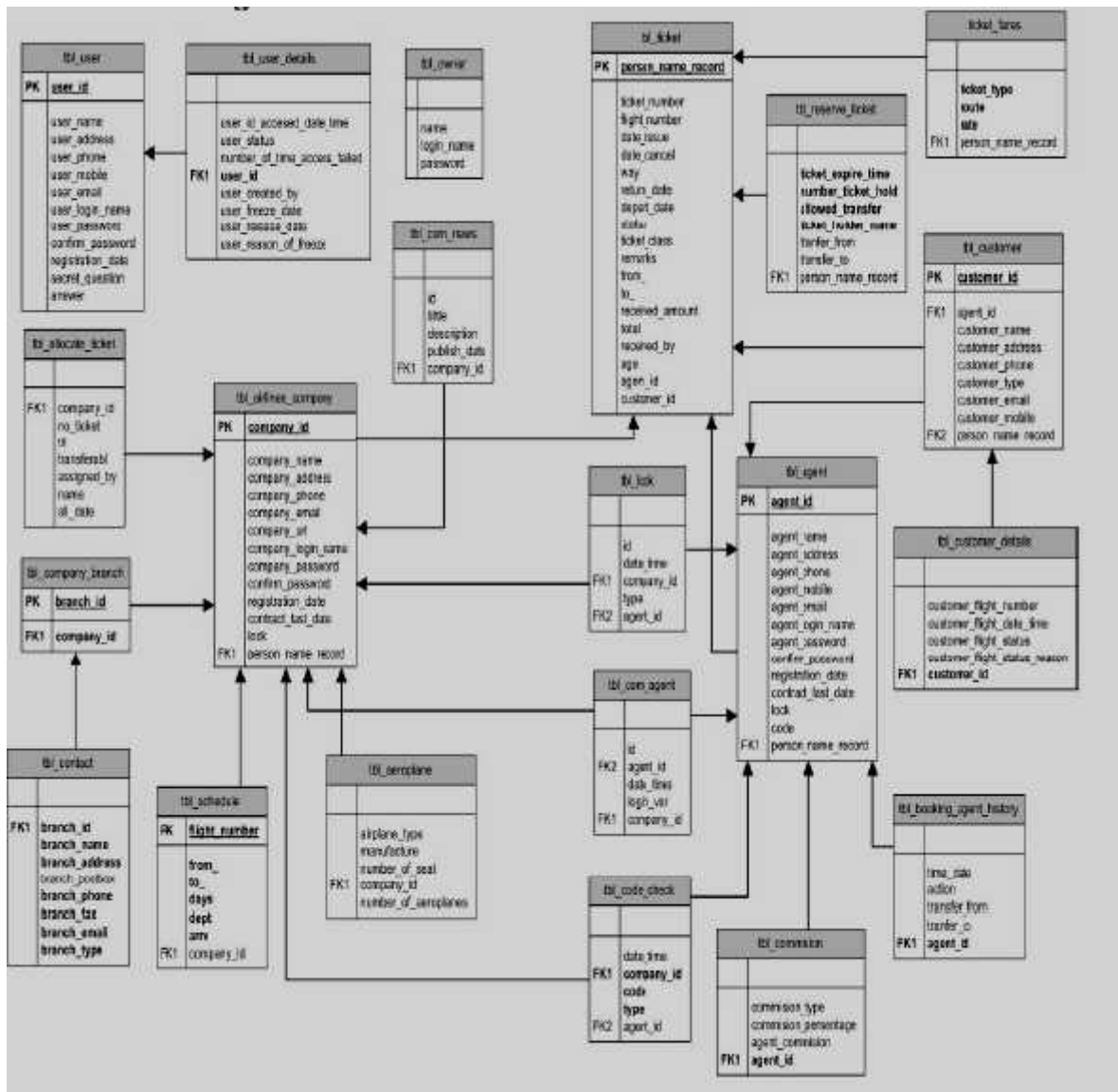


Fig 13: Schema Diagram

7.4 Context Level Diagram

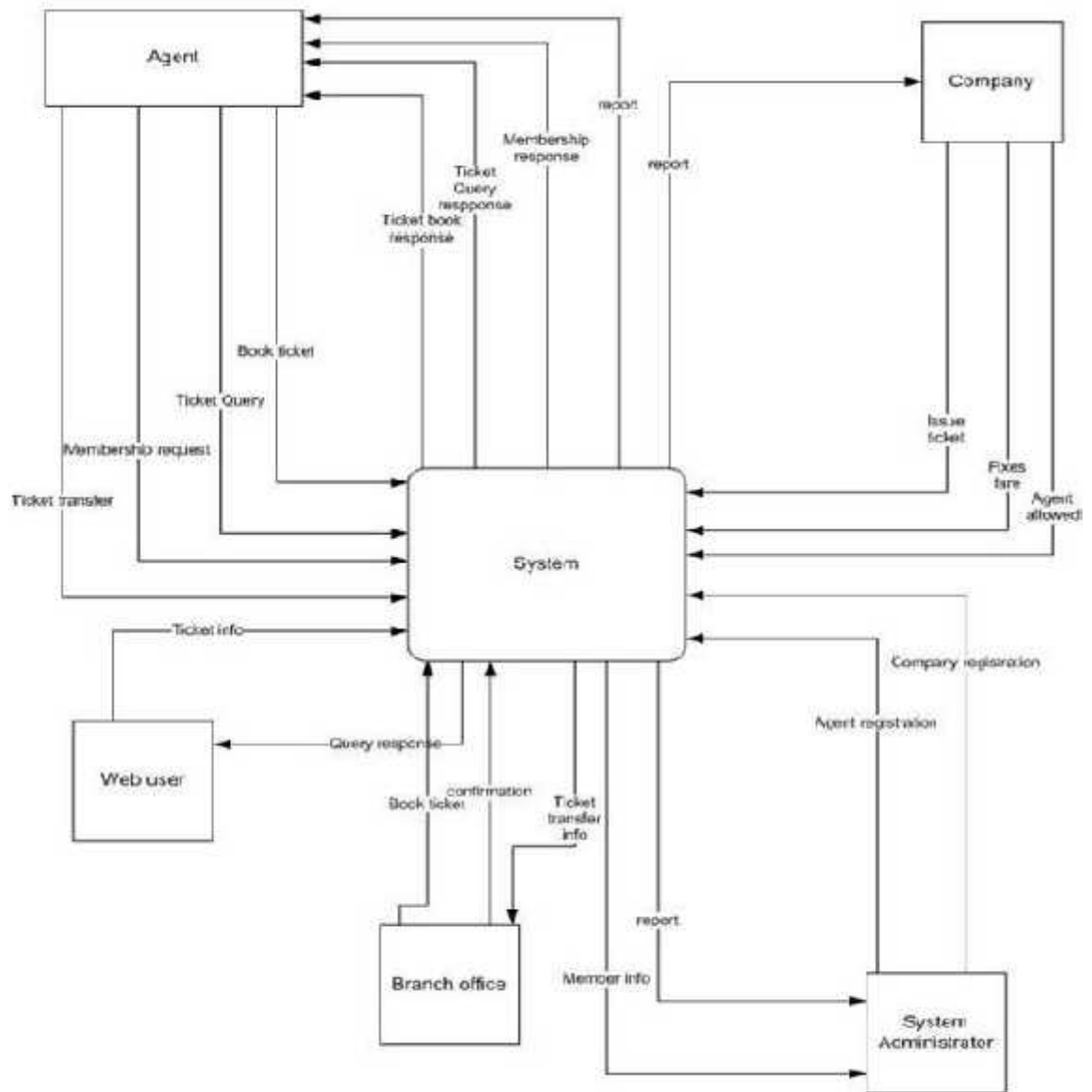


Fig 14:Context level Diagram

7.5 Dataflow Diagram

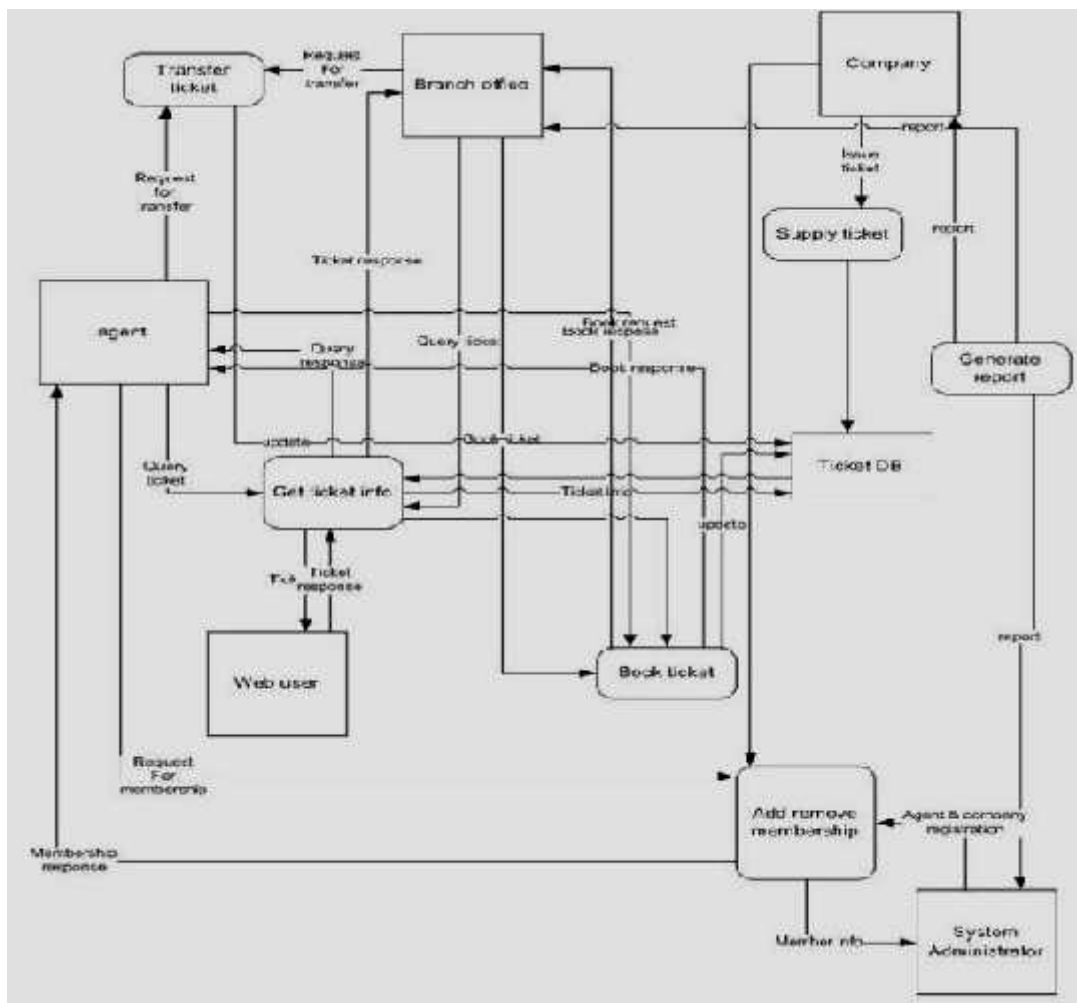


Fig 15: AeroWeb DFD

7.6 Class Diagram

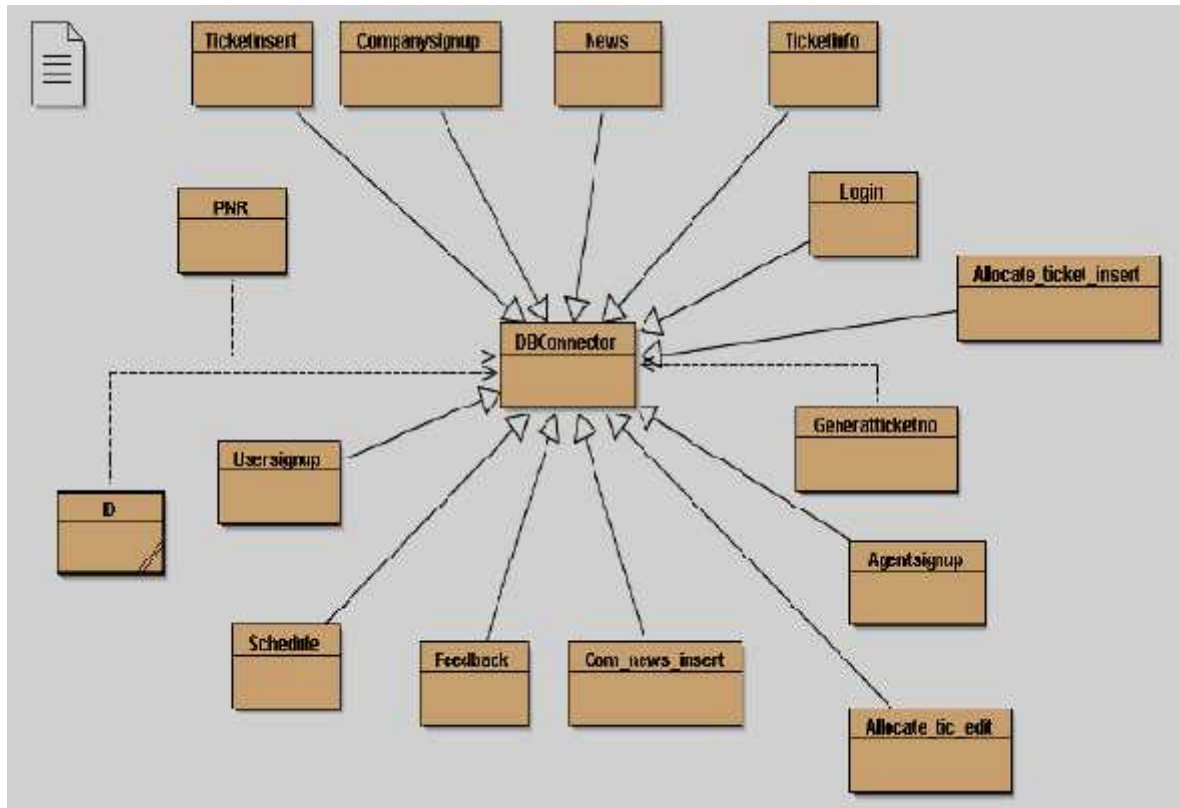


Fig 16: Class Diagram

Chapter 8

Conclusion and Future Recommendation

8.1 Introduction

This project work deals about the airline integration system. The software performs the semi automatic process to reserve the ticket in the aviation industry. Many different types of software tools have been used to develop the software. From this software all the airline lines are integrated in the single platform. Price comparison and route determination can be done qualitatively. Customer can prefer the selective airline company by comparing cost of the ticket and route needed for the customer.

