WILLFLY

*Project Report Submitted by*

# SALINI K B

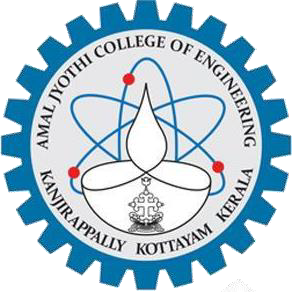
**Reg. No.: AJC21MCA-2091**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS**

**(MCA TWO YEAR)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with „A‟ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

# 2021-2023

## DEPARTMENT OF COMPUTER APPLICATIONS

### AMAL JYOTHI COLLEGE OF ENGINEERING

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**Willfly**- **Online Flight Booking”** is the bona fide work of **SALINI K B(Regno: AJC21MCA-2091)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

**Ms Ankitha Philip Ms. Meera Rose Mathew**

**Internal Guide Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department External Examiner**

**// CERTIFICATE ON PLAGIARISM CHECK**

**DECLARATION**

This is to certify that the Project report, “**Willfly”** is the bona fide work of **SALINI K B (Reg. No: AJC21MCA-2091)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

**Date: Salini K B**

**KANJIRAPPALLY Reg: AJC21MCA-2091**

# ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Ankitha Philip** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

.

SALINI K B

# ABSTRACT

.

Flight booking system is the website the system allows the airline passengers to search for flights that are available between the two travel cities namely departure city and available city. The objective of the airline system to manage the details of airline ticket,flights,customers details etc The airline ticket booking system that we are proposing ensures the gives total freedom for passengers ,where passenger can use his own personal computer to log on the website and can book his ticket..The airline ticket booking system allows only registered users or passengers to reserve the flights tickets,view flight timing and cancel there tickets. Here passengers can send their queries and suggestions through a feedback form and user can predict the fares. User friendliness provided in the application with various controls.The system always provides offer for regular customer in attractive manner.

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. NO** | | **TOPIC** | **PAGE NO** | |
| **1** | | **INTRODUCTION** | **1** | |
| **1.1** | | **PROJECT OVERVIEW** | **2** | |
| **1.2** | | **PROJECT SPECIFICATION** | **2** | |
| **2** | | **SYSTEM STUDY** | **3** | |
| **2.1** | | **INTRODUCTION** | **4** | |
| **2.2** | | **EXISTING SYSTEM** | **5** | |
| **2.3** | | **DRAWBACKS OF EXISTING SYSTEM** | **5** | |
| **2.4** | | **PROPOSED SYSTEM** | **5** | |
| **2.5** | | **ADVANTAGES OF PROPOSED SYSTEM** | **6** | |
| **3** | | **REQUIREMENT ANALYSIS** | **7** | |
| **3.1** | | **FEASIBILITY STUDY** | **8** | |
| **3.1.1** | | **ECONOMICAL FEASIBILITY** | **8** | |
| **3.1.2** | | **TECHNICAL FEASIBILITY** | **9** | |
| **3.1.3** | | **BEHAVIORAL FEASIBILITY** | **9** | |
| **3.1.4** | | **FEASIBILITY STUDY QUESTIONNAIRE** | **9** | |
| **3.2** | | **SYSTEM SPECIFICATION** | **11** | |
| **3.2.1** | | **HARDWARE SPECIFICATION** | **11** | |
| **3.2.2** | | **SOFTWARE SPECIFICATION** | **11** | |
| **3.3** | | **SOFTWARE DESCRIPTION** | **11** | |
| **3.3.1** | | **PHP** | **11** | |
| **3.3.2** | | **MYSQL** | **11** | |
| **4** | | **SYSTEM DESIGN** | **13** | |
| **4.1** | | **INTRODUCTION** | **14** | |
| **4.2** | | **UML DIAGRAM** | **14** | |
| **4.2.1** | | **USE CASE DIAGRAM** | **15** | |
| **4.2.2** | | **SEQUENCE DIAGRAM** | **17** | |
| **4.2.3** | | **STATE CHART DIAGRAM** | **19** | |
| **4.2.4** | | **ACTIVITY DIAGRAM** | **21** | |
| **4.2.5** | | **CLASS DIAGRAM** | **24** | |
| **4.2.6** | | **OBJECT DIAGRAM** | **27** | |
| **4.2.7** | | **COMPONENT DIAGRAM** | **28** | |
| **4.2.8** | | **DEPLOYMENT DIAGRAM** | **29** | |
| **4.2.9** | | **COLLABORATION DIAGRAM** | **30** | |
| **4.3** | | **USER INTERFACE DESIGN USING FIGMA** | **31** | |
| **4.4** | | **DATABASE DESIGN** | **33** | |
| **5** | | **SYSTEM TESTING** | **42** | |
| **5.1** | | **INTRODUCTION** | **43** | |
| **5.2** | | **TEST PLAN** | **44** | |
| **5.2.1** | **UNIT TESTING** | | **44** |
| **5.2.2** | **INTEGRATION TESTING** | | **45** |
| **5.2.3** | **VALIDATION TESTING** | | **45** |
| **5.2.4** | **USER ACCEPTANCE TESTING** | | **45** |
| **5.2.5** | **AUTOMATION TESTING** | | **46** |
| **5.2.6** | **SELENIUM TESTING** | | **46** |
| **6** | **IMPLEMENTATION** | | **57** |
| **6.1** | **INTRODUCTION** | | **58** |
| **6.2** | **IMPLEMENTATION PROCEDURE** | | **59** |
| **6.2.1** | **USER TRAINING** | | **59** |
| **6.2.2** | **TRAINING ON APPLICATION SOFTWARE** | | **59** |
| **6.2.3** | **SYSTEM MAINTENANCE** | | **60** |
| **6.2.4** | **HOSTING** | | **60** |
| **7** | **CONCLUSION & FUTURE SCOPE** | | **63** |
| **7.1** | **CONCLUSION** | | **63** |
| **7.2** | **FUTURE SCOPE** | | **64** |
| **8** | **BIBLIOGRAPHY** | | **65** |
| **9** | **APPENDIX** | | **67** |
| **9.1** | **SAMPLE CODE** | | **68** |
| **9.2** | **SCREEN SHOTS** | | **86** |

## List of Abbreviation

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language.

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

# CHAPTER 1

# INTRODUCTION

### PROJECT OVERVIEW

Flight Ticket Booking System will keep track record of online booked tickets, tickets available,and their important details related to flight tickets. It deals with the recording and proccessing employee data so that the executives can easily manage the organizational operations. The project “ Willfly” Flight booking system is the website the system allows the airline passengers to search for flights that are available between the two travel cities namely departure city and available city.The objective of the airline system to manage the details of airline ticket,flight customers details etc.

### PROJECT SPECIFICATION

The system is a booking website in which user can book there flights online . User can make payment online. There are various modules in the system and associated functionalities for the users.

The system includes 2 users. They are:

##### **Admin**

Admin manages all the functionalities of Flight and the user. Admin can add new flights and user can view and book the flights if needed and admin can perform sentimental analysis based on the customer feedback .

##### **User**

Users has access to various functionalities of the system like Booking,feedback, payment etc. users can find that available between two places. They can search for various flights and predict the fare of airlines. They can make online payment.

# CHAPTER 2

# SYSTEM STUDY

### INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of anysystem development process. The system is studied to the minute‟s detail and analyzed. The system analyst plays the role of the interrogator and dwells deep intothe working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires,ticket booking areas etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

### EXISTING SYSTEM

In the existing system booking the ticket at some particular registered ticket counters in online. Even above approaches make a ticket booking online it was not completely done on online. So passenger may not have much freedom over this approach as it includes manual intervention like travelling to airport for booking his ticket .No proper coordination between different for application and fewer user friendly.

**2.2.1 NATURAL SYSTEM STUDIED**

The current flight booking system requires passengers to book their tickets either through specific registered ticket counters or by visiting the airport in person. While there are online booking options available, they still involve some manual intervention and may not provide passengers with complete freedom to book their tickets online. Additionally, there may be coordination issues between different applications, and the system may not be very user-friendly.

**2.2.2 DESIGNED SYSTEM STUDIED**

The airline ticket booking system that we are proposing ensures the gives total freedom for passengers where passenger can use his own personal computer to log on the website and can book his ticket.the airline ticket booking system Allows only registered users or passengers to reserve the flights tickets,view flight.Timing and cancel there tickets. Here passengers can send their queries and suggestions through a feedback form user friendliness provided in the application with various controls.The system always provides offer for regular customer in attractive manner.

### DRAWBACKS OF EXISTING SYSTEM

* + - Poor interface and the system is not user friendly.
    - It requires lot of Manpower
    - Less flexible to users.
    - This system cannot be used for personal and quick reference.

### PROPOSED SYSTEM

That we are proposing ensures the gives total freedom for passengers where passenger can use his own personal computer to log on the website and can book his ticket.the airline ticket booking system Allows only registered users or passengers to reserve the flights tickets,view flight.Timing and cancel there tickets. Here passengers can send their queries and suggestions through a feedback form user friendliness provided in the application with various controls Users can create an account and save their search preferences, and the website offers a price alert feature that notifies users when the price of a flight changes..The system always provides offer for regular customer in attractive manner.

### ADVANTAGES OF PROPOSED SYSTEM

* + - User friendly and secure interface.
    - Ensure data accuracy.
    - It is made in a quick and referential manner.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The results of the feasibility study should be a report that recommends whether or not it is worth carrying on with the requirements engineering and system development process.

If a system does not support the business objectives, it has no real value to the business. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

* To analyse whether the software will meet organizational requirements.
* To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
* To determine whether the software can be integrated with other existing software.

The information assessment phase identifies the information that is required to answer the three questions set out above. Once the information has been identified, you should question information sources to discover the answers to these questions.

### Economical Feasibility

It determines whether the required software is capable of generating financial gains for organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. The system being developed is economic with the flight booking point of view. Firstly web based system eliminate the use of paper works completely. The data generated will be error free and accessible anytime. The required timeframe would need to be set by the organization. Since this system can be used in Airline field, it will help passengers to book tickets easily avoiding the time constraints. So it is economically feasible.

### Technical Feasibility

It assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. The Online flight ticket booking for customer is a complete web based applicatons. The main technologies used in this project are PHP, HTML, CSS, JavaScript,Ajax. All the above technologies are freely available and the skills required to use these technologies are manageable. Other resources includes Programming device, Programming tools and Various IDE that support all technologies. Hosting Space is also freely available. The specified requirements can be implemented using the available hardware and software. Time limitaton of the system development and the ease of implementing using these technologies are manageable. So the proposed project is technically feasible.

### Behavioral Feasibility

### It assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system effciently. The user must not feel unfriendly by the system, instead must accept it as a necessity for them. The acceptance by the users depends on the methods that are provided to know the user about the system and to make them familiar with it.The proposed mode of operation make maximum use of available resources, including people, time, and flow of form.

### Feasibility Study Questionnaire

* + - 1. **Is there any cash back offers to increase the purchase rebate?**

Yes,you can earn cash back, in addition to miles on most airline purchases. How much you‟ll earn depends where you book.

##### **What about the seating arrangement in flight?**

For airplanes having a seating capacity of more than 50 but less than 101 passenger two flight attendants. For airplanes having a seating capacity of more than 100 passenger two fligattendants plus one additional flight attendant for each unit (or part of a unit) of 50 passenger seats above a seating capacity of 100 passengers.

##### **What about the availability of food packages ?**

Yes,the airlines to serve pre-packed snacks, meals and beverages on domestic and international flights.

##### **How long does it take to book a flight?**

Less than 15 minutes pass between their starting to fill in the search form and a ticket appearing in their mailbox.

##### **5 . Is it government organized system?**

Yes,government organized system.

##### **How flight reservation system boosts the sales of airlines?**

Reservation System shoots up the sales of an airline company and gives a competitive edge. The primary reason for choosing flight reservation system is to increase the number of bookings of the flight ticket. Online Flight Reservation System is very easy to integrate with the website.

##### **What Are the Differences Between Travel Classes on Airline Flights?**

There are four classes of travel you can find on airline flights all over the world: economy, premium economy, business, and first.

##### **Free cancelation of flight ticket is available or not?**

YES,you can cancel your flight for free within 24 hours.

##### **What is RPK?**

That shows the number of kilometers traveled by paying passengers.

##### **What are customer looking airline booking?**

Timeless,meals,helpfulness,comfort.

## 3.2 SYSTEM SPECIFICATION

### Hardware Specification

Processor - I5 Processor

RAM - 4 G B

Hard disk - 1 T B

### Software Specification

Front End - HTML, CSS,JS

Backend - MYSQL,PHP

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS,PYTHON

## SOFTWARE DESCRIPTION

### Eg.PHP

PHP started out as a small open source project that evolved as more and more people found out how useful it was. PHP is a recursive acronym for "PHP: Hypertext Preprocessor". PHP is server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites. In this project the backend is implemented using PHP along with MySQL database.

### Eg. MySQL

MySQL is one of the most recognizable technologies in the modern big data ecosystem. Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it‟s clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL. MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

### Eg.PYTHON

Python is a high-level, interpreted programming language that was first released in 1991. It is designed to be easy to read, write, and understand, making it a popular language for beginners and experts alike. Python is used for a wide variety of applications, including web development, scientific computing, data analysis, machine learning, and more.

One of the key strengths of Python is its simplicity and ease of use. Its syntax is clean and straightforward, with an emphasis on readability and reducing the amount of code needed to accomplish a task. Python also has a large and active community of developers, which has led to a vast ecosystem of third-party libraries and tools that make it easy to build complex applications.

