A

Project Report On

Airlines ticket Booking System

In Partial Fulfilment of

Bachelor of Computer Applications (B.C.A.)

Gujarat University (2025-26)

Developed By

Parmar Smita(5102)

Solanki Kinjal(5053)

Zanzarukiya Vidhi(5110)

(Sem-V Bachelor of Computer Applications)

Navgujarat College of Computer Applications

Ashram Road, Ahmedabad

Internal Guide

Pro.Ankita Patel

Acknowledgement

The acknowledgement sample for the university project is mostly attached after the dedication page thanking all the faculty members of the department, HOD, the dean and the project guide.

We are extended our sincere thanks to our collage – Navgujarat collage of computer applications , our respected principle Mr. Paresh Patel and our respected faculties for giving us a golden opportunities to work on the project of "**Airlines Ticket Booking System**" website and providing us their precious time , deep and valuable knowledge to make a project success.

We extend our warm thanks to our project guide prof. Ankita Patel for her guidance and constant encouragement throughout the semester for providing their valuable time , advise for various problems whenever we needed for deciding the objectives for our reports.

We are thankful to our parents, all staff members, our friends , seniors who helped directly or indirectly in our project..

<div align="center">

Parmar Smita

Solanki Kinjal

Zanzarukiya Vidhi

</div>

Introduction

The Airline Ticket Booking System is a software application designed to simplify and automate the process of booking flight tickets. In the traditional system, customers often have to visit travel agencies or airline offices to check flight availability, compare fares, or reserve seats. This method is time-consuming, prone to errors, and lacks real-time updates. To overcome these challenges, an efficient computerized system is required.

This project aims to develop a reliable, user-friendly, and secure platform that allows passengers to search for available flights, check schedules, compare prices, and book tickets online. The system also enables administrators to manage flights, update schedules, monitor bookings, and handle cancellations. By automating the entire workflow, the system reduces manual effort, increases accuracy, and provides a faster booking experience.

The Airline Ticket Booking System incorporates essential modules such as Passenger Management, Flight Management, Booking Management, Payment Processing, and Admin Controls. It improves communication between

airlines and customers by offering real-time information and instant confirmation.

SYSTEM FLOW DIAGRAM

System flow diagram is a basically a graphical and sequential representation of the major steps involved in a systematic process.

Symbols of system flow diagram

❖ **Start/End point**

An oval represents a start or end point. The terminator symbol marks the starting or ending point of the system. It usually contains the word "Start" or "End".

❖ **Arrows**
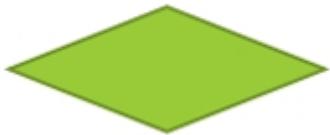
A line is a connector that shows relationships between the representative shapes.

