AIRLINE RESERVATION SYSTEM

BY

KAAVYA KUPPA

Bachelor of Engineering, Jawaharlal Nehru Technological University, India, 2005

A REPORT

submitted in partial fulfillment of the requirements for the degree of

MASTER OF SOFTWARE ENGINEERING

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2008

Approved by:

[Major Professor]
Dr. Daniel Andresen
ABSTRACT

The objective of the project is to design an Airline Reservation System application which enables the customers to search and book flights, packages and hotels. The project has been designed in C#.NET technology and consists of a SQL server which acts as the database for the project.

My motivation for the project came from my enthusiasm and strong urge to learn C# and .NET which is one of the fastest growing technologies in today’s world. The Airline Reservation System project mainly consists of two types of users. The customers who access the information provided by the website and the administrator who modifies and updates the information available in the website. All the data needed for the application is stored in the form of tables in the SQL server 2000.

The report contains the details of all the tasks carried out during the entire software development life cycle of the Airline Reservation Project. This document depicts all the details of the project starting from the project design to testing.
# Table of Contents

List of Figures ................................................................................................................................. 9
List of Tables ...................................................................................................................................... 11
List of Tables ...................................................................................................................................... 11
Acknowledgements.......................................................................................................................... 12
Dedication.......................................................................................................................................... 13

CHAPTER 1 - VISION DOCUMENT ................................................................................................. 14
  1. INTRODUCTION ................................................................................................................ 14
  1.1 PURPOSE AND MOTIVATION ....................................................................................... 14
  1.2 PROJECT OVERVIEW ..................................................................................................... 14
  1.3 REFERENCES .................................................................................................................. 14

2. OVERALL PRODUCT DESCRIPTION ....................................................................................... 15
  2.1 PRODUCT PERSPECTIVE ............................................................................................... 15
  2.2 PRODUCT FEATURES ..................................................................................................... 15
    2.2.1 Customer / End user activities ..................................................................................... 15
    2.2.2 Administrator Activities ............................................................................................... 17
  2.3 USER CHARACTERISTICS ............................................................................................. 18
  2.4 CONSTRAINTS ................................................................................................................. 18
  2.5 ASSUMPTIONS AND DEPENDENCIES ........................................................................ 19

3. SPECIFIC REQUIREMENTS ................................................................................................. 19
  3.1 EXTERNAL INTERFACES .............................................................................................. 19
    3.1.1 USER INTERFACE .................................................................................................... 19
    3.1.2 SOFTWARE INTERFACE ......................................................................................... 20
  3.1.3 SOFTWARE INTERFACE ............................................................................................ 20

4. FUNCTIONAL REQUIREMENTS ....................................................................................... 20
  4.1 Use Case name: User Registration ...................................................................................... 21
  4.2 Use Case name: User Login ............................................................................................... 21
  4.3 Use Case name: Contact the company ................................................................................ 21
  4.4 User Case name: Booking Instructions ............................................................................... 22
  4.5 Use Case name: Book Flights ............................................................................................ 22
4.6 Use Case name: Book Motel ........................................................................................................ 22
4.7 Use Case name: Booking Packages .......................................................................................... 23
4.8 Use Case name: Login/Logout .................................................................................................. 23
4.9 Use Case name: Add/Delete or Modify Customer information .................................................. 23
4.10 Use Case name: Add/Delete or Modify flight information ....................................................... 24
4.11 Use Case name: Cancellation of Reservations ......................................................................... 24
4.12 Use Case name: E-mail confirmations ..................................................................................... 24
4.13 Use Case name: Modifying details of webpage ...................................................................... 25
4.14 Use Case name: Add/Delete or Modify Motel information .................................................... 25
4.15 Use Case name: Add/Delete or Modify package information .................................................. 25
5. PERFORMANCE REQUIREMENTS ............................................................................................... 26

CHAPTER 2 - PROJECT PLAN .................................................................................................... 27
1. TASK BREAKDOWN ................................................................................................................ 27
1.1 INCEPTION PHASE ............................................................................................................. 27
1.2 ELABORATION PHASE ......................................................................................................... 27
1.3 PRODUCTION PHASE .......................................................................................................... 28
2. COST ESTIMATE ..................................................................................................................... 29
2.1 COCOMO MODEL .................................................................................................................. 29
2.2 GANTT CHART ....................................................................................................................... 32
3. ARCHITECTURE ELABORATION PLAN ............................................................................... 32
3.1 REVISE VISION DOCUMENT ............................................................................................. 32
3.2 REVISE PROJECT PLAN ....................................................................................................... 33
3.3 ARCHITECTURE DESIGN .................................................................................................... 33
3.4 DEVELOPING PROTOTYPE ................................................................................................. 33
3.5 TEST PLAN ............................................................................................................................ 33
3.6 FORMAL TECHNICAL INSPECTORS .................................................................................. 33
3.7 FORMAL REQUIREMENTS SPECIFICATION ........................................................................ 33

CHAPTER 3 - ARCHITECTURE DESIGN .................................................................................. 34
1. INTRODUCTION ..................................................................................................................... 34
2. ARCHITECTURE OF THE AIRLINE RESERVATION SYSTEM ............................................. 34
3. PRESENTATION TIER .............................................................................................................. 36
8. TOOLS, TECHNIQUES AND METHODOLOGIES .......................................................... 56
9. RECORDS COLLECTION, MAINTENANCE AND RETENTION ................................. 56
10. DELIVERABLES ........................................................................................................... 56

CHAPTER 8 - TEST PLAN ................................................................................................. 58
1. TEST PLAN IDENTIFIER ................................................................................................. 58
2. INTRODUCTION ............................................................................................................. 58
2.1 OBJECTIVES ................................................................................................................. 58
2.2 REFERENCES ............................................................................................................... 58
2.3 DEFINITIONS ............................................................................................................... 59
3. TEST ITEMS .................................................................................................................... 59
4. APPROACH ..................................................................................................................... 59
4.1 UNIT TESTING .............................................................................................................. 60
4.2 INTEGRATION TESTING .............................................................................................. 60
4.3 REGRESSION TESTING ............................................................................................... 60
4.4 ACCEPTANCE TESTING ............................................................................................. 61
4.5 SYSTEM TESTING ...................................................................................................... 61
5. TEST CASES .................................................................................................................. 61
5.1 TEST CASE 1 – USER LOGIN .................................................................................... 61
5.2 TEST CASE 2 – USER REGISTRATION ................................................................. 61
5.3 TEST CASE 3 – USER REGISTRATION ..................................................................... 62
5.4 TEST CASE 4 – SEARCH AND BOOK FLIGHTS ................................................... 62
5.5 TEST CASE 5 – SEARCH AND BOOK PACKAGES .............................................. 62
5.6 TEST CASE 6 – SEARCH AND BOOK HOTELS .................................................... 63
6. PASS OR FAIL CRITERIA ............................................................................................... 63
7. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS .............................. 63
7.1 SUSPENSION CRITERIA ............................................................................................ 63
7.2 RESUMPTION REQUIREMENT ................................................................................. 64
8. TEST DELIVERABLES ................................................................................................... 64

CHAPTER 9 - ASSESSMENT EVALUATION ...................................................................... 65
1. INTRODUCTION ............................................................................................................ 65
2. TEST CASE RESULT SUMMARY .................................................................................. 65
3. DETAILED TEST RESULTS ........................................................................................................ 66

3.1 MANUAL TESTING .................................................................................................................. 66

3.1.1 TC # 1 – USER LOGIN ..................................................................................................... 66

3.1.2 TC # 2 - USER REGISTRATION ...................................................................................... 66

3.1.3 TC # 3 – SEARCH AND BOOK FLIGHTS ....................................................................... 67

3.1.4 TC # 4 – SEARCH AND BOOK PACKAGES ..................................................................... 68

3.1.5 TC # 5 - SEARCH AND BOOK HOTELS ......................................................................... 68

3.1.6 TC # 6 - ADMINISTRATOR SIGN IN .............................................................................. 69

3.1.7 TC # 7 – NEW FLIGHT/PACKAGE/HOTEL ADDITION ..................................................... 70

3.1.8 TC # 8 – UPDATING FLIGHT, PACKAGE, MOTEL DETAILS ........................................... 70

3.2 PERFORMANCE TESTING ..................................................................................................... 71

3.2.1 Home page of Airline Reservation System website ....................................................... 72

3.2.2 Login Page of the Airline Reservation System ............................................................... 74

3.2.3 Flight Search and booking page of the Airline Reservation System ......................... 75

4. OBSERVATIONS ...................................................................................................................... 76

CHAPTER 10 - USER MANUAL .................................................................................................... 81

1. INTRODUCTION ...................................................................................................................... 81

2. INSTALLATION AND SETUP ............................................................................................... 81

2.1 REQUIRED HARDWARE .................................................................................................... 81

2.2 REQUIRED SOFTWARE ..................................................................................................... 81

2.3 REQUIRED NETWORK CONFIGURATION .......................................................................... 82

2.4 SOFTWARE SET-UP ............................................................................................................. 82

3. AIRLINE RESERVATION SYSTEM USAGE FOR CUSTOMERS .......................................... 88

3.1 REGISTER FEATURE .......................................................................................................... 88

3.2 FLIGHT SEARCH AND BOOKING ...................................................................................... 90

3.3 HOTEL SEARCH AND BOOKING .................................................................................... 92

3.4 PACKAGE SEARCH AND BOOKING ................................................................................ 95

CHAPTER 11 - PROJECT EVALUATION ..................................................................................... 98

1. INTRODUCTION ...................................................................................................................... 98

2. PROBLEMS ENCOUNTERED ................................................................................................. 98

2.1 SOLUTION DOMAIN RESEARCH AND LEARNING .......................................................... 98
List of Figures

Figure 1 - Customer use case diagram .............................................................. 15
Figure 2 - administrator use case diagram ........................................................ 18
Figure 3 - gantt chart ........................................................................................ 32
Figure 4 - Three Tier Architecture .................................................................. 35
Figure 5 - page flow diagram for customer ...................................................... 38
Figure 6 - class diagram .................................................................................... 39
Figure 7 - sequence diagram for customer ...................................................... 39
Figure 8 - class diagram .................................................................................... 47
Figure 9 - user class ......................................................................................... 48
Figure 10 - administrator class ........................................................................ 48
Figure 11 - customer class ................................................................................ 49
Figure 12 - package class .................................................................................. 50
Figure 13 - flight class ...................................................................................... 50
Figure 14 - hotel class ....................................................................................... 51
Figure 15 - Performance and Maintenance in Control Panel............................. 83
Figure 16 - IIS ................................................................................................. 83
Figure 17 - Creating a New Virtual Directory .................................................. 84
Figure 18 - Virtual Directory Creating Wizard ................................................ 85
Figure 19 - Virtual Directory Alias .................................................................. 86
Figure 20 - Home Page of Airline Reservation System ..................................... 88
Figure 21 - Customer Registration Page ........................................................... 89
Figure 22 - Customer Home Page ..................................................................... 90
Figure 23 - FlightSearch Page ......................................................................... 91
Figure 24 - Flight Booking Confirmation Page ................................................. 92
Figure 25 - Hotel Search page ......................................................................... 93
Figure 26 - Hotel Booking Confirmation ........................................................... 94
Figure 27 - Package Search page ...................................................................... 95
Figure 28 - Package Confirmation page ............................................................ 96
Figure 29 - figure showing the task breakdown for each phase ................................................. 100
### List of Tables

Table 1 - adjustment factor table .................................................................................................. 30
Table 2- ASP.NET webforms and their purpose ........................................................................ 37
Table 3 - inspection checklist ..................................................................................................... 41
Table 4- user/customer pages testing summary......................................................................... 65
Table 5 - Administrator pages testing summary ....................................................................... 65
Table 6 - tc # 1 - user login......................................................................................................... 66
Table 7 - tc # 2 - user registration ........................................................................................... 66
Table 8 - tc # 3 search and book flights .................................................................................... 67
Table 9 - tc # 4 search and book packages .............................................................................. 68
Table 10 - tc # 5 search and book hotels .................................................................................. 68
Table 11 - tc # 6 Administrator sign in ..................................................................................... 69
Table 12 - new flight /hotel/.package addition ......................................................................... 70
Table 13 - updating flight, package, motel details..................................................................... 70
Table 14 - Project BreakDown and Duration .......................................................................... 100
Table 15 - : Activities and allotted time for all the Phases of MSE project ......................... 100
Acknowledgements

I would like to thank Dr. Daniel Andresen my major professor who has helped me and guided me through the entire life cycle of the project. I would also like to thank Dr. Torben Amtoft and Dr. Mitchell L. Neilsen for accepting to serve on my committee.