# CHAPTER 4

# SYSTEM DESIGN

* 1. **INTRODUCTION**

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

## UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML includes the following nine diagrams.

* Class diagram
* Object diagram
* Use case diagram
* Sequence diagram
* Activity diagram
* State chart diagram
* Deployment diagram
* Component diagram

## USE CASE DIAGRAM

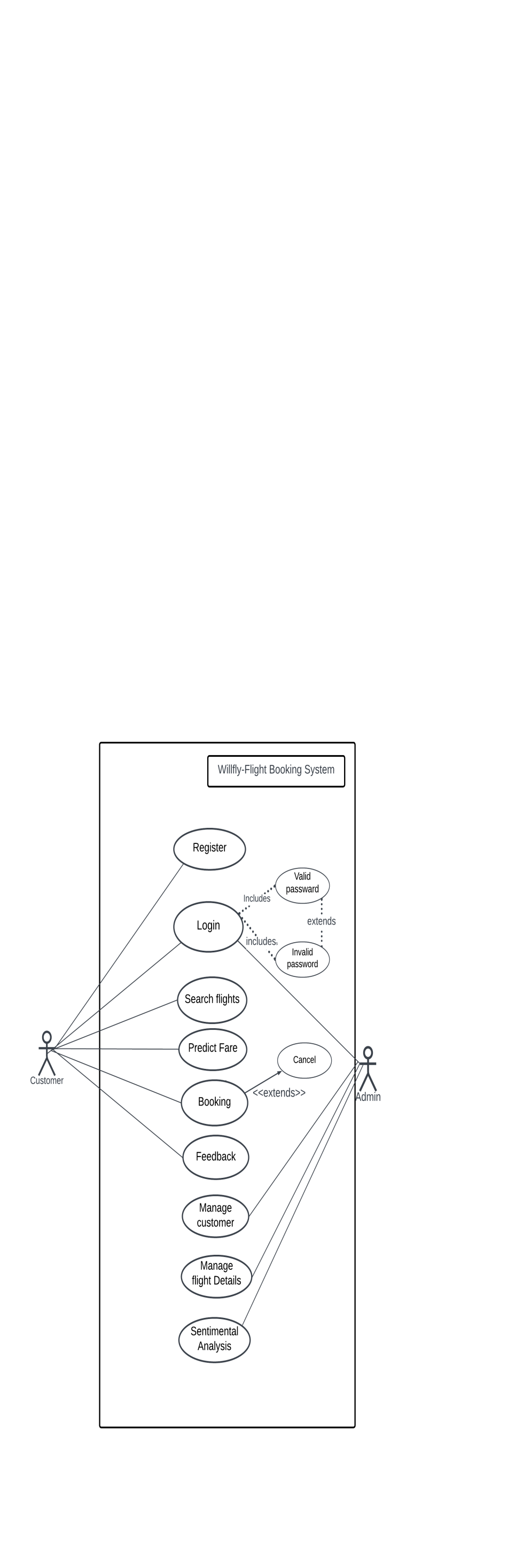
A use case diagram is a graphic depiction of the interactions among the elements ofa system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation forthe modeling of real-world objects and systems.

A usecase diagram contains four components.

* The boundary, which defines the system of interest in relation to the world around it.
* The actors, usually individuals involved with the system defined according to their roles.
* The use cases, which are the specific roles are played by the actors within and around the system.
* The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram.

* The name of a use case is very important. The name should be chosen in such away so that it can identify the functionalities performed.
* Give a suitable name for actors.
* Show relationships and dependencies clearly in the diagram.
* Do not try to include all types of relationships, as the main purpose of thediagram is to identify the requirements.
* Use notes whenever required to clarify some important points.

****

## 

## *Fig. 4.2.1 Use case diagram for Flight Booking*

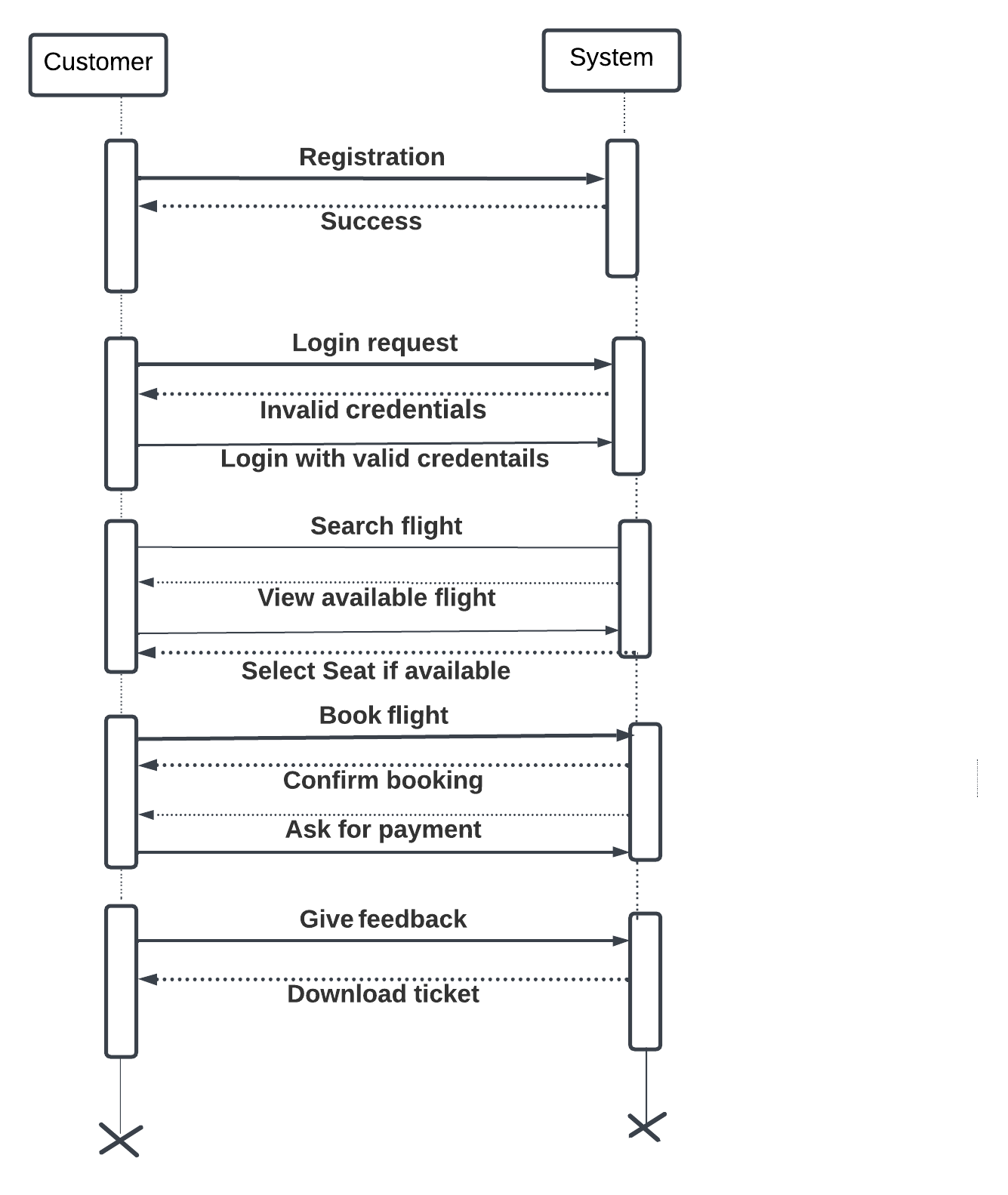
## 

## 4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

1. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model usingthe UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UMLdiagram using a stick person notation. We can have multiple actors in a sequence diagram.
2. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements arelocated at the top in a sequence diagram
3. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
4. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

****

*Fig.4.2.2 Sequence Diagram for Flight Booking*

## 4.2.3 State Chart Diagram

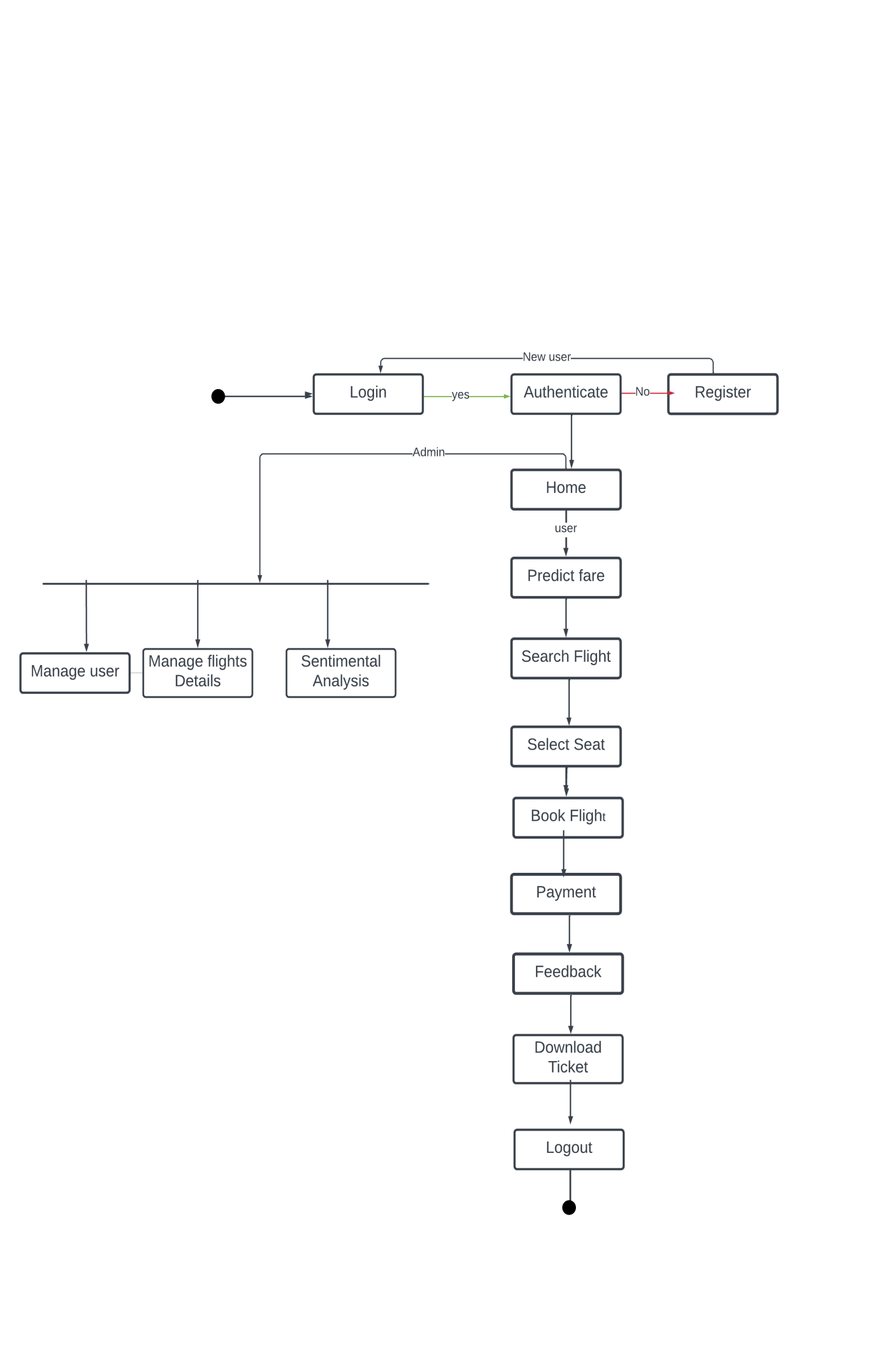
The state machine diagram is also called the State chart or State Transition diagram, which shows the order of states underwent by an object within the system. It captures the software system's behavior. It models the behavior of a class, a subsystem, a package, and a complete system.

It tends out to be an efficient way of modeling the interactions and collaborations in the external entities and the system. It models event-based systems to handle the state of an object. It also defines several distinct states of a component within the system. Each object/component has a specific state.

Notation of a State Machine Diagram

Following are the notations of a state machine diagram enlisted below:

1. **Initial state:** It defines the initial state (beginning) of a system, and it is represented by a black filled circle.
2. **Final state:** It represents the final state (end) of a system. It is denoted by a filled circle present within a circle.
3. **Decision box:** It is of diamond shape that represents the decisions to be made on the basis of an evaluated guard.
4. **Transition:** A change of control from one state to another due to the occurrence of some event is termed as a transition. It is represented by an arrowlabeled with an event due to which the change has ensued.
5. **State box:** It depicts the conditions or circumstances of a particular object of a class at a specific point of time. A rectangle with round corners is used to represent the state box.

****

*Fig.4.2.3 State Chart Diagram for Flight Booking*

## 4.2.4 Activity Diagram

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc. It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

Components of an Activity Diagram

Following are the component of an activity diagram:

1. Activities

The categorization of behavior into one or more actions is termed as an activity.In other words, it

can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes. The control flow of activity is represented by control nodes and object nodes that illustrates the objects used within an activity. The activities are initiated at the initial node and are terminated at the final node

1. Activity partition /swimlane

The swimlane is used to cluster all the related activities in one column or one row. It can be either vertical or horizontal. It used to add modularity to the activity diagram. It is not necessary to incorporate swimlane in the activity diagram. But it is used to add more transparency to the activity diagram.

1. Forks

Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters. Whenever a data is received at an inward edge, it gets copied and split crossways various outward edges. It split a single inward flow into multiple parallel flows.

1. Join Nodes

Join nodes are the opposite of fork nodes. A Logical AND operation is performed on all of the inward edges as it synchronizes the flow of input acrossone single output (outward) edge.

