

**Tribhuvan University**

**Faculty of Humanities and Social Science**

**AIRLINE TICKET RESERVATION SYSTEM**

**A PROJECT REPORT**

**Submitted to**

**Department of Computer Application**

**Kathmandu BernHardt College**

***In partial fulfillment of requirements for the Bachelors in Computer Application***

**Submitted by**:

|  |  |
| --- | --- |
| Name | Roll No. |
| Nischal Adhikari | 4610531 |
|  |  |
|  |  |

September, 2022

Under supervision of

**Anil Verma**

# ABSTRACT

As the innovation advancements are immediately created including mobile innovation, a web application for reservation services can upgrade people's life, make it less complex. This looks into presents a reservation system for airline organizations to rearrange the way toward booking a flight. The clients, as a rule, purchase their ticket manually furthermore, perhaps hold up in a line to check the required flight. In this paper, we presented a survey on airline reservation system. The proposed system executed to conquer manual system issue by making a booking from home or somewhere else and save time and effort.

Airline Reservation system is used for searching available flights and booking tickets. It’s an easy for Admin to retrieve the data of flights if it is available for the location, and also if there are available seats on searched flights. It aims to provide high level security to user information.

The system was achieved by developing an airline reservation using the three-tier software architectural model. The System is implemented using web-based technologies which include CSS, JS, HTML, MySQL, Laravel framework and runs on Windows operating system. Algorithms is also implemented onto the system as it plays a very vital role for giving the nearest empty parking lot. The overall functionality of the system shows that it works satisfactory and the result obtained shows that the system is error free, faster and allows customers to securely purchase their tickets.

**Keywords:** *Flight, Destination, Laravel, HTML, MySQL.*

# ACKNOWLEDGEMENT

I would like to express our sincere gratitude to Project Supervisor **Mr. Anil Verma,** he continuously helped and guided us through different problems faced during this whole coding process. Flight Management System on Laravel PHP Framework. Without his support, it would have been difficult to work on. The satisfaction and success of the completion of this task would be incomplete without heartfelt thanks to people whose constant guidance, support and encouragement made this work successful.

I would like to thank Lecturer **Mr. Sagar KC** and Head of Department **Mr. Ram Babu Mahato** sir for assisting in the different scope of the programs. I would like to thank Kathmandu BernHardt College for providing all the resources and ideal environment for the project developing process and encourage us to develop such a project. I would also like to share our gratitude to Tribhuvan University for providing the opportunity to evaluate our programming skills through this project.

I sincerely express our whole hearted thanks to the principal Mr. Sahadev Sigdel for his constant encouragement and moral support during the course of this project. I owe our sincere thanks for furnishing every essential facility for doing this project. I sincerely thank our guides for valuable help and guidance throughout the project. I wish to express our sincere thanks to the project coordinator and all staff members, Department of Computer Application for their valuable help and guidance rendered to us throughout the project. Above all I am grateful to all our classmates and friends for their friendly cooperation and their exhilarating company.

Nischal Adhikari

6-2-461-66-2018

Table OF CONTENTS

**Page No.**

**Title**

[ABSTRACT i](#_Toc112600628)

[ACKNOWLEDGEMENT ii](#_Toc112600629)

[LIST OF ABBREVIATIONS v](#_Toc112600630)

[LIST OF FIGURES vi](#_Toc112600631)

[LIST OF TABLES vii](#_Toc112600632)

[CHAPTER 1: INTRODUCTION 1](#_Toc112600633)

[1.1 Introduction 1](#_Toc112600634)

[1.2 Problem Statement 1](#_Toc112600635)

[1.3 Objectives 2](#_Toc112600636)

[1.4 Scope and Limitations 2](#_Toc112600637)

[1.4.1 Scope of Project 2](#_Toc112600638)

[1.4.2 Limitations 2](#_Toc112600639)

[1.5 Development Methodology 2](#_Toc112600640)

[1.6 Organization of Report 3](#_Toc112600641)

[CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW 5](#_Toc112600642)

[2.1 Background Study 5](#_Toc112600643)

[2.2 Literature Review 5](#_Toc112600644)

[CHAPTER 3: SYSTEM ANALYSIS AND DESIGN 7](#_Toc112600645)

[3.1 System Analysis 7](#_Toc112600646)

[3.1.1 Requirement Analysis 7](#_Toc112600647)

[3.1.2 Feasibility Analysis 9](#_Toc112600648)

[3.1.3 Object Modelling: Object & Class Diagram 10](#_Toc112600649)

[3.1.4 Dynamic Modelling: State & Sequence Diagram 11](#_Toc112600650)

[3.1.5 Process Modelling: Activity Diagram 12](#_Toc112600651)

[3.2 System Design 13](#_Toc112600652)

[3.2.1 Refinement of Classes and Object 13](#_Toc112600653)

[3.2.2 Component Diagram 14](#_Toc112600654)

[3.2.3 Deployment Diagram 15](#_Toc112600655)

[3.3 Algorithm Details 16](#_Toc112600656)