I would also like to specially thank my parents for their constant moral support and encouragement throughout the project. It is they who have helped me achieve all my dreams through their blessings throughout my life.
Dedication

I would like to dedicate the MSE project Airline Reservation System to my mother Mrs. A.Uma Devi and my father Mr.K.V.S.Prasad.
CHAPTER 1 - VISION DOCUMENT

1. INTRODUCTION

1.1 PURPOSE AND MOTIVATION

The main purpose of this vision document is to list the requirements of the Airline Reservation System project. This document also helps us to collect and analyze the ideas gathered for the project. This vision document will be subject to change, if more requirements are added to the project. This document is mainly prepared to set stage for the design phase of the project. The document being prepared is the first version of vision document for the Airline Reservation System project.

C#.NET is a new technology which is being used a lot in the IT field. My interest to learn this new technology has prompted me to take up this project, which would set the stage for the applications I would be developing in the future.

1.2 PROJECT OVERVIEW

The Airline Reservation System project is an implementation of a general Airline Ticketing website like Orbitz, which helps the customers to search the availability and prices of various airline tickets, along with the different packages available with the reservations. This project also covers various features like online registration of the users, modifying the details of the website by the management staff or administrator of the website, by adding, deleting or modifying the customer details, flights or packages information. In general, this website would be designed to perform like any other airline ticketing website available online.

1.3 REFERENCES

Some of the references used for preparing the vision document include:

2. [www.orbitz.com](http://www.orbitz.com)
3. IEEE document for Software Requirements Specifications
4. E-Draw software is used to generate the use case diagrams
2. OVERALL PRODUCT DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The Airline Reservation System project uses the .NET framework 1.1 and is completely independent. The project itself is a bigger product and does not need to be introduced into a larger system. The application would be running on a Windows XP/2000 Operating system.

2.2 PRODUCT FEATURES

The Airline Reservation System has the following features:

This project is mainly intended for two types of audiences. One is the customer or the end user and the other is the administrator of the website. Some of the major functions of the product can be categorized under two different categories that are for the administrator and the user.

2.2.1 Customer / End user activities

Figure 1 - Customer use case diagram
The above use case diagram depicts all the functions or activities that a user or a customer can perform on the application. They can be discussed in detail as follows:

**Home Page:** Like all the other airline websites available online, the user can access the user home page of the Airline Reservation System website, after he logs into the system. Here, he can look up information regarding flights, packages and motels.

**Login and Register:** The Airline Reservation System also comes with the customer registration details page, where the customer can enter his details and register. He can also create a username and password. Moreover, he will also be able to modify the registration information in case of a change in his e-mail address or any other information.

**Booking Flights:** The customer can also search for the flights available and reserve his place on the flight by purchasing a ticket.

**Book Motels:** Different Airline websites like Orbitz also offer various deals for booking Motels along with the airline tickets. So, the customer will also be able to view this functionality on the website.

**Book Packages:** This functionality is similar to the motel booking function, except the fact that the customer can look up various touring packages available at the person’s destination.

**Contact the Company:** The Customer can also call the company if he has any concerns or questions related to the bookings he has made online.

**Booking Instructions:** The website also provides instructions to the customers on how to book airline tickets or motels along with the different packages.

Some of the functions of the Airline Reservation System, such as creating, maintaining and updating the database are available only to the administrator. The functions of the administrator, explained in detail are as follows:
2.2.2 Administrator Activities

**Login/Logout:** The administrator has to login first in order to be able to make changes to the Airline Reservation System, by adding, deleting or modifying the data in the Airline Reservation System database. After making the necessary changes, he then has to logout of the system, in order to prevent misuse of the data.

**Add/Modify Customer Information:** Daily the Airline Reservation System will have many customers registering with the website and many of them unsubscribing. Only the administrator will have the sole rights to modify the database accordingly.

**Add/Modify Flight Information:** The Administrator also has the sole rights to add, delete or modify the flight information. Sometimes, flights get cancelled for some reason, so such flights would be removed from the list of flights available to the customer. Similarly whenever any flight information has to be modified or if any new flights need to be added to the database, these operations are performed by the administrator.

**Add/Modify Motel Information:** The administrator can also add/delete or modify information related to different motels. This information can be the number of rooms available at the motel, the prices etc.

**Cancellation of Reservations:** Sometimes, after making a reservation, a customer might cancel the reservation he has made. So, the administrator also handles such special situations and sends the customer an e-mail confirmation after deleting the specific transaction.

**E-mail confirmations:** Whenever a customer makes or cancels a reservation, the administrator is responsible for sending confirmation e-mails to the customer, confirming the transaction.
2.3 USER CHARACTERISTICS

There are two kinds of users for the Airline Reservation System. One is the customer and the other is the administrator. The customers do not need to have any prior training to use the application. However, instructions for making flight and motel reservations would be provided to them on the airline website. The administrators would however need to be trained in order to use the application.

2.4 CONSTRAINTS

Incase of changes made to the database, the application should be able to show the updated information on the website, without much delay. The database for the project is designed to be of moderate size. Currently, the application is designed to be able to run in Internet
Explorer. The Airline Reservation system will be designed in such a way that, it can be run on a Windows XP/2000 and IIS server. The .NET technology will be used to code the project and SQL server 2000 will act as the database for the project. The project will run on Internet Explorer and it should be installed on User’s system.

2.5 ASSUMPTIONS AND DEPENDENCIES

There are no assumptions as of now. To be updated in later versions of the vision document.

3. SPECIFIC REQUIREMENTS

3.1 EXTERNAL INTERFACES

The different types of interfaces that we would come across while developing the Airline Reservation System application are as follows:

- User Interface
- Hardware Interface
- Software Interface

3.1.1 USER INTERFACE

There are two types of users for the Airline Reservation System project. One is the customer and the other is the administrator. Both the customer and administrator user interface would be a graphical user interface. The graphical user interface for the customer home page would be as follows:

![Image of the customer home page with the following options: About us, AdminPage, How to book, Contact Us, Home, Images, Email ID, Password, L, New]
Figure representing the sample Customer GUI

The Graphical User Interface would mainly consist of Hyperlinks, Data entry fields like the E-mail Id field, push down buttons like the Login button etc.

The administrator of the website would also have a similar Graphical User Interface. After an administrator logs onto the system, the home page for the administrator would be as follows:

Figure representing the sample Administrator Interface

3.1.3 SOFTWARE INTERFACE

The application should run on a Windows XP/2000 Operating System. Since the application needs a database to store all the customer details, airline, motel and package information, SQL server 2000 would be used.

Visual Studio.NET 2003 would be used for creating the application. All the coding will be done in C#.

4. FUNCTIONAL REQUIREMENTS

The functional requirements of the Airline Reservation System are divided among the customer and the administrator of the application.

These functional requirements can be explained in detail as follows:
4.1 Use Case name: User Registration

- **Description:** This use case describes the scenario where the user registers with the application by providing all the necessary details, in order to make reservations or bookings for flights, motels, special packages.
- **Actor:** User or the Customer
- **Input:** The user or the customer will have to provide all the necessary details present in the customer registration form of the application.
- **Output:** All the details entered in the customer registration page will be verified and accepted by the system into the database.

4.2 Use Case name: User Login

- **Description:** This use case describes the scenario where the user logs into the application, with the username and password he has provided while registering with the system.
- **Actor:** User or the Customer
- **Input:** The user or the customer creates a username and password at the time of registering with the system. He then uses them to logon to the system and make reservations or view any information.
- **Output:** The application then verifies the authenticity of the username and password that the customer has provided and allows the user to view the information available on the system, if the username and password are valid.

4.3 Use Case name: Contact the company

- **Description:** This use case describes the scenario where the user contacts the company for any information.
- **Actor:** User or the Customer
- **Input:** The customer can contact the airline company, requesting them for any information he needs.
- **Output:** The application verifies the authenticity of the username and password that the customer has provided and allows the user to view the contact information for the company.
4.4 User Case name: Booking Instructions

- **Description:** This use case describes the scenario where the user views the instructions for booking flights, packages, or motels.
- **Actor:** User or the Customer
- **Input:** After the customer logs onto the application with his username and password, he can look up the instructions posted on the website for booking flights, packages or motels.
- **Output:** The application verifies the authenticity of the username and password and displays the how to book instructions page.

4.5 Use Case name: Book Flights

- **Description:** This use case describes the scenario where the user books airline tickets.
- **Actor:** User or the Customer
- **Input:** After logging into the application, the customer looks up the information related to various airlines and checks the availability of seats on flights. If he finds that there are any available tickets, he then purchases them.
- **Output:** The application verifies the authenticity of the username and password and then displays information related to various flights to the customer.

4.6 Use Case name: Book Motel

- **Description:** This use case describes the scenario where the user books motels at the time of airline ticket reservation
- **Actor:** Customer or the user
- **Input:** After logging onto the application, the customer looks up the information for all the available motels at his destination.
- **Output:** The application verifies the authenticity of the username and password and then displays information pertaining to various motels at the customer’s destination.
4.7 Use Case name: Booking Packages

- **Description:** This use case describes the scenario where the user books different touring packages at the airline ticket reservation.
- **Actor:** Customer or the user
- **Input:** The customer looks up information regarding various touring packages available at his destination at the time of airline ticket reservation.
- **Output:** The application verifies the authenticity of the username and password of the customer and then displays information of various touring packages available at customer’s choice of place.

The administrator activities use cases will be described here:

4.8 Use Case name: Login/Logout

- **Description:** This use case describes the scenario where the administrator of the application, logs into the system and logs out after the work is done.
- **Actor:** Administrator
- **Input:** The administrator of the website logs into the application with the username and password provided to him.
- **Output:** The application verifies the authenticity and displays the home page of the administrator.

4.9 Use Case name: Add/Delete or Modify Customer information

- **Description:** This use case describes the scenario where the administrator adds, deletes or modifies customer information in the system database
- **Actor:** Administrator
- **Input:** The administrator of the applications logs onto the system with his username and password.
- **Output:** The application authenticates the administrator, and then displays the page where the administrator can add new customers to the database, or delete existing customers or modify details of customers in the database.
4.10 Use Case name: Add/Delete or Modify flight information

- **Description:** This use case describes the scenario where the administrator adds, deletes or modifies flight information in the application database
- **Actor:** Administrator
- **Input:** The administrator logs onto the system with the username and password provided to him.
- **Output:** The application authenticates the administrator, by verifying the username and password. Then the application displays the page where the administrator can add new flights to the database, delete the flights that have been cancelled or modify information for the flights.

