AIRLINE RESERVATION SYSTEM

NOOR NABIL IBRAHIM

UNIVERSITI TEKNOLOGI MALAYSIA

PSZ 19:16 (Pind. 1/13)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author’s full name : Noor Nabil Ibrahim

Date of Birth : 22 January 2001

Title : Airline Reservation System

Academic Session : Sem 5

I declare that this thesis is classified as:

|  |  |  |
| --- | --- | --- |
| ✓ | CONFIDENTIAL | (Contains confidential information under the Official Secret Act 1972)\* |
|  |  |  |
|  | RESTRICTED | (Contains restricted information as specified by the organization where research was done)\* |
|  |  |  |
| ✓ | OPEN ACCESS | I agree that my thesis to be published as online open access (full text) |

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Certified by: |  |
|  |  |  |  |  |
|  | SIGNATURE OF STUDENT |  | SIGNATURE OF SUPERVISOR |  |
|  |  |  |  |  |
|  | QU180136 |  | DR. Alaa Khalil Jumaa |  |
|  | **MATRIX NUMBER** |  | **NAME OF SUPERVISOR** |  |
|  |  |  |  |  |
|  | Date: 21 Oct 2021 |  | Date: 21 Oct 2021 |  |

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“ I hereby declare that we have read this thesis and in my

opinion this thesis is suffcient in term of scope and quality for the

award of the degree of Bachelor of Computer Science (Software Engineering)”



|  |  |  |  |
| --- | --- | --- | --- |
| Signature | : | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Name of Supervisor | : | DR. Alaa Khalil Jumaa |  |
| Date | : | 22 June 2022 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

ON-LINE RECOGNITION OF DEVELOPING CONTROL CHART PATTERNS

NOOR NABIL IBRAHIM

A thesis submitted in fulfilment of the

requirements for the award of the degree of

Bachelor of Computer Science (Software Engineering)

School of Computing

Faculty of Engineering

Universiti Teknologi Malaysia

June 2022

DECLARATION

I declare that this thesis entitled *“Airline Reservation System”* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

|  |  |  |
| --- | --- | --- |
| Signature | : | ...... |
| Name | : | Noor Nabil Ibrahim |
| Date | : | 22 June 2022 |
|  |  |  |
|  |  |  |

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

During the preparation of this thesis, I was in contact with many people, researchers, academics, and professionals. they have added to my understanding and thoughts. In particular, I would like to thank my main supervisor for the doctoral thesis, assistant professor Dr. Alaa, and express my sincere gratitude for his encouragement, guidance, criticism, and friendship. I am also very grateful to my co-director Professor Dr.Adawiya for her guidance, advice, and motivation. Without their continued support and interest, this work would not have been the same as that presented here.

My fellow student should be recognized for/her support too. My sincere appreciation also extends to all my colleagues and others who have provided assistance on various occasions. They're very helpful indeed. It would be impossible to list all of them here. My family has been extremely supportive of me throughout my life and I am so thankful for their support

ABSTRACT

Many people are travelling by airplanes, either as daily transportation to and from work or when going on vacation. To make reservations for such travels, airline companies' websites hold the functionality for the user to book travel himself. A functionality that these websites lack is the option for the user to set up specific requirements for travel, such as; minimal travel time or travel distance. The purpose of this project is to develop an easy-to-use airline reservation system, which accommodates these functionalities. In addition, the system should also be of use for travel agencies within that they don’t have to visit the agency offices as well. So this project is to provide a user-friendly, efficient, and easy-to-use Online Reservation System that will help the airline agency to manage their flights in an efficient way and at the same time help the travellers to look for the right flight and to see if the time and the duration are available or not at the exact time they want also to check about all information that they need to know and keep records of various passengers and provide prompt services to the passengers. The methodology that has been implemented is Prototyping

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| **TITLE** | **PAGE** |

[DECLARATION iv](#_Toc2850685)

[DEDICATION v](#_Toc2850686)

[ACKNOWLEDGEMENT vi](#_Toc2850687)

[ABSTRACT vii](#_Toc2850688)

[TABLE OF CONTENTS Error! Bookmark not defined.](#_Toc2850690)i

[LIST OF TABLES xii](#_Toc2850691)

[LIST OF FIGURES 1](#_Toc2850692)

[LIST OF ABBREVIATIONS xiv](#_Toc2850693)

|  |  |  |
| --- | --- | --- |
| **CHAPTER 1** | **INTRODUCTION** | **1** |
| 1.1 | Introduction | 1 |
| 1.2 | Problem Background | 2 |
| 1.3 | Project Aim | 2 |
| 1.4 | Project Objectives | 3 |
| 1.5 | Project Scopes | 3 |
| 1.6 | Project Importance | 4 |
|  |  |  |
| **CHAPTER 2** | **LITERATURE REVIEW** | **5** |
| 2.1 | Introduction | 5 |
| 2.2 | Inter-organization case study | 5 |
| 2.3 | Current system analysis | 6 |
| 2.3.1 | Existing System 1 | 6 |
| 2.3.2 | Existing System 2 | 6 |
| 2.3.3 | Existing System 3 | 7 |
| 2.3.4 | Existing System 4 | 8 |

|  |  |  |
| --- | --- | --- |
| 2.3.5 | Existing System 5 | 8 |

|  |  |  |
| --- | --- | --- |
| 2.4 | Comparison between Existing Systems | 9 |
| 2.5 | Technology Used on Literature Review | 11 |
| 2.6 | Chapter Summary | 13 |
|  |  |  |
| **CHAPTER 3** | **SYSTEM DEVELOPMENT METHODOLOGY** | **13** |
| 3.1 | Introduction | 14 |
| 3.2 | Methodology Choice and Justification | 14 |
|  | 3.2.1 Waterfall Methodology | 15 |
|  | 3.2.2 Agile Methodology | 15 |
|  | 3.2.3 Prototyping Methodology | 16 |
|  | 3.2.4 Comparison between the Methodologie | 16 |
|  |  |  |
| 3.3 | Methodology Phases | 17 |
|  | 3.3.1 Requirements gathering and analysis | 17 |
|  | 3.3.1.1 Information Interview | 18 |
|  | 3.3.1.2 Internet | 18 |
|  | 3.3.2 Quick Design | 18 |
|  | 3.3.2.1 Software Design | 19 |
|  | 3.3.3 Phase 3: Build a Prototype | 19 |
|  | 3.3.4 Phase 4: Initial User Evaluation | 19 |
|  | 3.3.5 Phase 5: Refining Prototype | 19 |
|  | 3.3.6 Phase 6: Implement Product and Maintain | 19 |
|  |  |  |
|  |  |  |
|  | 3.3.7 Information Gathering | 20 |

|  |  |  |
| --- | --- | --- |
|  | 3.3.7 Information Gathering | 21 |
| 3.4 | Technology Used | 24 |
| 3.5 | System Requirement Analysis | 26 |
|  | 3.5.1 Hardware Requirement | 26 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.5.2 | | Software | Requirement | 26 |
| 3.6 | Chapter Summary | |  | 27 |
|  |  | |  |  |
| **CHAPTER 4** | **REQUIREMENT ANALYSIS AND DESIGN** | | | **28** |
| 4.1 | Introduction | | | 28 |
| 4.2 | Requirement Analysis | | | 28 |
|  | 4.2.1 Use Case Model | | | 29 |
|  | 4.2.1.1 Use Case Description | | | 30 |
|  | 4.2,2 Activity Diagram | | | 31 |
|  | 4.2.3 Sequence Diagram | | | 33 |
| 4.3 | Project Design | | | 34 |
|  | 4.3.1 UML Class Diagram | | | 35 |
| 4.4 | Database Design | | | 36 |
| 4.5 | Interface Design | | | 37 |

|  |  |  |
| --- | --- | --- |
| 4.6 | Chapter Summery | 43 |
| **CHAPTER 5** | **CONCLUSION** | **44** |
| 5.1 | Introduction | 44 |
| 5.2 | Project Development | 44 |
| 5.3 | Achievement of Project Objective | 44 |
| 5.4 | Suggestions Plan for Project Implementation (PSM2) | 45 |

**Appendix A**  Gantt Chart PSM I 47

**Appendix B** Software Requirement Specification 51

**Appendix C** Software Design Document 80

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NO. | TITLE | PAGE |

Table 2.1 Hardware requirements 9

Table 2.2 Comparation between existing works 9

Table 3.1 The Comparation between the Methodologies Table 17

Table 3.2 Interview with Employees 23

Table 3.3 Minimum Hardware Requirement 26

Table 3.4 Minimum Software Requirement 27

Table 4.1 Actor Description Table 30

Table 4.2 Use Case Table 30

Table 4.3 Database Table of Customer 37

Table 4.4 Database Table of Booking 37

Table 4.5 Database Table of Flight 38

Table 4.6 Database Table of Airport 38

Table 4.7 Database Table off Book Flight 38

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO. | TITLE | PAGE |

Figure 2.1 Apache web server example 13

Figure 3.1 Prototyping model phases 18

Figure 3.2 Information Gathering 22

Figure 3.3 XAMPP control panel 24

Figure 4.1 Use Case Diagram of Airline Reservation System 29

Figure 4.2 Activity Diagram of Book Ticket 29

Figure 4.3 Sequence Diagram of Search Ticket 30

Figure 4.4 Architecture for Airline Reservation System 32

Figure 4.5 Class Diagram UML for Airline Reservation System 33

Figure 4.6 ERD for Airline Reservation System 36

Figure 4.7 Welcome Interface of Airline Reservation System 39

Figure 4.8 Login Interface for Employee 40

Figure 4.9 Interface for Employee actions Inside the System 40

Figure 4.10 Interface for Employee Managin Booking and Cancellatio 41

Figure 4.11 Login Interface for Customer 41

Figure 4.12 Home Page Interface for Customer 42

Figure 4.13 Interface for Review Customer Booking List 42

Figure 4.14 Interface for Customer to Book Flight 43

Figure A.1 Gantt Chart for PSM I 42

LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| ARS | - | Airline Reservation System |
| ERD | - | Entity Relationship Diagram |
| SDD | - | Software Design Documentation |
| MVC | - | Model View Controller |
| SRS | - | Software Requirements Specification |

**CHAPTER 1  
  
  
INTRODUCTION**

**1.1 Introduction**

With the rapid improvement in computer technology, software program initiatives have become increasingly larger and more complex. Software professionals have recently attempted to adopt a more systematic and formal approach to the design, development, and implementation of their software. This new approach has become necessary because traditional methods of systems development often produced software characterized by late diversity, cost, unreliability, lack of maintainability, and non-usability. Considering the possibility, or likely creativity, of making such a task different and unique, we thought about how to develop the software. This project aims to support and computerize flight reservations. Keep in mind that users will find a simpler and more user-friendly interface to carry out their tasks. The software will be so easy to use that anyone with no computer experience can easily use it. Currently (in Kurdistan), booking flight tickets are done manually with the travellers walking into the airline offices and asking for available flight tickets and just booking them without knowing if there are other flights from another origin, Also Booking a flight online gives you the freedom to save at a great price. Travelers don't get that kind of freedom by booking flights through a single travel agent or by calling an airline. So, you have to take a look at several airlines before making your decision.

The main purpose of the flight reservation system is to help guests to bespeak flight tickets online. Airline reservation systems are used to being stand-alone systems. Every airline had its own system, is separate from other airlines, and is only used by a specific group of airline workers. trip agents supported access to airline systems in the 1970s. (CRS) accessible to multiple airlines and trip agencies the airline reservation system aims to increase delicacy and improve security and communication, and performance by transubstantiating the prevailing homemade system into the perpetration of motorized instruments. Currently (in Kurdistan), the management and booking of flight tickets are done manually with the passenger walking into the airline offices and asking for available flights and just booking them without knowing if there’s another better offer for them or a better flight from another airline, and for the staff, it’s hard to keep the information of all those customers in his mind and it’s hard to know how many seats left in the plane also might some mistakes happen and give the wrong time into the passenger. Unfortunately, this method is very slow and has many flows which could ruin the travel of the passengers and staff jobs in the airline’s offices. An efficient and better way is proposed the ARS (Airline Reservation System). The purpose of an online airline reservation system is to avoid any mistakes and make it easier to book a flight ticket, and efficient in the airline office by automating the existing manual system through the implementation of computerized equipment

**1.2 Problem Background**

Many people are travelling by airplanes, either as daily transportation to and from work or when going on vacation. To make reservations for such travels, airline companies' websites hold the functionality for the user to book travel himself. A functionality that these websites lack is the option for the user to set up specific requirements for travel, such as; minimal travel time or travel distance. The purpose of this project is to develop an easy-to-use airline reservation system, which accommodates these functionalities. In addition, the system should also be of use for travel agencies within that they don’t have to visit the agency offices only for receiving the money because the system dose not provide an online payment methods so the customer after book their flight through the system then for the last step the go to the airline office to do the payment.

**1.3 Project Aim**

The main aim of this project is to develop a user-friendly, efficient, and easy-to-use Online Reservation System that will help the airline agency to manage their flights in an efficient way and at the same time help the travelers to look for the right flight and to see if the time and the duration are available or not at the exact time they want also to check about all information that they need to know and keep records of various passengers and provide prompt services to the passengers.

**1.4 Project Objectives**

The objective of this project is the development of web-based software for an effective flight reservation system. The following goals are achieved with this system:

1. Develop Airline Reservation System base on the requirement analysis and the information gathering.
2. Choosing the right Methodology for the system and design an interface for the system.
3. Define the technology tool that are used to develop this system and the software/Hardware requirements.
4. To ensure the availability of flights and maintain the database of available options to update and ensure the fast searching, deleting, and updating of booking and manage the information of flights and airlines ticket.
5. Test the Airline Reservation System using one of the tools.

**1.5 Project Scope**

The scopes of the project are:

We will start the project by contacting some of the Travel Agency offices directly or by preparing an electronic questionnaire, this will help in collecting the main requirements and needs of those offices, after that, we need to analyse those requirements and identify the challenges and obstacles that may encounter it during the project design and implementation process.

The project will be developed using PHP, MYSQL, CSS, HTM, JavaScript, Bootstrap, J Query, and of course database to build the website and Apache web server.

**1.6 Project Importance**

The importance of this project comes from a personal experience, 3 years ago I went on vacation to one of the neighboring countries, the purpose of my speech about my trip is that I had some difficulties in finding the right flight for me and had many choices of airlines, which I couldn't easily tell which ones because I had to visit the airline offices to find the right flight for me at the right price and time for me, the problem was that I couldn't visit offices. I was busy with my work and couldn't find time to go to the offices. I called and booked my travel ticket and someone else received it for me. In this situation, I would like to be able to see flights, discounts, and all the information myself through a website or application that is available to me anywhere and anytime.