[CHAPTER 4: IMPLEMENTATION AND TESTING 17](#_Toc112600657)

[4.1 Implementation 17](#_Toc112600658)

[4.1.1 Tools Used (CASE Tools, Programming Languages, Database Platforms) 17](#_Toc112600659)

[4.1.2 Implementation Details of Modules (Description of procedures/functions) 19](#_Toc112600660)

[4.2 Testing 23](#_Toc112600661)

[4.2.1 Test Cases for Unit Testing 23](#_Toc112600662)

[4.2.2 Test Cases for System Testing 23](#_Toc112600663)

[CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS 25](#_Toc112600664)

[5.1 Lesson Learnt/Outcome 25](#_Toc112600665)

[5.2 Conclusion 25](#_Toc112600666)

[5.3 Future Recommendations 26](#_Toc112600667)

APPENDIX

REFERENCES

# LIST OF ABBREVIATIONS

CSS Cascading Style Sheet

HTML Hypertext Markup Language

OOP Object Oriented Programming

PHP Hypertext Preprocessor

SQL Structured Query Language

UML Unified Modeling Language

UI User Interface

# 

# LIST OF FIGURES

Figure 1: Iterative Model in Airline Ticket Reservation System …..…………...………. 3

Figure 2: Use Case Diagram of Airline Ticket Reservation System .…………...……… 7

Figure 3: Class Diagram of Airline Ticket Reservation System ..…………...……..…. 10

Figure 4: State & Sequence Diagram of Airline Ticket Reservation System ….……. 11

Figure 5: Activity Diagram of Airline Ticket Reservation System …………....……. 12

Figure 6: Component Diagram of Airline Ticket Reservation System ...…………….. 14

Figure 8: Deployment Diagram of Airline Ticket Reservation System …...………….. 15

# LIST OF TABLES

[**Table 1:** Test Cases for Unit Testing 23](#_Toc81773535)

[**Table 2:** Test Cases for System Testing 24](#_Toc81773536)

# CHAPTER 1: INTRODUCTION

## Introduction

The world is growing at a very fast pace with high level of technological advancements. Globalization is now reaching to almost every corner of the world at a very rapid rate.

In today’s airline industry users or customers can reserve seat or book flight from any place in the world as long as they are connected to internet. The Innovation of technology has made traveling in the air easier for customers with airline reservation or booking just a click away. The fastest means of transportation today is by air. Thousands of people flock the airline industry these days so that they can arrive their destination within a short period of time. Vehicles, train etc. cannot drive across the Atlantic Ocean or the Pacific Ocean, so customers tend to travel to other state, countries, and continents using air transportation there by with an hour’s interval the customers would have arrive their destination.

In science and technology, the need for development is a consistent subject which triggers progressions. This is obvious in each implication and the airline industry isn't an exception. Airline reservation systems were first presented in the late 1950s as moderately straightforward independent systems to control flight stock, keep up flight plans, and situate assignments and airplane stacking. Today current airline reservation systems are far-reaching suites of items to give systems that help an assortment of airline administration assignments and service customer needs from the season of introductory reservation through the finish of the flight [1].

Airline reservation system, is a web base app in which you can book, buy or have knowledge on upcoming flights and regarding their tickets availability.

## Problem Statement

1. In the current context, People are more focused on living a smart life. In most of flight reservation system if a person wants to book a flight ticket he either manually goes to airport or fill the ticket form on system and get the print out on paper document to submit it at airport. Or even the people can book the ticket from a certain registered ticket counters in online. Even this approaches make it online, it was not done completely online. Passengers may not have much more freedom to this approach. Hence passengers may or may not be satisfied with this approach as it includes manual intervention like travelling to airport for booking his ticket. This proposed system ensures the complete freedom for users, where user at his own system can login to this website and can book his ticket. Our proposed system allows only registered users book their tickets, view timing and cancel their tickets. In this system entire work is done online and ticket with id is provided for passengers as print document. Here passengers can send their queries and suggestion through feedback form.

## Objectives

The objectives of project are:

#### To develop user friendly system with various system controls

#### To provide high level security to user information

* To make booking of ticket easier and safe for payments

## Scope and Limitations

### 1.4.1 Scope of Project

Airline Reservation System is one the modifications that were carried out in the Passenger Service System so that the working and availability of Service area can be broadened. With the help of this system, customers can view all the different flight's availability with different timings for a particular date and it also allows them to reserve a seat, cancel a reservation or modify it.