4.11 Use Case name: Cancellation of Reservations

- **Description:** This use case describes the scenario where the administrator handles the cancellation of reservations by the customers.
- **Actor:** Administrator
- **Input:** The administrator logs onto the system with the given username and password.
- **Output:** The application authenticates the administrator and then displays the page where the administrator looks up the id of the customer who has requested cancellation of reservation. After canceling the reservation, the administrator then sends a confirmation e-mail to the customer.

4.12 Use Case name: E-mail confirmations

- **Description:** This use case describes the scenario where the administrator sends e-mail confirmations to the customers of the application.
- **Actor:** Administrator
- **Input:** The administrator logs onto the application with the username and password provided.
- **Output:** The application then authenticates the administrator and displays the page where the administrator can send e-mail confirmations to the customer. These e-mail
confirmations may be sent in cases where the customer has cancelled a reservation or changed the personal information available on the website.

4.13 Use Case name: Modifying details of webpage

- **Description:** This use case describes the scenario where the administrator logs onto the application to modify the details of the airline website.
- **Actor:** Administrator
- **Input:** The administrator logs onto the application with the username and password provided to him.
- **Output:** After verifying the username and password of the administrator, the application then allows the administrator to login. The administrators can then browse through the website and change the details of any webpage in the Airline Reservation system application.

4.14 Use Case name: Add/Delete or Modify Motel information

- **Description:** This use case describes the scenario where the administrator adds, deletes or modifies motel information in the database.
- **Actor:** Administrator
- **Input:** The administrator logs onto the system with the username and password provided to him.
- **Output:** The application authenticates the administrator, by verifying the username and password. Then the application displays the page where the administrator can add new motels to the database, delete a specific motel from the list of motels, since there are no more available rooms there. He can even update the price per room of each motel. To make it easier, the administrator might assign a unique id for each hotel.

4.15 Use Case name: Add/Delete or Modify package information

- **Description:** This use case describes the scenario where the administrator adds, deletes or modifies package information in the application database.
- **Actor:** Administrator
• **Input:** The administrator logs onto the system with the username and password provided to him.

• **Output:** The application authenticates the administrator, by verifying the username and password. Then the application displays the page where the administrator can add new packages to the database, delete the packages that are no longer available or modify information for any particular package.

### 5. PERFORMANCE REQUIREMENTS

The Airline Reservation System application should be able to respond to the queries submitted by the customer without much delay. When a user searches for a flight leaving from a particular place to another place, the application should not take much time to return the results, similarly for the motel and package information. Considering that the application is of moderate size, it should be able to display 10 results at a time on each page, when the customer looks up for any particular data. Since the Airline Reservation websites have much traffic, the user should also be able to logon to the system using high speed internet. Most of the requests sent to the application should be answered in less than 5 seconds.
CHAPTER 2 - PROJECT PLAN

1. TASK BREAKDOWN

1.1 INCEPTION PHASE

The Inception Phase is the first phase of a software development life cycle. The main objective of the inception phase is to establish the business case for the system and define the scope of the system. The inception phase of the Airline Reservation System project mainly focuses on defining the project requirements. The primary documents created in the inception phase consist of the Vision document, Software Quality Assurance Plan and the Project Plan.

The Vision document, which is one of the outcomes of the inception phase, mainly focuses on core project requirements and the key features. It also discusses the main features of the project along with the interfaces of the project. The next document which is an outcome of the inception phase is the project plan document. The project plan document as the name says is mainly used to document the schedule of the project as well as the time required for each phase of the project. This plan, gives us an estimate of when the project will be completed. The Software Quality Assurance Plan, which is also an outcome of the inception phase documents the standards and conventions that need to be followed in order to ensure good quality of the end product.

At the end of the inception phase, the developer will give a presentation to the supervisory committee after submitting all the documents necessary. This phase will be marked complete, once all the documentation for the phase I is reviewed and approved by the committee.

1.2 ELABORATION PHASE

The next phase in the software development lifecycle is the elaboration phase. The main purpose of the elaboration phase is to establish a strong architectural foundation for the Airline Reservation System project. It is the most critical phase of all the phases of the software development lifecycle. The entire architectural design of the Airline Reservation System will be documented using the appropriate UML diagrams. In this phase, revisions on the initial versions of the Vision document, Project Plan document will be made based on the suggestions made by the Supervisory Committee members. In the elaboration phase, each component in the Airline Reservation System architecture will be documented at the interface level. Another deliverable
of this phase is the Test Plan which documents all the testing activities that will be carried out on the project and also states how to report and track the test results. The two technical inspectors of the project also perform an architectural review on the project and provide feedback by submitting the formal technical inspection letters which are based upon their findings.

At the end of this phase, the developer will demonstrate an executable prototype of the project and submit all the documentation required for the phase II of the project. The deliverables of this phase can be stated as follows:

- Vision Document 2.0
- Project Plan 2.0
- Formal Requirement Specification
- Architecture Design
- Test Plan
- Formal Technical Inspection – submitted by two individual MSE students
- Executable Architecture Prototype

Once all the above documents and the prototype is reviewed and accepted by the Supervisory Committee, the Elaboration phase is said to be complete.

1.3 PRODUCTION PHASE

The production phase of the Airline Reservation System project mainly focuses on the implementation and testing of the project. In this phase, the entire working code for the project will be constructed and all the documentation pertaining to the project is completed. In this phase, the entire code for the Airline Reservation System project will be tested to ensure that it has met all the requirements. All the test results will also be analyzed and documented. A user manual is also produced for the project which describes how to install, run and use the tool efficiently.

At the end of the production phase, the developer of the project will give a presentation to the supervisory committee and will also demonstrate the entire working of the software product. This presentation is the last presentation, and this phase of the project will be completed once the committee members have reviewed and approved all the documentation and the working code of
the Airline Reservation System project. Once this phase is said to complete, then the MSE project is also said to be complete.

2. COST ESTIMATE

2.1 COCOMO MODEL

The Constructive Cost Model known as the COCOMO Model, has been designed in 1981 by Barry Boehm, to give an estimate of number of man months it will take to develop a software product. The model also estimates the development schedule for the project in months and gives us a schedule distribution for all the major phases of a project.

The COCOMO models are developed for three classes of software projects. They are as follows:

- **Organic Projects** - These are relatively small and simple software projects in which small teams with good application experience work towards a set of less than rigid requirements.

- **Semi – Detached Projects** – These are intermediate size software projects in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

- **Embedded Projects** – These are software projects that must be developed within a set of tight hardware, software and operational constraints.

The Airline Reservation System project has an average complexity and fair flexibility. Thus, this project is classified as an organic project under the COCOMO model.

The equations as they are modified for the organic projects are as follows:

\[
\text{Effort} = 3.2 \times \text{EAF} \times (\text{Size})^{1.05}
\]

\[
\text{Time} = 2.5 \times (\text{Effort})^{0.38}
\]

where

- **Effort** = number of staff months (PM)
- **EAF** = effort adjustment factor
- **Size** = number of lines of code for completed product. It is measured in KLOC (thousands of lines of code)
- **Time** = total number of months
The Effort Adjustment Factor mentioned above is the product of 15 adjustment parameters. Each adjustment parameter is categorized as very low, low, nominal, high or very high. Each adjustment factor along with the range of values it lies within is shown in the following table:

**Table 1 - adjustment factor table**

<table>
<thead>
<tr>
<th>IDENTIFIER</th>
<th>EFFORT ADJUSTMENT FACTOR</th>
<th>RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RELY</td>
<td>Required Reliability</td>
<td>0.75 – 1.40</td>
</tr>
<tr>
<td>DATA</td>
<td>Database Size</td>
<td>0.94 – 1.16</td>
</tr>
<tr>
<td>CPLX</td>
<td>Product Complexity</td>
<td>0.70 – 1.65</td>
</tr>
<tr>
<td>TIME</td>
<td>Execution Time Constraint</td>
<td>1.00 – 1.66</td>
</tr>
<tr>
<td>STOR</td>
<td>Main Storage Constraint</td>
<td>1.00 – 1.56</td>
</tr>
<tr>
<td>VIRT</td>
<td>Virtual Machine Volatility</td>
<td>0.87 – 1.30</td>
</tr>
<tr>
<td>TURN</td>
<td>Computer Turnaround Time</td>
<td>0.87 – 1.15</td>
</tr>
<tr>
<td>ACAP</td>
<td>Analyst Capability</td>
<td>1.46 – 0.71</td>
</tr>
<tr>
<td>AEXP</td>
<td>Applications Experience</td>
<td>1.29 – 0.82</td>
</tr>
<tr>
<td>PCAP</td>
<td>Programmer Capability</td>
<td>1.42 – 0.70</td>
</tr>
<tr>
<td>VEXP</td>
<td>Virtual Machine Experience</td>
<td>1.21 – 0.90</td>
</tr>
<tr>
<td>LEXP</td>
<td>Language Experience</td>
<td>1.14 – 0.95</td>
</tr>
<tr>
<td>MODP</td>
<td>Use of Modern Practices</td>
<td>1.24 – 0.82</td>
</tr>
<tr>
<td>TOOL</td>
<td>Use of Software Tools</td>
<td>1.24 – 0.83</td>
</tr>
<tr>
<td>SCED</td>
<td>Required Development Schedule</td>
<td>1.23 – 1.10</td>
</tr>
</tbody>
</table>

The adjustment factors for the Airline Reservation System are as follows:

- RELY as nominal with a value of 1.00
- DATA as nominal with a value of 1.00
- CPLX as low with a value of 0.75
- TIME as nominal with a value of 1.00
• STOR as low with a value of 1.00
• VIRT as nominal with a value of 1.03
• TURN as low with a value of 0.88
• ACAP as high with a value of 0.9
• AEXP as nominal with a value of 0.9
• PCAP as nominal with a value of 0.9
• VEXP as nominal with a value of 1.00
• LEXP as nominal with a value of 1.00
• TOOL as high with a value of 0.9
• SCED as nominal with a value of 1.00

Being a beginner to the C#.NET technology, my experience with the technology is not very high, but only nominal, so I have assigned a value of 0.9 for the applications experience. Since I will be implementing the project while learning C#.NET,

With the help of above stated values, the EAF for the Airline Reservation System project is calculated as EAF = 0.45. Also by estimating the size of the project we have the value 3.0.

Since we already have the formulae for Effort and Time, we can calculate the values as follows:

\[
\text{Effort} = 3.2 \times 0.45 \times 3.0^{1.05} = 4.56 \text{ staff months}
\]
\[
\text{Time} = 2.5 \times 4.56^{0.38} = 4.44 \text{ months (development time)}
\]
2.2 GANTT CHART

The Gantt chart for the Airline Reservation System project can be depicted as follows:

Figure 3- gantt chart

<table>
<thead>
<tr>
<th>Number</th>
<th>Task</th>
<th>Start</th>
<th>End</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Phase I</td>
<td>5/1/2008</td>
<td>6/11/2008</td>
<td>29</td>
</tr>
<tr>
<td>3</td>
<td>Project Plan</td>
<td>5/19/2008</td>
<td>5/30/2008</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>SQL Plan</td>
<td>6/1/2008</td>
<td>6/6/2008</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Prototype I</td>
<td>6/5/2008</td>
<td>6/6/2008</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Phase II</td>
<td>6/11/2008</td>
<td>7/11/2008</td>
<td>22</td>
</tr>
<tr>
<td>8</td>
<td>Action Items</td>
<td>6/11/2008</td>
<td>6/12/2008</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>Project Plan revision</td>
<td>6/13/2008</td>
<td>6/24/2008</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>Architectural Design</td>
<td>6/19/2008</td>
<td>6/21/2008</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>Formal Technical Inspections</td>
<td>6/25/2008</td>
<td>6/26/2008</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>Prototype III</td>
<td>6/15/2008</td>
<td>7/9/2008</td>
<td>17</td>
</tr>
<tr>
<td>16</td>
<td>Presentation II</td>
<td>7/8/2008</td>
<td>7/10/2008</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>Phase III</td>
<td>7/10/2008</td>
<td>7/30/2008</td>
<td>14</td>
</tr>
<tr>
<td>18</td>
<td>Action Items</td>
<td>7/10/2008</td>
<td>7/11/2008</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>Testing</td>
<td>7/21/2008</td>
<td>7/25/2008</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>Initial Documentation</td>
<td>7/25/2008</td>
<td>7/28/2008</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>Presentation III</td>
<td>7/29/2008</td>
<td>7/30/2008</td>
<td>1</td>
</tr>
</tbody>
</table>

3. ARCHITECTURE ELABORATION PLAN

The following tasks have to be completed by the end of Phase II of the Airline Reservation System project.