8.2 Conclusion

The software that can be utilized commercially in the various airlines companies, their branches and their agents for the better performance, better competition and easier and reliable information supply to the information consumers. Hence the development of the system always starts with the establishment of the environment system that shows the operating system requirement and the system considered for the development and the present running system. This project has been developed with the complete approach of the Software Engineering from the start to the end. Different Object Oriented Programming tools have been incorporated during the accomplishment of this project, which eased the project development to greater extent.

For the project development process various software and platform are needed and that requirements are fulfilled through the use of the most reliable and the happening software in the worlds' market. The different technologies incorporated as requirement as well as the cutting edge technologies such as XML also have been incorporated. It can be executed in any operating system with enough memory.

8.3 Recommendation

The software prepared from this project is working in the Centralized database system in computer Network. This can further be extended in the Distributed database system in mobile computing device where the will use mobile device to query about the ticket booking, ticket cancellation, flight number, etc by doing SMS. The ticket reservation system can not only used in aviation industry but also used in hotel management industry as well as travel industry in Nepal.

References

1. Roger S. Pressman, “Software Engineering - A Practitioner’s Approach”, Fifth Edition, McGraw-Hill Higher Education 2001.
2. Jeffery L. Whitten, Lonnie D. Bentley, Kevin C. Dittaman, “System Analysis and Design”, Fifth Edition, Galgotia Publication pvt. ltd. 2002 .
3. Arnold, Goslina and Holms, “The Java Programming Language”, The Java Series.
4. John Zukowski, “Mastering Java2”, BPB Publications.
5. Larnie Pekowsky, “Java Server Pages”, Pearson Education Asia.
6. Cay S. Horstmann, Gary Cornell, “Core Java Volume I and II”, The Sun Microsystems Press.
7. Martin Gruber , “ Mastering SQL”, BPB Publications, New Delhi.
8. Ian Sommerville, “Software Engineering”, sixth Edition , Pearson Education .
9. Korth, Silberschatz and Sudharshan, “Database System Concepts”, Third Edition, McGraw Hill .
10. Bernard Van Haecke, “Java DataBase Connectivity 3”, Wiley – Dreamtech .
11. Kal Ahmead, Sudhir Ancha, Jay Coususs, Jerney Crostie, John Davies, Kyle Gabhart, “Professional Java XML”, Wrox Publication.
12. <http://www.faa.gov/library/reports/medical/oamtechreports/2000s/media/0208.pdf>
13. <http://www.hf.faa.gov//docs/freeflt.doc>
14. <http://www.alpa.org/>
15. <http://www.atsb.gov.au/admin/redirect.aspx>
16. <http://frwebgate.access.gpo.gov>
17. http://www.flightsafety.org/amb_home.html