### 1.4.2 Limitations

There are some limitations related to the project:

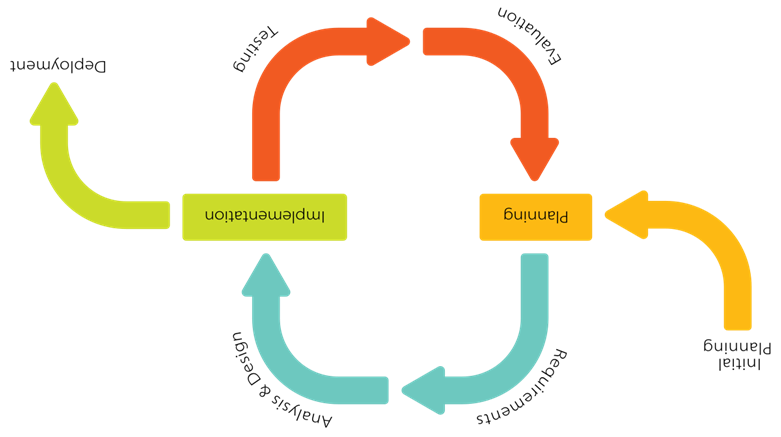
* This project is just a prototype.

## Development Methodology

A software development methodology or system development methodology is a framework that is used to structure, plan and control the process of developing a system.

The Iterative Model is used in our system because the initial requirements are already clearly defined and more features are added to the base software product with the ongoing iterations until the final system is created.

At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).



**Figure 1.1. Iterative Model**

Unlike the more traditional waterfall model, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process.

* Planning & Requirements

As with most any development project, the first step is go through an initial planning stage to map out the specification documents, establish software or hardware requirements, and generally prepare for the upcoming stages of the cycle.

* Design & Development

Once planning is complete, the design stage occurs here, establishing any technical requirements (languages, data layers, services, etc.) designing software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.

* Coding

With the planning and analysis out of the way, the actual implementation and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

* Testing

After completing the coding stage, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

* Implementation and Evaluation

In the implementation stage, requirements are written in the coding language and transformed into software. It is time for a thorough evaluation of development up to this stage once all prior stages have been completed. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.

## Organization of Report

The material presented in the project is organized into five chapters.

Chapter 1 represents the problem statement, objectives, scope and limitations of the project.

Chapter 2 describes the fundamental theories and concepts as well as information about existing system, journals and references.

Chapter 3 summarizes the keynote on system analysis and design where description of use case diagram, performance and reliability, different feasibility analysis, diagrams, database as well as architectural design are set out.

Chapter 4 provides an account on implementation and testing, tools used for preparation of the project. Test cases for unit testing as well as integration testing are done. Implementation details of modules are traced.

Chapter 5 presents brief summaries of outcome of the project, conclusion, reviews as well as future recommendations, improvements that can be done on upcoming days and feedback of systems, stability of the project. The implementation of web-based application in the project are described.

# CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

## Background Study

In 1946, American Airlines began experimenting with the first automated booking system, the Electromechanical Reservoir. In 1952, the airline introduced a newer system in order to gain access to inventory in real time from across their network. The system, called the Magnetronic Reservoir, was able to store up to 1,000 flights, a major feat at the time. However, it did have some obvious flaws including that it required a team of agents, multiple phone calls and considerable effort to book even a single seat. All of the systems that create part of the airline industry’s foundation today is made up of much more than back in the 1950s and 1960s. The average airline’s systems are nothing short of works of art. There are the customer-facing aspects – the app, the website, the kiosks at the airport – on top of everything behind the scenes – crew scheduling, network planning, inventory and even finance. Everything has to mesh perfectly for anything to run smoothly. The slightest issue can lead to cancellations and delays that cause problems for days.

Many people are travelling with airplanes, either as means of daily transportation to and from worker when going on vacation, to mention a few. To make reservations for such travels, airline companies' websites holds the functionality for the user to book a travel himself. A functionality which these websites lack is the option for the user to set up specific requirements for a travel, such as; minimal travel time or travel distance (Jarvenpaa L. S, 1996). The purpose of this project is to develop an easy-to-use airline reservation system, which accommodates these functionalities [2].

## Literature Review

In the arena of global competition, organizations all over the world are competing through the use of the most comprehensive and advanced technological features. The most common example of innovation is in the area of information technology and communication. Various industries are using technology and the advancements of software and internet to maintain and monitor their business transactions. In the application of the informative systems, the airline industry is the most commonly used system. This chapter explores the concept of reservation information system, their history, components, types and their applications in real world situation to solve problems.

The history of airline reservations systems began in the late 1950s when American Airlines required a system that would allow real-time access to flight details in all of its offices, and the integration and automation of its booking and ticketing processes. As a result, Sabre (Semi-Automated Business Research Environment) was developed and launched in 1964(C. Winston,1995). Sabre's breakthrough was its ability to keep inventory correct in real time, accessible to agents around the world. Prior to this, manual systems required centralized reservation centers, groups of people in a room with the physical cards that represented inventory, in this case, seats on airplanes. The deregulation of the airline industry, in the Airline Deregulation Act, meant that airlines, which had previously operated under government-set fares ensuring airlines at least broke even, now needed to improve efficiency to compete in a free market (Shon, 2003). In this deregulated environment the ARS and its descendants became vital to the travel industry. In the early days of American commercial aviation, passengers were relatively few, and each airline's routes and fares were tightly regulated by the Civil Aeronautics Board. These were published in a volume entitled. The Official Airline Guide, from which travel agents or consumers could construct an itinerary, then call or telex airline staff, which would mark the reservation on a card and file it. As demand for air travel increased and schedules grew more complex, this process became impractical. This system was used in the hospitality industry [3].

# CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

## System Analysis

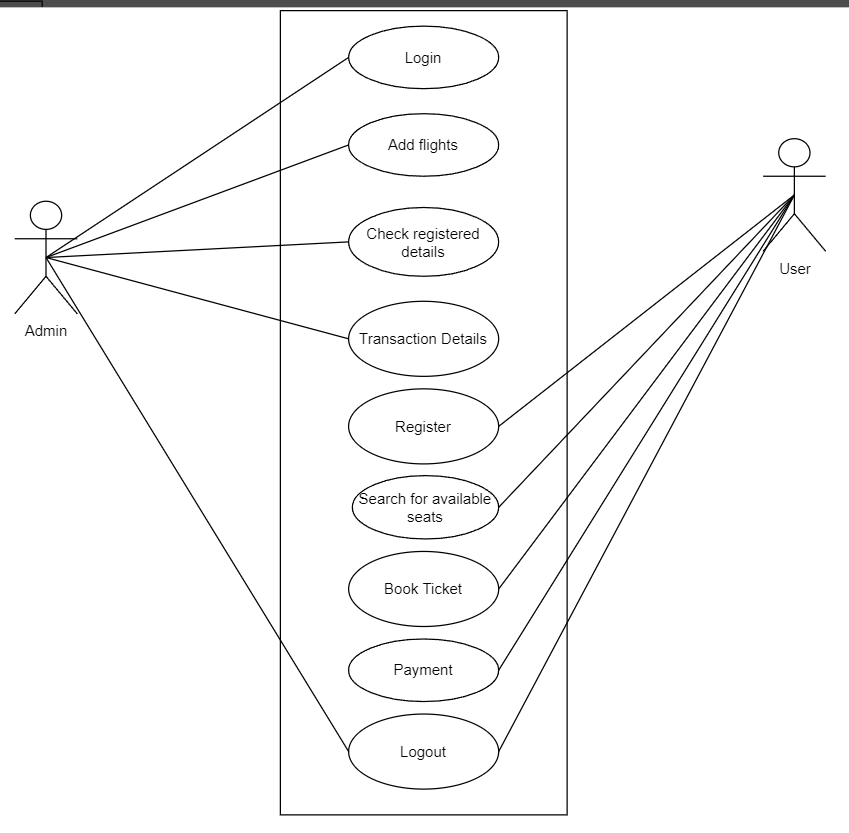
In the object-oriented approach, the focus is on capturing the structure and behaviour of information systems into small modules that combines both data and process. The main aim of Object-Oriented Design (OOD) is to improve the quality and productivity of system analysis and design by making it more usable.

### Requirement Analysis

Requirement analysis results in the specification of operational characteristics of software: indicates interface of software with other system elements and establishes constrains the software must meet. The requirement analysis is mainly categorized into two types functional and non-functional:

#### **Functional Requirement**

* **Register:** One customer has to register on this interface with his/her user credentials and then press enter to be saved in the database. Further, move on to login.
* **Login:** The customer inputs its secure user email and password and enters the system. The user can enter their own personal account.
* **Add, Update and delete:** The admin can add flights details, update the flight details and available seats and delete unnecessary details.
* **Search Flights:** Users can search flights for their wanted location and book it.
* **Payment Gateway:** The System uses card Payment Gateway.
* **Search user data:** The admin can search for all the registered flights.
* **Logout:** In the end, the user can logout the system to release the resources held for the purpose of anything else.



**Figure 3.1. USE CASE Diagram of Airline Ticket Reservation**

#### **Non-functional Requirement**

* **Usability:** Efficiency to use because many task users can complete without any help. Simple to understand the interface.
* **Scalability:** This system can further be modified in future.
* **Security:** The admin can only view the number of users and the flight details.

### Feasibility Analysis

#### **Technical Feasibility:**

Technical feasibility assesses the current resources (hardware and software) and technologies, which are required to accomplish user requirements. It requires a computer with python anaconda installed. Today every organization has computer, so it is not an extra cost.

#### **Economic Feasibility**

In this project work, the system developed is a web based application; which requires all the basic hardware and software support as required by other application. To integrate real surveillance may require software and manpower with developing skills. The proposed model is cost effective.

#### **Schedule Feasibility**

The project to be completed, realistic and achievable under a deadline according to strategy. It is developed within time limit. Hence, it is feasible in respective schedule.

#### **Operational Feasibility**

The proposed system performs effective outcome. It used to dig into the data and quickly conduct experiments to establish baseline performance on a task. The system recognizes a person without wearing a mask which requires no human involvements.