Notation of an Activity diagram

Activity diagram constitutes following notations:

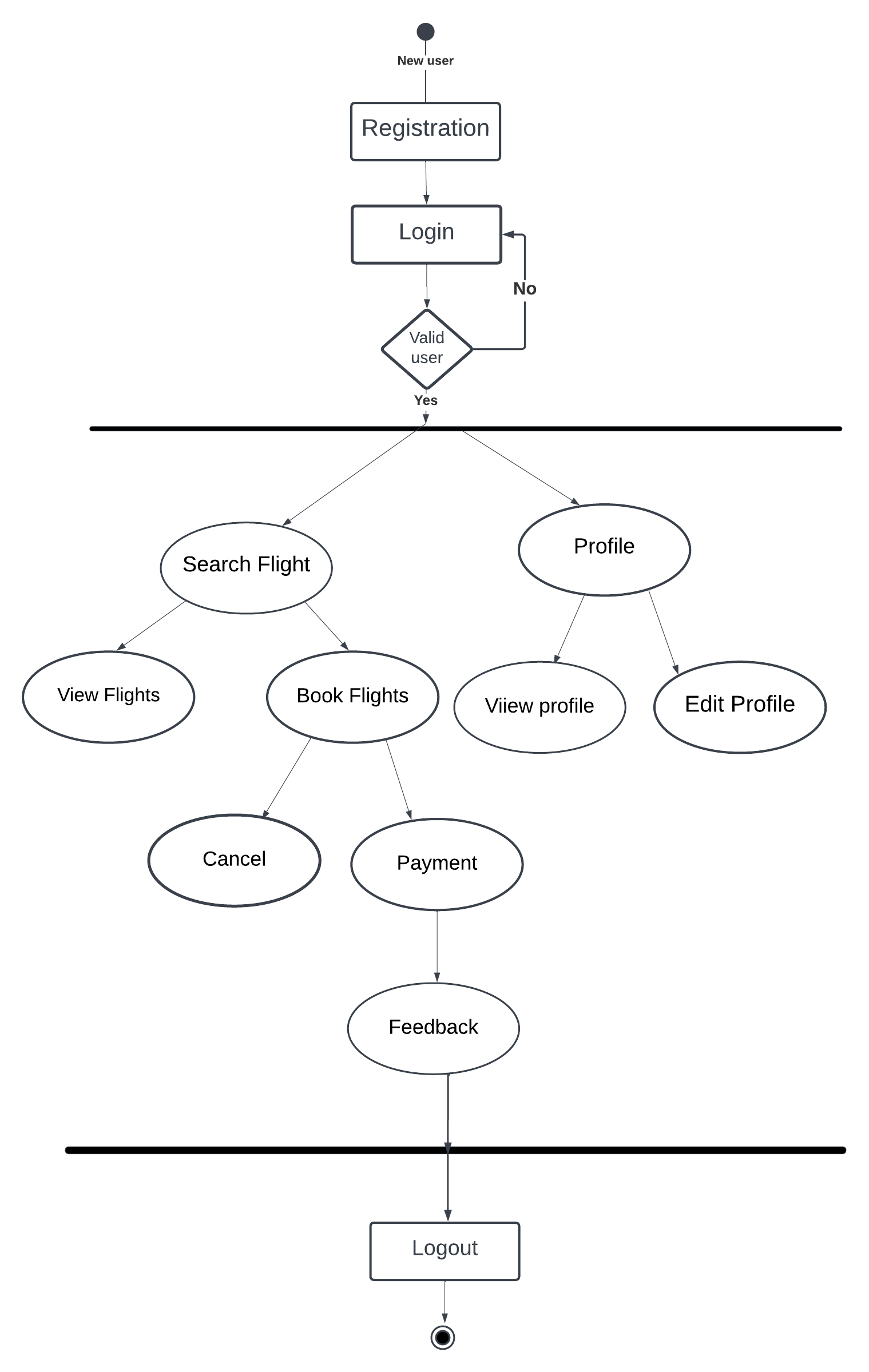
**Initial State:** It depicts the initial stage or beginning of the set of actions.

**Final State:** It is the stage where all the control flows and object flows end.

**Decision Box:** It makes sure that the control flow or object flow will follow only one path.

**Action Box:** It represents the set of actions that are to be performed.

**4.2.4.1 User Activity**

****

*Fig.4.2.4 Activity Diagram for Flight Book*

## Class Diagram

The class diagram depicts a static view of an application. It represents the typesof objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of thesystem, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

Components of a Class Diagram

The class diagram is made up of three sections:

**Upper Section:** The upper section encompasses the name of the class. A class is a representation of similar objects that shares the same relationships attributes, operations, and semantics. Some of the following rules that should betaken into account while representing a class are given below:

1. Capitalize the initial letter of the class name.
2. Place the class name in the center of the upper section.
3. A class name must be written in bold format.
4. The name of the abstract class should be written in italics format.

**Middle Section:** The middle section constitutes the attributes, which describe the quality of class. The attributes have the following characteristics:

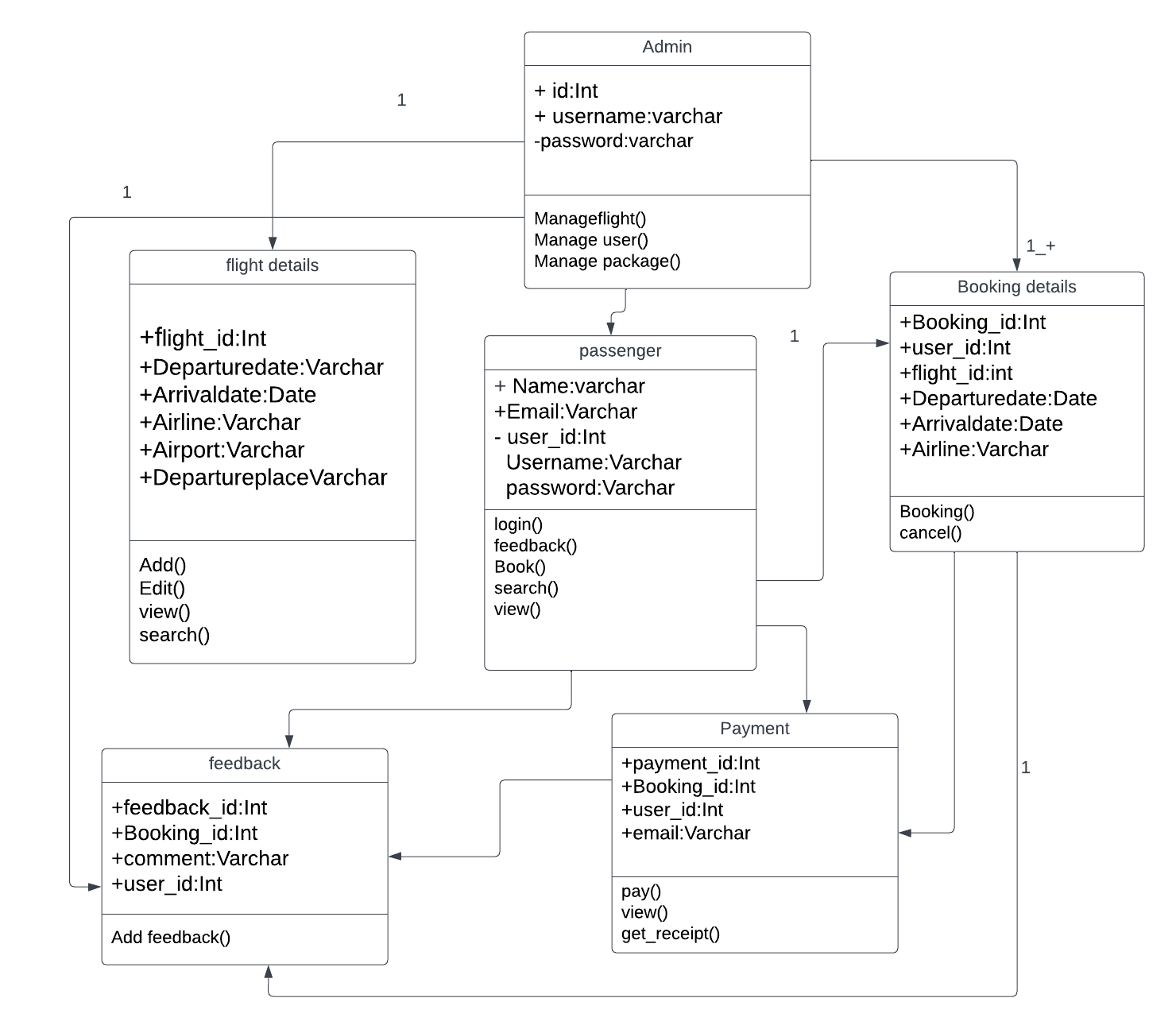
* The attributes are written along with its visibility factors, which are public (+),private (-), protected (#), and package (~).
* The accessibility of an attribute class is illustrated by the visibility factors.
* A meaningful name should be assigned to the attribute, which will explain its usage inside the class.

**Lower Section:** The lower section contain methods or operations. The methodsare represented in the form of a list, where each method is written in a single line. It demonstrates how a class interacts with data.

##### **Relationships**

In UML, relationships are of three types:

* **Dependency:** A dependency is a semantic relationship between two or more classes where a change in one class cause changes in another class. It forms a weaker relationship.
* **Generalization:** A generalization is a relationship between a parent class (superclass) and a child class (subclass). In this, the child class is inherited fromthe parent class.
* **Association:** It describes a static or physical connection between two or more objects. It depicts how many objects are there in the relationship.
* **Multiplicity:** It defines a specific range of allowable instances of attributes. In case if a range is not specified, one is considered as a default multiplicity.
* **Aggregation:** An aggregation is a subset of association, which represents has a relationship. It is more specific then association. It defines a part-whole or part- of relationship. In this kind of relationship, the child class can exist independently of its parent class.
* **Composition:** The composition is a subset of aggregation. It portrays the dependency between the parent and its child, which means if one part is deleted, then the other part also gets discarded. It represents a whole-part relationship.

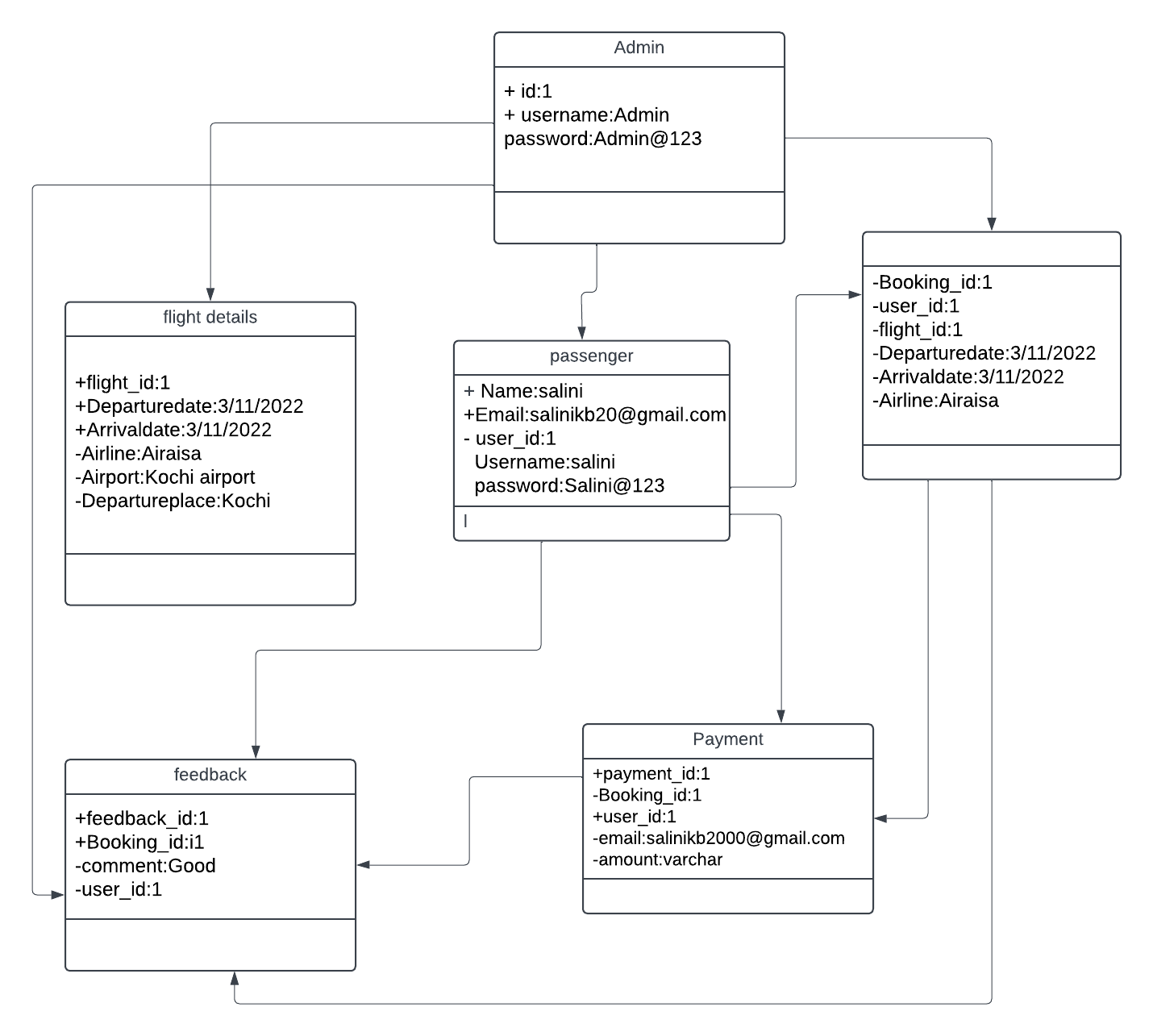


*Fig.4.2.5 Class Diagram for Flight Booking*

## Object Diagram

Object diagrams are dependent on the class diagram as they are derived from the class diagram. It represents an instance of a class diagram. The objects help in portraying a static view of an object- oriented system at a specific instant.

Both the object and class diagram are similar to some extent; the only difference is that the class diagram provides an abstract view of a system. It helps in visualizing a particular functionality of a system.