Annex A



Fig 17: Homepage

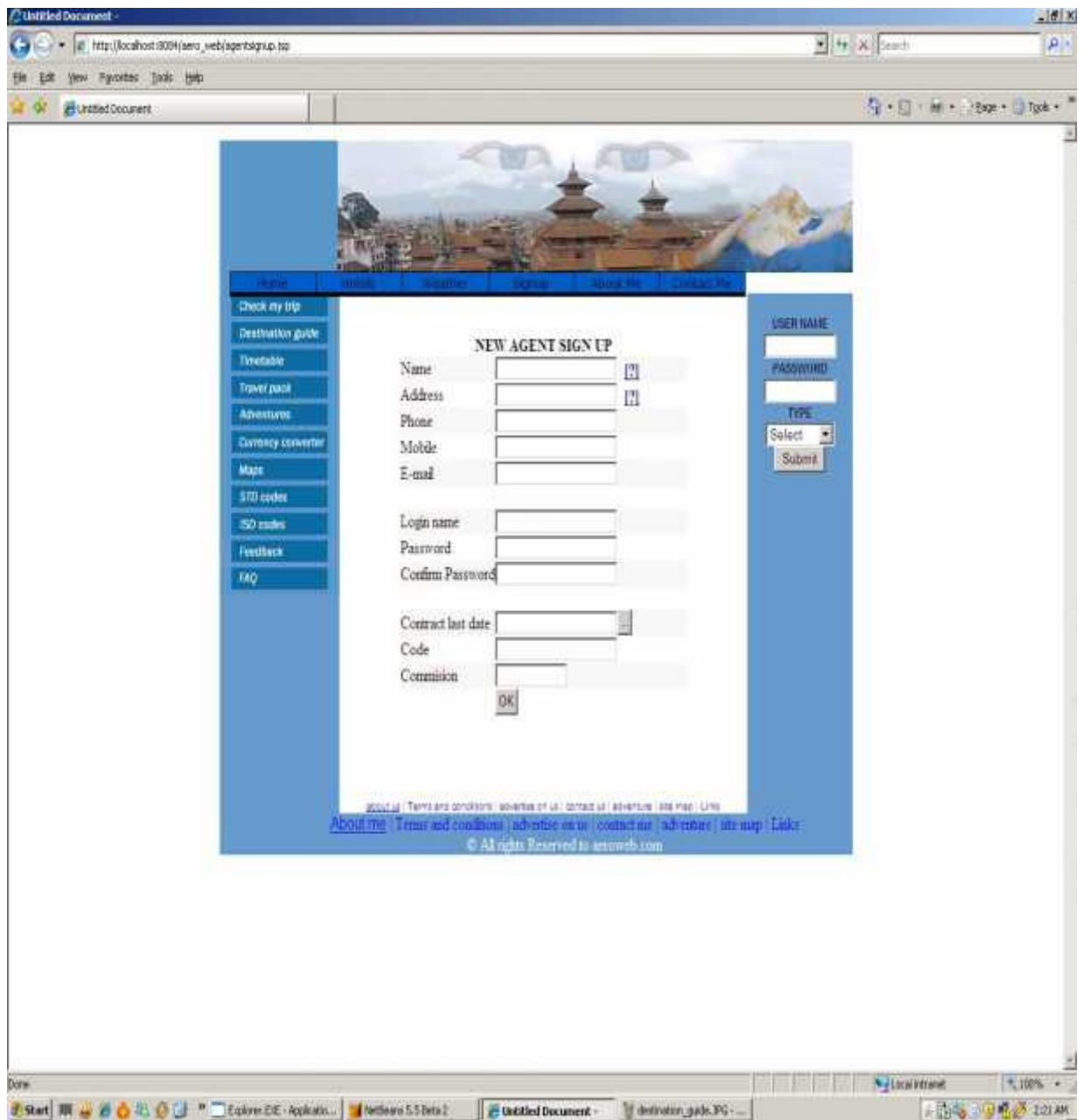


Fig 18: New account sign-up Interface

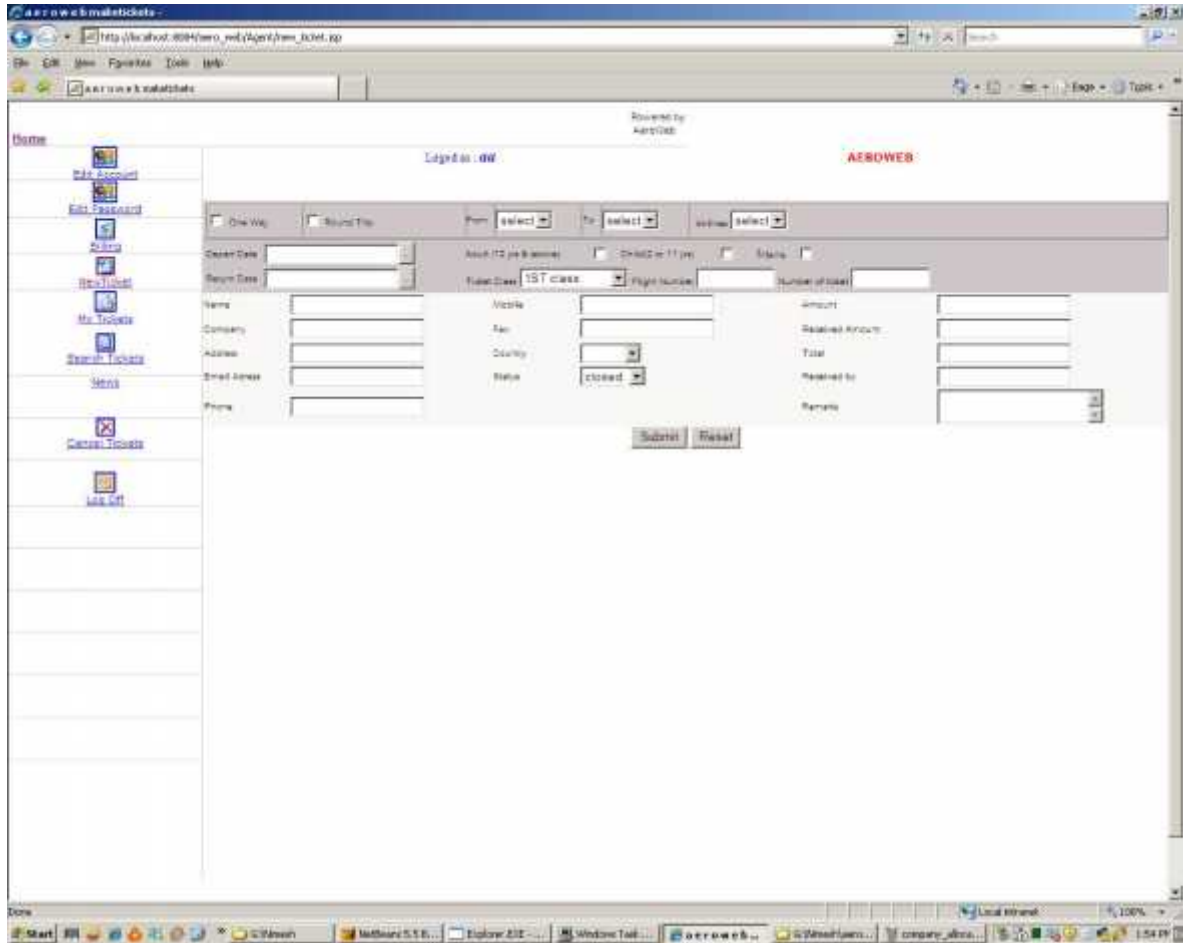


Fig 19: Ticket booking Section

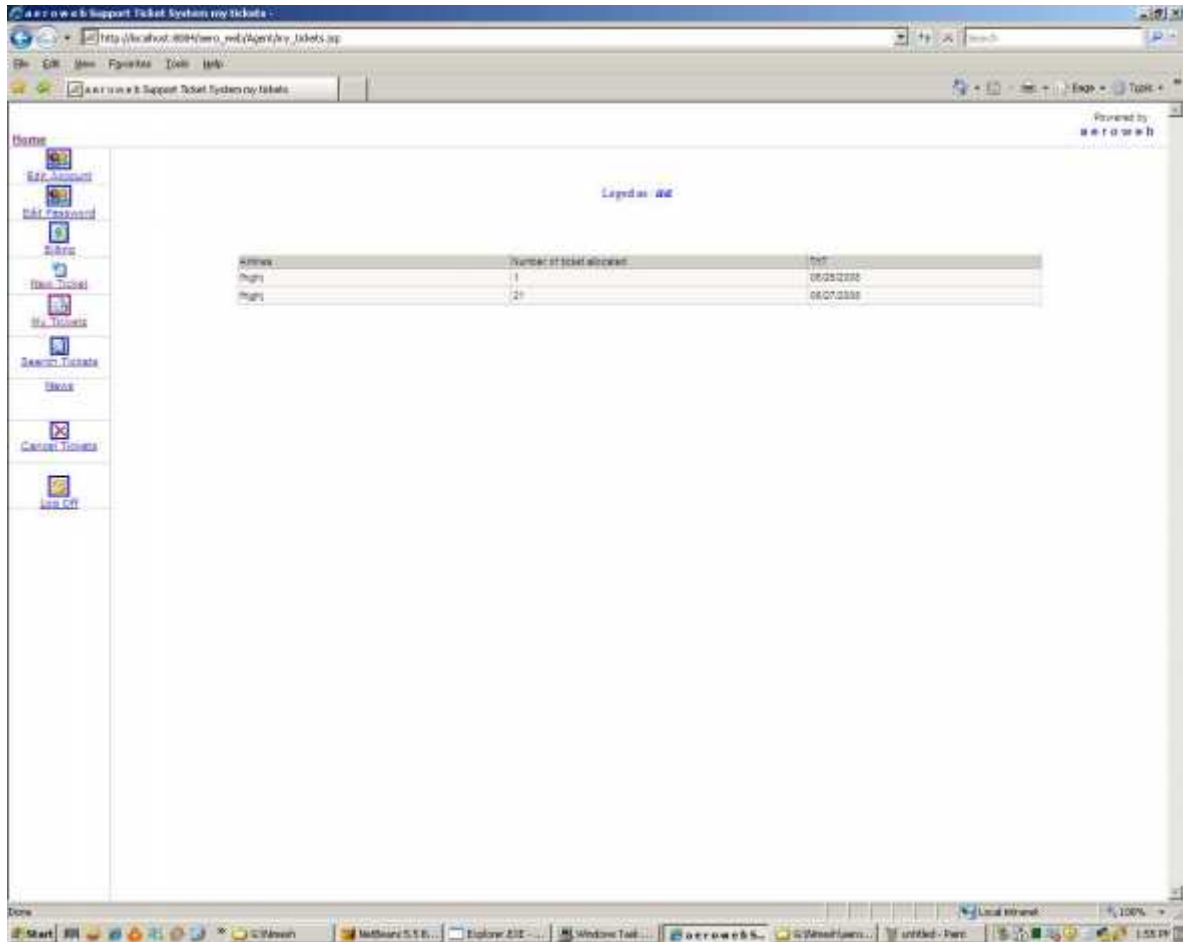


Fig 20: Active Agent Display Page

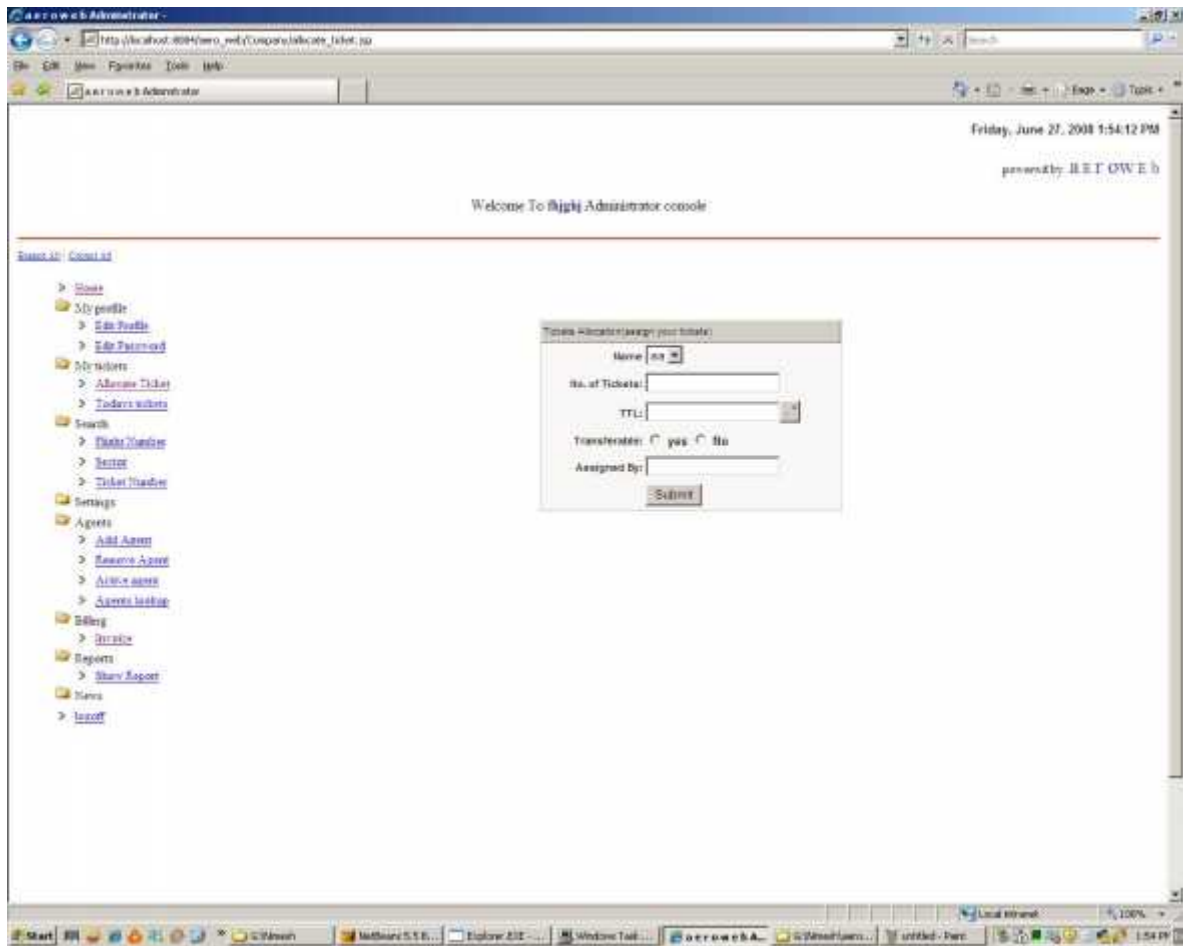


Fig 21: Company Search Page

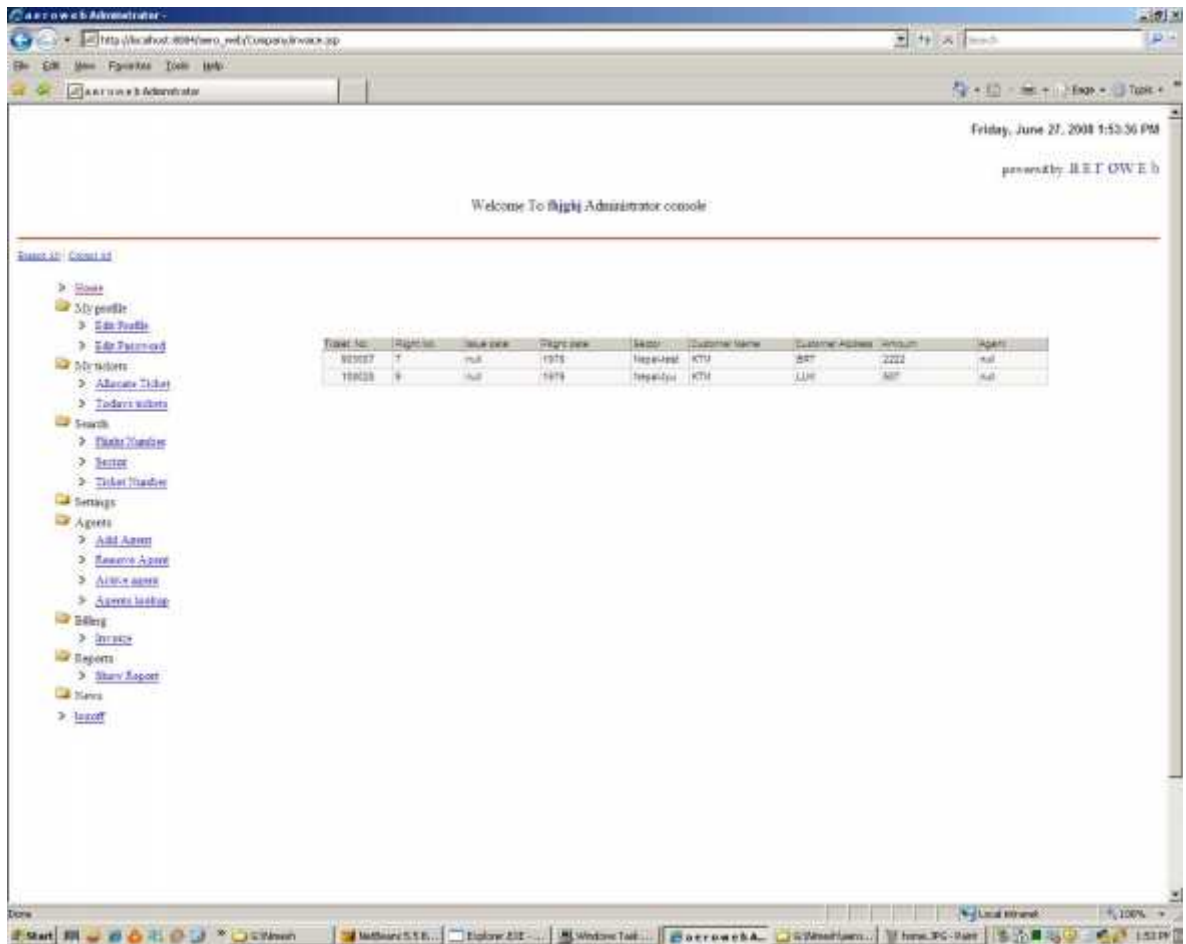


Fig22: Administrative Console

Annex B

Code to validate ConnectionString with aeroweb database:

```
<%@ page import ="java.sql.*" %>
<%@ page import ="java.sql.ResultSet.*" %>
<%@ page import ="java.io.*"%>

<%! Connection con; Statement st; ResultSet r; %>
<%!
    public void jspInit()
        {
            try
            {
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                con=DriverManager.getConnection("jdbc:odbc:aeroweb","","");