3.1 REVISE VISION DOCUMENT

The vision document which is produced at the end of Phase I will be revised in order to make sure that all the requirements for the project have been represented. All the requirements will be ranked in order of their importance. At the end of phase I of the project, the supervisory
committee members will provide suggestions to the developer regarding improving the vision document. All the changes mentioned by the committee will be made in the revised document and the correct document will be submitted at the end of Phase II for approval.

**3.2 REVISE PROJECT PLAN**

Similar to the vision document, the project plan document, submitted at the end of Phase I, will also be reviewed by the supervisory committee and the updated document will be submitted by the developer at the end of Phase II. This updated document will provide us with an updated estimate of cost, size and effort required for the project. Finally, this document will be submitted to the major professor for approval.

**3.3 ARCHITECTURE DESIGN**

The complete project design of the Airline Reservation System project, will be documented using the UML diagrams.

**3.4 DEVELOPING PROTOTYPE**

The architecture executable prototype of the Airline Reservation System project will be built to illustrate that all the requirements stated in the project’s vision document have been met.

**3.5 TEST PLAN**

A test plan will be developed at the end of the Phase II, which documents all the testing activities that will be performed on the Airline Reservation system project, to ensure that the project has satisfied all the requirements that have been mentioned in the vision document. This test plan will also include evaluation criteria for all the critical use cases of the project.

**3.6 FORMAL TECHNICAL INSPECTORS**

The architecture design of the Airline Reservation system project will be inspected by two other MSE students, Sandhya Bathini and Srunokshi Neelakantan.

**3.7 FORMAL REQUIREMENTS SPECIFICATION**

The Object Constraint Language will be used to define and verify the formal specification of the product.
CHAPTER 3 - ARCHITECTURE DESIGN

1. INTRODUCTION

The main purpose of the Architecture Design document is to discuss the architectural design for the Online Reservation System project in a clear and concise form. This design document will give a detailed description of the presentation tier, the middle tier which consists of the class diagrams, sequence diagrams for the Airline Reservation System and finally the data tier. The Unified Modeling Language (UML) is a standardized visual specification language for object modeling. Thus, the class diagrams and the sequence diagrams depicted in the Architecture design document will be developed according to the UML standard notation.

2. ARCHITECTURE OF THE AIRLINE RESERVATION SYSTEM

The architecture of the Airline Reservation System is based on the three-tier architecture. This three-tier architecture mainly consists of three layers namely:

- Presentation Tier
- Business Tier
- Data Access Tier

The Presentation Tier converts and displays information into a human legible form. This tier displays information related to services such as browsing the Airline website, purchasing tickets etc. It communicates with the other tiers by outputting results to the browser/client tier and all the other tiers. The Business Logic tier is mainly responsible for information exchange between the user interface and the database of the project. The final layer of the three tiered architecture is the Data Access tier, which mainly consists of the Database servers. The information related to the Airline Reservation System is stored and retrieved from here.
A simple representation of the three-tier architecture would be as follows:

**Figure 4 - Three Tier Architecture**

- **Presentation tier**: The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