*Fig.4.2.6 Object Diagram for Flight Booking*

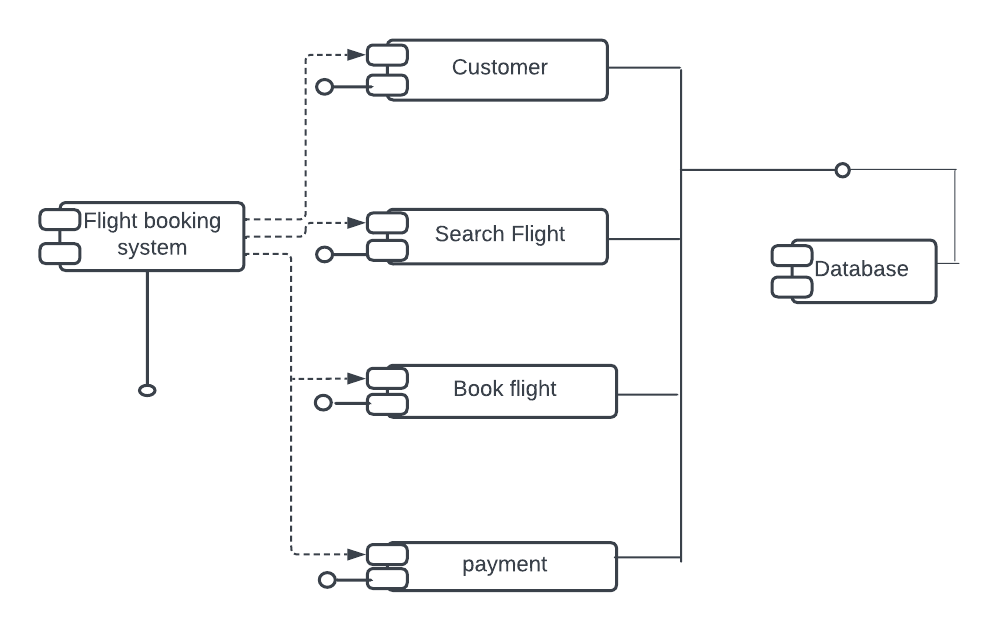
## Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

Notation of a Component Diagram

1. A component
2. A node



**4.2.8 Deployment Diagram**

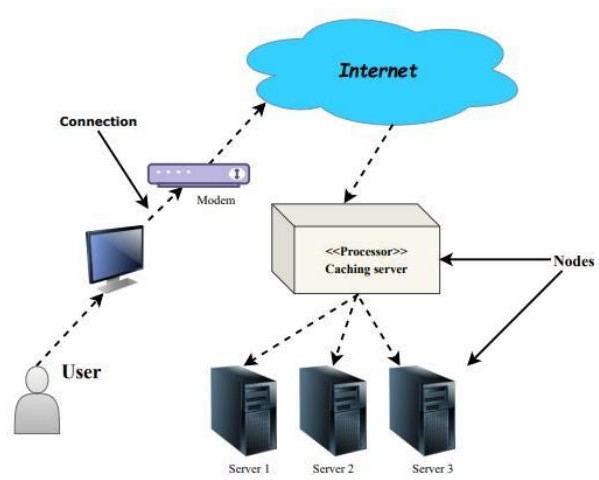
The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

Notation of Deployment diagram

The deployment diagram consists of the following notations:

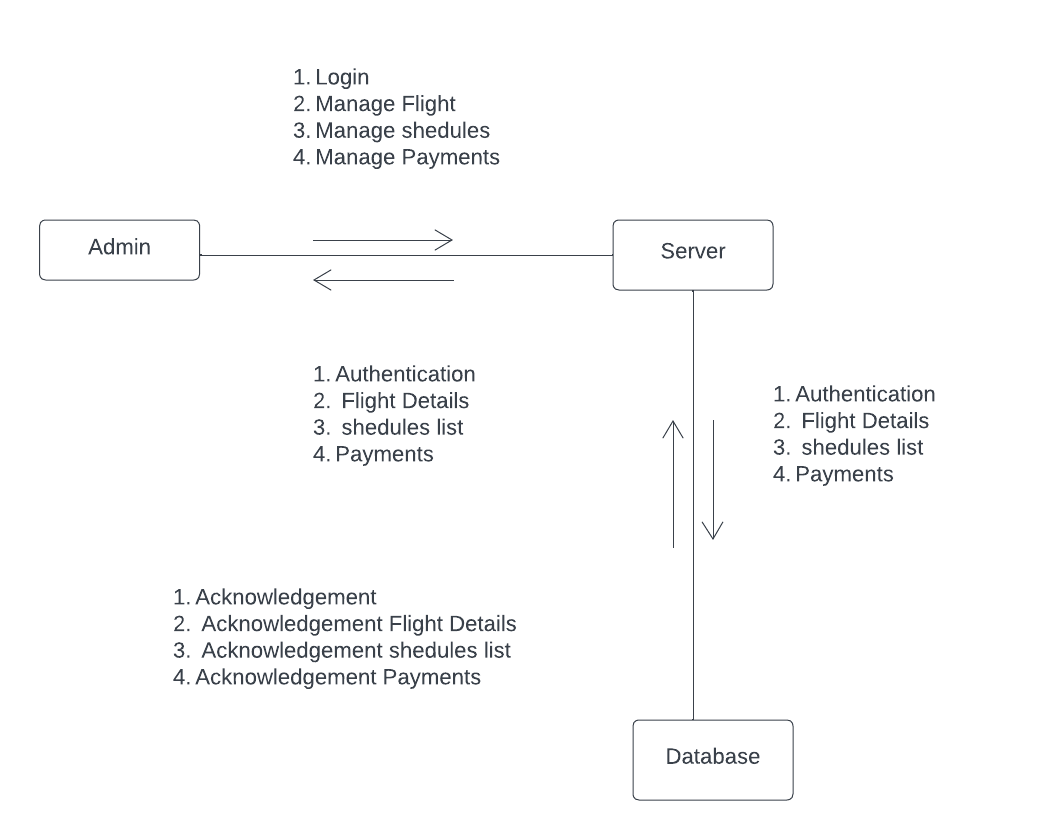
1. A component
2. An artifact
3. An interface
4. A node



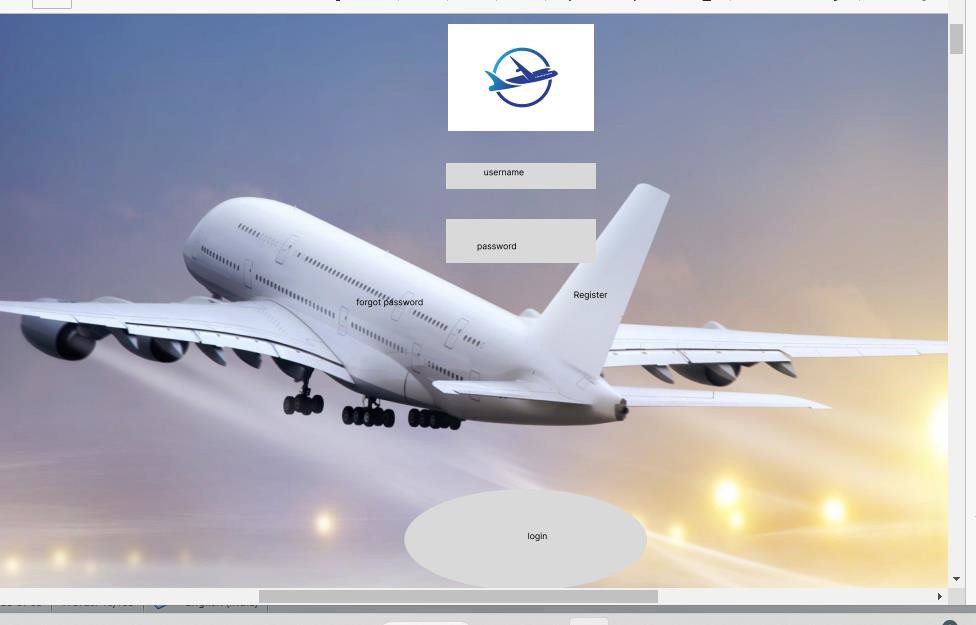
**4.2.9 Colloboration Diagram**

A collaboration diagram, also known as a communication diagram, is a type of diagram used in software engineering to visualize the interactions between objects or components within a system. It depicts how objects or components interact with each other to achieve a particular functionality.

In a collaboration diagram, the objects or components are represented as rectangular boxes, and the interactions between them are represented by arrows or lines. The arrows indicate the flow of messages or signals between the objects or components, and they may be labeled to indicate the specific message being transmitted.

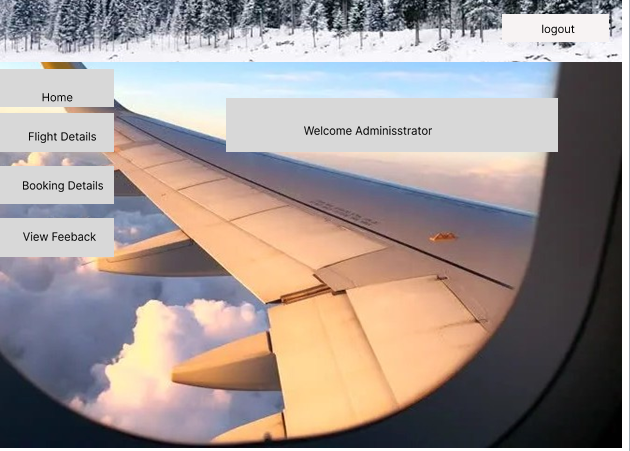


## 4.3 USER INTERFACE DESIGN USING FIGMA

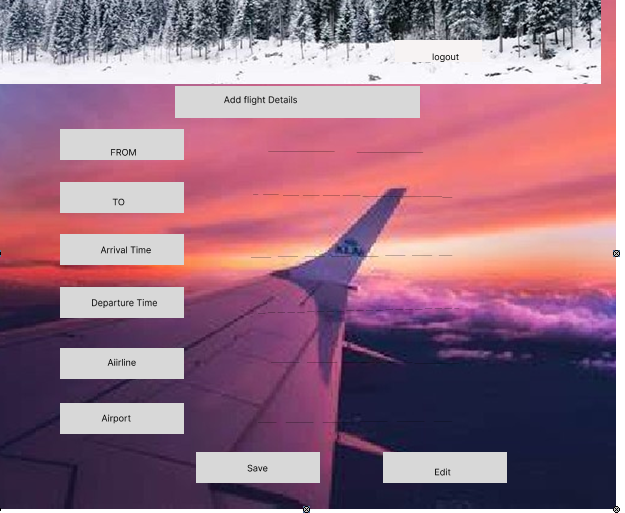
**Form Name: Login**

 **Form Name: Search flight**

**Form Name: Admin home**

****

**Form Name: Add flights**



**4.4 DATABASE DESIGN**

### 4.4.1 Relational Database Management System (RDBMS)

A database is a structured system that allows for the storage and retrieval of information in an efficient manner. Protecting the data is essential. The process of database design involves two levels.

The first level is the Information Level Design, where the user requirements are gathered, and a database is designed to meet these requirements as clearly as possible. This step is independent of any specific database management system (DBMS).

The second level is the Physical Level Design, where the Information Level Design is transformed into a design for a specific DBMS that will be used to implement the system. This level is concerned with the characteristics of the particular DBMS to be used. Database design occurs concurrently with system design. The objective of organizing the data in the database is to achieve two primary goals.

Data Integrity

Data independence

### Relations, Domains, Attributes

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In relational model, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name.

A tuple is an ordered collection of n elements, and the columns of a table are called attributes. To ensure the integrity of both referential and entity relationships, relationships are established between every table in the database. Also, domain D is defined as a set of atomic values, and every value in a relation is considered atomic, meaning it cannot be further decomposed **Relationships**

* Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
* Entity Integrity and Referential Integrity enforces that no Primary Key can have null values.
* Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

### 4.4.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. All the tables have been normalized up to the third normal form.

Normalization is the process of arranging data in a structured and logical way. Its primary goal is to organize data into appropriate tables and columns, which can be easily understood and correlated by the user. By eliminating repeating groups in data, normalization helps to avoid data redundancy, which can be a significant burden on computer resources. The benefits of normalization include:

* + - * Data normalization should be performed.
      * Appropriate names should be chosen for tables and columns.
      * The data should be given a suitable name.

#### **First Normal Form**

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form.

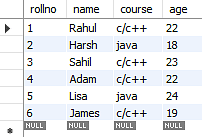


Fig 4.4.2.1 Table not in 1NF

In the students record table, you can see that the course column has two values. Thus, it does not follow the First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table.

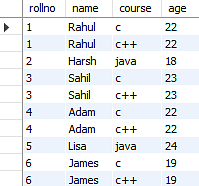


Fig 4.4.2.2 Table in 1NF

#### **Second Normal Form**

#### According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

#### **Consider the table Location**:

#### normalizationinsql_4

Fig 4.4.2.3 Table not in 2NF

The Location table possesses a composite primary key cust id, storied. The non-key attribute is store location. In this case, store location only depends on storied, which is a part of the primary key. Hence, this table does not fulfill the second normal form.

To bring the table to Second Normal Form, you need to split the table into two parts. This will give you the below tables:

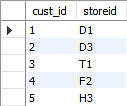
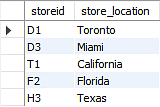
 

Fig 4.4.2.4 Table in 2NF

#### **Third Normal Form**

#### According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key.

Below is a student table that has student id, student name, subject id, subject name, and address of the student as its columns where stu\_id determines sub\_id, and sub\_id determines sub. Therefore, stu\_id determines sub via sub\_id. This implies that the table possesses a transitive functional dependency, and it does not fulfill the third normal form criteria.

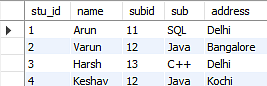


Fig 4.4.2.5 Table in 3NF

Now to change the table to the third normal form, you need to divide the table as shown below:

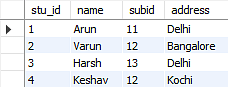
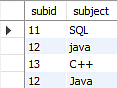
 

Fig 4.4.2.6 Table not in 3NF

As you can see in both the tables, all the non-key attributes are now fully functional, dependent only on the primary key. In the first table, columns name, sub\_id, and addresses only depend on stu\_id. In the second table, the sub only depends on sub\_id.

### 4.4.3 Sanitization

Sanitizing data means removing any illegal character from the data. Sanitizing user input is one of the most common tasks in a web application. To make this task easier PHP provides native filter extension that you can use to sanitize the data such as e-mail addresses, URLs, IP addresses, etc.

PHP filters are used to sanitize and validate external input. The PHP filter extension has many of the functions needed for checking user input, and is designed to do data sanitization easier and quicker. This function, when using the flag in the example, is making sure that the code remove all characters except letters, digits and the following characters !#$%&‟\*+-=?\_`{|}~@.[] . Many web applications receive external input. External input/data can be:

* User input from a form
* Cookies
* Web services data
* Server Variables
* Database query result

**4.4.4 Indexing**

The index stores the value of a specific field or set of fields, ordered by the value of the field. The ordering of the index entries supports efficient equality matches and range-based query operations. Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records. Indexes support the efficient execution of queries in PHP. An "index" can improve the speed of operation in a table. MySQL automatically creates an index for primary key, foreign key, and unique constraints. In addition, you may want to create "indexes" for other columns that are frequently used in joins or search conditions.The user cannot see indexes. The indexes are used to speed up database queries and also used by a database engine to locate records very quickly.