❖ **Process**
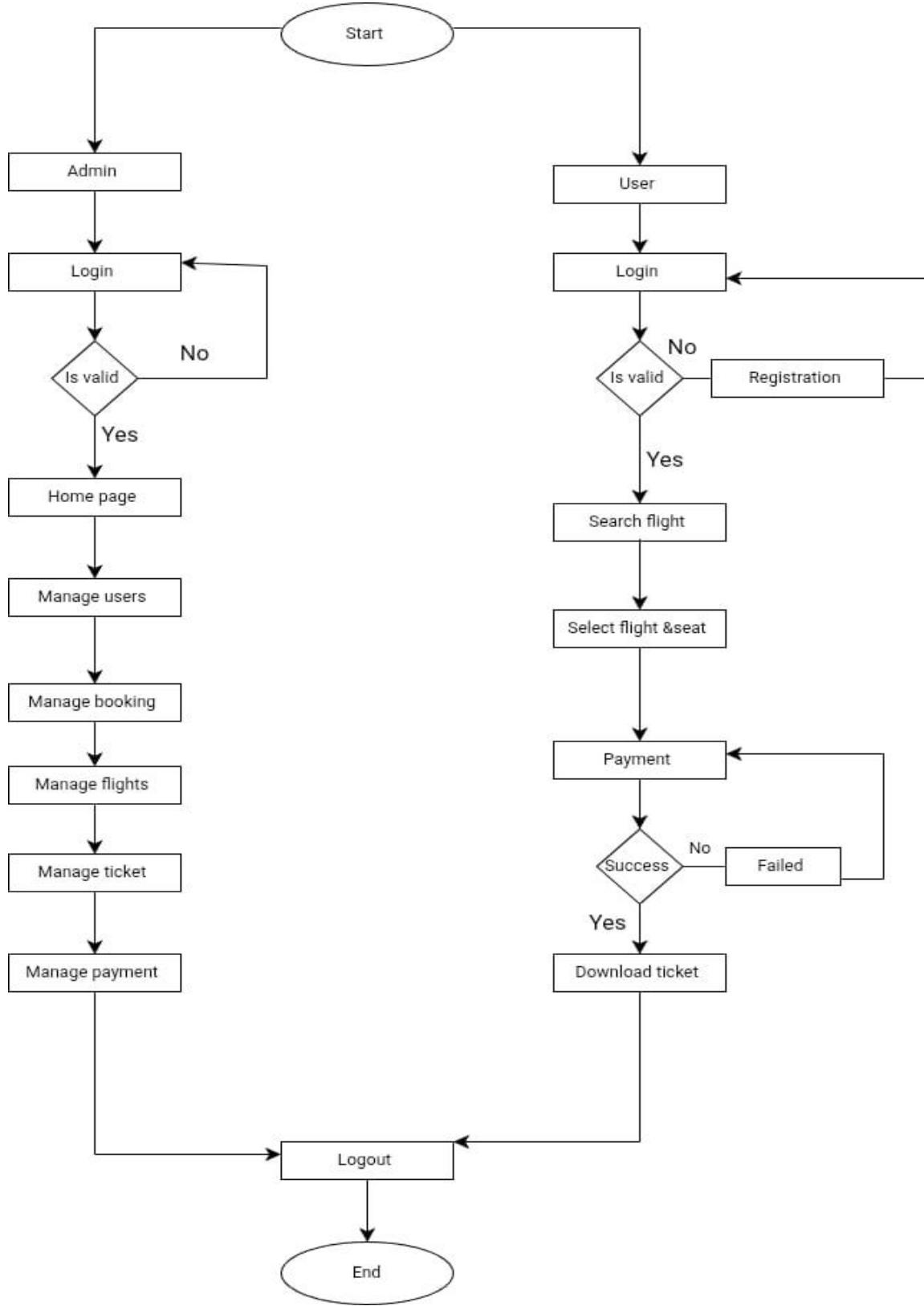
A rectangle represents a process.

❖ **Decision**

A diamond indicates a decision. It appoints where the outcome of a decision dictates the next step. There can be multiple outcomes, but often there are just two – yes and no or true – false, then branches to different parts of the system flow diagram.

SYSTEM FLOW DIAGRAM

```
                              Start

        Admin                              User

        Login          No                  Login         No
                                                                      Registration
        Is valid                           Is valid
            Yes                                Yes

        Home page                          Search flight

        Manage users                       Select flight &seat

        Manage booking                     Payment

        Manage flights                     Success    No    Failed

        Manage ticket                         Yes

        Manage payment                     Download ticket

                              Logout

                              End
```

UML(UNIFIED MODELLING LANGUAGE)

• What is UML and it's use?

Unified Modelling Language (UML) diagrams are graphical representation used for system design to illustrate various aspects of system. The overall goal of UML diagrams is to allow teams to visualize how a project is or will be working. UML can be used to develop diagrams and provide users with ready-to-use expressive modelling examples.

• What is the need of UML Diagrams?

An UML Diagrams are basically used in software engineering and other business processes where modelling is useful. UML becomes essential to communicate with non- programmers' essential requirements, functionalities, and processes of the system**.**

Here are some advantages of UML Diagrams:

UML provides a standard language that communicates design information among software developers which saves time and money. Using this language can help developers avoid potential errors in their programs, UML allows developers to reuse code. UML enhances collaboration and provides for a more efficient design process. It also helps identify potential problems early in the design process**.**

Types of UML Diagram:-

• Use Case Diagram:

An UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped.

• Activity Diagram:

An activity diagram is used by developers to understand the flow of programs on a high level.