- **Logic tier**: This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

- **Data tier**: Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

```
The architecture of the Airline Reservation System can be depicted as follows:

![Diagram of three-tier architecture]

This includes the ASP.NET web forms and the ASP.NET user controls for the Airline Reservation System.

This includes the C# classes or the C# business components for the Airline Reservation System project. SQL server 2000 is

The three tier architecture would be discussed in detail in the following sections:

**3. PRESENTATION TIER**

The presentation tier is the top most layer of the Airline-Reservation system application. The presentation tier is mainly responsible for the user interface of the application which deals with the presentation of data to the user. The presentation tier of the Airline Reservation System is mainly formed by the ASP.NET web forms. In the case of the Airline Reservation system project, I have used the Visual Studio 2003.NET to create the web forms. Each web form will have the extension .aspx and there are several web forms created for the user and the administrator of the website.

The web pages of the Airline Reservation System project are as follows for the User as well as the Administrator. The following table shows the ASP.NET web forms for the users of the Airline Reservation System:
Table 2- ASP.NET webforms and their purpose

<table>
<thead>
<tr>
<th>ASP.NET Web Forms</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home.aspx</td>
<td>The home page for the Kansas Air Airline Reservation System website</td>
</tr>
<tr>
<td>Register.aspx</td>
<td>The page provided for the Customer Registration</td>
</tr>
<tr>
<td>Customer.aspx</td>
<td>The home page that appears after the customer logs in</td>
</tr>
<tr>
<td>FlightSearch.aspx</td>
<td>The page which helps the customer to search for the available flights</td>
</tr>
<tr>
<td>FlightBooking.aspx</td>
<td>The page which enables the customer to make reservations for the flights available online.</td>
</tr>
<tr>
<td>PackageSearch.aspx</td>
<td>The page which helps the customer to search for the packages available.</td>
</tr>
<tr>
<td>PackageBooking.aspx</td>
<td>The page which enables the customer to make reservations for the packages available online.</td>
</tr>
<tr>
<td>HotelSearch.aspx</td>
<td>The page which helps the customer to search for the available hotels and rooms in each hotel.</td>
</tr>
<tr>
<td>HotelBooking.aspx</td>
<td>The page which enables the customer to make reservations for the hotel rooms available online.</td>
</tr>
<tr>
<td>About.aspx</td>
<td>This page gives the customer some information about the Airline Reservation System website.</td>
</tr>
</tbody>
</table>
3.1 PAGE FLOW DIAGRAM

The page flow diagram for the users of the Airline Reservation System would be as follows:

**Figure 5 - page flow diagram for customer**

The home page of the Airline Reservation System has the login id and the password field for the user to login. So the home page can also be used as the login page for the customer.

4. BUSINESS LOGIC TIER

The Business Logic Tier is the middle tier of the three-tier architecture. The business logic for the Airline Reservation System would be present here. In the case of my project, the C# classes would be performing the duty of the business logic. This is the layer which is responsible for the information exchange between the user interface and the database.

The Airline Reservation system mainly consists of the Users, which can be further classified into the customer and administrator of the Airline Reservation System website. The class diagram for the Airline Reservation System would be as follows:
4.1 CLASS DIAGRAM

Figure 6 - class diagram

4.2 SEQUENCE DIAGRAM

The sequence diagram for the Customer of the Airline Reservation System would be as follows:

Figure 7 - sequence diagram for customer
5. DATABASE TIER

The database tier is the final and last tier of the three-tier architecture. All the data related to the Airline Reservation System project is stored and retrieved from here. For this project I have used the Microsoft SQL server to create the database. To be specific, Microsoft SQL server 2000 is being used. It is very easy to work with and makes creation and maintaining of tables very easy.
CHAPTER 4 - INSPECTION CHECKLIST

1. INTRODUCTION

The major purpose of the document is to provide a checklist to the formal technical inspectors of the Airline Reservation System project. The technical inspectors of the project will be provided with a checklist which they will use to verify the correctness and the consistency of the Airline Reservation System project. This helps the developer to understand any shortcomings present in the project.

2. ITEMS TO BE INSPECTED

The items that are to be inspected by the Formal Technical Inspectors are as follows:

- Vision Document
- Architecture Design Document
- UML diagrams like the class diagram, sequence diagram
- Formal specification document

3. FORMAL TECHNICAL INSPECTORS

The formal technical inspectors for the Airline Reservation System project are as follows:

- Srunokshi Neelakantan
- Sandhya Bathini

4. FORMAL TECHNICAL INSPECTION CHECKLIST

The formal technical inspection checklist for the Airline Reservation System project would be as follows:

<table>
<thead>
<tr>
<th>Items to be inspected</th>
<th>Pass/Fail/Partial</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The document follows the MSE portfolio standard which is described at:</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://mse.cis.ksu.edu/mse-portfolio.htm">http://mse.cis.ksu.edu/mse-portfolio.htm</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. All the classes used in the USE model are represented in the class diagram

3. The multiplicities have been correctly depicted in the class diagram

4. All the symbols used in the class diagram are according to the UML standards

5. Classes are clearly depicted in the class diagram

6. The symbols used in the sequence diagram correspond to the UML standards

7. Sequence diagram matches the class diagram

8. All the diagrams are correct and match the documentation presented to the technical inspectors
model AirlineReservationSystem

----- CLASSES

class User
attributes
  userid: string
  password: string
  emailid: string
  name: string
  loginstatus : Boolean
operations
  VerifyLogin(email : string , password : string) : Boolean = user.allInstances -> exists (u: user | u.emailid = emailid and u.password = password)
end

class Customer < User
attributes
  userid : string
  password : string
  customername : string
  emailid : string
  address : string
  phonenumber : integer
  creditcardnumber : integer
  age : integer
operations
  register()
  login() : Boolean
searchflights()
bookflights()
searchpackages()
bookpackages()
searchhotels()
bookhotels()
browse()

end

class flight()
attributes
  flightnumber : integer
  noofseats : integer
  source : string
  destination : string
end

class hotel()
attributes
  hotelname : string
  hotelid : integer
  noofrooms : integer
end

class package()
attributes
  packageid : integer
end

class seat()
attributes
seatnumber : integer

end

---- ASSOCIATIONS

-- This association is being written to indicate that there should be some number of seats on the plane greater than zero.
association planeseats between
Flight[1] role belongto
Seat[10..300] role has
end

-- A customer can book any no.of seats on the flight, depending upon the capacity of the flight.
Association customerandseats between
Customer[1] role bookedby
Seat[1..*] role books
End

-- A customer should be seated in the flight in only one seat.
Association customerseated between
Customer[1] role holds
Seat[1] role isheld
End

---- CONSTRAINTS

-- This constraint is written to indicate that each customer who registers to the Airline Reservation system website should have a unique id.
Context user
inv uniqueid:
user.allInstances -> forAll( u1,u2 | u1 <> u2 implies u1.userid <> u2.userid)
-- This constraint is written to indicate that each user will have a unique e-mail id.
Context user
inv uniqueemail:
user.allInstances -> forAll( us1,us2 | us1 <> us2 implies us1.emailid <> us2.emailid)
end

-- This constraint is written to indicate that each customer’s e-mail id, should be unique
Context customer
inv uniqueemail:
customer.allInstances -> forAll(c1,c2| c1<>c2 implies c1.emailid <> c2.emailid)
end

-- This constraint is written to indicate that the number of seats in the flight should be
greater than zero
Context flight
inv noofseats:
flight.allInstances -> forAll (f1 | f1.nooseats >= 1)
end

-- This constraint is written to ensure that no two flights have the same number.
Context flight
Inv uniquename:
flight.allInstances -> forAll (f1,f2| f1<>f2 implies f1.flightnumber <> f2.flightnumber)
end

-- This constraint is written to ensure that a hotel can have multiple rooms.
Context Hotel
Inv hotelrooms:
Hotel.allInstances -> forAll(h1| h1.noofrooms ) ->size() >=1
end
CHAPTER 6 - COMPONENT DESIGN

1. INTRODUCTION AND PURPOSE

The main purpose of the Component Design document is to provide and explain the component design for the Airline Reservation System project in detail.

2. CLASS DIAGRAM

The class diagram of the Airline Reservation System project would be as follows:

Figure 8 - class diagram
2.1 CLASS DESCRIPTIONS

2.1.1 USER

![User class diagram](image)

Figure 9 - user class

The user class is responsible for handling all the user functions. This user class is the super class or the base class for two types of users namely the customer and the administrator. The user logging into the system will be mainly verified for their username and password., which is the email id for the customer and a username for the administrator.

2.1.2 ADMINISTRATOR

![Administrator class diagram](image)

Figure 10 - administrator class

The administrator class is the sub class of the user class. This class is mainly depicted here to define all the attributes and functions carried out by the administrator of the Airline Reservation System website. The attributes for the administrator of the Airline Reservation System website are the userid and password, which the administrator uses to log on to the Airline Reservation system website. The administrator would be able to perform the tasks of adding or updating the information for flights, motels and packages etc.
2.1.3 CUSTOMER

The customer class has been depicted here as it is a sub class for the super class user of the Airline Reservation System project. The attributes for the customer would be the userid, password, customer name, email id, credit card number, address, phone number.

The functions that the customer of the website would be able to perform are:

Register() – the customer would be able to register onto the system
Login() - the customer can login with the email id and password
Logout() – the customer would be able to logout of the system
Browse () – the customer can browse through the website
Searchflights() and bookflights() – the customer would be able to search and book the flights available on the website.
Searchpackages() and bookpackages() - the customer would be able to search and book packages available on the website
Searchhotels() and bookhotels() - the customer would be able to search and book for the hotels available on the website
2.1.4 PACKAGE

The class package is depicted in the class diagram to represent the set of packages present in the application. The customer can book packages using the PackageSearch page. Package generally has the following attributes:
- Package id – a unique id given to a package
- Package name – the name of the package
- Package location - the location of the package…it indicates the place where the package is available
- Package price – the price of the package

2.1.5 FLIGHT

The class flight is depicted in the class diagram to represent the set of flights present in the application. The customer can book packages using the FlightSearch page. A flight present in the airline reservation system, generally has the following attributes:
- Flight number - A unique number is given to the flight
airline name – the name of the airline that the flight belongs to
source – the place where the flight has as its beginning point
destination – the place where the flight has as its end point
price – the price of the airline ticket of a particular flight

2.1.6 HOTEL

Figure 14 - hotel class

The class hotel is depicted in the class diagram to represent the set of hotels present in the application. The customer can book hotels using the HotelSearch and HotelBooking page. A Hotel available in the Airline Reservation System application generally has the following attributes:
hotel id – a unique id given to a hotel
hotel name – the name of the hotel
location - the location of the hotel
number of rooms – the number of rooms available at each hotel
price – the price of each room in a hotel
CHAPTER 7 - SOFTWARE QUALITY ASSURANCE PLAN

1. PURPOSE

The main purpose of the Software Quality Assurance plan is to ensure production of high quality end software product according to the specific requirements stated. The Software Quality Assurance plan of the Airline Reservation System establishes the goals, processes and responsibilities required to ensure high quality and on-time delivery of the project. The results of the reviews and audits conducted in the Software Quality Assurance plan would be provided to the appropriate management of the project, so that they can track and assess the progress being made on the project.

2. REFERENCE DOCUMENTS

- IEEE standard for Software Quality Assurance Planning
- IEEE guide for Software Quality Assurance Planning
- Project Plan document for the Airline Reservation System

3. MANAGEMENT

3.1 ORGANIZATION

The organization consists of the supervisory committee, major professor, developer and two formal technical inspectors.

**Supervisory Committee**

The supervisory committee consists of:

- Dr. Daniel Andresen
- Dr. Torben Amtoft
- Dr. Mitchell L. Nielsen

**Major Professor**

- Dr. Daniel Andresen
Developer
The developer for this project is Kaavya Kuppa.

Formal Technical Inspectors
The formal technical inspectors of the Airline Reservation Project are:
- Sandhya Bathini
- Srunokshi Neelakantan

3.2 RESPONSIBILITIES

Supervisory Committee
The supervisory committee will be responsible for attending the presentations and submitting their reviews at the end of each phase. After each presentation, the committee will provide feedback and suggestions pertaining to each phase.

Major Professor
The major professor of the project will supervise and evaluate the work of the developer on a regular timely basis. Along with the other supervisory committee activities, he will also measure the progress being made by the developer at each meeting.

Developer
The developer of the project will be responsible for all the documentation and software development tasks of the Airline Reservation System project. The developer will also meet the major professor on a timely basis to report the progress of the project.

Formal Technical Inspectors
The formal technical inspectors are necessary to ensure good quality of the software design of the project. They will be responsible for a technical review of the architecture design artifacts and the formal requirements specifications and will also be required to submit a formal report based on their observations.
These formal technical inspectors will be provided with a formal technical inspection checklist, which will contain all the items that need to be inspected. After inspecting the project against all the items in the checklist, the two technical inspectors will provide their report, which will be included in the documentation for the project.

3.3 TASKS

All the tasks performed during the Airline Reservation System project are documented in the project plan. The project plan along with the software quality assurance plan will be reviewed at the end of phase 1 by the supervisory committee and all the changes necessary will be incorporated in the documents and will be submitted at the end of phase 2.

4. DOCUMENTATION

The documentation for the Airline Reservation System project consists of documents submitted at the end of each phase of the project. They consist of the vision document, project plan, software quality assurance plan, architecture design, test plan, formal technical inspection, prototype, user manual, component design, source code, assessment evaluation, project evaluation, references and finally the formal technical inspection letters. The supervisory committee will review all the documentation submitted at the end of each phase for final approval. All the documentation prepared for the Airline Reservation System project will be uploaded at the developer’s website at:

http://people.cis.ksu.edu/~kaavya/MSE%20PROJECTPAGE.htm

5. STANDARDS, PRACTICES, CONVENTIONS AND METRICS

Documentation Standard
The IEEE standards are used as a guideline for all the documentation of the project.

Coding Standard
The project follows the guidelines in the C# coding standards and style guide.

Commentary Standard
Comments are used in the project to give a brief description of the code, which mainly focuses on the functionality and purpose of the commented areas. Each block of statements will be well-commented. Each routine will also have a comment which will be placed above the specific routine.

**Testing Standard**
The IEEE standard for software test documentation will be used for the Airline Reservation System project.

**Metrics**
The basic COCOMO model will be used to estimate the project time and effort.

6. REVIEWS AND AUDITS

The main purpose of the reviews and audits is to check the quality of the application as it develops. Apart from the audits and reviews conducted periodically, all the project documentation will also undergo some technical inspections which will ensure that the quality of the documentation is in compliance with the project. The developer of the project will give three formal presentations, one at the end of each phase. The supervisory committee members will conduct periodic reviews on the project code and documentation and will also evaluate the performance of the developer at the end of each phase. So, the committee members will also evaluate the software prototype at the end of each presentation and suggest changes that need to be incorporated in the project code or documentation. The two formal technical inspectors of the project, assess the architecture design artifacts and submit a formal report based on their observations.

7. TEST AND PROBLEM REPORTING

The developer of the Airline Reservation System will develop a test plan which will outline all the test activities. All the tests and their results will be evaluated and documented. These documents will also be reviewed by the supervisory committee members. All the unresolved problems will be reported to the committee members.
8. TOOLS, TECHNIQUES AND METHODOLOGIES

The following are the tools that will be used for coding, testing and documentation:

- Microsoft Visual Studio .NET 2003 – for coding
- C# - for coding
- HTML – for coding
- JavaScript for coding
- IIS – for web server
- ASP.NET – for web forms
- SQL server 2000 – for database server
- MS WORD 2003 – for documentation
- XML – for coding
- NUnit – for unit testing
- JMeter for performance testing
- User Testing

9. RECORDS COLLECTION, MAINTENANCE AND RETENTION

Three copies of the design documentation are to be produced. One will be kept in the Kansas State University Library, one with the major professor and the third with the developer himself. A softcopy of the entire source code, documentation and web pages of the Airline Reservation System project will be submitted to the major professor. A copy will also be kept with the developer.

10. DELIVERABLES

The following set of deliverables will be submitted at the end of each phase:

Phase I
Vision Document 1.0
Project Plan 1.0
Software Quality Assurance Plan

Phase II
Action items identified during phase I
Vision Document 2.0
Project Plan 2.0
Formal Requirement Specification
Architecture Design
Test Plan
Formal Technical Inspection – submitted by two individual MSE students
Executable Architecture Prototype

**Phase III**
Action items identified during phase II
User Manual
Component Design
Source Code
Assessment Evaluation
Project Evaluation
Test Results
References
Formal Technical Inspection- submitted by two individual MSE students
CHAPTER 8 - TEST PLAN

1. TEST PLAN IDENTIFIER
Airline Reservation System – V 1.0

2. INTRODUCTION
The main purpose of the test plan for the Airline Reservation System is to discuss the testing details of the use cases of the Airline Reservation System. The software project test plan also describes the objective, scope and approach of the software testing effort for the Airline Reservation System project. The test plan for the Airline Reservation System also indicates the personnel responsible for each task and also specifies the risks associated with the test plan.

2.1 OBJECTIVES
The main objectives of the test plan for the Airline Reservation System are as follows:
- To identify the features of the system that will be tested.
- To identify and define all the activities necessary to prepare for and conduct the testing process on the Airline Reservation System
- To define the pass/fail criteria for each item that will be tested
- To identify the deliverables of the testing phase.
- To define any suspension criteria and resumption techniques
- To discuss the testing techniques being used to test the Airline Reservation System.

2.2 REFERENCES
The following references have been used in the preparation of the Test Plan document for the Airline Reservation System:
- Deliverables of the Phase I for the Airline Reservation System.
- MSE Portfolio lectures online
- Wikipedia
2.3 DEFINITIONS

The following are some of the terms and definitions that are related to the test plan of the Airline Reservation System:

- **Pass/Fail criteria**: Decision rules that are used to determine whether a software item passes or fails a test.
- **Test**: A collection of one or more test cases
- **Test Item**: A software item that is an objective of testing.
- **Test Plan**: A document describing the scope, approach, resources and schedule of the intended testing activities.
- **Test Summary Report**: A document summarizing the testing activities and results.
- **Testing**: The process of analyzing a software item to detect the differences between the existing and required conditions.

3. TEST ITEMS

This section of the test plan lists all the items of the Airline Reservation System project that will be tested:

- Login
- Search and book flights
- Search and book packages
- Search and book hotels
- Register

4. APPROACH

This section of the test plan describes the overall approach for testing the Airline Reservation System project. The approach followed for testing the Airline Reservation System ensures that the major features of the project are adequately tested. All the testing will be done with the help ANTS (Advanced .NET Testing System). The testing would be carried out on the Airline Reservation System while logging into the system as a customer or a normal user of the system.
4.1 UNIT TESTING

The Unit Testing is a test that tests each single module of the software to check for errors. This is mainly done to discover errors in the code of the Airline Reservation System. The main goal of the unit testing would be to isolate each part of the program and to check the correctness of the code. In the case of the Airline Reservation System, all the web forms and the C# classes will be tested. There are many benefits for this unit testing:

- The unit testing facilitates change in the code.
- It allows testing to be done in a bottom up fashion.

At the same time, unit testing has some disadvantages such as, it might not identify each and every error in the system.

4.2 INTEGRATION TESTING

In Integration Testing, the individual software modules are combined and tested as a whole unit. The integration testing generally follows unit testing where each module is tested as a separate unit. The main purpose of the integration testing is to test the functional and performance requirements on the major items of the project.

All the modules of the project developed individually would be combined together and tested as a whole system in the integration testing.

4.3 REGRESSION TESTING

The Regression Testing is generally done whenever modifications are made to the source code of a project. The Regression Testing can also be defined as the process of testing changes made to the computer program and also makes sure that the older programming still works with the new changes.

So, before any new version of a software product is released, the old test cases for the project will be run against the software with the changes made, to make sure that the old functionalities of the project still work.
4.4 ACCEPTANCE TESTING

This testing is generally performed when the project is nearing its end. This test mainly qualifies the project and decides if it will be accepted by the users of the system. The users or the customers of the project are responsible for the test.

4.5 SYSTEM TESTING

The system testing is mainly done on the whole integrated system to make sure that the project that has been developed meets all the requirements. The test cases for the system testing will be the combination of unit and integration tests.

5. TEST CASES

The following are the test cases for the Airline Reservation System:

5.1 TEST CASE 1 – USER LOGIN

- **Incorrect Input:** Incorrect username, which is the email-id in the case of the Airline Reservation System.
- **Pass Criteria:** An appropriate message should be generated to indicate that an invalid username has been typed.
- **Correct Input:** The correct input would be a valid e-mail id of the user and a correct password associated with the email-id which he uses to log in.
- **Pass Criteria:** The user should be directed to the webpage that the customer is intended to go to after he logs into the system.

5.2 TEST CASE 2 – USER REGISTRATION

- **Incorrect Input:** Wrong format entered in the input fields for the registration page.
- **Pass Criteria:** An appropriate message should be generated to the user saying that he has entered the wrong format in the specific input field.
- **Correct Input:** The correct input would a correct format entered by the customer into the input fields of the registration page.
• **Pass Criteria:** The pass criteria for this test case would be a successful registration of the customer into the Airline Reservation System website. The system would log the user into the system after this.

5.3 TEST CASE 3 – USER REGISTRATION

• **Incorrect Input:** The data fields left out empty in the registration page.
• **Pass Criteria:** An error message should be generated to the user saying that he has to fill out those fields in order to be registered into the system.
• **Correct Input:** The correct input in this case, would be that the customer would enter the data in all the fields in the registration form.
• **Pass Criteria:** The pass criteria for the system would be that it accepts all the customer details and then registers the customer and helps him log into the system.

5.4 TEST CASE 4 – SEARCH AND BOOK FLIGHTS

• **Incorrect Input:** Incorrect input in this case, would be incorrect search criteria entered or incorrect format of data entered into the data entry fields of the flight search and booking page.
• **Pass criteria:** A message has to be generated to the user indicating the wrong entry that he has made in the fields.
• **Correct Input:** A correct input would be entering the data into the data entry fields in a correct format.
• **Pass Criteria:** The pass criteria for this test case would be that the search would return valid results and then when the customer made the booking, the system has to generate a confirmation to the customer and indicate that an e-mail has been sent to the customer.

5.5 TEST CASE 5 – SEARCH AND BOOK PACKAGES

• **Incorrect Input:** Incorrect input in this case, would be incorrect search criteria entered or incorrect format of data entered into the data entry fields of the package search and booking page. In this case, a wrong input would be an incorrect package id etc.
• **Pass criteria:** A message has to be generated to the user indicating the wrong entry that he has made in the fields.

• **Correct Input:** A correct input would be entering the data into the data entry fields in a correct format.

• **Pass Criteria:** The pass criteria for this test case would be that the search would return valid results and then when the customer made the booking, the system has to generate a confirmation to the customer and indicate that an e-mail has been sent to the customer.

5.6 TEST CASE 6 – SEARCH AND BOOK HOTELS

• **Incorrect Input:** Incorrect input in this case, would be incorrect search criteria entered or incorrect format of data entered into the data entry fields of the hotel search and booking page. In this case, an incorrect input would be an incorrect hotel id, or an incorrect format of date entered in the input field for the date.

• **Pass criteria:** A message has to be generated to the user indicating the wrong entry that he has made in the fields.

• **Correct Input:** A correct input would be entering the data into the data entry fields in a correct format.

• **Pass Criteria:** The pass criteria for this test case would be that the search would return valid results and then when the customer made the booking, the system has to generate a confirmation to the customer and indicate that an e-mail has been sent to the customer.

6. PASS OR FAIL CRITERIA

The test cases executed on the Airline Reservation System will pass if they meet the specific requirements mentioned in the Vision document of the project. A test case is said to fail, if the desired functionality is not satisfied by the system.

7. SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

7.1 SUSPENSION CRITERIA
Testing for all the dependant features will be suspended if a test case fails. The failed test case will be logged onto the test log which contains the description for the error.

7.2 RESUMPTION REQUIREMENT

The test cases which are not dependant on the case where the bug is reported will be executed in parallel with the bug fixing. Once the failed test case has been taken note of and has been identified and fixed then the testing for the failed test case will resume.

8. TEST DELIVERABLES

The following documents will be produced after the testing phase for the Airline Reservation System has been completed.

- Test Plan
- Test Cases
- Test Log
CHAPTER 9 - ASSESSMENT EVALUATION

1. INTRODUCTION

The Assessment Evaluation document presents the results obtained by testing the Airline Reservation System. The test cases are in reference to the test cases defined in the Test Plan document from the Phase II.

2. TEST CASE RESULT SUMMARY

The summary of the test case result has been depicted in the table shown below:

These are the test cases for the User/Customer pages. The major part of testing has been concentrated on the customer pages, as they would be the main clients of the Airline Reservation System website.

Table 4- user/customer pages testing summary

<table>
<thead>
<tr>
<th>TEST CASE #</th>
<th>DESCRIPTION</th>
<th>RESULTS/COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC # 1</td>
<td>User Login</td>
<td>Passed</td>
</tr>
<tr>
<td>TC # 2</td>
<td>User Registration</td>
<td>Passed</td>
</tr>
<tr>
<td>TC # 3</td>
<td>Search and Book Flights</td>
<td>Passed</td>
</tr>
<tr>
<td>TC # 4</td>
<td>Search and Book Packages</td>
<td>Passed</td>
</tr>
<tr>
<td>TC # 5</td>
<td>Search and Book Hotels</td>
<td>Passed</td>
</tr>
</tbody>
</table>

The below table represents the summary of results of testing on the Administrator pages. The results have been explained in detail later in the document.

Table 5 - Administrator pages testing summary

<table>
<thead>
<tr>
<th>TEST CASE #</th>
<th>DESCRIPTION</th>
<th>RESULTS/COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC # 6</td>
<td>Administrator sign in</td>
<td>Passed</td>
</tr>
<tr>
<td>TC # 7</td>
<td>New Flight/Package/ Hotel addition</td>
<td>Passed</td>
</tr>
<tr>
<td>TC # 8</td>
<td>Updating Flight/Package/Hotel details</td>
<td>Passed</td>
</tr>
</tbody>
</table>
3. DETAILED TEST RESULTS

3.1 MANUAL TESTING

To start with, I have performed manual testing on the Airline Reservation System website. Manual Testing is one of the oldest and rigorous methods of software testing. This testing strategy gives the best opportunity to check every page thoroughly and make sure it works in the expected manner. Due to the complexity of the various automation tools and the time available for testing the entire web application, I preferred to use manual testing based on the fact that it is one of the best methods of testing suggested for a beginner.

All the test cases mentioned in the Test Plan document of Phase II were tested here. The results of the manual testing are represented in the following tables:

3.1.1 TC # 1 – USER LOGIN

Table 6 - tc # 1 - user login

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log In Button</td>
<td>An invalid username(which is the e-mail id in this case) or password is entered by the user</td>