You must have used a "CREATE INDEX" statement to create an index for one or more columns of a table. To create an index, write the table name and column names after the "on" clause. You can also use "UNIQUE" keywords to specify that an "index" has only unique values. You can also specify "ASC" and "DESC" keywords with a column name to indicate whether you want the "index" stored in ascending or descending order. If you do not specify "asc" or "desc", then "asc" is the default same as the "order by" keyword (which is also able to sort columns in "asc" or "desc"order)

### 4.5 TABLE DESIGN

1. **Tbl\_login**

Primary key: **loginid**

Foreign key **register\_id** references table **Tbl\_users\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype | Key Constraints | Description |
| 1 | login\_id | varchar(10) | Primary Key | Unique id of the table |
| 2 | register\_id | varchar(10) | Foreign Key | Email for login |
| 3 | password | varchar(20) | Not Null | Password for login |
| 4 | username | varchar(10) | Not Null | Name of the user |
| 5 | type | varchar(10 | Not Null | Type of the user |

1. **Tbl\_Register**

Primary key: **register\_id**

Foreign key **register\_id** references table **tbl\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No  : | Fieldname | Datatype | Key Constraints | Description |
| 1 | register\_id | varchar(10) | Primary Key | Unique id of the table |
| 2 | u\_name | varchar(20) | Not Null | User name |
| 3 | u\_email | varchar(20) | Not Null | Email of user |
| 4 | u\_gender | varchar(15) | Not Null | Gender of User |
| 5 | u\_nationality | varchar(50) | Not Null | Nationality |
| 6 | u\_phone | varchar(20) | Not Null | Phone number |
| 7 | u\_password | varchar(10) | Not Null | password |
| 8 | u\_type | varchar(20) | Not Null | Type of the User |
| 12 | status | bool | Not Null | Status |

1. **Tbl\_bookings**

Primary key: **book\_id**

Foreign key **login\_id** references table **tbl\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype | Key Constraints | Description |
| 1 | book\_id | varchar(5) | Primary Key | Unique id of the table |
| 2 | login\_id | varchar(5) | Foreign Key | Foreign Key of tbl\_bookings |
| 3 | from\_place | varchar(10) | Not Null | Departure place |
| 4 | to\_place | varchar(15) | Not Null | Arrival place |
| 5 | preferred\_airline | varchar(10) | Not Null | Name of the Airline |
| 6 | preferred\_seating | varchar(7) | Not Null | Preffered seating  Arrangement |
| 7 | departure\_date | date | Not Null | Date of Departure |
| 8 | departure\_time | time | Not Null | Time of Departure |
| 9 | adult | varchar(7) | Not Null | Passenger Adult |
| 10 | children | varchar(15) | Not Null | Passenger Children |
| 11 | infant | varchar(5) | Not Null | Passenger infant |
| 12 | oneorround | varchar(10) | Not Null | Way of Travel |

1. **Tbl\_feedback**

Primary key: **feedback\_id**

Foreign key **book\_id** references table **Tbl\_booking**

Foreign key **register\_id** references table **Tbl\_register**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype | Key Constraints | Description |
| 1 | feedback\_id | varchar(5) | Primary Key | Unique id of the table |
| 2 | book\_id | varchar(5) | Foreign Key | Foreign Key of feedback  table |
| 3 | register\_id | varchar(20) | Foreign Key | Registration id of the user |
| 3 | comments | varchar(20) | Not Null | About the Bookings |

##### Tbl\_flights

Primary key: **flight\_id**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype | Key Constraints | Description |
| 1 | flight\_id | varchar(5) | Primary Key | Unique id of the table |
| 2 | f\_airline | varchar(5) | Not Null | Foreign Key of tbl\_bookings |
| 3 | flight\_no | varchar(10) | Not Null | Flight code |
| 4 | departlocation | varchar(15) | Not Null | Departure place |
| 5 | arrivallocation | Varchar(10) | Not Null | Arrival place |
| 6 | departdate\_time | Datetime(6) | Not Null | Preffered seating Arrangement |
| 7 | arrivaldate\_time | Datetime(6) | Not Null | Date and time of Departure |
| 8 | seats | Varchar(10) | Not Null | Number of seats |
| 9 | price | varchar(7) | Not Null | price |
| 10 | status | bool | Not Null | status |

**6.Tbl\_payment**

Primary key: **payment\_id**

Foreign key **book\_id** references table **Tbl\_booking**

Foreign key **login\_id** references table **Tbl\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No: | Fieldname | Datatype | Key Constraints | Description |
| 1 | payment\_id | varchar(5) | Primary Key | Unique id of the table |
| 2 | book\_id | varchar(5) | Foreign Key | Foreign Key of feedback  table |
| 3 | login\_id | varchar(5) | Foreign Key | Login id of the user |
| 4 | email | varchar(20) | Not Null | About the Bookings |
| 5 | amount | varchar(20) | Not Null | About the Bookings |

# CHAPTER 5

# SYSTEM TESTING

* 1. **INTRODUCTION**

Software Testing is the process of executing software in a controlled manner, in order to answer the question Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

* + - A good test case is one that has high possibility of finding an undiscovered error.
    - A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met. There are three ways to test program.

* + - For correctness
    - For implementation efficiency
    - For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable. terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan. The levels of testing include:

* Unit testing
* Integration Testing
* Data validation Testing
* Output Testing

### Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm‟s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to rerouteor cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

### Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continuesin a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

### Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly knownas Black Box testing or System tests.Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions thatwill fully exercise all functional requirements for a program.Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

### Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm‟s need. The software should keep in touch with perspective system; user at the time ofdeveloping and making changes whenever required. This done with respect to the following points:

* + - * Input Screen Designs,
      * Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system understudy is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

* + 1. **Automation Testing**

Automated testing is a process that validates if software is functioning appropriately and meeting requirements before it is released into production. This software testing method uses scripted sequences that are executed by testing tools. UI automation testing is a technique where these testing processes are performed using an automation tool. Instead of having testers click through the application to verify data and action flows visually, test scripts are written for each test case. A series of steps to follow when the verifying data is then added. Automatic testing is required when you want to run the same test cases across multiple machines at the same time.

* + 1. **Selenium Testing**

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

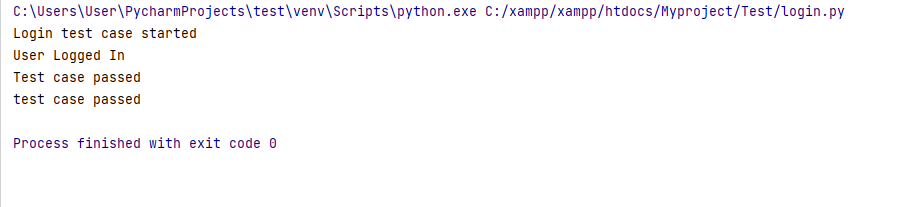
Selenium Integrated Development Environment (IDE) is the simplest framework in the Selenium suite and is the easiest one to learn. It is a Chrome and Firefox plugin that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a prototyping tool. If you want to create more advanced test cases, you will need to use either Selenium RC or WebDriver.

**Test Case 1**

**Code**

from selenium import webdriver  
import time  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.action\_chains import ActionChains  
from selenium.webdriver.common.by import By  
from selenium.webdriver.support.ui import Select  
print("Login test case started")  
options=webdriver.ChromeOptions()  
options.add\_experimental\_option('excludeSwitches',['enable-logging'])  
driver = webdriver.Chrome(options=options)  
driver.maximize\_window()  
driver.get("http://localhost/Myproject/index1.php")  
driver.find\_element(By.ID, "sub1").click()  
  
driver.find\_element("id", "phone").send\_keys("salinikb2000@gmail.com")  
time.sleep(3)  
driver.find\_element("id", "pass").send\_keys("Salini@123")  
time.sleep(3)  
submit=driver.find\_element(By.ID,"sub")  
submit.click()  
  
print("User Logged In")  
print("Test case passed")

**Output**

****

**Test Report**

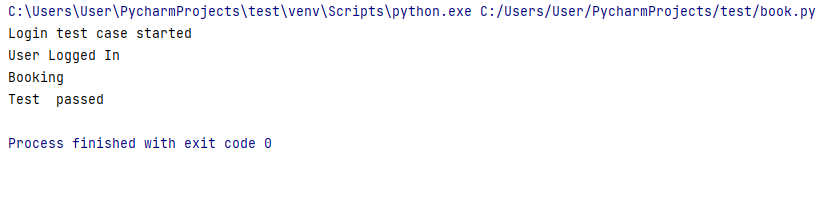
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case 1** | | | | | |
| **Project Name: Willfly- Online Flight Booking** | | | | | |
| **Login Test Case** | | | | | |
| **Test Case ID: Test\_1** | | | **Test Designed By:** Salini K B | | |
| **Test Priority(Low/Medium/High):**High | | | **Test Designed Date:** 11/05/2023 | | |
| **Module Name**: Login Page | | | **Test Executed By :** MS. Ankitha Philip | | |
| **Test Title :** Verify login with email and password | | | **Test Execution Date:** | | |
| **Description:** Testing the login page | | |  | | |
| **Pre-Condition :**User has valid username and password | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigation to Login page |  | Login Page should be displayed | Login Page displayed | Pass |
| 2 | Provide valid email | Username:salinikb2000@gmail.com | User should be able to login | User logged in and navigated to dashboard | Pass |
| 3 | Provide valid password | Salini@123 |
| 4 | Click on Login Button |  |
| 5 | Provide Invalid Email id or password | Email:gg123@gmail.com  Password: ggg123 |  |  |  |
|  |  |  | User should not be able in Login | Message for enter valid email id or password displayed | Pass |
| 6 | Provide null email id or password | Email:null  Password: null |
| 7 | Click on Login In button |  |  |  |  |
|  |  |  |  |  |
| **Post-Condition: User is validated with database and login to the website. The account session details are logged in database** | | | | | |

**Test Case 2**

**Code**

from selenium import webdriver  
import time  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.action\_chains import ActionChains  
from selenium.webdriver.common.by import By  
from selenium.webdriver.support.ui import Select  
print("Login test case started")  
options=webdriver.ChromeOptions()  
options.add\_experimental\_option('excludeSwitches',['enable-logging'])  
driver = webdriver.Chrome(options=options)  
driver.maximize\_window()  
driver.get("http://localhost/Myproject/index1.php")  
driver.find\_element(By.ID, "sub1").click()  
  
driver.find\_element("id", "phone").send\_keys("riyajaceykurian99@gmail.com")  
time.sleep(3)  
driver.find\_element("id", "pass").send\_keys("Abc@1234")  
time.sleep(3)  
submit=driver.find\_element(By.ID,"sub")  
submit.click()  
  
print("User Logged In")  
driver.get("http://localhost/Myproject/user12/asyourneed.php")  
time.sleep(3)  
driver.find\_element("xpath","/html/body/div[1]/div[2]/div[2]/ul/li[5]/a").click()  
  
time.sleep(3)  
driver.find\_element("xpath", '/html/body/div[7]/section/div/div[2]/div[1]/div/button/a/h5').click()  
time.sleep(3)  
print("Test passed")

**Output**



**Test Report**

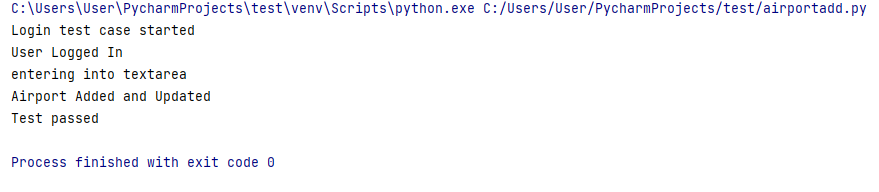
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case 2** | | | | | | |
| **Project Name: Willfly- Online Flight Booking** | | | | | | |
| **Book Flight Test Case** | | | | | | |
| **Test Case ID: Test\_2** | | | | **Test Designed By:** Salini K B | | |
| **Test Priority(Low/Medium/High):**High | | | | **Test Designed Date:** 11/05/2023 | | |
| **Module Name**: Booking Page | | | | **Test Executed By :** Ms Ankitha Philip | | |
| **Test Title :** Book Flight | | | | **Test Execution Date:** | | |
| **Description:** Testing the Book Flight | | | |  | | |
| **Pre-Condition :**Require added schedules | | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigation to Login page and login to user page |  | User dashboard should be displayed | | User Dashboard displayed | Pass |
| 2 | Navigate to booking page |  | User should be able to book flight | | User navigated to booking page | Pass |
| 3 | Click on corresponding Book Button | Flight ID  User ID |
| 4 | Select details and click book button |  | User should be able to book flight | | User book Flight | Pass |
| **Post-Condition: User is logged in to the website. User select booking details and click on book your seats button.** | | | | | | |

**Test Case 3**

**Code**

from selenium import webdriver  
import time  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.action\_chains import ActionChains  
from selenium.webdriver.common.by import By  
from selenium.webdriver.support.ui import Select  
print("Login test case started")  
options=webdriver.ChromeOptions()  
options.add\_experimental\_option('excludeSwitches',['enable-logging'])  
driver = webdriver.Chrome(options=options)  
driver.maximize\_window()  
driver.get("http://localhost/Myproject/index1.php")  
driver.find\_element(By.ID, "sub1").click()  
  
driver.find\_element("id", "phone").send\_keys("admin123@gmail.com")  
time.sleep(3)  
driver.find\_element("id", "pass").send\_keys("Admin@12")  
time.sleep(3)  
submit=driver.find\_element(By.ID,"sub")  
submit.click()  
  
print("User Logged In")  
driver.get("http://localhost/Myproject/admin12/index2.php")  
time.sleep(3)  
driver.find\_element("xpath","/html/body/div/nav/div[2]/div/ul/li[3]/a").click()  
time.sleep(3)  
*# Enter text in textarea*print("entering into textarea")  
driver.find\_element("id", "myairport").send\_keys("cochin")  
time.sleep(3)  
driver.find\_element("id", "location").send\_keys("Kochi")  
time.sleep(3)  
driver.find\_element("xpath", '/html/body/div/div/div[2]/div/div/div[1]/form/div/div[3]/div/div/input').click()  
time.sleep(3)  
  
print("Airport Added and Updated")  
  
print("Test passed")