• Sequence Diagram:

sequence diagrams is used to document and understand the requirements to establish new systems or learn about existing systems.

• Class Diagram:

Class Diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

USE-CASE DIAGRAM

Symbols of use case diagram

❖ Use-Cases

Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

❖ Actors
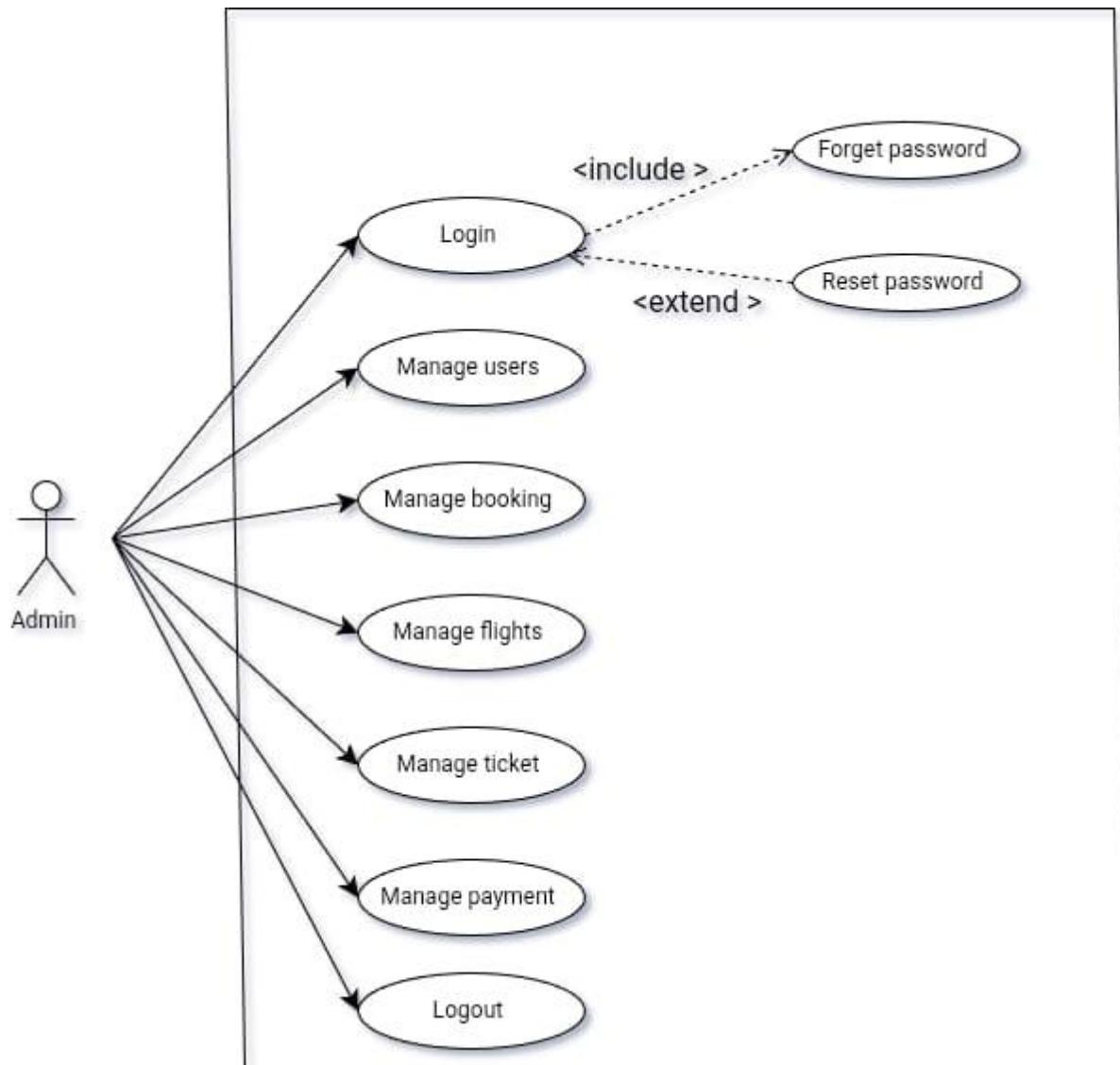
Actors are the users of a system. When one system is the actor of another system label the actor system with te actor stereotype.