**CHAPTER 2  
  
  
INTRODUCTION**

* 1. **Introduction**

Due to airlines' interest in customer convenience, automation of airline management services cannot be ignored. The Airline Reservation System is a complete submission workflow management system designed to improve accuracy and efficiency. Most airlines still do their work manually. This manual system requires employees or workers to manually monitor all processes and verify the existence of all data in the office. This chapter presents current systems and recent work used to solve problems in airlines and management offices. explained and commented. In addition, we identify the weaknesses and strengths of this work. Additionally, the concepts, tools, and techniques used in this project are explained succinctly and key points are highlighted.

* 1. **Case Study (If any)**

This section discusses the techniques and systems used today in airline offices in our region. Most offices still use paper-based accounting methods or data storage software, e.g. Microsoft Excel. In order to manage the data in the airline office, the staff needs to check the data from time to time and save some information, and that is difficult for the staff because the data access in the file processing system is not convenient and efficient. For security reasons, the database must restrict access for users. Each user should only be able to access the data relevant to them. Again, this method is paper-based as staff write down the amount of each reservation and check for availability each time. It's still there. Both the paper-based method and the simple computer-aided method have disadvantages, although the latter is better than the former, it doesn't solve the whole problem but part of the problem.

**2.3 Current System Analysis**

This chapter will discuss the available systems that have been researched or developed in the past, the discussion will include the item's name and a simple explanation for each including its weaknesses and strengths, and at the end, I will compare them:

**2.3.1 Airline Reservation System [1}**

This airline reservation system is an implementation of a general airline ticket website that helps customers to search for the availability and prices of various airline tickets as well as various packages available at the time of booking. Airline Reservation System is a web application project for which C# was chosen and SQL Server acts as the database for the project. This project also includes various functions such as online user registration, admin staff, or website administrators can change

website details, and add, delete or change customer details, flights, or information about packages. In general, this website is designed to work like any other airline. Ticketing websites are available online but using this website is not easy, it has security issues and the system GUI platform is weak as it is developed with C# for the same flexibility.

**2.3.2 Airline Reservation System [2]**

In this system, we have the possibility to add customer data from the ticket manager using predefined formats and drop-down lists. Ticket manager overhead was reduced with this system. Easy access to customer data was another advantage. The ticket may be removed if deemed unnecessary. The system mainly deals with customers who book and cancel flights with airlines. This ticket system allows customers to reschedule flights as well. This project provides all the necessary features to complete the task. Java was used to develop the entire component.

**The advantages of the system:**

1. User-friendly interface
2. Faster database access
3. Fewer errors
4. More storage capacity
5. Environmental look and feel
6. Fast transaction

**2.3.3 A Distributed Airline Reservation System [3]**

The system is designed to meet and achieve the goals set for the airlines in Nigeria. This is possible thanks to overcoming all the problems related to the old operating system. The outcome of this study will be a basis for developing the appropriate approach to solving problems related to flight operations related to the Airline Flight Information System (AFIS). The system was designed using the Structured System Design and Analysis Methodology (SSADM) and developed using PHP, JavaScript, and HTML as programming languages, while the database was developed with MySQL.

**The system works as follows:**

* The customer must register in order to preserve the ticket service.
* The customer must enter all required data during the registration process.
* After successful registration, the customer can proceed to book the flight.

**The primary weakness of the (SSADM) system used in the system is that it is**:

Time-consuming. The lengthy analysis required by this approach can delay the delivery of the desired information system by several months.

**2.3.4 Online Airline Ticketing System [4]**

Users can easily buy an electronic ticket by logging into the ticketing website, searching and selecting the travel destination, entering details such as name, travel type, baggage information, and dates, and finally paying by bank card, bank transfer, or online. payment company. The electronic ticket is then sent to the customer's phone via email or SMS. With the advent of the Internet and airline dist., The database is integrated into the airline's passenger service system, which allows the airline to share real-time flight information with airports and travel agencies, which has been upgraded. Various software and programs were used to develop the exemplary online reservation system.

The programs used in the project included Python, Eclipse, JavaScript, HTML, and SQLite. All programs required installation, while basic understanding of programming was necessary to complete the project.

**2.3.5 Online Flight Reservation System [5]**

The project was developed to fulfill these purposes. It claims to check all available airline databases and return a set of results that can help you with your travel plans. The goal of this project is to create a reservation system for airline flights that allows users to obtain all flight information in one location, including prices and times. When the customer calls the counter attendant for his travel needs, the counter attendant enters the customer details (flight requirements) into the system. The system displays all available airlines, flight schedules, and prices. This system would help the airline meet its customers' needs better. The project was developed to fulfill these purposes. The goal of this project is to create an online flight reservation system that allows customers to request information about available flights, costs, and schedules in one place. The website would store information in a database, including fare and availability data.

**Hardware Configuration:**

2.1 Hardware requirements

|  |  |
| --- | --- |
| **Hardware** | **Requirement** |
| Processor | Pentium 2 |
| Hard disk | 4 GB |
| RAM | 64 MB |
| Dot matrix printer | 16 Pin |

**Software Configuration: -**

1. ORACLE as Back End.
2. Visual Basic 6.0 as Front End.

**2.4 Comparison between existing systems**

Table 2.2 Comparation between existing works.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Achieved Objectives | Drawbacks | Type | Tools Used |  | Features |
| 1 | Customers search for and compare the prices of different airline tickets. | The system is not easy to use. And Weak platform GUI. | Website | C#/.NET  SQL |  | Login and register, Book flight, Book Mote, book a package, Contact the Company, Booking Instructions. |
| 2 | The system mainly deals with customers who book and cancel flights with airlines. In this ticket system, customers can even postpone their flights. | Java is slow and has a poor performance, memory-consuming also poor GUI. | Software | Java 2 Runtime Environment |  | Sign in, Book flight, cancel flight, Postpone flight, Exit. |
| 3 | The system is able to handle the transfer of airline tickets from one user to another ahead of flight time and the system offers users the ability to reschedule them  flights and allows them to have more control over their flight time. | The SSADM methodology is time-consuming. There is a long delay between the beginning of a project and the delivery of a system. | Website | PHP  HTML  SQL  (SSADM) |  | Search flight, view customer details, view available flight, View customer information, Edit flight information, login to the system. |
| 4 | Users buy tickets from the website and then fill in their personal details and book flights, accommodation and transfers by making a payment. | Security issues | Web Application | Python  HTML  SQL  JavaScript |  | Flight search, Contact page with feedback form, booking the selected flight. |
| 5 | The goal of this project is to create a flight reservation system that allows customers to request flight information based on their travel dates. The counter attendant enters customer details into the system when he or she assists the customer. | The system is subject to overloading, but only when there are a lot of users on the site. | Web Application | Oracle  HTML  SQL (DBMS)  Microsoft visual basic |  | LOGIN: add user, delete user, change password. MASTER: shows the flight information by branch, service, about us, airbus, fare, flight schedule, control, route, exit. TRANSACTION. |

**2.5** **Literature Review of Technology Used**

This section will discuss the technologies used in this project from which database, programming languages, and web server, and we clarify why those were chosen [6].

1. Database management system: MySQL server, a database management system & DBMS was chosen as the database server in this project because it offers extensive support for any application development need, it is flexible and scalable, it is highly available which means it is used by many people, which means 24/7 uptime and support when needed, strong Data security, easy to manage. Using MYSQL in this project is to create spreadsheets that store data like flight planning, crew planning, workforce planning, pricing, revenue management, passenger reservations, online reservations.
2. Programming Languages:The project will use PHP for its backend connection, as PHP is a popular and versatile programming language. One of the language's main advantages is that it is platform-independent, running on Mac operating systems, Windows, and Linux. It is compatible with most web browsers. It is also compatible with all major web hosts, making it easy to deploy on different systems and platforms with minimal additional costs. HTML and CSS are used to build web pages while JavaScript is used to add functionality. Web Server: Apache web server is used in this project because Apache is open-source software that gives its users permission to publish their website on the Internet. The Apache web server is one of the oldest, most widely used, and most trusted web servers. Apache serves as a bridge between servers and browsers. Since the structure of the program is client-server, data is constantly being transferred.

The Apache webserver was chosen for this project for the following reasons:

1. Free and open source, also for commercial use.
2. It is fast, reliable, and highly secure.
3. Flexible due to modular design.
4. Easy to configure, easy to use for beginners.
5. Optimal deliverability for static files and compatibility with every programming language.
6. Large community and support immediately available in case of problems.

While it has many advantages, it also has some disadvantages, which are:

* Performance issues on extremely large websites
* Too many configuration options can lead to security vulnerabilities

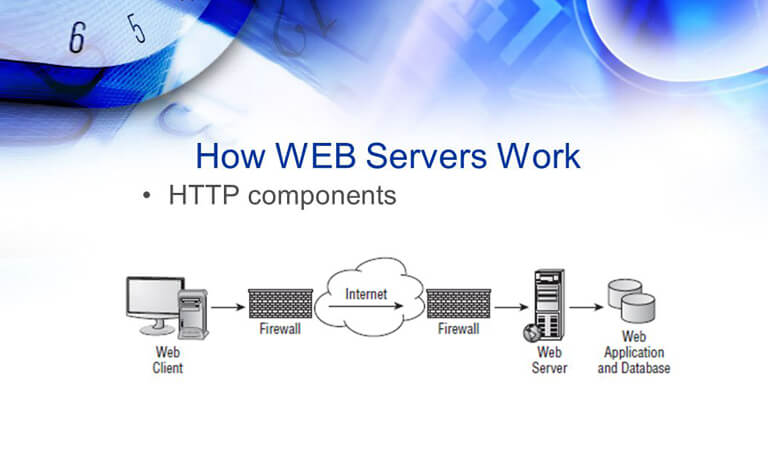


Figure 2.1: Apache Web Server example

**2.6 Chapter Summary**

In this chapter, we discussed the literature review which included many sections which include:

* Introduction: In this part, we talked in general about what we will do in this chapter and what we are trying to design.
* Inter-Organization (Case Study): Here we gave an explanation of the traditional way of the Airline systems and management view and discussed why is not efficient and shouldn’t be used anymore in our society.
* Current system analysis: In this section, the current or recent systems available besides the traditional way those systems can be found in the local and international airline systems are explained, and their advantages and drawbacks are identified as well.
* Compression between current systems: In this section, we made a comparison among the recent systems in terms of advantages, drawbacks, type, and tools.
* Technologies Used: In this section, we talked about the tools that are used in the project (MySQL server, PHP, HTML …, Apache Web Server) and why we used them.

**CHAPTER 3  
  
  
SYSTEM DEVELOPMENT METHODOLOGY**

* 1. **Introduction**

A system development methodology is a process of structuring, planning, and controlling the development of an information system. The acceptable specified technique has a number of advantages, including better flexibility. An effective risk identification system is one that involves the implementation of a security system and a clear understanding of the task at hand. Agile methods, iterative development cycles, spiral development, and waterfall models are types of effective approaches.

Choosing the appropriate methodology is crucial. Alternatively, as time passes, a growing number of worries emerge. As an illustration. The created systems do not satisfy the users' expectations, causing project delays and occasionally cost overruns. As a result, adhering to the process reduces these occurrences.

* 1. **Methodology Choice and Justification**

Software development is all about using the right methodology. Today, other professional software development firms use a variety of approaches. Each approach has advantages and disadvantages that must be considered when choosing which one to use for Airline Reservation System.

* + 1. **Waterfall Model**

A waterfall model is a software develop­ment method that may be used to develop software. Usually, the software development process flows linearly through a series of steps, each step depending on the completion of the preceding one. As a result, after a phase is completed, this approach does not allow for a return to a prior phase. Its flexibility has been challenged by advocates for the flexible model.

Waterfall models have several advantages. The orderly sequence of phases, including strict controls for reviewing documentation and design documents improves the maintainability, quality and reliability of the developed software. It's also suitable for inexperienced managers who don't know much about their teams. There are some disadvantages associated with this approach and changing the method is difficult. If you're going to pay for an adjustment, you might as well get it done sooner rather than later.

**3.2.2 Agile Methodology**

You can use agile methodologies to alter your reaction to unexpected situations that arise during the course of a project. An example of an Agile methodology is Scrum. Scrum works best for complex and innovative projects Scrum development begins whenever there is a need for improvement. It continues until all the improvements have been made. It's an ideal way of tackling projects because it quickly gets even the most sluggish project moving again.

Scrum has a number of advantages for example, it is possible to break down a large and scalable project into manageable jobs. This technique also recognizes when a change needs to be made to the project's direction, or field because each sprint is reviewed at the end of each sprint. Despite the fact that Scrum adoption may be used for nearly every type of project, it is most commonly associated with software development projects. in large teams is difficult because the methodology can only be successful with experienced team members.

**3.2.3 Prototyping Software Development Method**

Prototyping allows developers to create a mere mock-up of a solution so they may show it to their users before developing the actual solution. Prototyping techniques allow users to be involved in the development process by creating prototypes that they can use.[7]

Prototype methods have several advantages including that they can be used for realistic modelling of critical components of a system throughout its entire lifecycle. Involvement refers to the degree to which the user is involved in the design and development of the system. It helps promote user participation in system development. Because the user needs to be involved in resolving unclear goals and verifying user requirements, this is why we need to involve them. This technique can be useful for demonstrating the functionality of a system, but it may also result in false assumptions about the system itself. This is because clients assume that the system is completed when it is not. The system doesn't appear to be good in terms of the user interface, but its functionality is lacking.[7]

**3.2.4 Comparison between the Methodologies**

System development methods are important components of software development. Prototyping was chosen to develop an airline reservation system. A throwaway prototype is the best option since its purpose is to gain a better understanding of the requirements. This prototype was created according to the current requirements derived from the problem statement, in order to satisfy its users.

* 1. **Phases of the Chosen Methodology**

Table 3.1 The Comparison between the Methodologies Table

|  |  |
| --- | --- |
| Waterfall Method  The good and bad aspects | Under firm control. A good fit for people who aren't familiar with the technology. It's not possible to change something that has already happened. Reduce manageability because fewer iterations |
| Agile Method  The good and bad aspects | Large projects can be divided into smaller ones which can then be further divided into even smaller tasks. Reviews are done at every sprint. Need experienced developers. It is important to have non-stop reviews and meetings. And it requires solid resources. |
| Prototype Method  The good and bad aspects | Can show you the prototype. To avoid failure, encourage people to take risks. It asks for too many changes at once. |

There are several phases in Prototype. The methodology phase starts with determine objectives, developing the prototype, show the prototype to the user for the evaluation process, refining the prototype after getting the feedback from users, and proceed to the last phase which implements the proposed system and maintain it. Figure 3.1 Prototyping Model Phases shows the step to be done when implement Prototype methodology.