**Output**

****

**Test report**

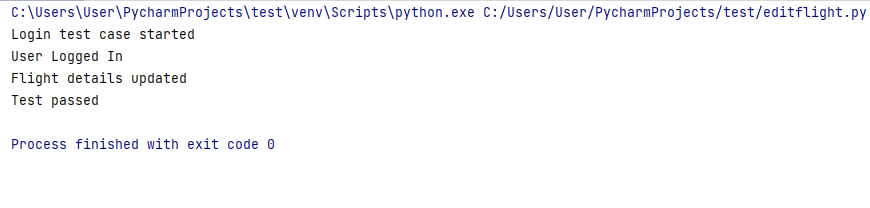
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case 3** | | | | | | |
| **Project Name:Willfly- Online Flight Booking** | | | | | | |
| **Add Schedule Test Case** | | | | | | |
| **Test Case ID: Test\_3** | | | | **Test Designed By: Salini KB** | | |
| **Test Priority(Low/Medium/High):High** | | | | **Test Designed Date: 11/05/2022** | | |
| **Module Name**: Add Airport | | | | **Test Executed By : Ms Ankitha Philip** | | |
| **Test Title : Add Airport Details** | | | | **Test Execution Date:** | | |
| **Description: Testing the Airport page** | | | |  | | |
| **Pre-Condition :**Require added Airports | | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigation to Login page and login to Admin page |  | Admin dashboard should be displayed | | Admin Dashboard displayed | Pass |
| 2 | Navigate to Airport page |  | Admin should be able to edit page | | Admin navigated to edit page | Pass |
| 3 | Give details of the Airport | Airport ID |
| 4 | Update details and click update button |  | Admin should be able to update data | | Admin updated data | Pass |
| **Post-Condition: Admin is validated with database and login to the website. Then add airport details and edit.** | | | | | | |

**Test Case 4**

**Code**

from selenium import webdriver  
import time  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.action\_chains import ActionChains  
from selenium.webdriver.common.by import By  
from selenium.webdriver.support.ui import Select  
print("Login test case started")  
options=webdriver.ChromeOptions()  
options.add\_experimental\_option('excludeSwitches',['enable-logging'])  
driver = webdriver.Chrome(options=options)  
driver.maximize\_window()  
driver.get("http://localhost/Myproject/index1.php")  
driver.find\_element(By.ID, "sub1").click()  
  
driver.find\_element("id", "phone").send\_keys("admin123@gmail.com")  
time.sleep(3)  
driver.find\_element("id", "pass").send\_keys("Admin@12")  
time.sleep(3)  
submit=driver.find\_element(By.ID,"sub")  
submit.click()  
  
print("User Logged In")  
driver.get("http://localhost/Myproject/admin12/index2.php")  
time.sleep(3)  
driver.find\_element("xpath","/html/body/div/nav/div[2]/div/ul/li[4]/a").click()  
time.sleep(3)  
*# Enter text in textarea*driver.find\_element("xpath", '/html/body/div/div/div[2]/div/div[2]/div/table/tbody/tr[3]/td[4]/b/font/a').click()  
time.sleep(3)  
variable=driver.find\_element("id", "myairline")  
variable.clear  
variable.send\_keys("jetairways")  
time.sleep(3)  
driver.find\_element("xpath", '/html/body/div/div/div[2]/div[2]/form/div[9]/input').click()  
time.sleep(3)  
  
  
print("Flight details updated")  
print("Test passed")

**OUTPUT**

****

**Test Report**

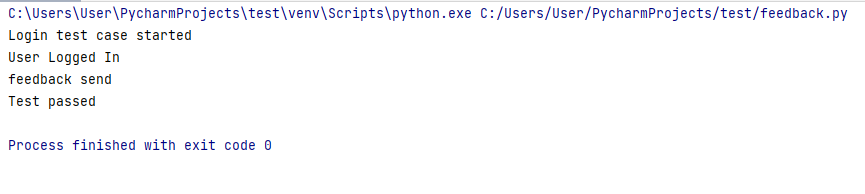
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case 4** | | | | | | |
| **Project Name: Willfly- Online Flight Booking** | | | | | | |
| **Flight Details Updated** | | | | | | |
| **Test Case ID: Test\_4** | | | | **Test Designed By: Salini K B** | | |
| **Test Priority(Low/Medium/High):High** | | | | **Test Designed Date: 11/05/2023** | | |
| **Module Name**: Flight Upadate Page | | | | **Test Executed By : Ms. Ankitha Philip** | | |
| **Test Title : Update the details** | | | | **Test Execution Date:** | | |
| **Description: Testing the Report Page** | | | |  | | |
| **Pre-Condition :**Require added schedules | | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigation to Login page and login to Admin page |  | Admin dashboard should be displayed | | Admin  Dashboard displayed | Pass |
| 2 | Navigate to Flight data page |  | Admin should able to add values | | Admin entered the value and added it to the table | Pass |
| 3 | Enter the values for each items | Data values |
| 4 | Updated new details in the Flight page |  | Admin should be able to view the new data | | Admin updated data | Pass |
| **Post-Condition: Admin is validated with database and login to the website. The value are inserted into the database and admin can view the flight details.** | | | | | | |

**Test Case 5**

**Code**

from selenium import webdriver  
import time  
from selenium.webdriver.common.keys import Keys  
from selenium.webdriver.common.action\_chains import ActionChains  
from selenium.webdriver.common.by import By  
from selenium.webdriver.support.ui import Select  
print("Login test case started")  
options=webdriver.ChromeOptions()  
options.add\_experimental\_option('excludeSwitches',['enable-logging'])  
driver = webdriver.Chrome(options=options)  
driver.maximize\_window()  
driver.get("http://localhost/Myproject/index1.php")  
driver.find\_element(By.ID, "sub1").click()  
  
driver.find\_element("id", "phone").send\_keys("riyajaceykurian99@gmail.com")  
time.sleep(3)  
driver.find\_element("id", "pass").send\_keys("Abc@1234")  
time.sleep(3)  
submit=driver.find\_element(By.ID,"sub")  
submit.click()  
  
print("User Logged In")  
driver.get("http://localhost/Myproject/user12/asyourneed.php")  
time.sleep(3)  
driver.find\_element("xpath","/html/body/div[1]/div[2]/div[2]/ul/li[10]/a/span").click()  
time.sleep(3)  
driver.find\_element("xpath","/html/body/div[6]/form/div/ul/li[1]").click()  
*# Enter text in textarea*driver.find\_element("id", "comment").send\_keys("good")  
time.sleep(3)  
driver.find\_element("xpath", '/html/body/div[6]/form/div/input').click()  
time.sleep(3)  
print("feedback send")  
print("Test passed ")

**OUTPUT**



**Test Report**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case 5** | | | | | | |
| **Project Name: Willfly- Online Flight Booking** | | | | | | |
| **Adding feedback Test Case** | | | | | | |
| **Test Case ID: Test\_5** | | | | **Test Designed By: Salini K B** | | |
| **Test Priority(Low/Medium/High):High** | | | | **Test Designed Date: 11/5/2023** | | |
| **Module Name**: Adding Feedback page | | | | **Test Executed By : Ms Ankitha Philip** | | |
| **Test Title : Login Test case Started** | | | | **Test Execution Date:** | | |
| **Description: Testing the feedback page** | | | |  | | |
| **Pre-Condition : User should have access to the portal** | | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | | **Actual Result** | **Status(Pass/**  **Fai l)** |
| 1 | Navigation to Login page and login to user page |  | User dashboard should be displayed | | User  Dashboard displayed | Pass |
| 2 | Navigate to add feedback page |  | User should able to insert new feedback details like good,bad,excellent | | New feedback details are entered | Pass |
| 3 | Enter the feedback | Data values |
| 4 | Click send feedback button |  | New feedback details are added. | | New feedback details are added | Pass |
| **Post-Condition: User is validated with database and login to the website. New feedback is added.** | | | | | | |

# CHAPTER 6

# IMPLEMENTATION

## INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned intoa working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is beingtrained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual useis called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented onlyafter through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the systemanalysis and design effort required to implement the three main aspects: education and training, system testing and changeover. The implementation state involves the following tasks:

* Careful planning.
* Investigation of system and constraints.
* Design of methods to achieve the changeover.

## IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

* The active user must be aware of the benefits of using the new system. Their confidence in the software is built up.
* Proper guidance is imparted to the user so that he is comfortable in using theapplication. Before going ahead and viewing the system, the user must know that for viewing the result, the

server program should be running in the server. If the server object is not up running on the server, the actual process won‟t take place.

### User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

### System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "FindingMistakes".

### Hosting

Hosting a website refers to the process of making a website accessible and available on the World Wide Web. The hosting service provider is responsible for keeping the server up and running, ensuring that the website or application is available to users at all times, and providing technical support. The steps involved in hosting a website typically include choosing a hosting provider, selecting a hosting plan, registering a domain name, configuring DNS settings, uploading website files, setting up databases and email accounts, and configuring security settings. Overall, hosting is an important aspect of website development and maintenance, as it allows users to access the website or application from anywhere in the world.

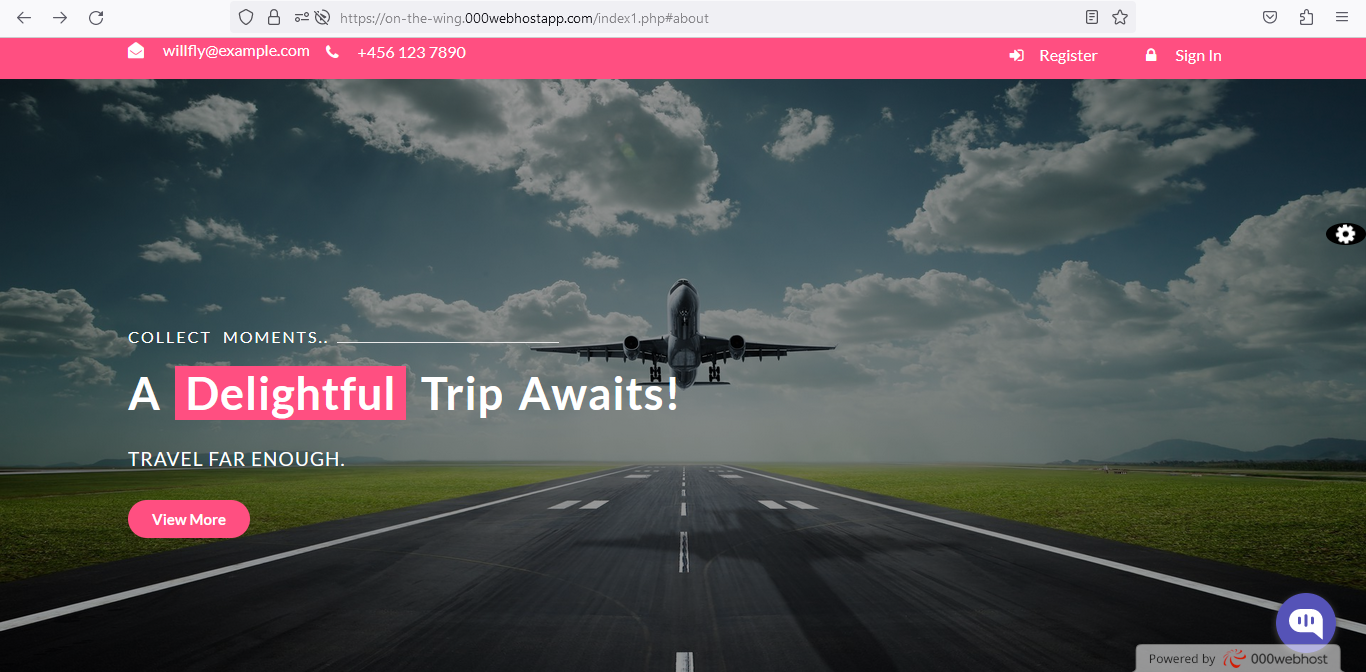
The steps to host the website named Willfly- Online flight Booking on 000webhost:

1. **Create an account on 000webhost:** The first step is to create an account on the 000webhost website. Provide your basic information such as name, email address, and password.
2. **Verify your email address:** After creating the account, verify your email address by clicking on the verification link sent to your registered email.
3. **Login to the 000webhost cPanel:** Once you have verified your email address, login to the 000webhost cPanel using your registered email and password.
4. **Add your domain name:** After logging in to the cPanel, add your domain name by clicking on the 'Add Website' button. Enter the domain name you want to use for your website.
5. **Upload the website files:** After adding the domain name, upload the website files to the 000webhost server using FTP or File Manager.
6. **Configure the website settings:** After uploading the website files, configure the website settings such as the database, email, and SSL.
7. **Publish the website:** After configuring the website settings, publish the website by clicking on the 'Publish Website' button.
8. **Test the website:** Finally, test the website to ensure it is working properly. Check all the links, forms, and functions of the website.

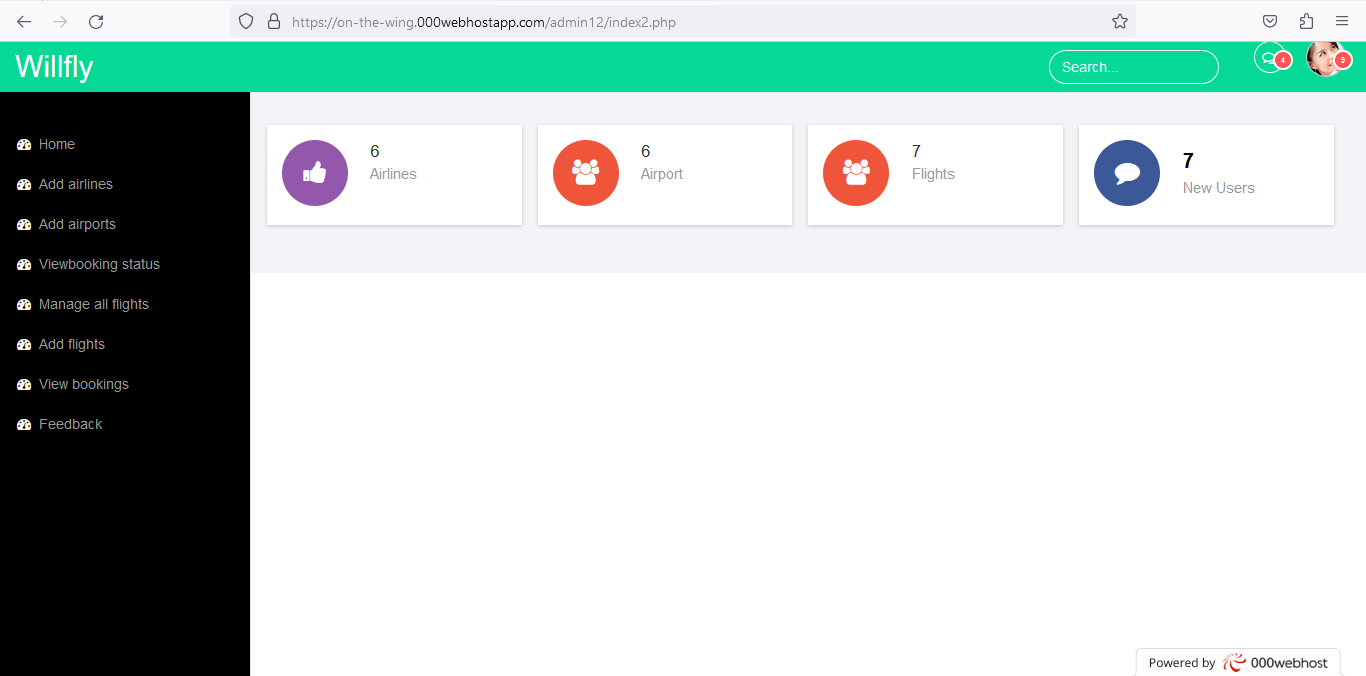
**Hosted Link:** [**https://on-the-wing.000webhostapp.com**](https://on-the-wing.000webhostapp.com)