### 3.1.3 Object Modelling: Object & Class Diagram

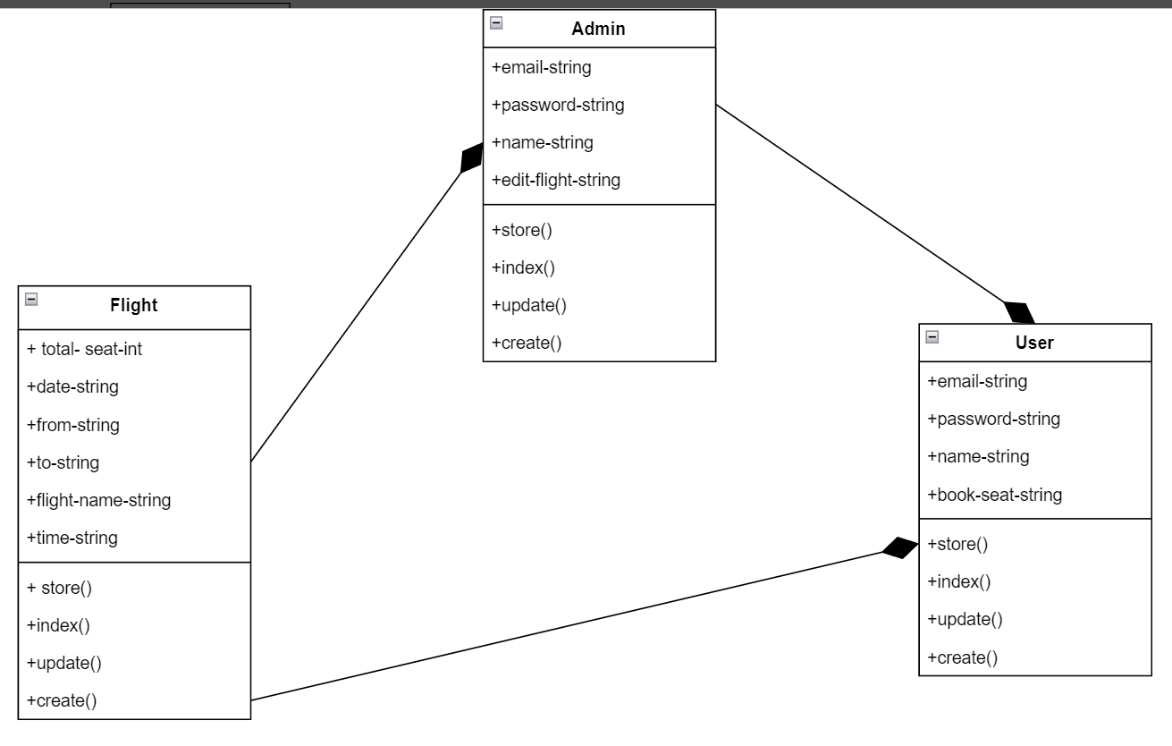
******Class diagram shows your classes and their relationships. An Object Model Diagram shows the interaction between objects at some point, during run time. A Class Diagram will show what the Objects in your system consist of (members) and what they are capable of doing (methods) mostly static.

Figure 3: Class Diagram of Airline Ticket Reservation System

### 3.1.4 Dynamic Modelling: State & Sequence Diagram

A sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when. Sequence Diagrams are interaction diagrams that detail how operations are carried out. It describes interactions among classes in terms of an exchange of messages over time.

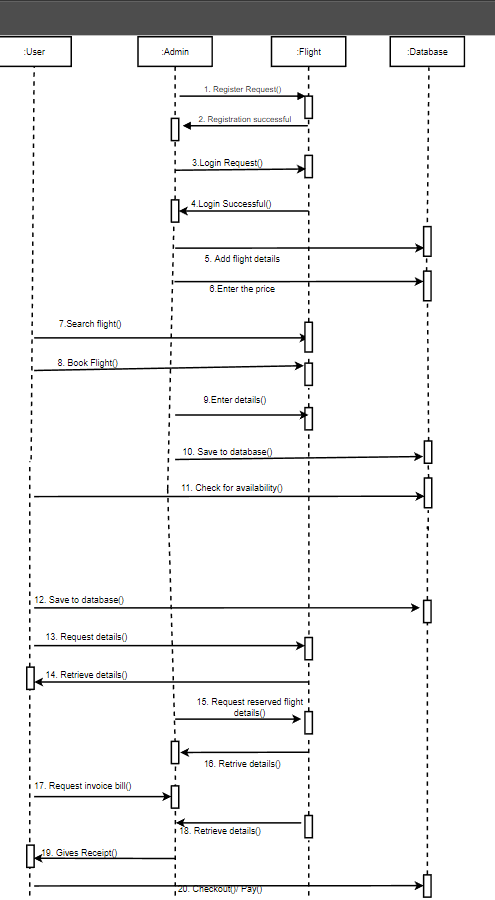
******

Figure 4: State & Sequence Diagram of Airline Ticket Reservation System

### 3.1.5 Process Modelling: Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. It describes how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modelling how a collection of use cases coordinates to represent business workflows. Model workflows between/within use cases. Model complex workflows in operations on objects. Model in detail complex activities in high-level activity. Activity diagrams are often used in business process modelling. They can also describe the steps in a use case diagram.