Figure 3.1 Prototyping Model Phases

**3.3.1 Phase 1: Requirements gathering and analysis**

The needs of the system are defined in depth during requirement analysis, which is the first step in the prototyping process. During the process, company administrators were questioned to learn what they expected from the system. There are various methods that have been used, such as the internet.

**3.3.1.1 Informational Interview**

The purpose of the interview was to gather the requirements directly from the administrators to understand the needs better. During the interview, the users told the manual operation they usually would do, problems arise during the process, the weakness of the manual operation, and their expectation of the proposed system.

**3.3.1.2 Internet**

The goal of searching the internet is to gain a general understanding and knowledge of the Airline Reservation System.

**3.3.2 Phase 2: Quick Design**

The second phase is a rapid design, in which the system's basic design is created.

**3.3.2.1 Software Design**

Together with the corporate managers, we came up with the idea of transforming the requirements into software and recommending the finest tools for achieving a fantastic outcome, such as programming languages, libraries, frameworks, and databases. PHP was selected as the cire programming language, while JavaScript was mainly programming language for the discussion. MySQL was used as a database service with the implementation of the phpMyAdmin tool.

**3.3.3 Phase 3: Build a Prototype**

In this stage, a prototype has been built by writing simple code based on the requirements gathered from phase 1. This is early version only and was shown to the users for them to give their comments and current progress of Airline Reservation System.

**3.3.4 Phase 4: Initial User Evaluation**

After building the small prototype, a user study was conducted to evaluate the proposed system. Early discovery of the strength and weaknesses of a functional prototype helps you to lower the likelihood that the prototype will fail.

**3.3.5 Phase 5: Refining Prototype**

If users are unsatisfied with the current prototype, it will be improved based on their feedback and suggestions to ensure that the system meets their needs. Following the approval of the refined prototype by the users committee, another feature was added, and the procedure returned to the previous phase, which included the evaluation. This phase didn't end until all of the users' requirements were met.

**3.3.6 Phase 6: Implement Product and Maintain**

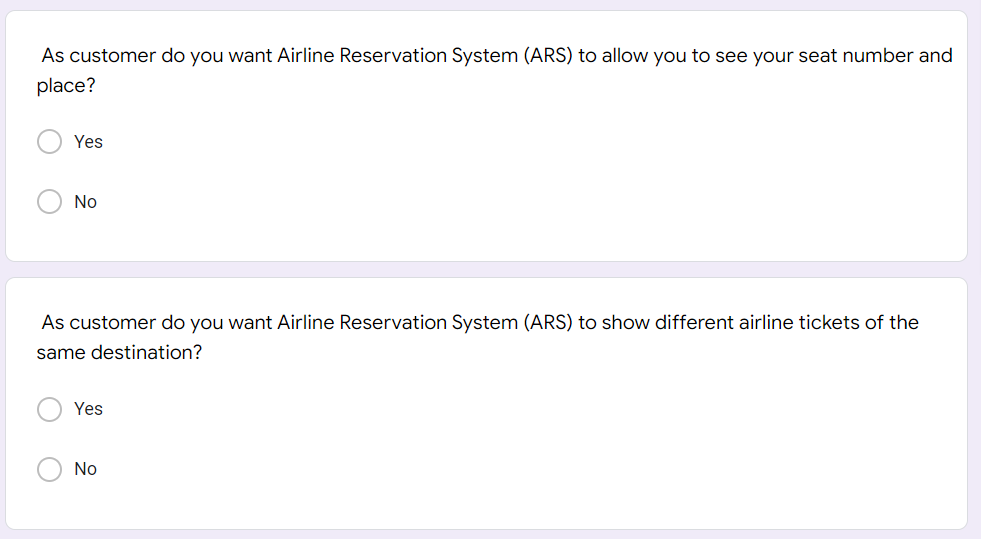
After users approved the authorized version of the final prototype, a final version was created. It has been thoroughly tested and is now ready for use. Users who have used the system had the opportunity to provide feedback. In addition, the system is regularly maintained. so, you don't experience major failures and don't fail in large numbers.

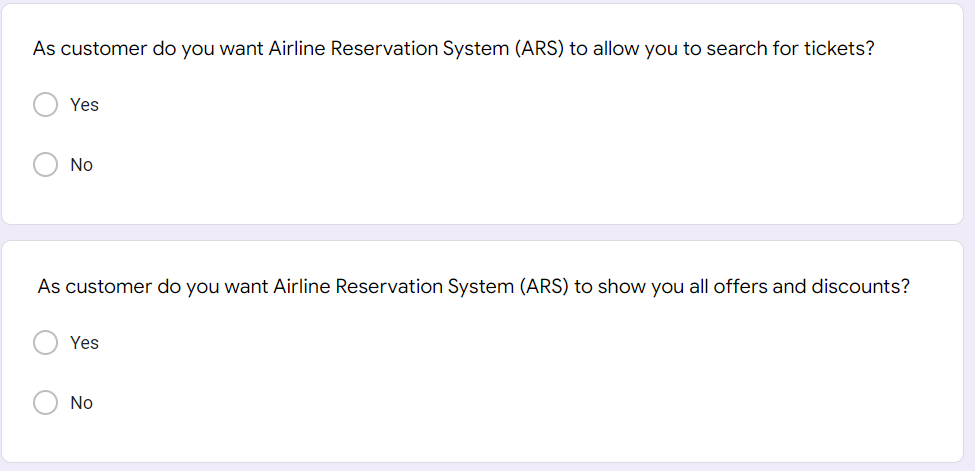
**3.3.7 Information Gathering**

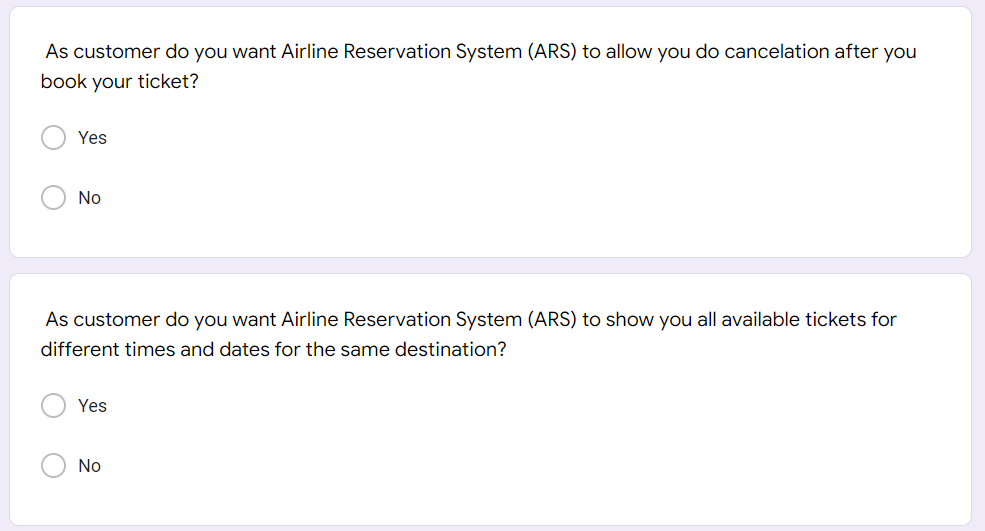
**We have two stakeholders to gather information from:**

**1- Customers**

According to the existing system that have been mention in chapter 2, we conducted a survey for our users to get information about what the user feels about ARS.







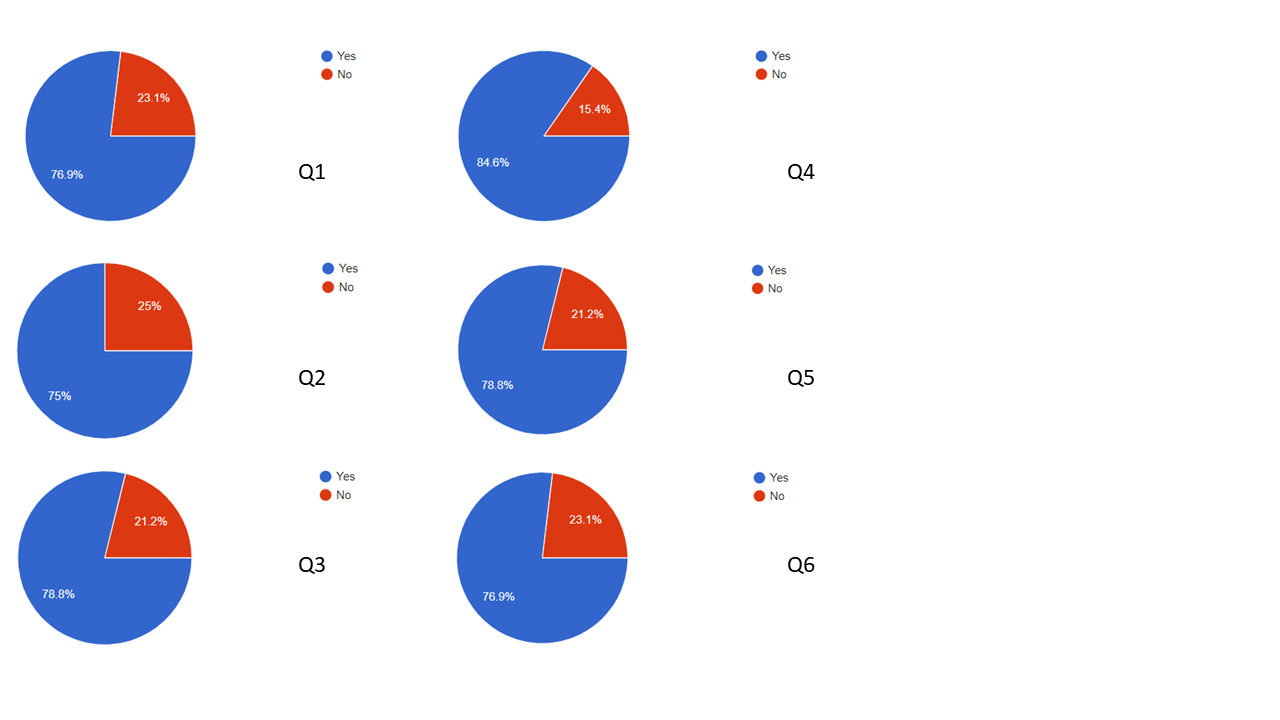
****

Figure 3.2 Information Gathering

**As stated in the result of the survey:**

* 76% of customers want (ARS) to allow them to see seat number and their place.
* 75% of customers want (ARS) to show different airline tickets of the same destination.
* 78% of customers want (ARS) to allow them to search for tickets.
* 84% of customers want (ARS) to show them all offers and discounts.
* 78% of customers want (ARS) to allow them to do cancelation after book their ticket
* 76% of customers want (ARS) to show them all available tickets for different times and dates for the same destination.

1. Employees

By making an interview with the airline agency employees I gathered some information from them which help me to develop my system:

Table 3.2 Interview with Employees

|  |  |
| --- | --- |
|  | The Dialogue |
| INTERVIEWER | Hello, could you help me to gather some information? I will ask you a few questions If you don’t mind. |
| EMPLOYEE 1 | Hi, Yes Sure go ahead. |
| INTERVIEWER | Alright, so what are the difficulties that you face as an employee working at an airline agency? |
| EMPLOYEE 1 | The load on the place especially on holidays makes me nervous and makes me lose control sometimes. |
| INTERVIEWER | Okay, so at this point what you would like to change at the place you work? |
| EMPLOYEE 1 | I prefer using an application or a website to manage the number of people I think that would be fun and less stressful than dealing with people face to face. |
| INTERVIEWER | Alright, Thank You so much for collaborating. |
| EMPLOYEE 1 | You are welcome. |
|  |  |
| INTERVIEWER | Hi Sir may I ask you some questions and Information on travel time from various airlines? |
| EMPLOYEE 2 | Oh Hello, Of Course |
| INTERVIEWER | How do you manage Information on travel time from various airlines? |
| EMPLOYEE 2 | I do split the information and time and also the Airlines type into an Excel table so whenever a customer comes to ask about the information I go search through the table and give the information they need. |
| INTERVIEWER | Do you think this way to store information is easy to manage and is it secure? |
| EMPLOYEE 2 | Not at all it is not secure neither easy to manage |
| INTERVIEWER | what is the reason? |
| EMPLOYEE 2 | Because it takes time to look for while the customers sitting there waiting to receive the information from us also is not secured because everyone can access the excel file while I am not there and sometimes, we lose information somehow. |
| INTERVIEWER | What would you rather use to store data instead of an excel table? |
| EMPLOYEE 2 | Using an application or a website would satisfy us as employees and customers as well. |
| INTERVIEWER | Nice, thank you so much for letting me do an interview with you. |
| EMPLOYEE 2 | No worries, I'm glad to help. |

## **Technology used**

For this project, I'll use the XAMMP tool. XAMPP is an open source cross-platform web server that includes MySQL, Apache, FileZilla, Mercury and Tomcat bundled up into one package. Allows creating websites locally on a PC. Apache and MySQL are the two essential components of XAMPP. Figure 3.4 represents the XAMPP control Panel. [8]

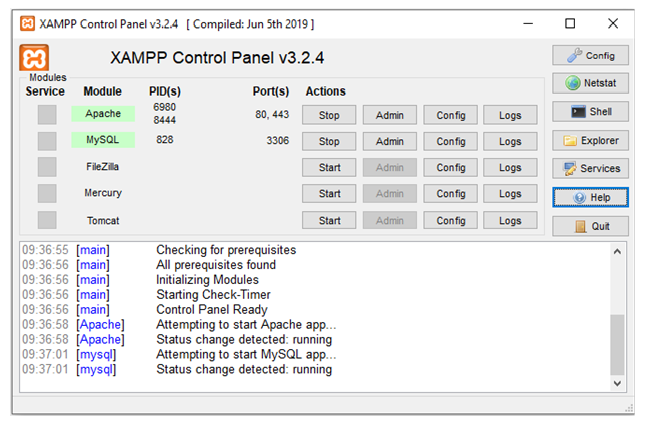


Figure 3.3: XAMPP Control Panel

There are many varieties of languages that are powerful to build the proposed system such as PHP, Bootstrap and JavaScript. Below is a short description of languages that will be using to develop EMP.

Visual Studio Code: It features a lighting fast source code editor, it’s a perfect editor which supports hundreds of languages, and it helps with instantly productivity with syntax highlighting, racket-matching, auto-indentation, box-selection, snippets and etc.