**Screenshot**

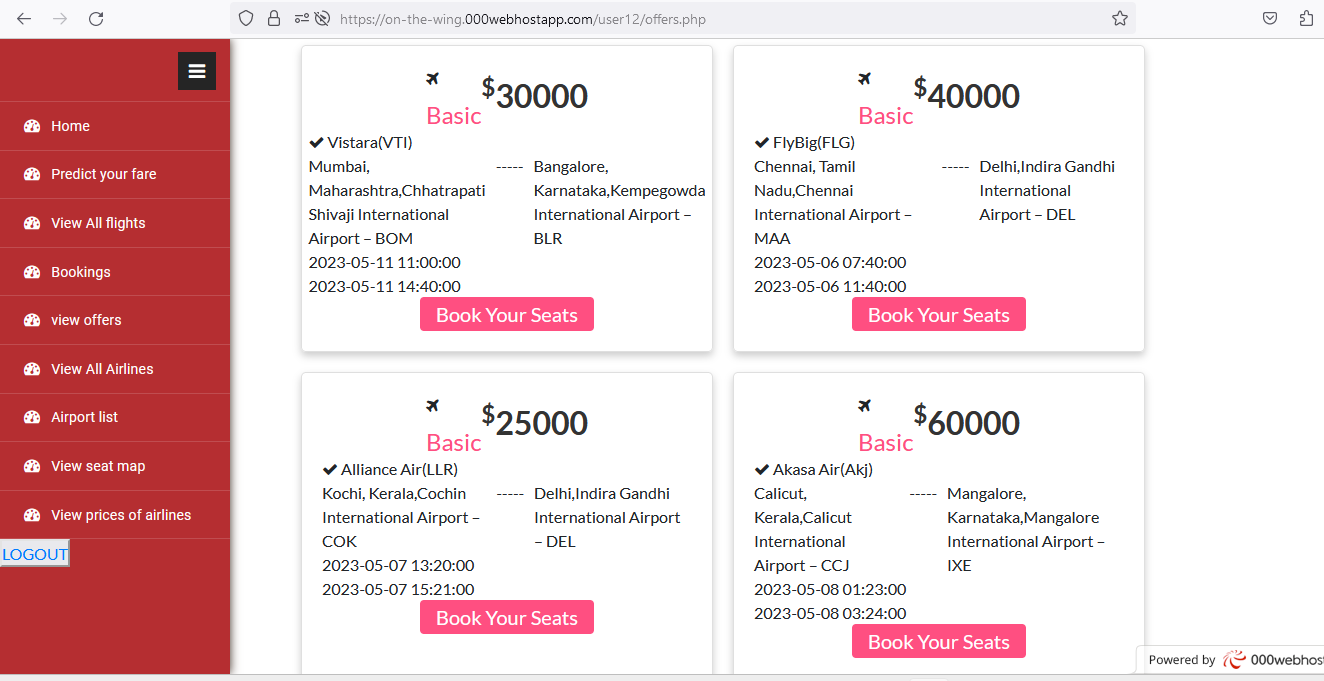
**Fig 6.2.4 : Login Page of Hosted Website**

****

**Fig 6.2.4 : Dashboard of Admin of Hosted Website**

****

**Fig 6.2.4 : Dashboard of user of Hosted Website**



# CHAPTER 7

# CONCLUSION AND FUTURE SCOPE

## CONCLUSION

## 

Flight booking system is the website the system allows the airline passengers To search for flights that are available between the two travel cities namely Departure city and available city.The objective of the airline system to manage the details of airline,ticket,flights,customers details etc. The main objective of this application is to decrease the manual mistakes engaged in the commercial airline booking process and create it practical for the clients to book the routes as when they require such that they can implement this application to create bookings, change bookings or terminate a particular booking.

The name of the application is “Willfly”. This application provides alternatives for watching different routes available with different timings for a particular time frame and provides clients with the service to publication a admission, change or terminate a particular reservation but it does not offer the clients with information of price of the admission and it does not allow the client to alter a particular aspect of his reservation and he/she can change all his information. . This system will give an idea about the price trends and also provide a predicted price value which travelers can refer to before booking their flight tickets to save money. The software keep generating timely message regarding schedule related updates to users.

* 1. **FUTURE SCOPE**

The airline ticket booking system that we are proposing ensures the gives total freedom of passengers ,where passenger can use his own personal computer to log on the website and can book his ticket.The airline ticket booking system allows only registered users or passengers to reserve the flights tickets,view flight.Timing and cancel there tickets. Here passengers can send their queries and suggestions through a feedback form and user friendliness provided in the application with various controls.The system always provides offer for regular customer in attractive manner.Flight ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, duration of flights, various occasions such as vacations or festive season. Therefore, having some basic idea of the flight fares before planning the trip will surely help people save money and time. Proposed system in future can be implemented using ML based on the collected historical data of flights. This system will give an idea about the price trends and also provide a predicted price value which travelers can refer to before booking their flight tickets to save money.

# CHAPTER 8

# BIBLIOGRAPHY

#### REFERENCES

* + - Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
    - Roger S Pressman, “Software Engineering”, 1994.
    - PankajJalote, “Software engineering: a precise approach”, 2006.
    - James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
    - The Complete reference PHP by Steven Holzner
    - The Complete reference MySQL by Vikram Vaswani
    - CSS Cookbook by Christopher Schmitt

#### WEBSITES

* + - [www.jquery.com](http://www.jquery.com/)
    - <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
    - [www.agilemodeling.com/artifacts/useCaseDiagram.html](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)
    - [www.geeksforgeeks.com](http://www.geeksforgeeks.com/)
    - [www.tutorialspoint.org](http://www.tutorialspoint.org/)

https://[www.airindia.in/](http://www.airindia.in/)

# CHAPTER 9

# APPENDIX

## Sample Code

##### Login

<?php require\_once("db\_connect.php"); include("functions.php"); session\_start();

?>

<?php

if (isset($\_POST['submit'])) {

if (!empty($\_POST['fullname']) && !empty($\_POST['password'])) {

$fullname = mysqli\_real\_escape\_string($conn,$\_POST['fullname']);

$password = mysqli\_real\_escape\_string($conn,$\_POST['password']);

//$type = mysqli\_real\_escape\_string($conn,$\_POST['type']);

$login = indexx($fullname,$password,$conn);

}

else{

echo "Required All fields!";

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="preconnect" href="https://fonts.gstatic.com">

<link href="[https://fonts.googleapis.com/css2?family=Source+Sans+Pro:wght@300&display=swap"](https://fonts.googleapis.com/css2?family=Source%2BSans%2BPro%3Awght%40300&display=swap) rel="stylesheet">

<link href="[https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"](https://cdn.jsdelivr.net/npm/bootstrap%405.0.0-beta2/dist/css/bootstrap.min.css) rel="stylesheet"

integrity="sha384- BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" crossorigin="anonymous">

<link rel="stylesheet" href="css/style.css">

<title>Online flight booking Application</title>

<style> #invalidMsg{ display:none;

}

body{

background: url(../pictures/t2.jpg) no-repeat center;

}

</style>

</head>

<body>

<!-- header -->

<nav class="navbar header-nav navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

<a class="navbar-brand" href="#">Online flight booking</a>

<a id="register" href="signup.php">Sign Up</a>

<a id="register" href="index1.php">Home</a>

</div>

</nav>

<!-- header ends -->

<!-- body -->

<div class="container-fluid">

<div class="row">

<!-- container and row divs for responsive -->

<!-- leftComponent -->

<div class="leftComponent col-md-5">

<img src="">

</div>

<!-- leftComponent ends -->

<!-- rightComponent -->

<div class="rightComponent col-md-5">

<h3>Please login to continue. . .</h3>

<hr>

<form method="POST" class="loginForm" autocomplete="off">

<div class="alert alert-danger" id="invalidMsg">

<?php if(isset($\_POST['login'])){ if($login == false)

echo "<script type='text/javascript'>document.getElementById('invalidMsg').style.display = 'block';</script>";

echo "Invalid Username or Password";

}

else echo "";

?>

</div>

<div class="mb-3">

<input class="form-control" type="text" id="fullname" name="fullname" placeholder="Enter Username" required>

</div>

<div class="mb-3">

<input class="form-control" type="password" id="password" name="password" placeholder="Enter Password"

required>

</div>

<div>

<a href="changepassword.php" class="forgot">Forgot password?</a>

</div>

<input type="submit" class="btn btn-success" name="submit" value="Log In">

</form>

</div>

<!-- rightComponent ends -->

</div>

</div>

<!-- body ends -->

<div class="content2">

<p class="heading lead text-start">

</p>

<p class="text-start">

<pre>

</pre>

</p>

</div>

<footer class="footer navbar navbar-expand-lg navbar-light bg-light" style="color:white;">

</footer>

</body>

</html>

<?php ini\_set('display\_errors', true); error\_reporting(E\_ALL);

?>

##### **Booking**

<?php

include "db\_connect.php"; include 'nav6.php';

?>

<?php if(isset($\_POST["submit"])){

$from=$\_POST["from"];

$to=$\_POST["to"];

$preferedairline=$\_POST["preferedairline"];

$preferedseating=$\_POST["preferedseating"];

$Departuredate=$\_POST["Departuredate"];

$Departuretime=$\_POST["Departuretime"];

$adult=$\_POST["adult"];

$children=$\_POST["children"];

$infant=$\_POST["infant"];

$oneorround=$\_POST["oneorround"];

$class=$\_POST["class"];

//$image=$\_FILES["image"]["name"];

//move\_uploaded\_file($\_FILES["image"]["tmp\_name"],"pictures/".$image); mysqli\_query($conn,"INSERT INTO `bookasyour\_tb`(`from`, `to`, `preferedairline`,

`preferedseating`, `Departuredate`, `Departuretime`, `adult`, `children`,`infant`,`oneorround`,`class`) VALUES

('$from','$to','$preferedairline','$preferedseating','$Departuredate','$Departuretime','$adult','$childre n','$infant','$oneorround','$class')");

echo "<script language='javascript'>";

echo 'window.location.href = "paymentmake.php";'; echo "alert('flight Booked')";

echo "</script>";

}

?>

<!DOCTYPE HTML>

<html>

<head>

<title>Book As Your Need</title>

<!-- Meta tags -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<meta name="keywords" content="Airline Booking Form Responsive Widget,Login form widgets, Sign up Web forms , Login signup Responsive web form,Flat Pricing table,Flat Drop downs,Registration Forms,News letter Forms,Elements"

/>

<script type="application/x-javascript"> addEventListener("load", function () { setTimeout(hideURLbar, 0);

}, false);

function hideURLbar() { window.scrollTo(0, 1);

}

</script>

<!-- //Meta tags -->

<!-- Stylesheet -->

<link href="asyourcss/wickedpicker.css" rel="stylesheet" type='text/css' media="all" />

<link rel="stylesheet" href="asyourcss/jquery-ui.css" />

<link href="asyourcss/style.css" rel='stylesheet' type='text/css' />

<!-- //Stylesheet -->

<!--fonts-->

<link href="//fonts.googleapis.com/css?family=Roboto:300,400,500,700" rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Raleway:300,400,500,600,700" rel="stylesheet">

<link href="//fonts.googleapis.com/css?family=Megrim" rel="stylesheet">

<!--//fonts-->

</head>

<body>

<!--background-->

<h1>Book As Your Need</h1>

<div class="booking-form-w3layouts">

<!-- Form starts here -->

<form action="#" method="post">

<h2 class="sub-heading-agileits">Booking Details</h2>

<div class="main-flex-w3ls-sectns">

<div class="field-agileinfo-spc form-w3-agile-text1">

<select class="form-control" name="from" id="from" >

<option>From</option required>

<option value="DELHI-INDIRA GANDHI INTERNATIONAL AIRPORT(DEL)">DELHI- INDIRA GANDHI INTERNATIONAL AIRPORT(DEL)</option>

<option value="CHENNAI-CHENNAI INTERNATIONAL AIRPORT(MAA)">CHENNAI- CHENNAI INTERNATIONAL AIRPORT(MAA)</option>

<option value="GOA-INTERNATIONAL AIRPORT(GOI)">GOA-INTERNATIONAL

AIRPORT(GOI)</option>

<option value="PUNE-INTERNATIONAL AIRPORT(PNQ)">PUNE-INTERNATIONAL

AIRPORT(PNQ)</option>

<option value="JAIPUR-INTERNATIONAL AIRPORT(JAI)">JAIPUR-INTERNATIONAL

AIRPORT(JAI)</option>

<option value="AGRA-AIRPORT(AGR)">AGRA-AIRPORT(AGR)</option>

</select>

</div>

<div class="field-agileinfo-spc form-w3-agile-text2">

<select class="form-control" name="to" id="to">

<option>To</option required>

<option value="DELHI-INDIRA GANDHI INTERNATIONAL AIRPORT(DEL)">DELHI- INDIRA GANDHI INTERNATIONAL AIRPORT(DEL)</option>

<option value="CHENNAI-CHENNAI INTERNATIONAL AIRPORT(MAA)">CHENNAI- CHENNAI INTERNATIONAL AIRPORT(MAA)</option>

<option value="GOA-INTERNATIONAL AIRPORT(GOI)">GOA-INTERNATIONAL

AIRPORT(GOI)</option>

<option value="PUNE-INTERNATIONAL AIRPORT(PNQ)">PUNE-INTERNATIONAL

AIRPORT(PNQ)</option>

<option value="JAIPUR-INTERNATIONAL AIRPORT(JAI)">JAIPUR-INTERNATIONAL

AIRPORT(JAI)</option>

<option value="AGRA-AIRPORT(AGR)">AGRA-AIRPORT(AGR)</option>

</select>

</div>

</div>

<div class="main-flex-w3ls-sectns">

<div class="field-agileinfo-spc form-w3-agile-text1">

<select class="form-control" name="preferedairline" id="preferedairline" required>