<td>The system generates a message saying “invalid user id” or invalid password, whichever is the case.</td>
</tr>
<tr>
<td>Log In Button</td>
<td>A valid username and password is entered by the user</td>
<td>The system logs on the user and transfers him to the booking page</td>
</tr>
</tbody>
</table>

3.1.2 TC # 2 - USER REGISTRATION

Table 7 - tc # 2 - user registration

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>New User button (Used for Register)</td>
<td>Wrong format entered in the input fields of the registration page</td>
<td>The system prompts a message to the user saying that</td>
</tr>
</tbody>
</table>
he has entered a wrong format in the input fields.

New User button  | Passwords and Confirm Password fields do not match in the registration page  | The system generates a message to the user saying “please enter the confirm password field” again.

New User button  | Data Fields left out empty in the registration page.  | The system prompts a message to the user asking him to fill the empty fields he has left out.

New User button  | Correct data entered into the fields in the register page  | The system accepts the details of the customer and then logs him onto the system and displays the page where he can search and book for flights, packages and hotels.

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight booking</td>
<td>Wrong format of information entered into the data fields of the flight booking page</td>
<td>The system generates an error message to the user indicating that the wrong format of data is entered and to re-enter the data.</td>
</tr>
</tbody>
</table>
| Flight Booking | Wrong date format, in the date of journey data field, wrong flight number in the Flight Number | The system generates a message to the user saying that he has entered an invalid date.
fields etc,
(similarly for all the other data fields of the FlightSearch page)
format and incase of wrong flight number, indicates that his entry is invalid.

| Flight Booking | Correct format of data is entered into the data fields in the flight booking page. | The system allows the users to book the flight by providing details required and directs them to the booking confirmation page. |

3.1.4 TC # 4 – SEARCH AND BOOK PACKAGES

Table 9 - tc # 4 search and book packages

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package Booking</td>
<td>Wrong format of data is entered into the data fields of the package booking page</td>
<td>The system displays an error message to the user saying that invalid form of data has been entered into the data fields of the page.</td>
</tr>
<tr>
<td>Package Booking</td>
<td>Correct form of data is entered into the datafields of the PackageSearch.aspx page.</td>
<td>The system verifies the details entered by the customer, accepts the details and confirms package booking by redirecting the customer to the package confirmation page.</td>
</tr>
</tbody>
</table>

3.1.5 TC # 5 - SEARCH AND BOOK HOTELS

Table 10 - tc # 5 search and book hotels

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
</table>

Hotel Booking | Wrong format of data is entered into the data fields of the flight search page. | The system displays an error message to the user saying that he has entered invalid data into the data fields.

Hotel Booking | Correct data entered into the data entry fields of the HotelSearch.aspx Page | The system then verifies and accepts the data entered by the user into the data entry fields allowing the user to book hotel specified and then directs him to the hotel confirmation page.

All the above mentioned results are for the customer pages. On the whole, the user/customer pages have passed the manual testing phase. The manual testing results for the Administrator pages of the Airline Reservation System are as follows:

**3.1.6 TC # 6 - ADMINISTRATOR SIGN IN**

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator sign in feature</td>
<td>Wrong username/password entered into the username and password data fields</td>
<td>The system generates a message to the user saying that an incorrect username / password have been entered.</td>
</tr>
<tr>
<td>Administrator sign in feature</td>
<td>Correct username and password entered into the username and password fields in the Administrator sign in page.</td>
<td>The system verifies the details and allows the administrator to log on to the system.</td>
</tr>
</tbody>
</table>
3.1.7 TC # 7 – NEW FLIGHT/PACKAGE/HOTEL ADDITION

Table 12 - new flight /hotel/.package addition

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add new flight/package/hotel to the database</td>
<td>The administrator tries to add flight details already matching the details of a flight present in the database. In other words, a duplicate record is being created by the administrator.</td>
<td>The system generates a message to the Administrator saying that the Record already exists, thus avoiding duplicates.</td>
</tr>
<tr>
<td>Add new flight feature</td>
<td>The administrator enters new flight details into the form, that is those details are already not present in the database.</td>
<td>The system then verifies the details entered by the Administrator and then saves the entry into the table and displays a message that Record has been saved successfully.</td>
</tr>
</tbody>
</table>

3.1.8 TC # 8 – UPDATING FLIGHT, PACKAGE, MOTEL DETAILS

Table 13 - updating flight, package, motel details

<table>
<thead>
<tr>
<th>TEST UNIT</th>
<th>TEST CASE</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update flight/package/motel details button</td>
<td>The administrator enters the wrong format of data in the data fields of the flight/motel/package page and hits the update button</td>
<td>The system displays a message to the Administrator saying that wrong format of data has been entered into the data fields</td>
</tr>
<tr>
<td>Update flight/package/motel</td>
<td>The administrator enters the correct format of data in the data entry</td>
<td>The system verifies the details entered by the</td>
</tr>
</tbody>
</table>
Thus, the Administrator web pages have also passed the manual testing phase and thus the above results have been produced.

### 3.2 PERFORMANCE TESTING

In general, performance testing can be defined as one form of testing where we test the system to determine how fast the system performs under a particular workload. Several other features of the system such as scalability and security etc can also be tested under the performance testing phase. For the performance testing of the Airline Reservation System, I have used the JMeter tool. The Apache JMeter is a 100% pure Java desktop application which has been designed to load test functional behavior and measure performance. This tool has been originally designed for Web Applications, which has now been extended to a variety of other functions.

The Apache JMeter can be used to performance test both static and dynamic sources. With JMeter I was able to test various combinations of load types on the Airline Reservation System web application and analyze the strength of the system. The inputs to the JMeter tool, generally used for the performance testing of a web application would be as follows:

- **Number of Threads/Users** – The total number of people sending requests to a web page
- **Ramp-Up Period** - The Ramp Up period would be the time taken by JMeter to create and make sure that all the threads are up and running.
- **Loop Count** – It would be the value which determines the number of times that the test should be carried out.
The criteria selected to be the input to the JMeter tool for the Airline Reservation System would be explained later in the document. For the performance testing of the Airline Reservation System, I have chosen three kinds of pages.

1. Home Page of the Airline Reservation System - It deals with people just accessing the Home page of the Airline Reservation System. There is no database activity involved here. So this would be one good choice

   http://localhost/Kansas%20Air/Home.aspx

2. Login Page of the Airline Reservation System- This is the page where the user logs onto the home page of the system and is redirected to the Booking.aspx page after he logs in. So, here it involves authentication and some database activity. This would be another good choice.

3. Flight Search and Booking Page of the Airline Reservation System

   Here, the user searches for the available flights in the database of flights available and makes a reservation for a specific flight. There is a lot of database activity involved with the flight search page. The results of the JMeter testing are as follows for each of the above mentioned pages:

3.2.1 Home page of Airline Reservation System website

   http://localhost/Kansas%20Air/Home.aspx
Fig: Summary report for Home.aspx page

The above screenshot shows the result of running the JMeter test for the home page of the Airline Reservation System project. The above sample has been obtained as a result of the following load details applied in the JMeter:

<table>
<thead>
<tr>
<th>Thread Count</th>
<th>Ramp-Up Period</th>
<th>Loop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Thus the above result was obtained for a total of 1000 samples.

The following screen shot shows the response code 200 generated by the JMeter after it has accessed the page. Here the Response 200 code is generated in a response that the home page has been detected by the server.

The response time for the Home page was very less, and the number of requests handled per sec is higher when compared to the other pages of the project. Since there is no database activity involved. For the home page a throughput of about 100 requests/sec is recorded, which is the number of requests handled by the server per second.
3.2.2 Login Page of the Airline Reservation System

http://localhost/Kansas%20Air/Booking.aspx

The above screenshot is produced as a result of the JMeter test performed on the Booking.aspx page, which can be accessed by the user after he logs onto the website. The above result has been produced as a result of the following load being applied on the server:

<table>
<thead>
<tr>
<th>Thread Count</th>
<th>Ramp Up Period</th>
<th>Loop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Thus, for a total of about 1000 samples, the above results were recorded by JMeter. The response time for the Booking.aspx page is a little higher than that of the Home page. The Home page of the project does not have any database activity related to it. Where as for the Booking.aspx page, the customer has to login in order to access the page. So, the response time for the Booking.aspx page was a little higher when compared to the Home.aspx.

The result of the response created by JMeter while accessing the Booking.aspx page is as follows:
3.2.3 Flight Search and booking page of the Airline Reservation System

http://localhost/Kansas%20Air/FlightSearch.aspx

Fig: Summary report for the FlightSearch.aspx page
The above screenshot is produced as a result of the JMeter test performed on the FlightSearch.aspx page, which can be accessed by the user after he logs onto the website. The above result has been produced as a result of the following load being applied on the server:

<table>
<thead>
<tr>
<th>Thread Count</th>
<th>Ramp Up Period</th>
<th>Loop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Thus, for a total of about 1000 samples, the above results were recorded by JMeter. The throughput for the FlightSearch.aspx page is very less when compared to the Home.aspx page or the Booking.aspx page. It is because, the Flights available to the customer are displayed on this page from the database and then the customer searches for a desirable flight and book the flight. The response time for the FlightSearch.aspx page is very high when compared to the other two pages. The time taken for the page to load is also very high compared to the other two pages.

The result of the response created by the JMeter for the FlightSearch.aspx page is as follows:

4. OBSERVATIONS
Based on the results obtained from the JMeter, the following observations were made:

1. The system can accept only a total of 800 users at a time. But there are some exceptions to this too. If a combination of 800 users with a loop count of 1 was used for testing, then system responds well. But when a combination of 800 users with a greater loop count is used, it produced the following output in the response window of JMeter:

   ![JMeter Results]

   Ultimately, the page requested would not be available to any of the users and an error message would be displayed saying that there are too many people connected to the IIS server and that the access to the requested web page is forbidden.

There are many factors affecting the behavior of the server.
a. The Airline Reservation system project runs on a personal version of IIS and on a personal computer. So, the hardware capability of the system might be one of the reasons for the server to respond slowly.

b. JMeter also took a lot of time to respond to each request. This might be due to the fact that the server on the laptop is very slow. So, the results produced by JMeter also might not be accurate results.

2. The throughput obtained for the three pages are different. The three pages have been tested under the same condition that is for 100 users, the loop count being 10 and the ramp up period also being 10. The home page of the Airline Reservation System project does not have any database activity involved with it. So the throughput for it is very high when compared to the other pages.

<table>
<thead>
<tr>
<th>WEB PAGE</th>
<th>THROUGHPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home page</td>
<td>100/sec</td>
</tr>
<tr>
<td>Login page</td>
<td>91.0/sec</td>
</tr>
<tr>
<td>Flight Search and Booking page</td>
<td>62.3/sec</td>
</tr>
</tbody>
</table>

Table: Comparison of throughput for three pages

3. The response time for each page also increases with the number of users accessing the page. For the three pages the average response time with the number of users as 100, loop count as 10 and ramp up period as 10, are as follows:

<table>
<thead>
<tr>
<th>Web Page</th>
<th>Average Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home.aspx</td>
<td>3 ms</td>
</tr>
<tr>
<td>Booking.aspx</td>
<td>2 ms</td>
</tr>
<tr>
<td>FlightSearch.aspx</td>
<td>6 ms</td>
</tr>
</tbody>
</table>
Upon increasing the number of elements present in the customer database table, which is by adding around 5000 new rows of random data to the customer table, the results obtained for JMeter are as follows:

The throughput also decreases a lot and the number of requests handled by the server reduces to about 12.3 requests per second.

The average response time for the customer details page before adding large number of rows into the database table and after adding 5000 rows of data to the database table are as follows:

<table>
<thead>
<tr>
<th>WEB PAGE</th>
<th>BEFORE</th>
<th>AFTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerDetails.aspx</td>
<td>5 ms</td>
<td>3699 ms</td>
</tr>
</tbody>
</table>

The average response time of the page has shot up by a very high percentage. I have added 10,000 users to the customer table and tested the application with JMeter, the error rate obtained was very high and JMeter displayed a message in the response window saying that there are too many users accessing the website and the access has been forbidden.

4. The above results are also influenced by the fact that the IIS server and the database server run on the same machine. So, this excessive load might also affect the results being produced by the JMeter tool.

Based on the above observations the following conclusions can be made about the Airline Reservation System project:
1. The error percentage for each of the three pages is always 0%. So, the pages are very reliable to the customer.

2. A decent throughput has also been obtained for the tested pages and it is above 63 requests/sec. However by improving the hardware of the system and by testing under conditions where the overall performance of the system is very high also produces changes in the results.

3. Since the IIS server and the database server run on the same machine, the results obtained by JMeter might be affected by this factor. The performance and scalability of the Airline Reservation System could be further improved by using dedicated Database server and a dedicated web server.

Thus the above observations have been drawn based on results of testing of Airline Reservation System.
CHAPTER 10 - USER MANUAL

1. INTRODUCTION

The User Manual explains us the step by step procedure to set up and use the Airline Reservation System web application.