PHP: It is a free open-source programming language used for developing websites. It was one of the first server-side languages to be embedded into HTML, allowing it to easily add functionality to websites by accessing external files. was one of the first server-side programming languages that could be embedded into web pages, making it easy for developers to create dynamic sites. for web developers to add features to their websites without having to call up external files.

XAMPP: It’s a general-purpose scripting language that may be useful for creating interactive and dynamic websites. It was one of the first server-side languages that made it easy to add functionality directly to web pages without having to access other file types.

JavaScript: With JavaScript, you can use different built-in functions (such as loops), DOM access, etc., the code has reduced overhead by making it faster. SAAS: There could be several reasons why you might save a significant sum of money. One of the reasons why you should consider using a SaaS solution is because it removes both the initial cost of purchasing and installing the software, as well as ongoing expenses such as maintenance and upgrades (if needed). Software as a service (SaaS) application can be readily downloaded from the internet and maintained instead of having to spend a lot of money on installing hardware.

Bootstrap: is an open-source HTML, CSS, and JavaScript library for creating informative websites with a wealth of built-in features and a robust grid layout system (CSS Grid Layout) it's easier than ever to lay things out. With a variety of components and a robust grid system, CSS Grid Layout allows you to easily organize multiple HTML elements, from font styles to button styles.

* 1. **System Requirements Analysis**
     1. **Software Requirements**

The Airline Management System is run on a laptop or desktop computer. The minimum hardware and software necessary to run the system is required to achieve the best possible performance. Table 3.2 lists the hardware used in the system development as well as its specifications.

Table 3.3 Minimum Hardware Requirements

|  |  |  |
| --- | --- | --- |
| **No** | **Hardware** | **Hardware Specification** |
| **1.** | Laptop or desktop | Personal laptop or desktop |
| **2.** | Networking connection | Wi-Fi or 4G (data service) |

* + 1. **Software Requirements**

The need for software is critical in the development of a suggested system. Choosing the correct software specification will result in a smooth development process. Table 3.3 shows the bare minimum for software needs

Table 3.4 Minimum Software Requirements

|  |  |  |
| --- | --- | --- |
| **No** | **Software** | **Hardware Specification** |
| 1. | PHP | A programming language that has built-in web services support |
| 2. | Xampp | Abbreviation for cross-platform |
| 3. | phpMyAdmin | Database |
| 4. | Bootstrap | Design framework |
| 5. | JavaScript | Scripting language |
| 6. | Operating System | Windows version: 7 |

* 1. **Chapter Summary**

Finally, selecting the proper system technique is critical for keeping the project on track and within budget. Furthermore, adopting the appropriate approach will contribute to the development of the suggested system's success. Many aspects and comparisons must be examined when selecting a system approach, including project needs, intended end result, and project complexity.

Many things related to project methodology have been discussed in this chapter. The chapter started with: Introduction to this chapter and what to expect when this chapter talks about the parts of the chapter.

* The methodologies used in this project and the reasons why this methodology was chosen with the phases of the methodology can be seen in this chapter.
* The UML Diagram of the proposed system which included the classes and their features and relationships was explained.
* Gantt chart was also included in this chapter which included the timeline that each chapter needs to be submitted on.
* The technologies used in this system with Hardware requirement and Software requirement that are needed to run the system was explained.

**CHAPTER 4****REQUIREMENT ANALYSIS AND DESIGN**

* 1. **Introduction**

This chapter covers the requirements analysis and the design of the proposed system. The following parts will cover Requirements Analysis, which includes the system's use cases, sequence, and activity diagram. The System Design section contains the UML diagrams of the classes and the general architecture of your system. A Database Design explains the entity-relationship diagram (ERD) of a system database.

* 1. **Requirement Analysis**

This section analyses the system requirements for the proposed system, including the use case diagrams, \sequence diagrams and activity diagrams.

* + 1. **Use Case Model**

A use case diagram is a graphical representation of a user's possible interactions with a system. A use case diagram shows different use cases and different types of users that the system has, and is often accompanied by other types of diagrams as well. Figure 4.1 represent the Use case diagram of the proposed system showing the actors of the system and their functions/actions and the relationship between the functions/actions.

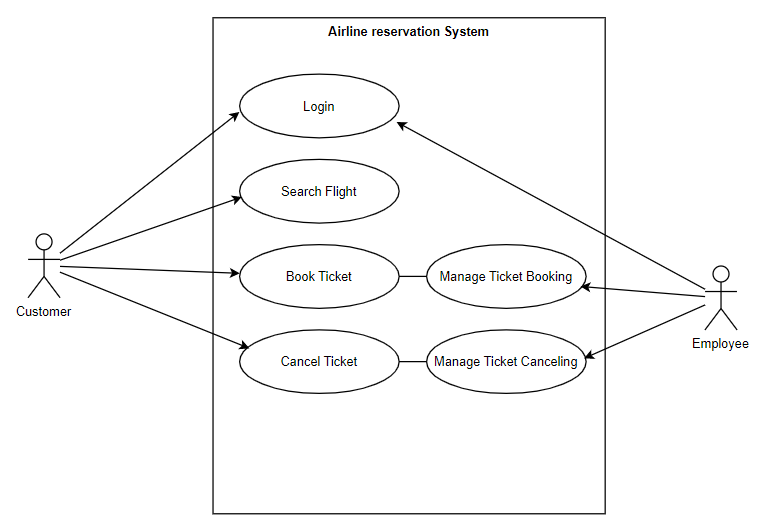
****

Figure 4.1 Use case Diagram of Airline Reservation System

* + - 1. **Use Case Description**

The table below shows that there are (2) actors whom are involved in using the proposed system: Customer and Employee. Those two actors play customer with full power over the system and employees who work on certain areas of the system.

Table 4.1 Actor Description Table

|  |  |  |
| --- | --- | --- |
| No. | **Actor** | **Rule** |
| 1 | Customer | Admin plays a main role in this project, they would be able to first login to the system then book tickets, cancel tickets and search flights |
| 2 | Employee | Employee is responsible for Manage Book Tickets and Manage Cancel Tickets. |

* + - 1. **Use Case Description**

Table 4.2 Use Case Table

|  |  |  |
| --- | --- | --- |
| No | Functionalities | Role |
| 1 | Search Flight | Customers would be able to search for available flight by searching the destination name. |
| 2 | Book Ticket | Customers would be able to book tickets |
| 3 | Manage Ticket Booking | Employee is managing the booking process |
| 4 | Payment | Customers would be able to make their payments through the system |
| 5 | Cancel Ticket | Customers would be able to cancel their ticket after booking it |
| 6 | Manage Ticket Cancelling | Employee is the one responsible to manage the cancel process |

* + 1. **Activity Diagram**

Activity Diagram for the Customer that would be able to book tickets whenever they want through the system. The complete activity diagram for each use case is already included in Software Requirements Specification (SRS). For further review of the activity, diagrams refer to Appendix B:

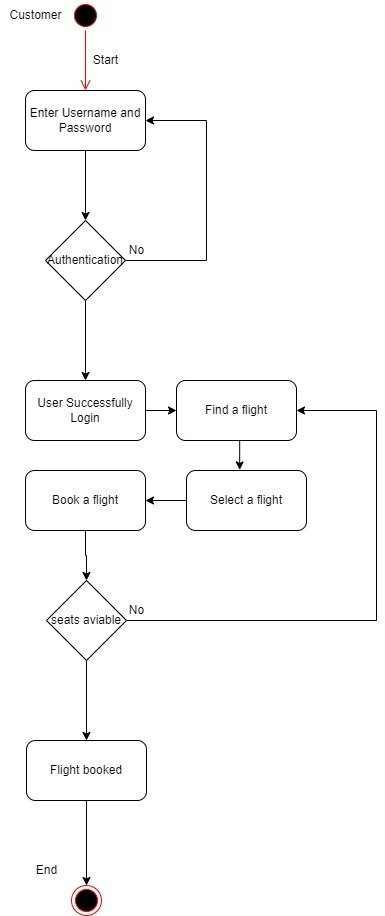


Figure 4.2 Activity diagram of Book Ticket

* + 1. **Sequence Diagram**

The Sequence Diagram for Customer that would be able to Book Ticket. The sequence diagrams show how the objects interact with each other in the system in a time. The complete sequence diagram for each use case is already included in Software Requirements Specification (SRS). For further review of the activity, diagrams refer to Appendix B:

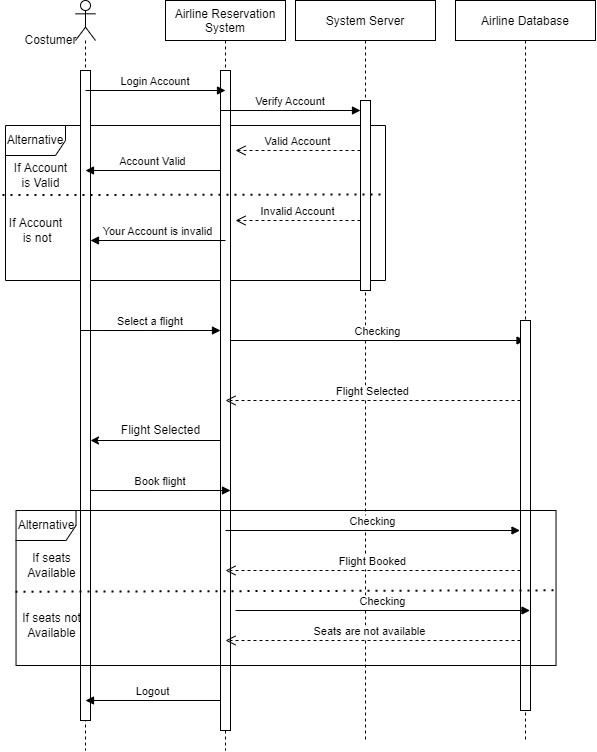


Figure 4.3 Sequence Diagram of Book Ticket

* 1. **Project Design**

This section explains the Proposed System Design in this work, including System Class Diagram, and System Architecture.

The architecture design of the proposed system can be seen in Figure 4.14 It shows all the system components (PCs for Employee and Customer, Web and MySQL Servers, and Network devices). Also, it shows how those components interact with each other as well. The Employee will access the system through the internet and then the Customer can access and book their flights that will be stored in the data in the server and the database.

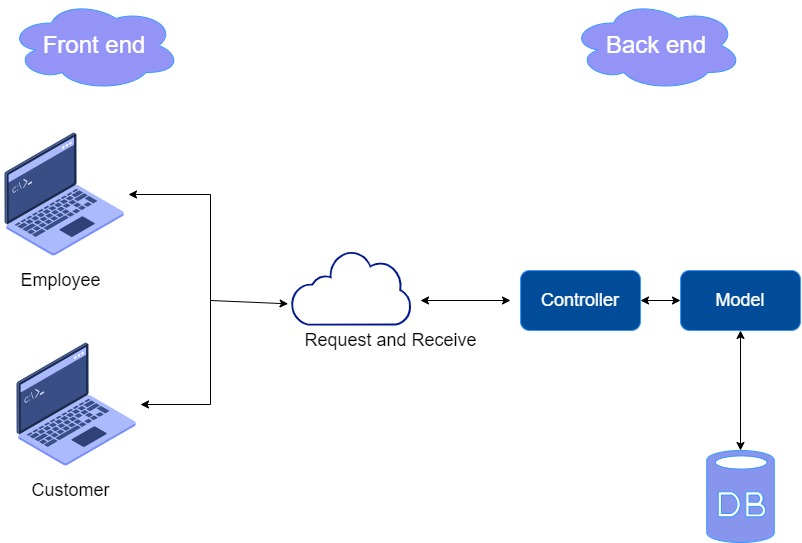


Figure 4.4 Architecture design for Airline Reservation System

**4.3.1 Project Design**

The class diagram for the Airline reservation system shows the information or data structures that are processed in the system. The classes will represent this data or information. Each of the classes has its attributes according to the methods it will use. Represented by a box with 3 partitions, the top part is the class name, the middle part is the attributes, and the bottom part is the methods. The arrows on it represent their relationships to each other ML class diagram. Then the classes to be made when making the Airline reservation system would be the "Customer, Booking, Book Flight, Flight and Airport" and this UML is changeable so to upcoming chapters might add some features it depends on the need for the system.

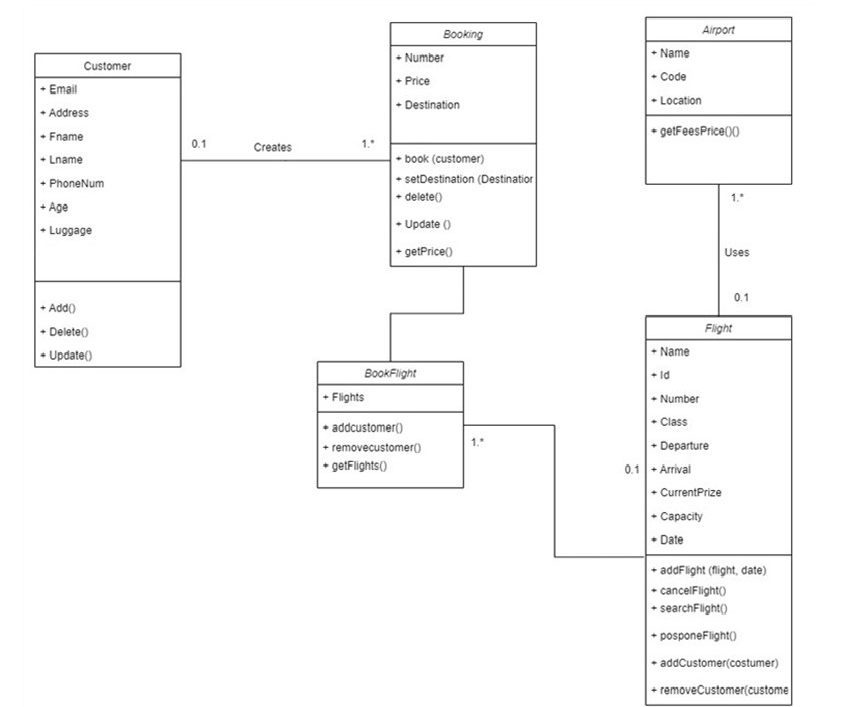
****

Figure 4.5 Class Diagram UML for Airline Reservation System.

**4.4 Database Design**

A database design is an arrangement of a database according to its database model (or schema) Let's take a quick look at what data needs to be stored and how the different data type relates to one another.