<option>Preferred Airline</option>

<option value="American Airline">American Airline</option>

<option value="Delta Airlines">Delta Airlines</option>

<option value="Frontier Airline">Frontier Airline</option>

<option value="Jet Blue">Jet Blue</option>

<option value="Southwest Airlines">Southwest Airlines</option>

</select>

</div>

<div class="field-agileinfo-spc form-w3-agile-text2">

<select class="form-control" name="preferedseating" id="preferedseating" required>

<option>Preferred Seating</option>

<option value="Window">Window</option>

<option value="Aisle">Aisle</option>

<!--<option value="Special">Special (Request note below)</option>-->

</select>

</div>

</div>

<div class="main-flex-w3ls-sectns">

<div class="field-agileinfo-spc form-w3-agile-text1">

<input id="datepicker" name="Departuredate" type="date" placeholder="Departure Date" value="Departuredate"

required="">

</div>

<div class="field-agileinfo-spc form-w3-agile-text2">

<input type="text" id="timepicker" name="Departuretime" class="timepicker form-control"

placeholder="Departure Time" value="Departuretime" required>

</div>

</div>

<div class="triple-wthree">

<div class="field-agileinfo-spc form-w3-agile-text11">

<select class="form-control" id="adult" name="adult">

<option value="">Adult(12+ Yrs)</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5+</option>

</select>

</div>

<div class="field-agileinfo-spc form-w3-agile-text22">

<select class="form-control" id="children" name="children">

<option value="">Children(2-11 Yrs)</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5+</option>

</select>

</div>

<div class="field-agileinfo-spc form-w3-agile-text33">

<select class="form-control" id="infant" name="infant">

<option value="">Infant(under 2Yrs)</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

<option value="4">4</option>

<option value="5">5+</option>

</select>

</div>

</div>

<div class="radio-section">

<h6>Select your Fare</h6>

<ul class="radio-buttons-w3-agileits">

<li>

<input type="radio" id="oneorround" name="oneorround" value="one way">

<label for="a-option">One Way</label>

<div class="check"></div>

</li>

<li>

<input type="radio" id="oneorround" name="oneorround" value="Round Trip">

<label for="b-option">Round-Trip</label>

<div class="check">

<div class="inside"></div>

</li>

</ul>

<div class="radio-section">

<h6>Select your Fare</h6>

<ul class="radio-buttons-w3-agileits">

<li>

<input type="radio" id="class" name="class" value="Bussiness">

<label for="a-option">Bussiness</label>

<div class="check"></div>

</li>

<li>

<input type="radio" id="class" name="class" value="Premium">

<label for="b-option">premium</label>

<div class="check">

<div class="inside"></div>

</div>

</li>

<li>

<input type="radio" id="class" name="class" value="Economy">

<label for="b-option">Economy</label>

<div class="check">

<div class="inside"></div>

</div>

</li>

</ul>

<div class="clear"></div>

</div>

<!--<div class="main-flex-w3ls-sectns">

<div class="field-agileinfo-spc form-w3-agile-text1">

<input id="datepicker1" name="Text" type="text" placeholder="Return Date" value="" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'mm/dd/yyyy';}"

required="">

</div>

<div class="field-agileinfo-spc form-w3-agile-text2">

<input type="text" id="timepicker1" name="Time" class="timepicker form-control" placeholder="Return Time" value="">

</div>

</div>

<div class="field-agileinfo-spc form-w3-agile-text">

<textarea name="Message" placeholder="Any Message..."></textarea>

</div>

<h3 class="sub-heading-agileits">Personal Details</h3>

<div class="main-flex-w3ls-sectns">

<div class="field-agileinfo-spc form-w3-agile-text1">

<input type="text" name="Name" placeholder="Full Name" required="">

</div>

<div class="field-agileinfo-spc form-w3-agile-text2">

<input type="text" name="Phone no" placeholder="Phone Number" required="">

</div>

<div class="field-agileinfo-spc form-w3-agile-text">

<input type="email" name="Email" placeholder="Email" required="">

</div>-->

<div class="clear"></div>

<input type="submit" value="Book Now" name="submit">

<!--<input type="reset" value="Clear Form">-->

<div class="clear"></div>

</form>

<!--// Form starts here -->

</div>

<!--copyright-->

<div class="copyright">

<!-- <p>&copy; 2018. Airline Booking Form . All Rights Reserved | Design by <a href="<http://w3layouts.com/>" target="\_blank">W3layouts</a> </p>-->

</div>

<!--//copyright-->

<script type="text/javascript" src="asyourjs/jquery-2.2.3.min.js"></script>

<!-- Time Js -->

<script type="text/javascript" src="asyourjs/wickedpicker.js"></script>

<script type="text/javascript">

$('.timepicker,.timepicker1').wickedpicker({ twentyFour: false

});

</script>

<!--// Time Js -->

<!-- Calendar Js -->

<script src="asyourjs/jquery-ui.js"></script>

<script>

$(function () {

$("#datepicker,#datepicker1,#datepicker2,#datepicker3").datepicker();

});

</script>

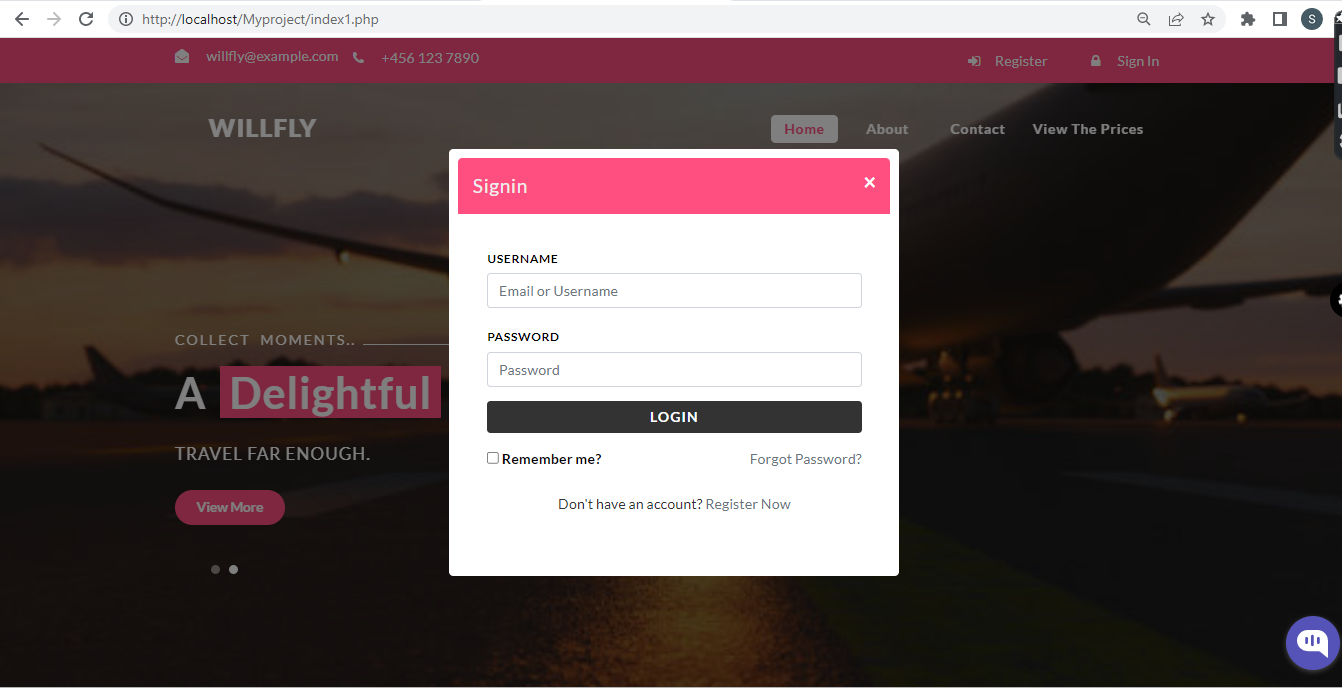
<!-- //Calendar Js -->

</body>

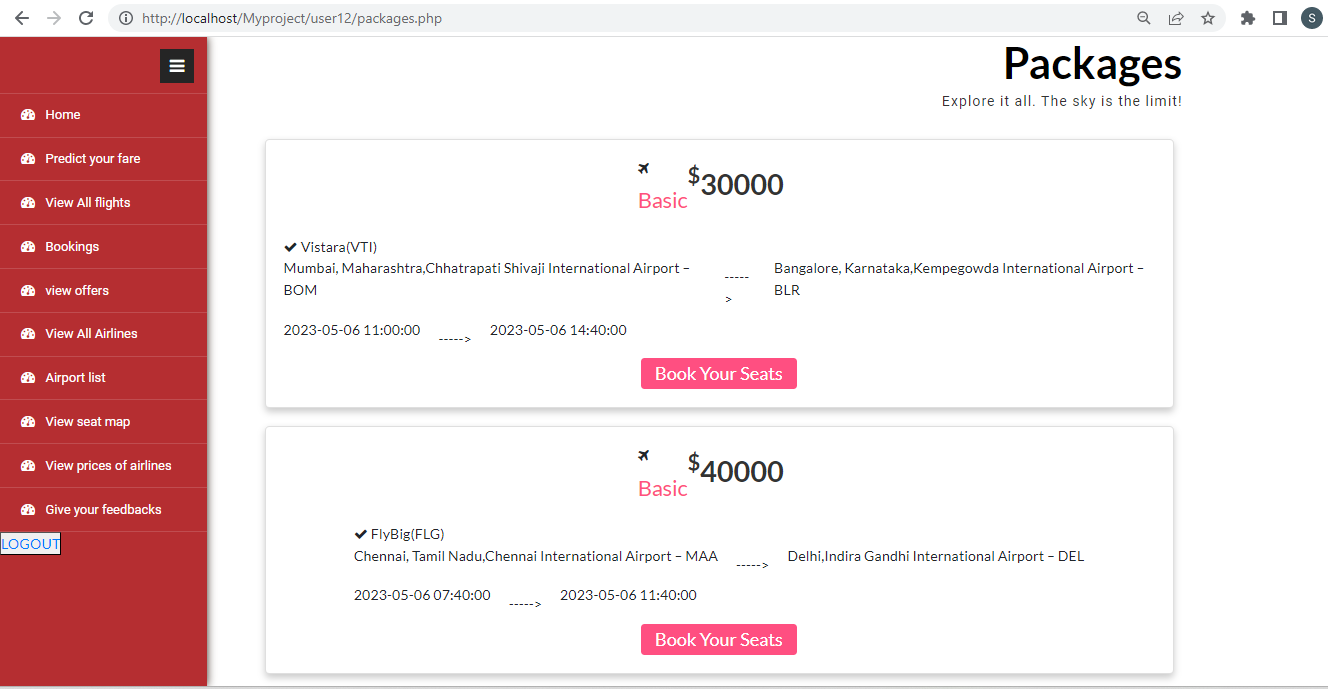
</html>

## Screen Shots

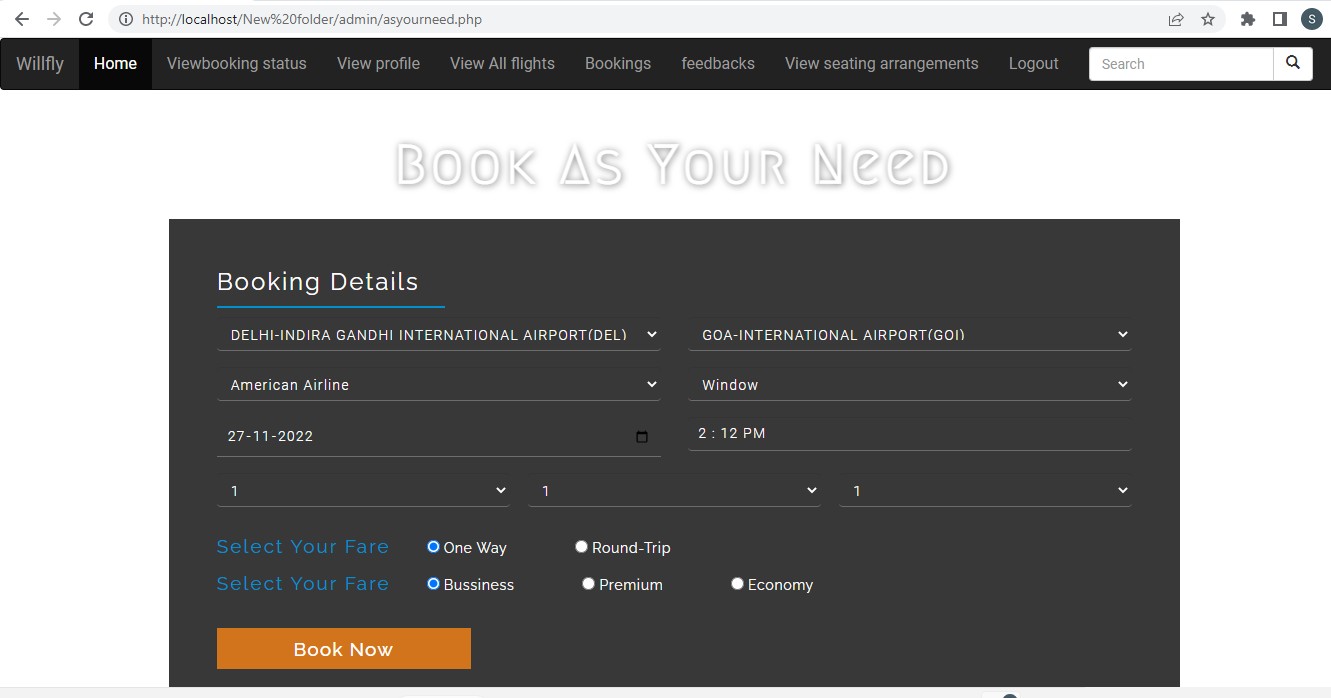
## login.php

****

**Home.php**

****

**Booking.php**



**flights.php**