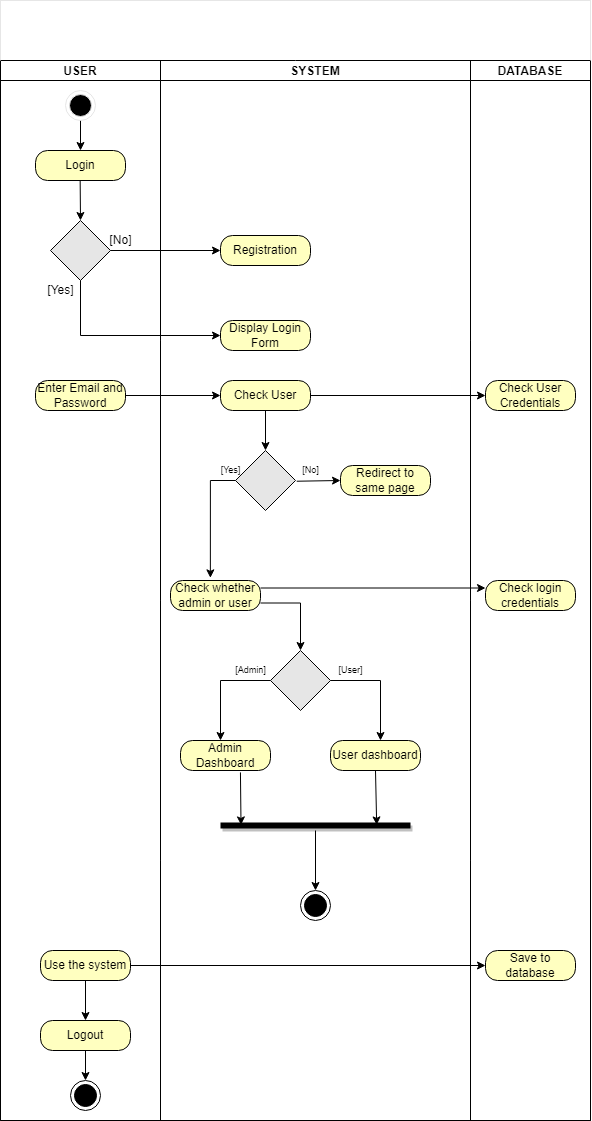


Figure 5: Activity Diagram of Airline Ticket Reservation System

# System Design

### Refinement of Classes and Object

Generalisation is the refinement of a class into more refined classes. Generalisation allows the developer to model objects into hierarchical structures based on their similarities. The class being refined is called super-class and the refined versions of it are called sub-classes. Each sub-class inherits the attributes and operations from their super-class. Methods and attributes can then be refined and the sub-class also adds specific attributes and operations. A discriminator is a variable of enumeration type, which indicate which property of an object is being abstracted. The most important use of inheritance is the conceptual simplification it makes trough the reduction of independent features in the system. By defining a feature with the same name a subclass can override a superclass feature. Overriding is a process of refining and replacing the overridden feature. Generalisation is used for extension and restriction.

### Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. The purpose of this diagram is different. Component diagrams are used during the implementation phase of an application. However, it is prepared well in advance to visualize the implementation details. Initially, the system is designed using different UML diagrams and then when the artifacts are ready, component diagrams are used to get an idea of the implementation.

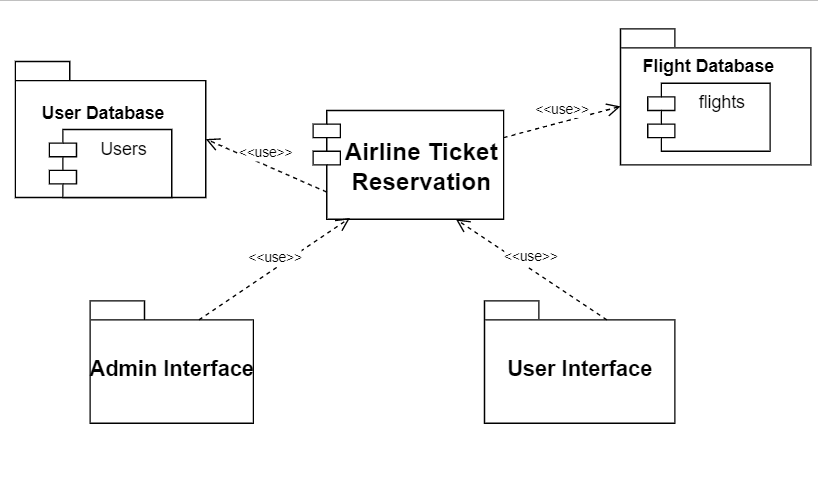


Figure 7: Component Diagram of Airline Ticket Reservation System

### 3.2.3 Deployment Diagram

In the context of Unified Modeling Language, A deployment diagram is a diagram that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. It consists of nodes and their relationships. An open-source python library, is used to design and construct a simple web app that allows users to submit a picture with a single click of a button and receive the outcome in a matter of seconds. They show the structure of the run-time system. They capture the hardware that will be used to implement the system and the links between different items of hardware. They model physical hardware elements and the communication paths between them. They can be used to plan the architecture of a system. They are also useful for Document the deployment of software components or nodes