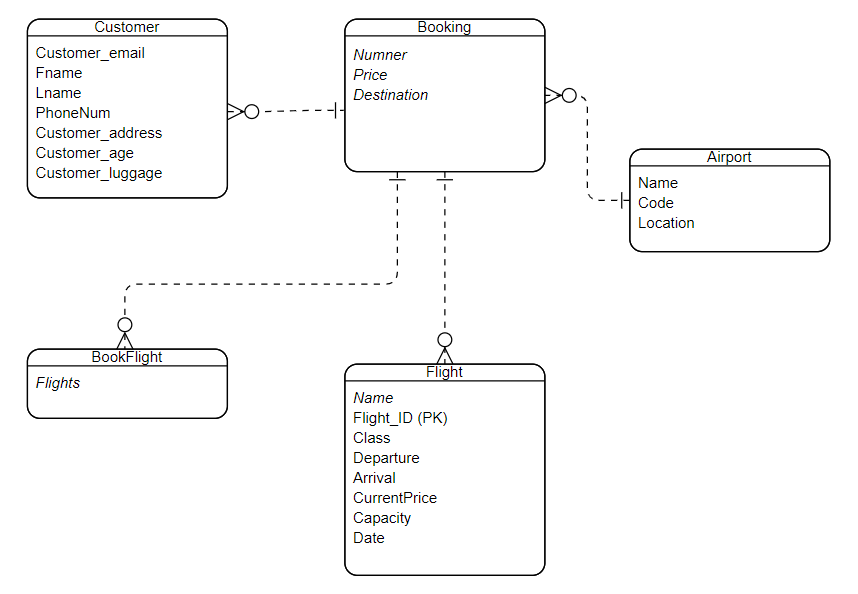


Figure 4.6 ERD for Airline Reservation System

**4.4.1 Database Table Description**

Below in the description of database in Airline Reservation System.

Table 4.3 Database Table of Customer

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Customer\_email | Varchar | 254 |
| Fname | Varchar | 254 |
| Lname | Varchar | 254 |
| PhoneNum | Int | 12 |
| Customer\_addess | Varchar | 12 |
| Customer\_age | Int | 12 |
| Customer\_luggage | Int | 12 |

Table 4.4 Database Table of Booking

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Number | Int | 12 |
| Price | Int | 12 |
| Destination | Varchar | 254 |

Table 4.5 Database Table of Flight

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Name | Varchar | 12 |
| Flight\_ID (PK) | Int | 254 |
| Class | Varchar | 254 |
| Departure | Int | 12 |
| Arrival | Int | 12 |
| CurrentPrice | Int | 12 |
| Capacity | Int | 12 |
| Date | Date |  |

Table 4.6 Database Table of Airport

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Name | Varchar | 12 |
| Code | Int | 12 |
| Location | Varchar | 254 |

Table 4.7 Database Table of Book Flight

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Flights | Varchar | 12 |

**4.5 Interface Design**

User contact with the Airline Reservation System through the interface design, which is an important aspect of the development process. Furthermore, the user interface design suggests the visual appeal for the user to utilize without additional training.



Figure 4.7 Welcome Interface of Airline Reservation System

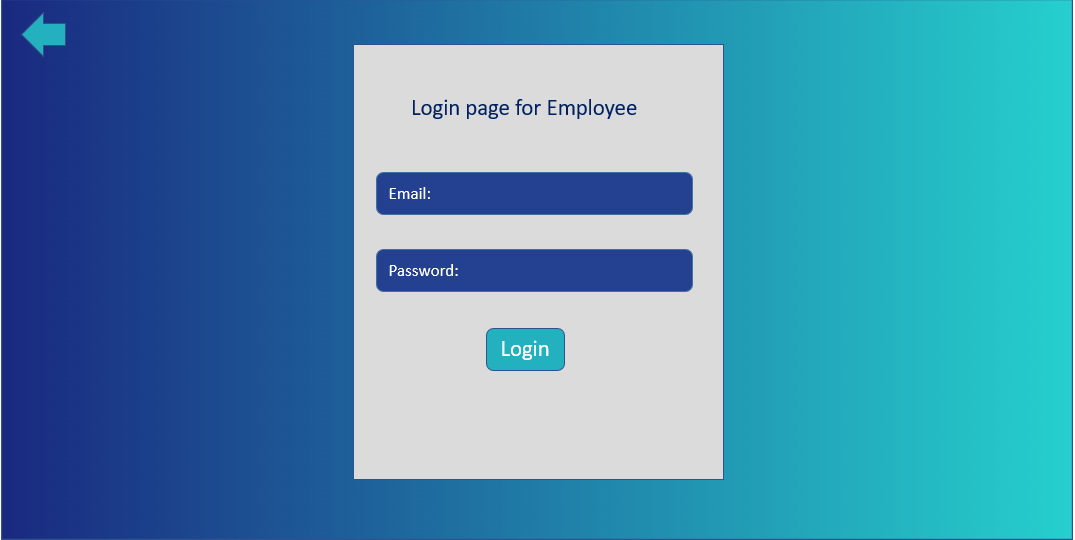


Figure 4.8 Login Interface for Employee



Figure 4.9 Interface for Employee actions Inside the system

****

Figure 4.10 Interface for Employee Managing Booking and Cancellation

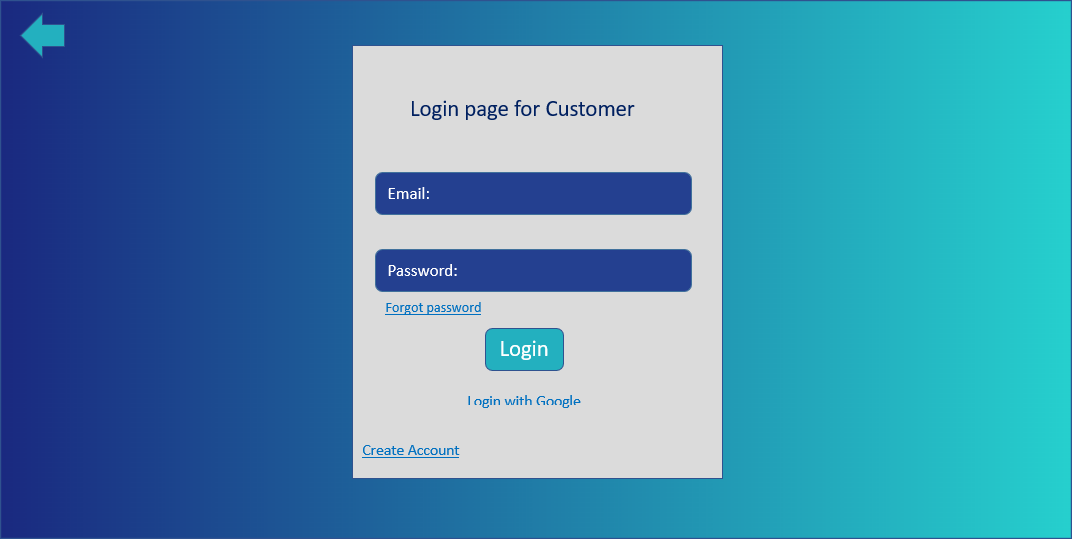


Figure 4.11 Login Interface for Customer

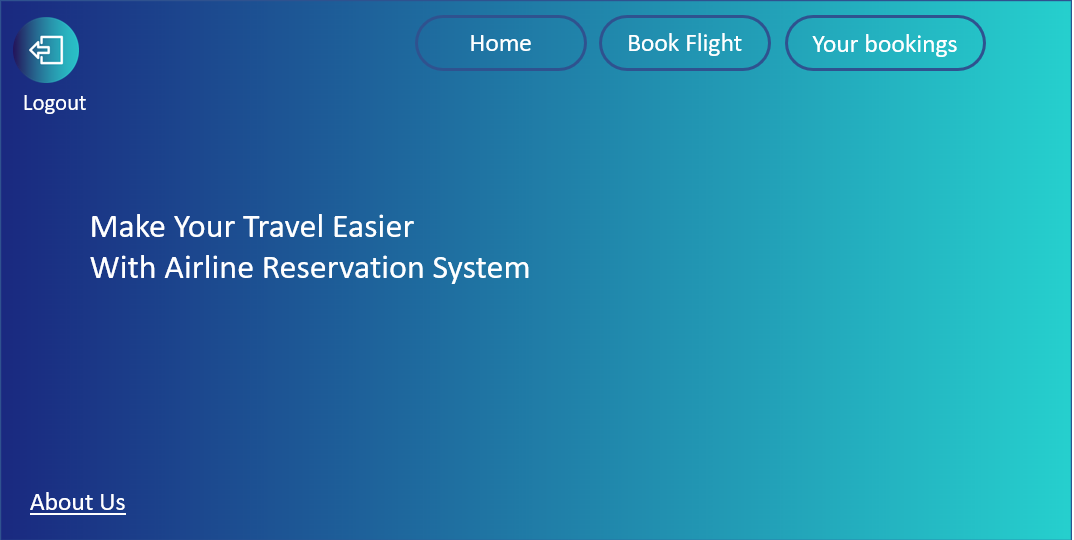


Figure 4.12 Home page Interface for Customer

****

Figure 4.13 Interface for Review Customer Bookings List

****

Figure 4.14 Interface for Customer to Book Flight

**4.6 Chapter Summary**

This chapter has discussed many things that are relevant to the design and requirements of the proposed system, including the introduction, use case diagram, sequence diagram, activity diagram, and the project architecture.

Then the design of the system is introduced in two forms: the class diagram and the system architecture. The database also was included in this chapter. Finally, the user interface was introduced, showing the prototype of the system. This interface might be different in the future according to the system's implementation and requirements.

**CHAPTER 5  
  
  
CONCLUSION**

**5.1 Introduction**

This chapter will discuss the overview of the Airline Reservation System. The discussion includes the results and achievements of the project objectives and recommendations for future implementations. The conclusion for the entire system will be briefly explained in this chapter.

**5.2 Project Development**

Airline Reservation System is developed to improve the current Airline Reservation system where the customers need to book their tickets using technology in order to make the system more accurate and not miss any part of their customers/passengers' data also makes it easy to manage by the employees. This system will provide information about the Flights and also show your bookings list so you as a customer can review it and at the same time you can retrieve any flight. The system allows you to search for any destination you want and it will provide the available flights.

**5.3 Achievement of Project Objectives**

Chapter 1: Introduces the project background, aims, objectives, and scopes. It explains how the system works and functions. After the study results were analysed, the problem was identified, and the solutions were achieved. Airline Reservation System is developed to help the customers/passengers to book their tickets and find the available and best flights for them easily through the system.

For Chapter 2: The literature review focused on the existing systems that provided the functionality to the system under development. It's important to figure out which aspects of the current system are beneficial and which ones aren't. The results of the existing system are discussed in this section, which was completed before the development stage.

Chapter 3: Prototyping methodology is used in developing an Airline Reservation System. Requirements gathering and analysis, fast design and prototyping, initial user evaluation, refining the prototype, implementing the product, and maintaining the product are the six phases of prototype development. Each phase plays its own role in developing the system. Technology, software, and hardware are also covered in this chapter.

Chapter 4: Explained how the system works and how it was designed. System architecture design is explained how the system is functioning. Database design is discussed so that you know how the data is related. Interface design is the design of how the system prototype looks like. The chapter Includes the use case, activity diagram, sequence diagram, class diagram and also entity relationship diagram for the database.

**5.4 Suggestions for Future Plan**

The methodology selected which is the prototype will be used in PSM 2. The project development will take place in phases, one at a time until the project is completed. After the development phase has been completed, both the white box test and the black box test will be run on the system. The user test and the user acceptance will be the last step in PSM 2 after all the phase is completed.

References:

[1] KUPPA, K "AIRLINE RESERVATION SYSTEM" Master Degree research for Kansas State University,2008

[2] ANKITSAAHIL. AIRLINE RESERVATION SYSTEM

[3] ElE, B.I., AGANA, M.A. and BUKIE, P.T" A DISTRIBUTED AIRLINE RESERVATION SYSTEMFOR NIGERIAN AIRLINE COMPANIES", COMPUSOFT, An international journal of advanced computer technology, 7(6), (Volume-VII, Issue-VI), June-2018

[4] GAUTAM, P., "Online Airline Ticketing System" Bachelor Degree research for CENTRIA UNIVERSITY OF APPLIED SCIENCES, 2015

[5] PARDEEP SINGH, "DESIGN AND IMPLEMENTATION OF AN ONLINE FLIGHT RESERVATION SYSTEM" Bachelor Degree research for Chandigarh University, 2015

[6] Article title: SDLC - Waterfall Model

Website title: Tutorialspoint.com <https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>

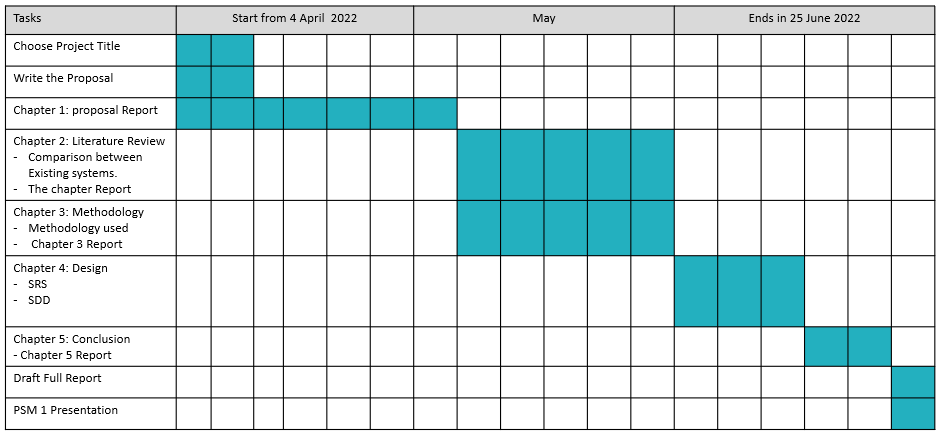
[7] Article title: Prototyping Model in Software Engineering: Methodology, Process, Approach

Website title: Guru99 <https://www.guru99.com/software-engineering-prototyping-model.html>

[8] Article title: XAMPP | Free and Open Source Web Server Solution Stack

Website title: Products.containerize.com <https://products.containerize.com/solution-stack/xampp/>

**Appendix A** **Gantt Chart PSM I**

****Figure A.1 Gantt Chart for PSM I

**Appendix B Software Requirements Specification**

**Software Requirements Specification**

Project Title:

Airline Reservation System

By:

Noor Nabil Ibrahim

Department and Faculty:

Computer Science department of Software Engineering

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **Introduction** | | | 52 |
|  | 1.1 | Purpose | | 52 |
|  | 1.2 | Scope | | 52 |
|  | 1.3 | Definitions, Acronyms and Abbreviations | | 52 |
|  | 1.4 | References 53 | |  |
|  | 1.5 | Overview 53 | |  |
| **2** | **Overall Description 54** | | |  |
|  | 2.1 | Product Perspective 56 | |  |
|  |  | 2.1.1 | System Interfaces 56 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  | 2.2 | Product Functions 56 | |  |
|  | 2.3 | User Characteristic 57 | |  |
|  | 2.4 | Constraints 57 | |  |
| **3** | **Specific Requirements 58** | | |  |
|  | 3.1 | External Interface Requirements 58  3.1.1 Software Interface Requirements 58  3.1.2 Hardware Interface Requirements 58 | |  |
|  | 3.2 | System Features 59 | |  |
|  |  | 3.2.1 | UC001: Use Case < Login > 59 |  |
|  |  | 3.2.2 | UC002: Use Case < Search Flight > 62 |  |
|  |  | 3.2.3 | UC003: Use Case < Book Ticket> 65 |  |
|  |  | 3.2.4 | UC004: Use Case < Cancel Ticket> 68 |  |
|  |  | 3.2.5 | UC005: Use Case < Manage Book Ticket > 71 |  |
|  |  | 3.2.6 | UC006: Use Case < Manage Ticket Cancellation > 74 |  |
|  | 3.3 | Performance Requirements 77 | |  |

Appendices (if any)

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NO. | TITLE | PAGE |

Table 2.1 Use Case Table 56

Table 2.2 Acrot Description Table 57

Table 2.3 Software Requirements in process of developing the ARS 57

Table 3.1 Use Case Description for < Login > 59

Table 3.2 Use Case Descriptiom for < Search Flight > 62

Table 3.3 Use Case Description for < Book Ticket > 65

Table 3.4 Use Case Description for < Cancel Ticket > 68

Table 3.5 Use Case Description for < Manage Book Ticket > 71

Table 3.6 Use Case Description for < Manage Ticket Cancelation > 74

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO. | TITLE | PAGE |

Figure 2.1 Use Case Diagram of Airline Reservation System 54

Figure 2.2 ERD for Airline Reservation System 55

Figure 3.1 Sequence Diagram of The System Login 60

Figure 3.2 Activity Diagram of The System Login 61

Figure 3.3 Sequence Diagram of Search Flight 63

Figure 3.4 Activity Diagram of Search Flight 64

Figure 3.5 Sequence Diaram of Book Ticket 66

Figure 3.6 Activity Diagram of Book Ticket 67

Figure 3.7 Sequence Diagram of Cancel Ticket 69

Figure 3.8 Activity Diagram of Cancel Ticket 70

Figure 3.9 Sequence Diagram of Manage Book Ticket 72

Figure 3.10 Activity Diagram of Manage Book Ticket 73

Figure 3.11 Sequence Diagram of Manage Ticket Cancelation 75

Figure 3.12 Activity Diagram of Manage Ticket Cancelation 70

1. **Introduction**

**1.1 Purpose**

A Software Requirements Specification (SRS) is a detailed document that outlines functional and technical specifications for a software system. This SRS is intended to outline the exact requirements for the Airline Reservation System. It also includes examples of how the user interacts with your system. This document includes various figures, including a sequence diagram and an activity diagram.

**1.2 Scope**

The scopes of the project are:

(a) We will start the project by contacting some of the Travel Agency offices directly or by preparing an electronic questionnaire, this will help in collecting the main requirements and needs of those offices, after that, we need to analyse those requirements and identify the challenges and obstacles that may encounter it during the project design and implementation process.

(b) We will use PHP, MYSQL, CSS, HTM, JavaScript, Bootstrap, J Query, and of course database to build the website and Apache web server.

**1.3 Definitions, Acronyms and Abbreviations**

ARS: Airline Reservation System.

SRS: Software Requirements Specification

**1.4 References**

We had a subject called Software Design Architecture for previous semesters and all explain System Design Architecture in details and I saved slides from E-learning system.

**1.5 Overview**

This specification will consist of three sections, the first one of which will introduce you to the entire document. The second section is a description of the system, including a description of the requirements for the system. These requirements will limit how the system will be constructed and operated. The third section is an extensive description of the system specifications.

1. **Overall Description**

The ARS is made up of five (6) modules which are:

* Login
* Search Flight
* Book Ticket
* Cancel Ticket
* Manage Book Ticket
* Manage Ticket Cancellation

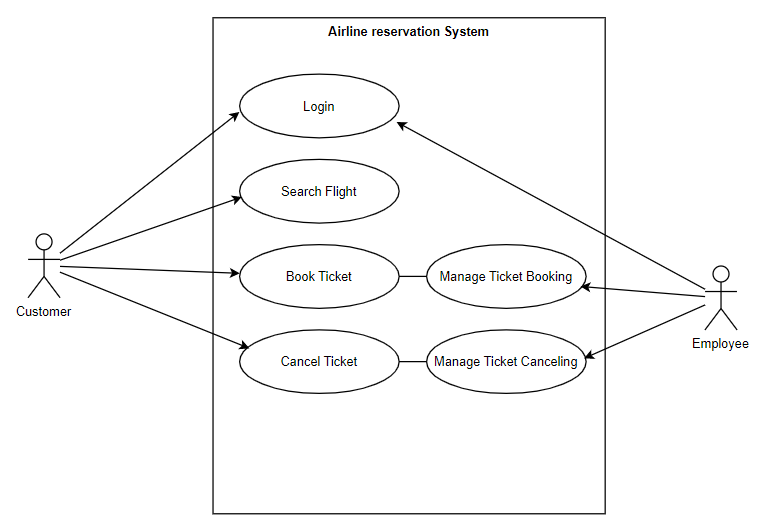
****

Figure 2.1 Use case Diagram of Airline Reservation System

**2.1 Product Perspective**

Airline Reservation System is a web-based system that is built for Employees and Customers for employee to manage the bookings and cancellations and for customers to book their tickets easily and also to search for the flight they want and they can cancel their flights also they can view their booking list and do retrieve of any of them as well.

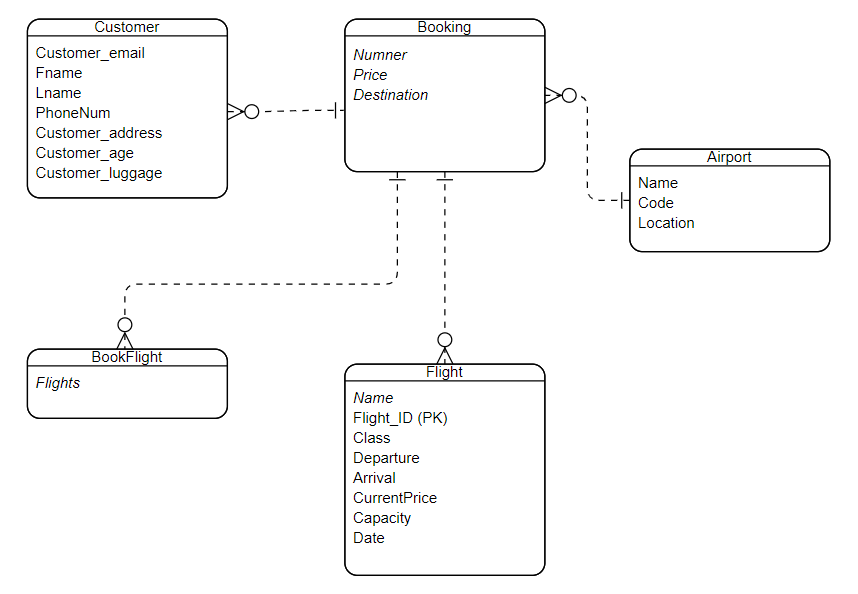


Figure 2.2 ERD for Airline Reservation System

**2.1.1 System Interface**

It consists of four modules and two (2) actors. Each actor has a specific role and function within the module. The user interface is easy to use and appealing for the users. It's also designed to be simple but attractive by using nice colours and icons to show each section.

**2.2 Product Functions**

There are 6 use cases that represent the major functions performed by the proposed solution.

Table 2.1 Use Case Table

|  |  |  |
| --- | --- | --- |
| No | Functionalities | Role |
| 1 | Search Flight | Customers would be able to search for available flight by searching the destination name. |
| 2 | Book Ticket | Customers would be able to book tickets |
| 3 | Manage Ticket Booking | Employee is managing the booking process |
| 4 | Payment | Customers would be able to make their payments through the system |
| 5 | Cancel Ticket | Customers would be able to cancel their ticket after booking it |
| 6 | Manage Ticket Cancelling | Employee is the one responsible to manage the cancel process |

**2.3 User Characteristic**

The table below shows that there are (2) actors whom are involved in using the proposed system: Customer and Employee. Those two actors play customer with full power over the system and employees who work on certain areas of the system.

Table 2.2 Actor Description Table

|  |  |  |
| --- | --- | --- |
| No. | **Actor** | **Rule** |
| 1 | Customer | Admin plays a main role in this project, they would be able to first login to the system then book tickets, cancel tickets and search flights |
| 2 | Employee | Employee is responsible for Manage Book Tickets and Manage Cancel Tickets. |

**2.4 Constraints**

Table 2.3 Software Requirements in process of developing the ARS

|  |  |  |
| --- | --- | --- |
| **Category** | **Software** | **Software Description** |
| Operating System | Windows Operating System | Platform to run the Airline Reservation System. |
| Source-code editor | Virtual Studio Code | Use for proposed system development by writing a code that implements PHP language. |
| Database | MySQL XAMMP | Responsible for the back-end of the system that manages and manipulates all the transaction of data into the database. |

1. **Specific Requirements**

**3.1 External Interface Requirements**

**3.1.1 Software Interface Requirements**

Provide the details for Minimum Requirements:

* Operating system: Windows 7 or later
* Processor: Intel Pentium 4
* Memory: 2GB minimum, 4GB recommended
* Screen Resolution: 1280x1024 or larger
* Application window size: 1024x680 or larger
* Internet Connection: Required
* Input Device: Mouse

**3.1.2 Hardware Interface Requirements**

Windows 11 pro was used for developing this System, and the code will be implementing using Visual Studio Code and for the database XAMPP was used.

**3.2 System Features**

**3.2.1** **UC001: Use Case < Login >**

Table 3.1 Use Case Description for < Login >

|  |  |
| --- | --- |
| **User Case** | **View Profile** |
| ID | UC01 |
| Actors | Employee and Customer |
| Description | Employee and Customer are allowed to login to the system |
| Pre-Condition | Employee and Customer logged into the system and get authenticated |
| Normal Flow | 1. The use case starts when the actor opens the system. 2. The Actor clicks the “Login” to get inside the system. |
| Alternative Flow | None |
| Exception Flow | 1. Warning message “Please fill the form” 2. Actor fills the form not left blank |
| Related Requirements | Possible Actions linked with another user case:  - |
| Post-Condition | 1. User successfully login to the system. |

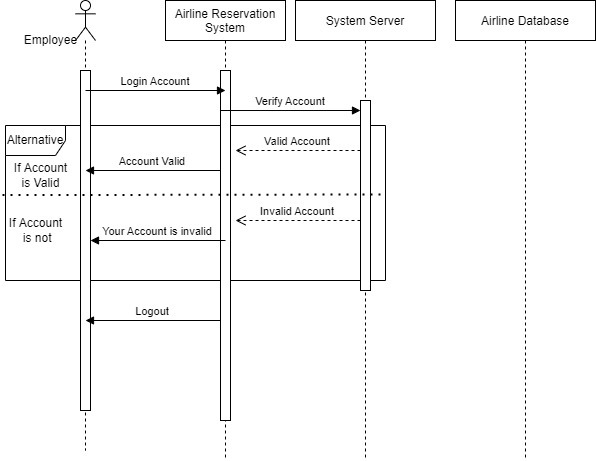


Figure 3.1 Sequence Diagram of The System Login

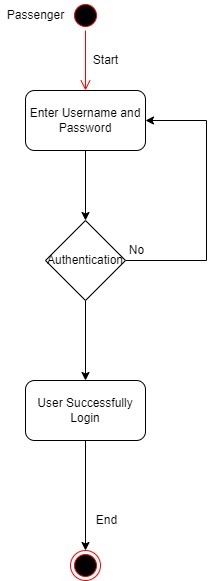


Figure 3.2 Activity diagram of the system login

**3.2.2 UC002: Use Case < Search Flight >**

Table 3.2 Use Case Description for < Search Flight >

|  |  |
| --- | --- |
| **User Case** | **View Profile** |
| ID | UC02 |
| Actors | Customer |
| Description | Customer are allowed to search for the flight they want by write the destination name. |
| Pre-Condition | 1. Customer logged into the system and get authenticated. 2. Customer already register into the system |
| Normal Flow | 1. The use case starts when the actor opens the system. 2. The Actor clicks the “Search bar” to do searching for any flight they want |
| Alternative Flow | None |
| Exception Flow | None |
| Related Requirements | Possible Actions linked with another user case:  - |
| Post-Condition | 1. User successfully login to the system. |

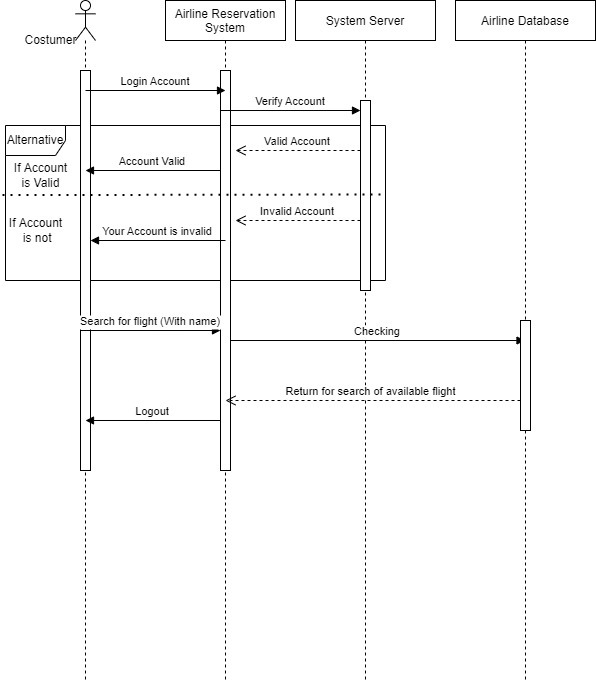
****

Figure 3.3 Sequence Diagram of Search Flight

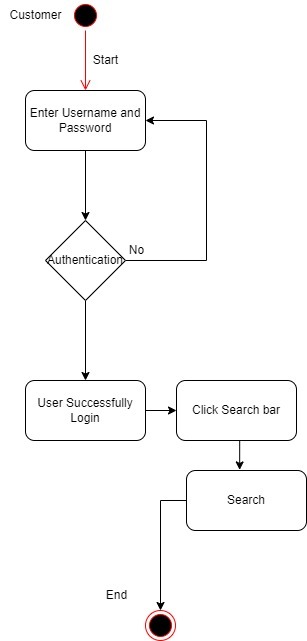


Figure 3.4 Activity diagram of Search Flight

**3.2.3** **UC003: Use Case < Book Ticket >**

Table 3.3 Use Case Description for < Book Ticket >

|  |  |
| --- | --- |
| **User Case** | **View Profile** |
| ID | UC03 |
| Actors | Customer |
| Description | Customer are allowed book tickets for any destination they want |
| Pre-Condition | 1. Customer logged into the system and get authenticated. 2. Customer already register into the system |
| Normal Flow | 1. The use case starts when the actor opens the system. 2. The Actor clicks the “Book Ticket” to get to the booking page and view all the flights 3. The Actor click “Book” to book the Ticket for the flight as the customer want. |
| Alternative Flow | None |
| Exception Flow | None |
| Related Requirements | Possible Actions linked with another user case:  - |
| Post-Condition | 1. User successfully login to the system. |