❖ System Boundary
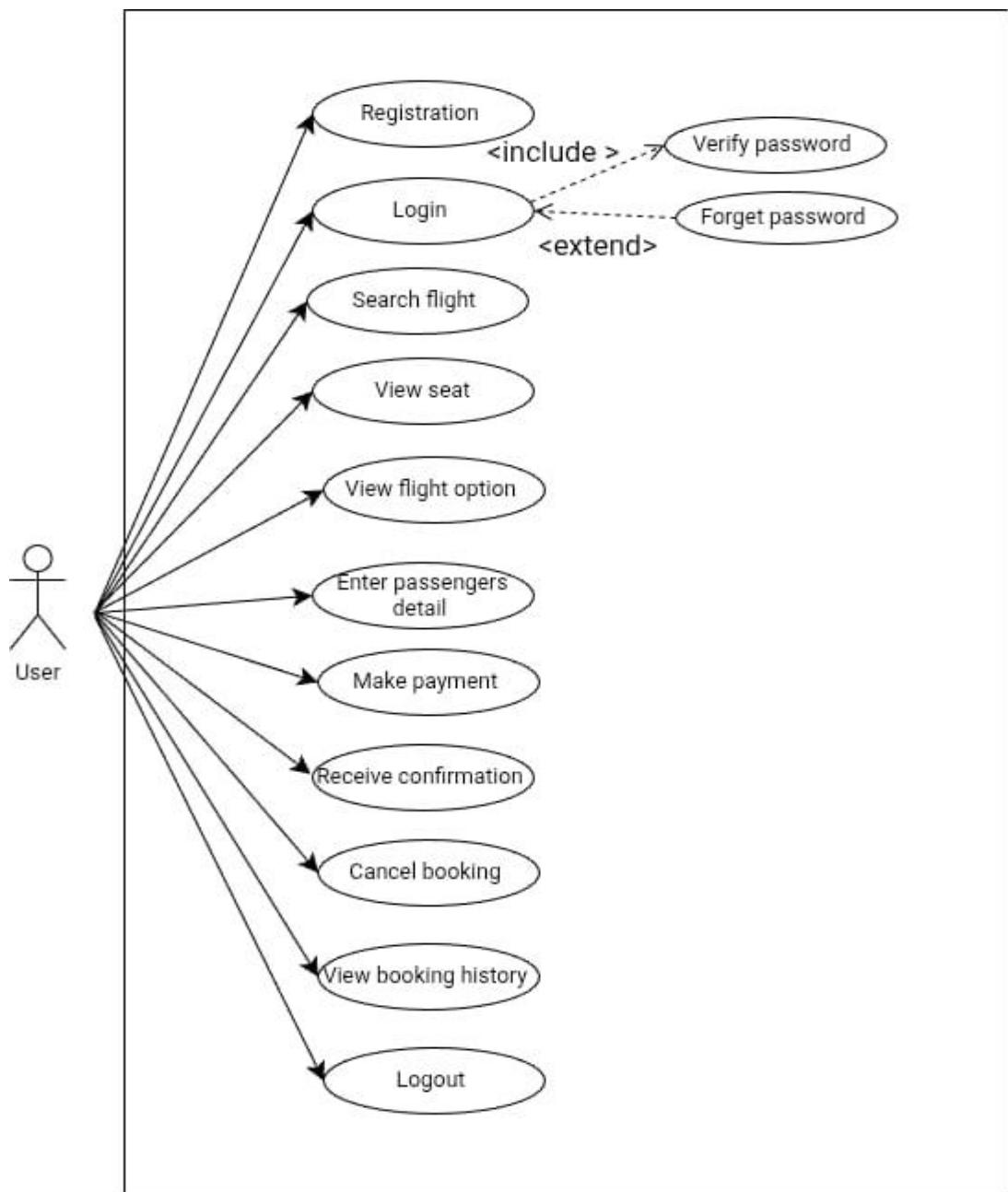
Draw the system's boundaries using a rectangle that contains use cases, place actor outsides the system's boundaries.