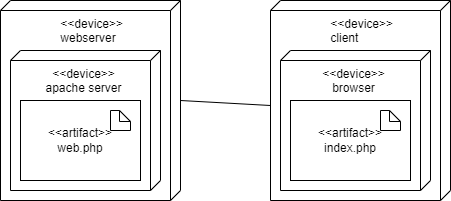
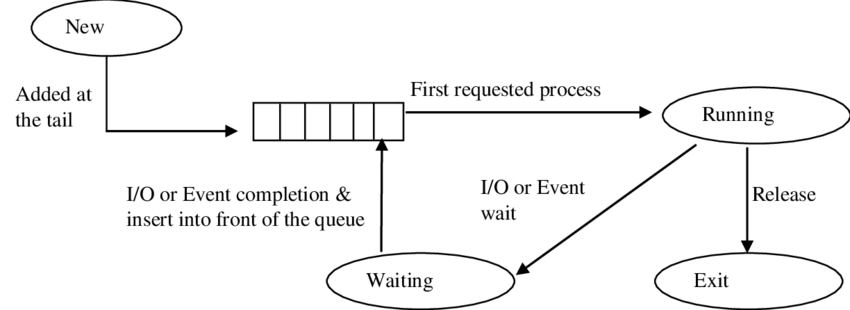


Figure 8: Deployment Diagram of Airline Ticket Reservation System

## Algorithm Details

In Airline Reservation System, the algorithms play a very vital role for searching the flights and recommending the nearest location for your destination. The algorithms that can be in airline reservation parking system are:

* Recommendation Algorithm
* First Come First Serve (FCFS)
* Linear Searching Algorithm

1. First Come First Serve (FCFS):

It is an operating system scheduling algorithm that automatically executes queued requests and processes in order of their arrival. It is the easiest and simplest scheduling algorithm. This is managed with a FIFO queue. It is easy to implement and use. The customer can search for available parking space according to FCFS algorithm. It searches up the available parking detail from database and sends response if the parking space is available or already booked.

**Figure: First Come First Serve Diagram of Airline Reservation System**

1. Linear Searching Algorithm

Searching Algorithms are designed to check f or an element or retrieve an element from any data structure where it is stored. Based on the type of search operation, these algorithms are generally classified into two categories:

* Linear Search (Sequential Search)
* Interval Search

Linear Search is the algorithm I have implemented in this system. A linear search is the simplest approach employed to search for an element in a data set. It examines each element until it finds a match, starting at the beginning of the data set, until the end. The admin can search for all the registered vehicles and reserved vehicles for parking.

The procedures for implementing linear search are as follows:

* + - 1. First, read the search element (Target element) in the array.
      2. Compare the search element with the first element in the array.
      3. If both are matched, display "Target element is found" and terminate the Linear Search function.
      4. If both are not matched, compare the search element with the next element in the array.
      5. In this step, repeat steps 3 and 4 until the search (Target) element is compared with the last element of the array.
      6. If the last element in the list does not match, the Linear Search Function will be terminated, and the message "Element is not found" will be displayed.

# CHAPTER 4: IMPLEMENTATION AND TESTING

## 4.1 Implementation

### 4.1.1 Tools Used (CASE Tools, Programming Languages, Database Platforms)

The different types of CASE tools used in the project are:

* **Diagram Tool:**

The components of the system, and the flow of the data and control between these components are demonstrated by diagram tools by using graphs.  “Draw.io” and “Creately” are the diagram tool used in the project.

* **Version Control Tool:**

An instance of software is released under one version. CASE tools help in this by automatic tracking, version management and release management. Git is used as configuration management tool.

* **Coding Tools:**

These tools assist in coding modules and algorithm with all allied elements like libraries, functions, graphic and so on. Visual Studio Code is used as development tools in the project.

The different tools & programing language used in project:

#### **Front End**

HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. HTML is used as a frontend tool for this project. Entire code is done in HTML.

CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML. Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. For designing purposes and the user interface of the project, CSS and bootstrap are used.

JavaScript

JavaScript (JS) is a scripting language, primarily used on the Web. It is used to enhance HTML pages and is commonly found embedded in HTML code. JS is used as both frontend and backend tool for the project. Various functions and actions are done through JavaScript.

#### **Back End**

Laravel

Laravel is a web application framework with expressive, elegant syntax. It is the PHP Framework for Web Artisans. Laravel is a back-end PHP-based and open-source framework used for building a wide range of custom web applications. It’s an entirely server-side framework that manages data with the help of Model-View-Controller (MVC) design which breaks an application back-end architecture into logical parts. For the backend, Laravel PHP Framework is used.