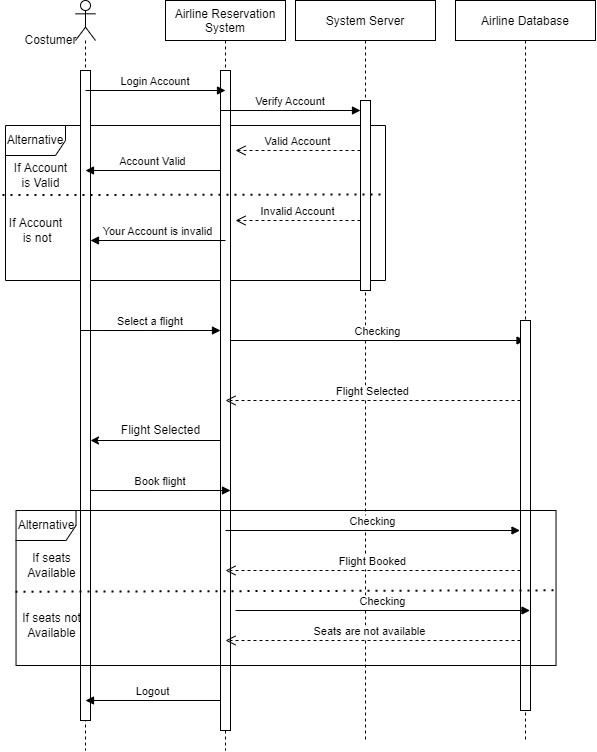


Figure 3.5 Sequence Diagram of Book Ticket

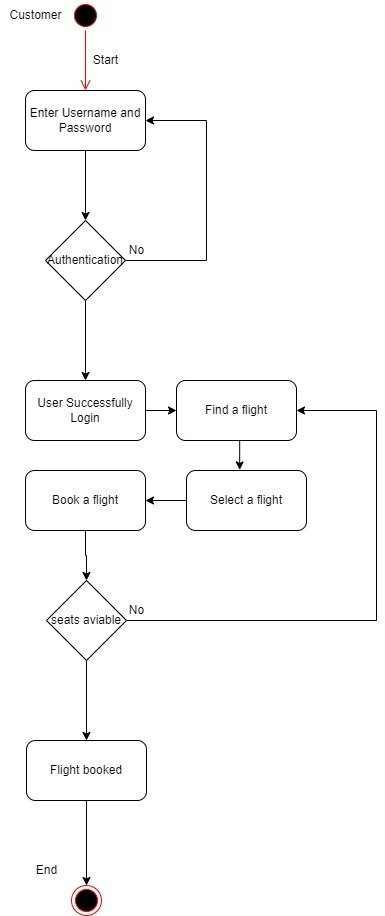


Figure 3.6 Activity diagram of Book Ticket

**3.2.4 UC004: Use Case < Cancel Ticket >**

Table 3.4 Use Case Description for < Cancel Ticket >

|  |  |
| --- | --- |
| **User Case** | **View Profile** |
| ID | UC04 |
| Actors | Customer |
| Description | Customer are allowed book tickets for any destination they want |
| Pre-Condition | 1. Customer logged into the system and get authenticated. 2. Customer already register into the system |
| Normal Flow | 1. The use case starts when the actor opens the system. 2. The Actor clicks the “Your Bookings” to view the booking list that already done. 3. The Actor click “Cancel” to cancel their flights after booking it. |
| Alternative Flow | None |
| Exception Flow | None |
| Related Requirements | Possible Actions linked with another user case:  - |
| Post-Condition | 1. User successfully login to the system. 2. User already booked a Ticket to be able to cancel it if they want. |

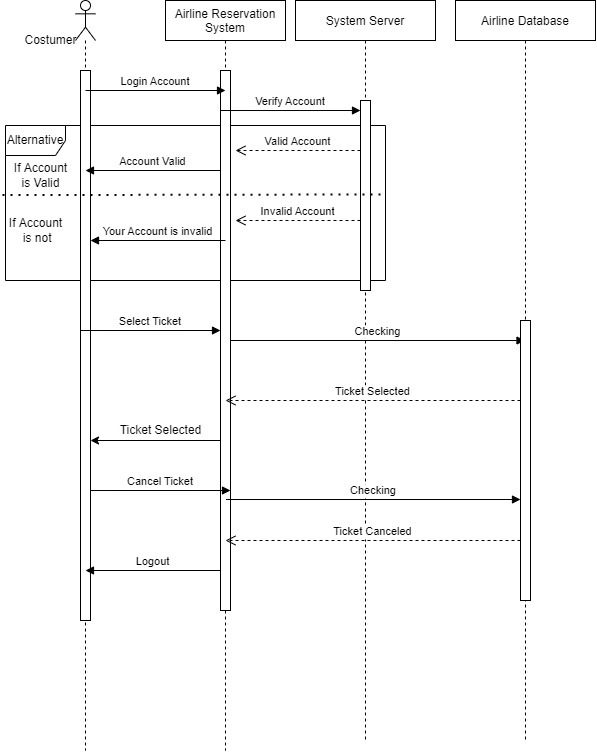


Figure 3.7 Sequence Diagram of Cancel Ticket

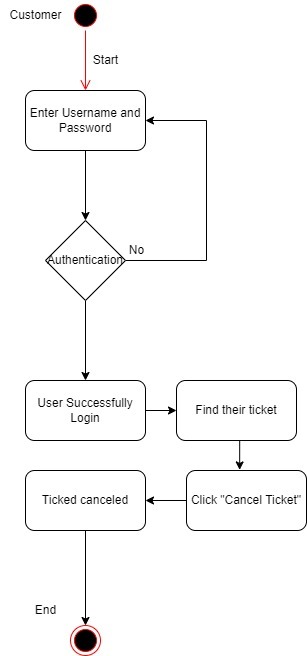
****

Figure 3.8 Activity diagram of Cancel Ticket

**3.2.5 UC005: Use Case < Manage Book Ticket >**

Table 3.5 Use Case Description for < Manage Book Ticket >

|  |  |
| --- | --- |
| **User Case** | **View Profile** |
| ID | UC05 |
| Actors | Employee |
| Description | Employee are allowed to manage the bookings |
| Pre-Condition | 1. Customer logged into the system and get authenticated. 2. Customer already register into the system |
| Normal Flow | 1. The use case starts when the actor opens the system. 2. The Actor clicks the “Your Bookings” to view the booking list that already done. 3. The Actor click “Cancel” to cancel their flights after booking it. |
| Alternative Flow | None |
| Exception Flow | None |
| Related Requirements | Possible Actions linked with another user case:  - |
| Post-Condition | 1. User successfully login to the system. |

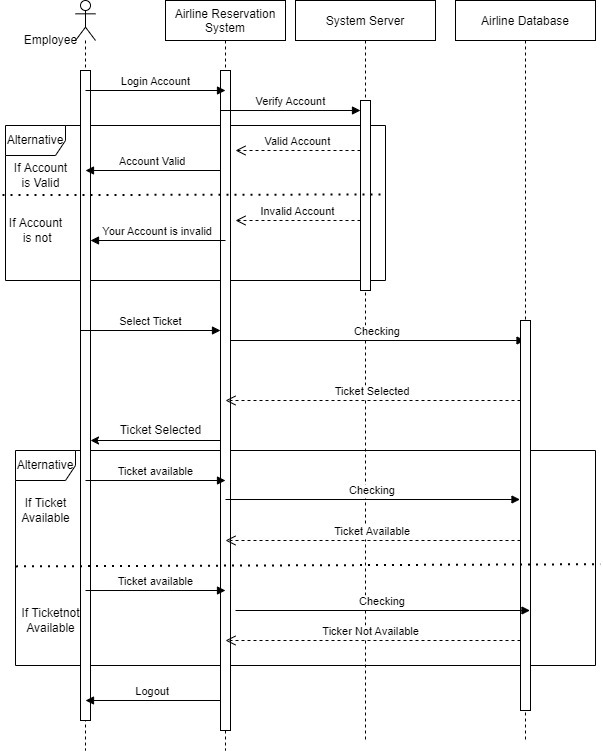
****

Figure 3.9 Sequence Diagram of Manage Book Ticket

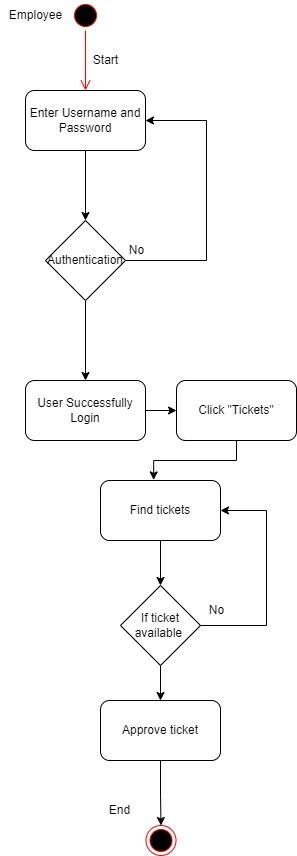


Figure 3.10 Activity diagram of Manage Book Ticket

**3.2.6 UC006: Use Case < Manage Ticket Cancelation >**

Table 3.6 Use Case Description for < Manage Ticket Cancelation >

|  |  |
| --- | --- |
| **User Case** | **View Profile** |
| ID | UC06 |
| Actors | Employee |
| Description | Employee are allowed to Manage Ticket canceling |
| Pre-Condition | 1. Customer logged into the system and get authenticated. 2. Customer already register into the system |
| Normal Flow | 1. The use case starts when the actor opens the system. 2. The Actor clicks the “Manage Booking & Cancelation” to view the booking list that already done by Customers. 3. The Actor click “Cancel” to cancel the flight for the customer after the customer already cancelled it. |
| Alternative Flow | None |
| Exception Flow | None |
| Related Requirements | Possible Actions linked with another user case:  - |
| Post-Condition | 1. User successfully login to the system. |

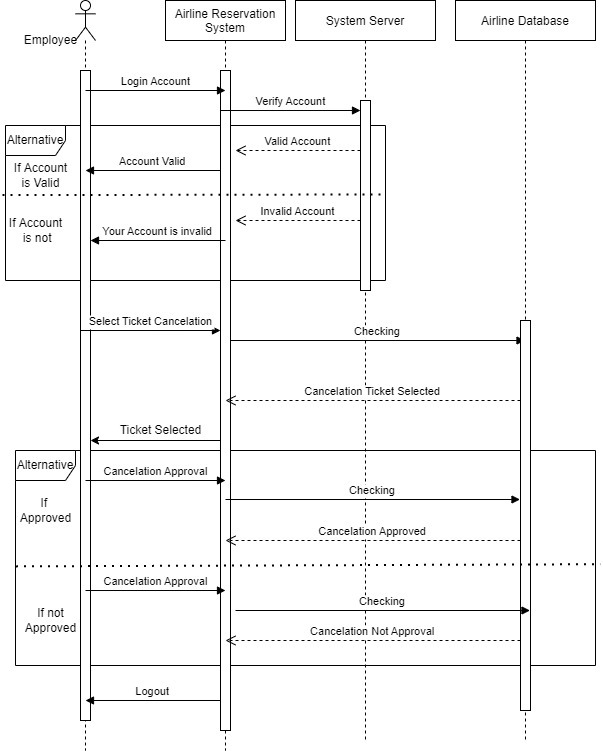
****

Figure 3.11 Sequence Diagram of Manage Ticket Cancelling

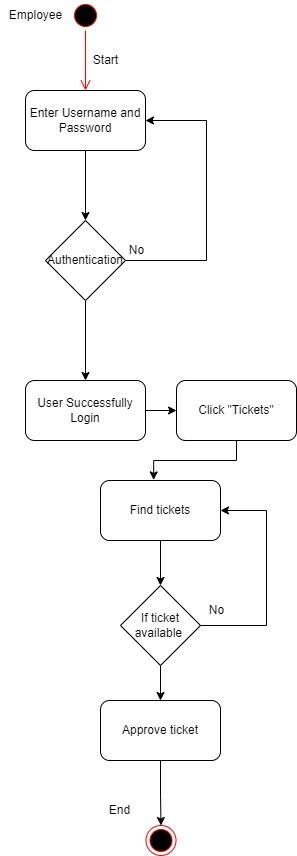


Figure 3.12 Activity diagram of Manage Ticket Cancelation

**3.3 Performance Requirements**

Availability: The user interface should be easy to use without requiring any additional training or support for the end user.

Performance: All the bookings shall be able to send to the airline company in a short time regardless of the number of customers.

Usability: The user interface should look good enough for a system that doesn't require any extra training or support for the user to understand.