            }
            catch(Exception ex)
            {
                System.out.println(ex+" caught in connect method");
            }
        }
%>
```


User Administration with session variable

```
<%  
String cid=(String) session.getValue("Username");  
if(cid != null)  
{  
    %>  
    {  
        alert("ur session has been started:")  
    }  
Else  
    {  
        %>  
        <script language="javascript">  
            alert("ur session has been expired");  
            location.href="support_login.jsp"  
        <%  
            }  
        %>
```

New ticket booking, cancellation and other function used in Aero Web

```
<% @ page import = "java.sql.*" %>
<% @ page import = "java.sql.ResultSet.*" %>
<% @ page import = "java.io.*"%>
<%! Connection con; Statement st;ResultSet r;String acr=""; String cnn="";String n; %>
<%! public void jspInit()
    {
        try{
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con=DriverManager.getConnection("jdbc:odbc:aeroweb", "", "");
        }
        catch(Exception ex){
            System.out.println(ex+" caught in connect method");
        }
    }
%>

<script type="text/javascript">

var dropboxleft=700 //set left position of box (in px)
var dropboxtop=100 //set top position of box (in px)
var dropspeed=10//set speed of drop animation (larger=faster)
var displaymode="always"
if (parseInt(displaymode)!=NaN)
var random_num=Math.floor(Math.random()*displaymode)
var ie=document.all
var dom=document.getElementById
function truebody()
{
return (document.compatMode == "CSS1Compat" || document.compatMode!="BackCompat"?
document.documentElement : document.body
```

```

}
function get_cookie(Name)
{
var search = Name + "="
var returnvalue = ""
if (document.cookie.length > 0) {
offset = document.cookie.indexOf(search)
if (offset != -1) {
offset += search.length
end = document.cookie.indexOf(";", offset)
if (end == -1)
end = document.cookie.length;
returnvalue=unescape(document.cookie.substring(offset, end))
}
}
return returnvalue;
}
if(displaymode=="oncepersession"    &&    get_cookie("droppedinv2")==""           ||
displaymode=="always" || parseInt(displaymode)!=NaN && random_num==0){
if (window.addEventListener)
window.addEventListener("load", initboxv2, false)
else if (window.attachEvent)
window.attachEvent("onload", initboxv2)
else if (document.getElementById || document.all)
window.onload=initboxv2
if (displaymode=="oncepersession")
document.cookie="droppedinv2=yes"
}

```