MYSQL

MySQL is an Oracle-backed open-source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. XAMPP server is used for databases. For various database queries, MySQL is used.

### 4.1.2 Implementation Details of Modules (Description of procedures/functions)

**a) Records Module**

**User Records**: -

This record helps for the authorization for using Airline Reservation System. It Provides the Username and Password for the User. It also includes the level of authority that means it separates the normal users and admin.

**Flight Records**: -

This most important record which focuses in our Airline Reservation System. It stores the essential Flight records like:

-Flight Name

-Flight Time

-Total Seats

-Flight Fare

## 4.2 Testing

It’s a process of executing a program or application with the intent of finding the software bugs. It can be also stated as the process of validating and verifying that a software program or application or product meets the business and technical requirements that guided its design and development. Testing can be done using varieties of level.

### 4.2.1 Test Cases for Unit Testing

The modules of application are tested alone in an attempt to find any errors in its code. In this web application, here is taken different procedures and interface as the small unit and tested individually.

Table 1: Test Cases for Unit Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.N | Function | Input | Expected Outcome | Result | Status |
| 1. | Start Laravel development server | Command: php artisan serve  http://127.0.0.1:8000 | Home Page | Home Page | True |
| 2. | Launch Register Page | http://127.0.0.1:8000/register | Register Page | Register Page | True |
| 3. | Register without filling all details | Null | Please fill out this field | Redirect to same page | True |
| 4. | Register using valid  credentials | Name: Griffin Conner  Email:mugehazi@mailinator.com  Password:123456789 | Redirection to the user dashboard | Redirected to the user dashboard | True |

### 4.2.2 Test Cases for System Testing

System testing during development involves integrating components to create a version of the system and then testing the integrated system. System testing checks that components are compatible, interact correctly, and transfer the right data at the right time across their interfaces.

Table 2: Test Cases for System Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.N | Function | Input | Expected Outcome | Result | Status |
| 1. | Launch register page | http://127.0.0.1:8000/register | Register Page | Register Page | True |
| 2. | Register new user | Name: Nischal Adhikari  Email: mugehazi@mailinator.com  Password: 123456789 | Successfully registered new user | Redirected to the user dashboard | True |
| 3. | Login using same credentials | Email: mugehazi@mailinator.com  Password: 123456789 | Successfully  Logged in | Redirected to the user dashboard | True |
| 4. | Register User details | Input his/her name | User registered successfully | Registered user displayed | True |
| 5. | Edit/delete flight  details | Click on edit/delete to update details | Details edited/deleted | Flight details updated |  |

# CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

## 5.1 Lesson Learnt/Outcome

Working with a project has been a great experience in learning new things and gaining practical knowledge. During the period of project, it is known that theory knowledge is incomplete without practical knowledge. But to work in practical environment, the theory knowledge is very essential. Some of the lessons learnt from the timeline of project are:

* Work in mentorship and make quality group decision.
* Importance of coordination and cooperation in the work environment.
* Understanding the difference between theoretical knowledge and practical world.

## 5.2 Conclusion

With the completion of the project, the experience of working in a team and maintaining cooperative behaviour has been got. The use and implementation of recognition algorithm has been learnt by the project. This project is done as Efficient as possible. The description of the background and the context of the project was thoroughly researched by the author.

The purpose, scope, applicability, and requirement specifications of the system have been accurately explained. The author has included features and operations in detail including screen layouts and the limitations on which the project is being developed. Finally, the system is implemented and tested according to test cases. After the development of the system finally, it was tested and the views about results were exchanged. After testing, the limitations of the existing system were discussed.

In conclusion, frontend tools like HTML, CSS, Laravel and backend tools like Laravel, MySQL and JavaScript were used in the development of my system. I would be glad to enhance and promote this project if given chance and help ourselves and society in the near future.

**5.3 Future Recommendations**

The applicability and usage of this project can be increased by enhancing the system based on the limitations.

The old (manual)and the newly designed software computerized for transaction of business between the company and their customers are being run. It’s just desktop app for now and can be developed as android and iOS application. This is to enable a sound understanding of the new system design for the effectiveness of this changeover; there are other tasks that must be accomplished.

They include the following.

* Installation of cooling devices such as the air condition for cooling the computer system
* The company should be sending their staff for training so that they acquire more skills and experience in operating the new designed system
* Provision of security measures for the system e.g. Alarm gadgets and burglary proof should be in place

**REFERNCES**

[1] https://www.ijariit.com/manuscripts/v4i3/V4I3-1422

[2]<https://www.academia.edu/37568974/AIRLINE_RESERVATION_SYSTEM_my_project>

[3] https://www.studymode.com/essays/Airline-Reservation-System-Introduction-

[4] https://www.provabtechnosoft.com/blog/introduction-to-airline-reservation-system

[5] <https://www.coursehero.com/file/46565347/Problem-Statement-Flight-Reservation>

[6] <https://www.accessone.io/Blog/post/airline-reservation-system>

<https://www.buddhaair.com/>

<https://www.taraair.com/>