**Appendix C Software Design Document**

**Software Design Documentation**

**for**

**Airline Reservation System**

By:

Noor Nabil Ibrahim

Department and Faculty:

Computer Science department of Software Engineering

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Introduction | | | 81 |
|  | 1.1 | Purpose | | 81 |
|  | 1.2 | Scope | | 81 |
|  | 1.3 | Definitions, Acronyms and Abbreviations | | 81 |
|  | 1.4 | References | | 82 |
|  | 1.5 | Overview | | 82 |
| 2 | System Overview | | | 82 |
| 3 | System Architectural Design  3.1 MVC Architecture | | | 83  83 |
| 4 | Data Design | | | 84 |
|  | 4.1 | Entity Relationship Diagram (ERD) | | 84 |
|  | 4.2 | Data Dictionary | | 85 |
| 5 | Interface Design | | | 87 |
|  | 5.1 | Overview of User Interface | | 87 |
|  | 5.2 | Screen Images | | 87 |
|  |  | 5.2.1 | Welcome Page | 87 |
|  |  | 5.2.2 | Login Page for Employee | 88 |
|  |  | 5.2.3 | A simple page for Employee | 88 |
|  |  | 5.2.4 | Manage Booking and Cancelation Page for Employee | 89 |
|  |  | 5.2.5 | Login Page for Customer | 89 |
|  |  | 5.2.6 | Home Page | 90 |
|  |  | 5.2.7 | My Bookings Page for Customers | 90 |
|  |  | 5.2.8 | Book Ticket Page for Customers | 91 |

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO. | TITLE | PAGE |

Figure 3.1 MVC Architecture Diagram 83

Figure 4.1 ERD for Airline Reservation System 84

Figure 5.1 Welcome Interface of Airline Reservation System 87

Figure 5.2 Login Interface for Employee 88

Figure 5.3 Interface for Employee actions Inside the System 88

Figure 5.4 Interface for Employee Managing Booking and Cancelation 89

Figure 5.5 Login Interface for Cuatomer 89

Figure 5.6 Home page Interface for Customer 90

Figure 5.7 Interface for Review Customer Bookings List 90

Figure 5.8 Interface for Customer to Book Flight 91

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| TABLE NO. | TITLE | PAGE |

Table 4.1 Use Case Table 85

Table 4.2 Acrot Description Table 85

Table 4.3 Software Requirements in process of developing the ARS 86

Table 4.4 Use Case Description for < Login > 86

Table 4.5 Use Case Descriptiom for < Search Flight > 86

1. **Introduction**

**1.1 Purpose**

The purpose of software design documentation (SDD) is to provide a clear overview of how the ARS is supposed to be built and a brief description of its flow. A software design description that includes an overview of the architecture, database, and user interface is also included in this documentation.

**1.2 Scope**

The scopes of the project are:

1. The system will focus on providing companies with a friendly use airline Reservation System.
2. Started the project by contacting some of the Travel Agency offices directly or by preparing an electronic questionnaire, this will help in collecting the main requirements and needs of those offices, after that, we need to analyze those requirements and identify the challenges and obstacles that may encounter it during the project design and implementation process.

**1.3 Definitions, Acronyms and Abbreviations**

ARS: Airline Reservation System.

MVC: Model View Controller.

SDD: Software Design Documentation.

ERD: Entity Relationship Diagram

**1.4 References**

MVC Framework Tutorial for Beginners: What is, Architecture & Example

Website title: Guru99 <https://www.guru99.com/mvc-tutorial.html>

* 1. **Overview**

This document describes the overview of the system in various sections:

* Introduction
* System Overview
* System Architecture
* Database Design
* Interface Design

**2. System Overview**

Airline Reservation System (ARS) is a web-based system that aims to make passenger booking tickets easier and also make employee jobs easier than working at airline offices without a website. This means this proposed system allows the target user which is an admin or an Airline company manager to use this.

**3. System Architectural Design**

**3.1 MVC Architecture**

MVC stands for “Model-View-Controller”. MVC is an architectural pattern that separates an application’s functionality into three separate components: Model, View, and Controller architecture is a common design pattern used for building both web applications and mobile applications. This architectural pattern works well with object-oriented programming due to distinct models, views, and controllers can be treated as objects and reused within applications.

There are three components in MVC. The first thing is the Model component. Its Components are responsible for storing data and its associated logic. This component represents data that is being transferred between different controllers. It responds to requests from the controllers. The view component is the second component. View represents the data displayed by means of displaying objects in the system. The last component is Controller. A controller acts as an intermediary between views and models. This component will receive input from the user and perform an action. Figure 3.1 illustrates the concept of MVC briefly.

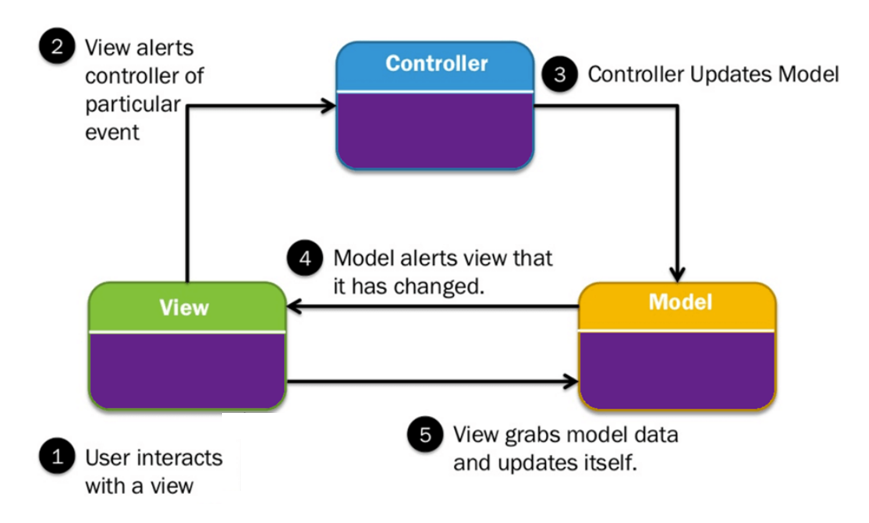
****

Figure: 3.1 MVC Architecture Diagram

**4. Data Design**

**4.1 Entity Relationship Diagram (ERD)**

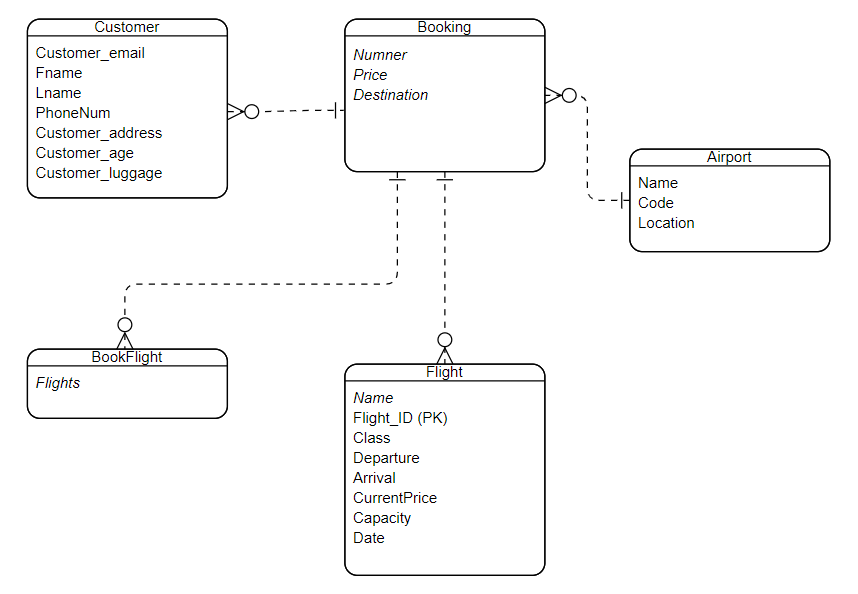


Figure 3.2 ERD for Airline Reservation System

**4.2 Data Dictionary**

Below in the description of database in Airline Reservation System.

Table 4.1 Database Table of Customer

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Customer\_email | Varchar | 254 |
| Fname | Varchar | 254 |
| Lname | Varchar | 254 |
| PhoneNum | Int | 12 |
| Customer\_addess | Varchar | 12 |
| Customer\_age | Int | 12 |
| Customer\_luggage | Int | 12 |

Table 4.2 Database Table of Booking

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Number | Int | 12 |
| Price | Int | 12 |
| Destination | Varchar | 254 |

Table 4.3 Database Table of Flight

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Name | Varchar | 12 |
| Flight\_ID (PK) | Int | 254 |
| Class | Varchar | 254 |
| Departure | Int | 12 |
| Arrival | Int | 12 |
| CurrentPrice | Int | 12 |
| Capacity | Int | 12 |
| Date | Date |  |

Table 4.4 Database Table of Airport

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Name | Varchar | 12 |
| Code | Int | 12 |
| Location | Varchar | 254 |

Table 4.5 Database Table of Book Flight

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Data Type** | **Length** |
| Flights | Varchar | 12 |

**5. Interface Design**

**5.1 Overview of User Interface**

This system is focused mainly on the desktop. A friendly user interface has been designed so that the target can use the system without any extra training.

**5.2 Screen Images**

**5.2.1 Welcome Page**



Figure 5.1 Welcome Interface of Airline Reservation System

**5.2.2 Login Page for Employee**

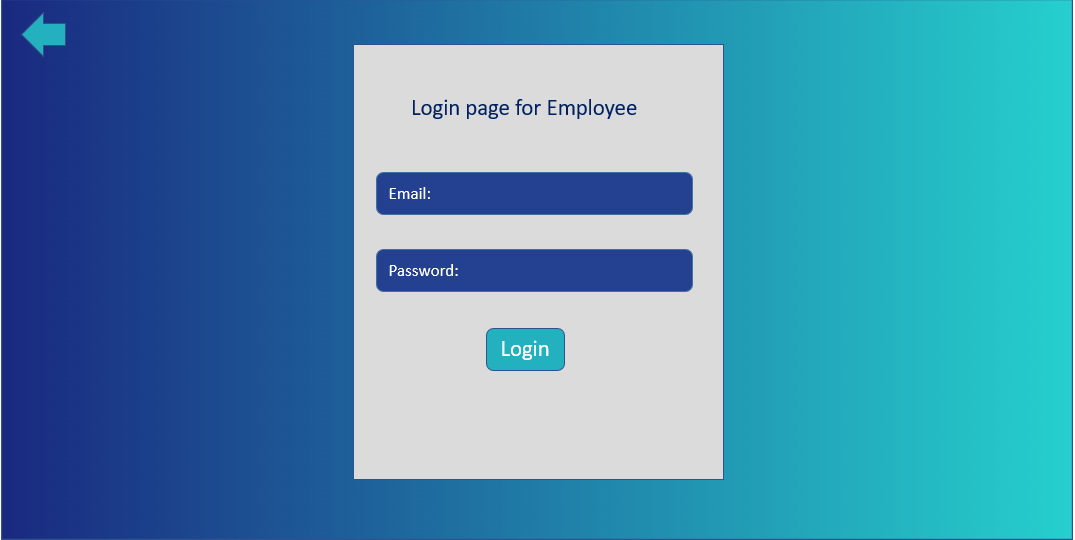


Figure 5.2 Login Interface for Employee

**5.2.3 A simple page for Employee**



Figure 5.3 Interface for Employee actions Inside the system

**5.2.4 Manage Booking and Cancelation Page for Employee**

****

Figure 5.4 Interface for Employee Managing Booking and Cancellation

**5.2.5 Login Page for Customer**

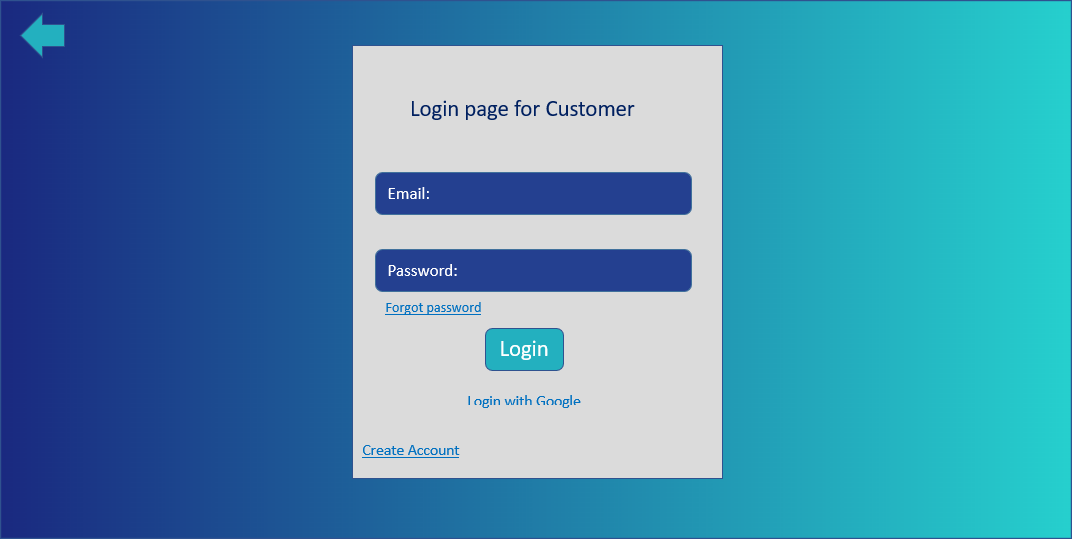


Figure 5.5 Login Interface for Customer

**5.2.6 Home Page**

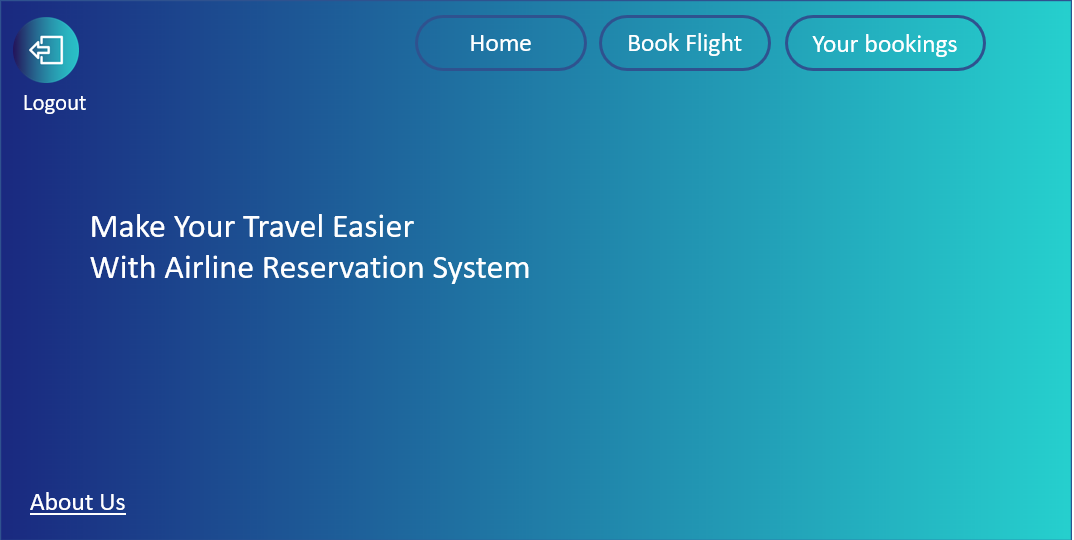


Figure 5.6 Home page Interface for Customer

**5.2.7 My Bookings Page for Customers**

****

Figure 5.7 Interface for Review Customer Bookings List

**5.2.8 Book Ticket Page for Customers**

****

Figure 5.8 Interface for Customer to Book Flight