2. INSTALLATION AND SETUP

This section describes the required software and hardware for the Airline Reservation System project.

2.1 REQUIRED HARDWARE

- A Microsoft SQL Database Server
  
  The Server should be equipped with a processor clock speed of over 1.5 GHz and a memory of over 512 MB.

- An IIS web-server
  
  The server should be equipped with a processor speed of over 1.5 GHz and a memory of over 512 MB.

2.2 REQUIRED SOFTWARE

The required software for the Airline Reservation System project would be as follows:

- Operating System: Microsoft Windows XP Professional

- Internet Information Server (IIS): The IIS Web Server is included in the Windows XP Professional installation CD. Since it is not automatically installed along with the XP Professional Installation, it has to be installed.

- .NET Framework 1.1 SDK. It has been installed with the help of an installation CD provided by the CIS department at Kansas State University

- Microsoft Visual Studio .NET 2003. It has been downloaded from the MSDNAA account provided by the CIS department at Kansas State University.

- Microsoft Internet Explorer 5.0 or higher (or) Mozilla Firefox for clients

- Microsoft SQL server 2000, which includes the SQL Query Analyzer and the Enterprise Manager. It can be downloaded from the following website:
2.3 REQUIRED NETWORK CONFIGURATION

A broadband connection with 10Mbps and above connection is needed to provide fast linkage to the server, thus providing faster service to the user/customer.

2.4 SOFTWARE SET-UP

The Windows XP Professional Operating System is installed in the D: / drive of the laptop.

- Unpack the software to the World Wide Web root of the IIS Server. (Generally it is the “C:\Inetpub\wwwroot” directory. But in this case, since I have the XP Professional Operating System installed on the D:\ drive it is the “D:\Inetpub\wwwroot”.
- Open the “Web.config” file and change the values of the < add name> property according to your connection configurations.
- Open the SQL server 2000
- Create a database and label it as KansasAir_DB
- Open the SQL Query Analyzer
- Paste the contents of the KansasAir SQL files
- Select the name of the database as Kansas Air_DB
- Select F5 and all the tables will then be populated and formed.
- Open the Internet Information Services
- Go to the Start -> Control Panel -> Performance and Maintenance as shown in the screen shot bellow:
Then select the administrative tools option and then select the Internet Information Services as shown in the figure below:

Figure 16 - IIS
• Now choose the option Web Sites
• Then right click on the Default Web Site option
• Select the New -> Virtual Directory Option as shown in the screenshot below:

Figure 17 - Creating a New Virtual Directory
• Now click on the Next Button which will then open the Virtual Directory Creating Wizard.
• Click the Next Button

Figure 18 - Virtual Directory Creating Wizard

• Enter the Alias Name as Kansas Air and then click the Next Button
• Now Browse the directory and select the folder where you have unpacked the software and then click on the Next Button.

• Now again click the Next Button
• Then the below screen shot appears:

• Now click the Finish Button. The following screen shot will appear:

• Now after this click on the Kansas Air folder and then select the “Home.aspx“

• Right click on “Home.aspx” and then select the Browse option.
You should then be able to see the home page of the Airline Reservation System website as follows:

**Figure 20 - Home Page of Airline Reservation System**

![Home Page of Airline Reservation System](image)

The customer of the Airline Reservation System website will then be able to logon to the system, to start using the features of the website. If the customer of the Airline Reservation System is a new user and does not have an account with the website, he can log on to the system only after registering with the web-site.

3. AIRLINE RESERVATION SYSTEM USAGE FOR CUSTOMERS

3.1 REGISTER FEATURE

After the user views the home page of the Airline Reservation System website, he can begin to browse the links available on the home page of the application. The main purpose of the Register feature is to help the customer of the website to create a login and password, which he will later use to log on to the system. The Register page will appear to the customer once he clicks the New User button on the home page of the application. The screen shot gives a description of the above explanation:

**Fig: New User button on Home page of the application**

![New User button on Home page of the application](image)
Once the user clicks the new user button, the Register page appears where the user can enter details into the system. The Registration page for the user would be as follows:

**Figure 21 - Customer Registration Page**
On this page, if the user does not enter the correct details into the data fields of the registration page, then the system prompts an error message to the user asking him to enter the correct format of data into the fields.

Once all the details have been entered into the system by the user, the customer then clicks the submit button on the registration page. All the details are then accepted and verified by the system and then the user is redirected to the home page for the customer, where he can then search and book the various flights, packages and hotels available to him.

The home page for the user after he logs onto the system would be as follows:

Figure 22 - Customer Home Page

From this page, the user can either go back to the home page of the Airline Reservation System website, or he can search and book flights, packages or hotels or logout of the application.

3.2 FLIGHT SEARCH AND BOOKING

The main feature of this part of the application is to enable the user to search and book for the flights available to him through the website. When the user clicks the flight booking link
available to him on the Booking.aspx page, the user would be redirected to the FlightSearch.aspx page where the user would be able to see the list of available flights. The screen shot of the FlightSearch.aspx page would be as follows:

**Figure 23 - FlightSearch Page**

The user can then search for a specific flight which has a choice of his source and destination. From the drop down list available to the user, the user can select a place of his choice from the source and a place of choice from the destination drop down box and then hit the search button on the page. If there is a flight in the database with the above choice of the customer, he can then view the results on the following page:

Then by filling in the required details on the page, like the data of journey, number of tickets and the flight number, he can then click the make booking button on the page and then the user would be directed to a confirmation page, which is as follows:
3.3 HOTEL SEARCH AND BOOKING

The main feature of this part of the application is to enable the user to search and book for the hotels available to the customer through the website. When the user clicks the HotelBooking link available to him on the Booking.aspx page, the user would be redirected to the HotelSearch.aspx page where the user would be able to see the list of available hotels. The screen shot of the HotelSearch.aspx page would be as follows:

The customers can logout from the system after this process by clicking the logout button on the FlightBooking.aspx page shown above.
The user can then search for a specific hotel of his choice. From the drop down list available to the user, the user can select a place of his choice and then hit the search button on the page. If there is a hotel in the database with the choice of the customer, he can then view the results on the following page:
Then by filling in the required details on the page, like the check in date, number of persons, number of days etc, he can then click the make booking button on the page and then the user would be directed to a confirmation page, which is as follows:

**Figure 26 - Hotel Booking Confirmation**

The customers can logout from the system after this process by clicking the logout button on the HotelBooking.aspx page shown above.
3.4 PACKAGE SEARCH AND BOOKING

The main feature of this part of the application is to enable the user to search and book for the packages available to the customer through the website. When the user clicks the PackageBooking link available to him on the Booking.aspx page, the user would be redirected to the PackageSearch.aspx page where the user would be able to see the list of available packages. The screen shot of the PackageSearch.aspx page would be as follows:

**Figure 27 - Package Search page**

The user can then search for a specific package of his choice. From the drop down list available to the user, the user can select a place of his choice and then hit the search button on the page. If there is a package in the database with the choice of the customer, he can then view the results on the following page:
Then by filling in the required details on the page, like the date of journey, number of persons, etc, he can then click the make booking button on the page and then the user would be directed to a confirmation page, which is as follows:

**Figure 28 - Package Confirmation page**
The customers can logout from the system after this process by clicking the logout button on the PackageBooking.aspx page shown above.
CHAPTER 11 - PROJECT EVALUATION

1. INTRODUCTION

This document mainly focuses on presenting the summary of experiences gained by me as an MSE student during the entire life cycle of the MSE project.

2. PROBLEMS ENCOUNTERED

This section of the project evaluation document describes all the difficulties that I have encountered during my MSE project.

2.1 SOLUTION DOMAIN RESEARCH AND LEARNING

Identifying a technology to work on for my MSE project was one of the basic difficulties that I have faced. But, to be honest, with my growing interest for C#.NET, I decided that it would be the best fit for my MSE project. Since my MSE project “The Airline Reservation System” is a web application project C# was chosen for the project. The different advantages offered by C#.NET technology and my craving for the technologies made me choose C#.NET.

2.2 LEARNING ASP.NET AND C# LANGUAGE

Learning and experimenting with new technologies and languages is of great interest to me. I wanted to take up the challenge of learning a new technology and then implementing it. Learning this new technology has taken quite some time for me. I have had a few difficulties finding a good resource for learning .NET. I have learnt .NET through many tutorials available online and also the complete reference book for .NET has helped me a lot. Since all the examples available online were very simple, I had to work hard to implement some of the features in the project.

2.3 SECURITY ISSUES

Installing the software necessary for project took some time for me. I couldn’t find the Visual Studio .NET 2003 CD and the Windows XP professional CD for a reasonable price online. I finally found the software that I needed from the CIS department. I also had some problems configuring the IIS server. A lot of research and hard work has helped me figure out the problem.
2.4 JMETER

JMeter installation and set-up was one of the problems I faced during the testing phase of the project. Initially I was able to set-up JMeter, but it kept crashing each and every time I put a heavy load on the server. So, I had to re-install it and then start the testing all over again.

3. SOURCE LINES OF CODE

The source line of code is a very important measure of the software project being developed. For my Airline Reservation System project, I have used a tool called the SLOC Metrics which counts the number of lines of code, based on the directory that we provide for search to the tool. We also need to indicate the types of files that the tool has to scan. So, based on the information provided by me to the tool, the following data was produced by the tool:

The numbers of lines of code in files with extension .cs are: 2310
The numbers of lines of code in the files with extension .aspx, which is the ASP.NET server page are: 1262
The numbers of lines of code in the files with extension .resx are: 962

Thus combining all these, the total lines of code would be: 4534

Thus nearly more than 50% of the coding consisted of C# coding. Most of the time spent for coding was for C#, since they are the files which have the actual logic to be implemented into the system. Nearly 28% of the coding was covered by the .aspx files, which are the ASP.NET server pages.

4. PROJECT DURATION

Initially I had planned to complete my MSE project by the end of the July 2008. But due to my health problems and also the availability of the committee members the project would be completed by the end of summer semester.

Initially I had estimated the effort required for the project as 4.56 staff months. In the project plan initially I had put in a total of 22 days for the Phase II of the project, but later on I had some coding problems and I had to extend the deadline for the Phase II of the project by 15 more days.

The following table would best depict the break down and the duration for each phase of the project:
Table 14 - Project BreakDown and Duration

<table>
<thead>
<tr>
<th>Phase</th>
<th>START TIME</th>
<th>FINISH TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase I</td>
<td>May 1\textsuperscript{st}, 2008</td>
<td>June 11\textsuperscript{th}, 2008</td>
</tr>
<tr>
<td>Phase II</td>
<td>June 12\textsuperscript{nd}, 2008</td>
<td>July 21\textsuperscript{st}, 2008</td>
</tr>
<tr>
<td>Phase III</td>
<td>July 21\textsuperscript{st}, 2008</td>
<td>Aug 12\textsuperscript{nd}, 2008</td>
</tr>
</tbody>
</table>

The breakdown of activities in each phase would also be presented in a table as follows:

Table 15 - : Activities and allotted time for all the Phases of MSE project

<table>
<thead>
<tr>
<th></th>
<th>Phase I hours</th>
<th>Phase II hours</th>
<th>Phase III hours</th>
<th>Total Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>25 hours</td>
<td>20 hours</td>
<td>40 hours</td>
<td>85 hours</td>
</tr>
<tr>
<td>Design</td>
<td>16 hours</td>
<td>22 hours</td>
<td>20 hours</td>
<td>58 hours</td>
</tr>
<tr>
<td>Coding</td>
<td>0 hours</td>
<td>80 hours</td>
<td>110 hours</td>
<td>190 hours</td>
</tr>
<tr>
<td>Testing</td>
<td>0 hours</td>
<td>0 hours</td>
<td>25 hours</td>
<td>25 hours</td>
</tr>
<tr>
<td>Documentation</td>
<td>20 hours</td>
<td>30 hours</td>
<td>30 hours</td>
<td>80 hours</td>
</tr>
<tr>
<td>Total Hours</td>
<td>61 hours</td>
<td>152 hours</td>
<td>225 hours</td>
<td>438 hours</td>
</tr>
</tbody>
</table>

The pie chart diagram showing the breakdown of each phase of the project would be as follows:

Figure 29 - figure showing the task breakdown for each phase
5. LESSONS LEARNT

My MSE project has been one of the rewarding experiences I have had. I have learnt a lot of things in the entire course of my Project. My desire to learn one of the fast evolving technologies like C# and .NET has been satisfied. I have learnt the various coding techniques in C# and .NET. In all the projects I have done till now, I have either taken up the front end or the back end responsibility. But, this is the first project where I have taken up both the front end and back end responsibility. I have experienced a situation similar to the real-time work environment, where programmers and developers work under pressure and a specified deadline. Being new to the technology I had some difficulties with the coding part initially. Going through the entire life-cycle of the software development has given me a lot of knowledge and experience which will be useful for my future. I have also improved my coding skills through this project.

The MSE project has also helped me realize that documentation for a project is as equally important as the coding of the project. Testing the Airline Reservation System project has helped me gain a lot of knowledge about the stress and load testing of the web applications. Even though I had a lot of problems initially getting JMeter to respond to a request, I was able to figure out the procedure with some research online. On the whole, I would like to thank my committee members for guiding
REFERENCES

The following references have been used by me, during all the phases of the MSE project:

1. http://www.w3schools.com/
2. www.msdn.microsoft.com
5. IEEE Recommended Practice for Software Requirements Specifications - IEEE Std 830-1998
6. SLOC Metrics Tool for .NET framework 1.1
   http://www.softpedia.com/get/Programming/Other-Programming-Files/SLOC-Metrics.shtml
13. Smart Draw software for the Gantt Chart