USE-CASE DIAGRAM FOR Admin

USE-CASE DIAGRAM FOR USER

SEQUENCE DIAGRAMS

• What is Sequence Diagram?

Sequence diagram is an interaction diagram which focuses on message interchange between number of timelines. It represents the flow of messages in the system and can also be termed as Event Diagram. It illustrates how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.

• Benefits of Sequence Diagram:

Following are some advantages of sequence Diagram:
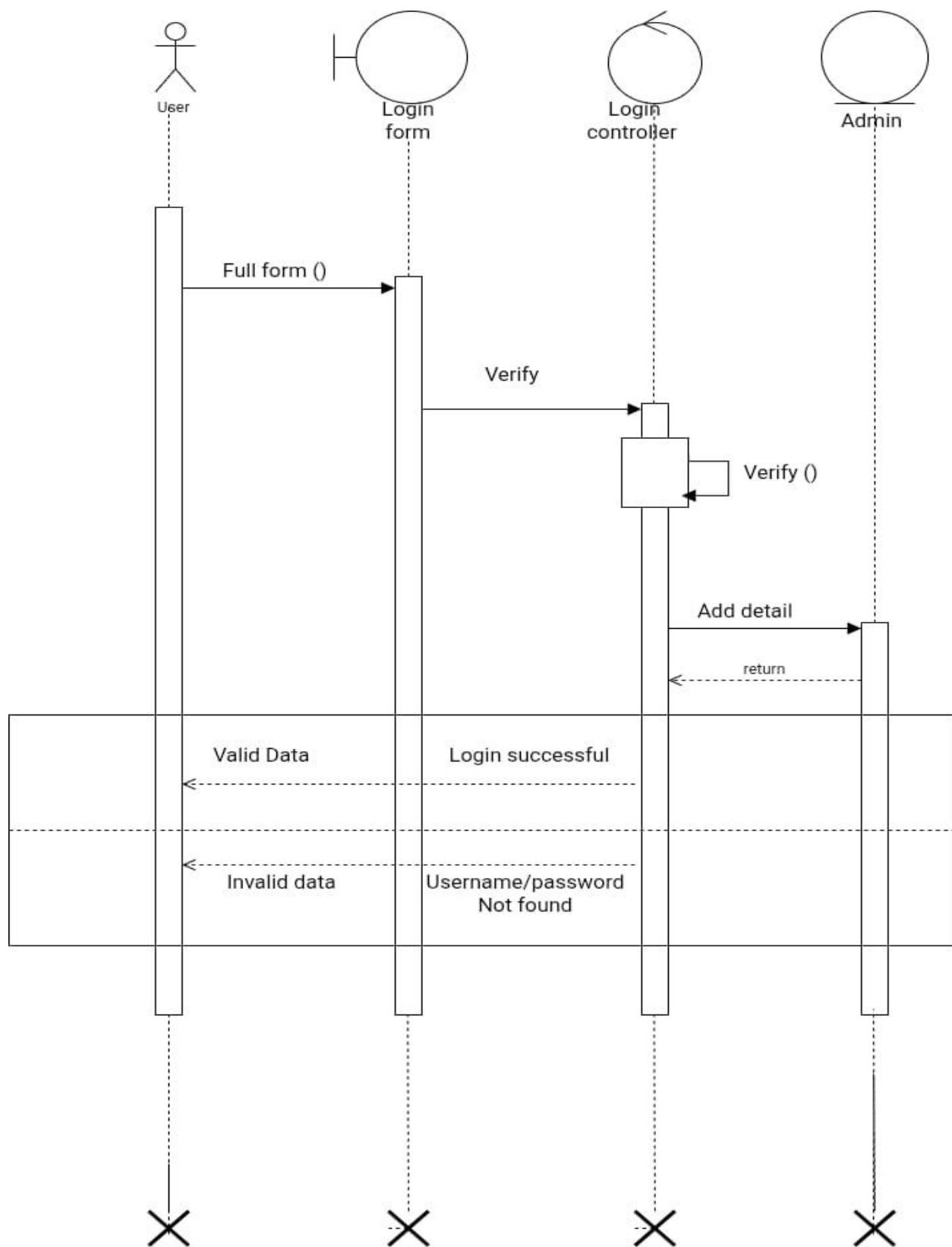
Sequence Diagram is used for high-level interactions hence helps in understanding developers design and complex system.

It has easy maintenance.
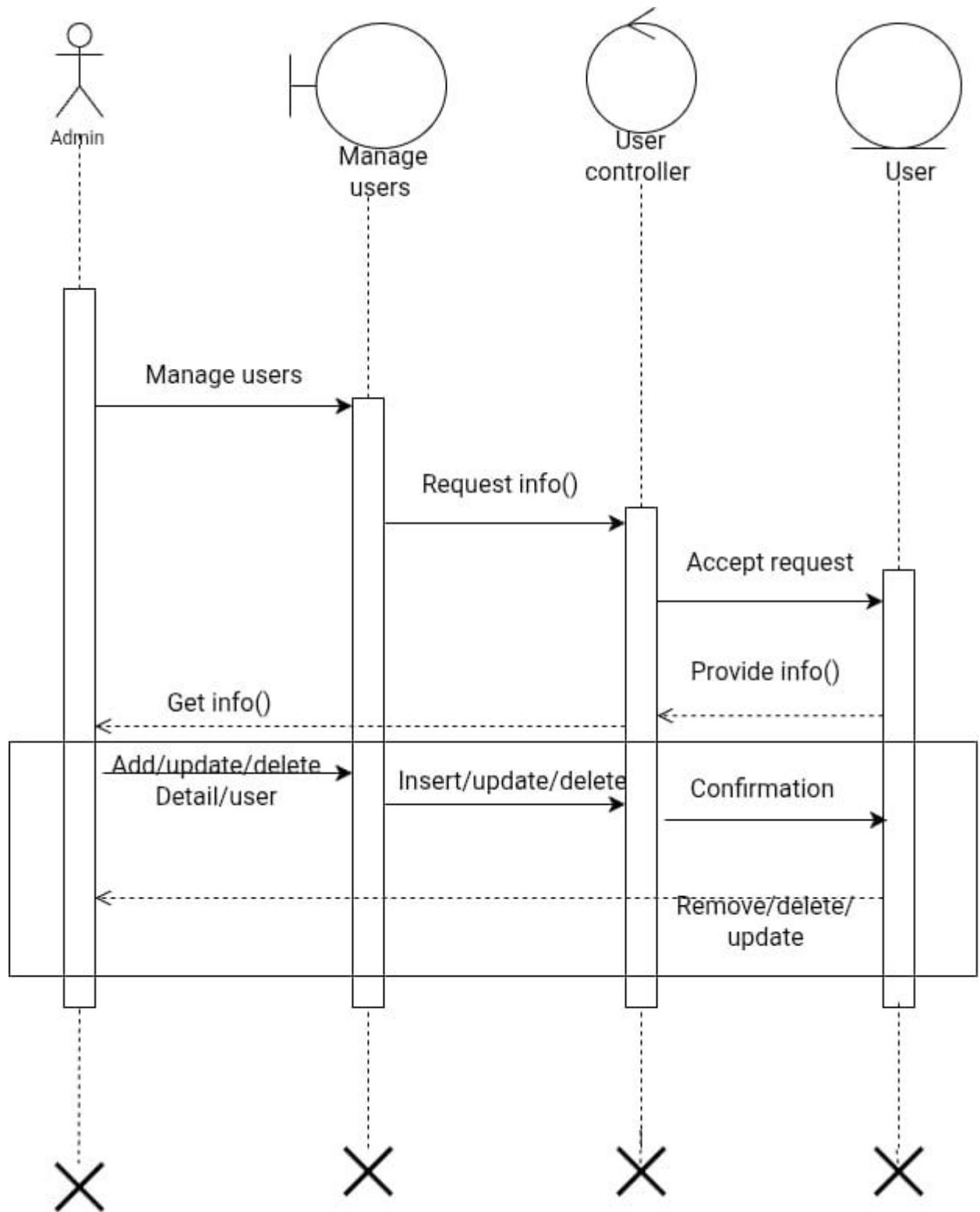
It depicts the message flow between the different objects.

It can easily update as per the new change in the system.
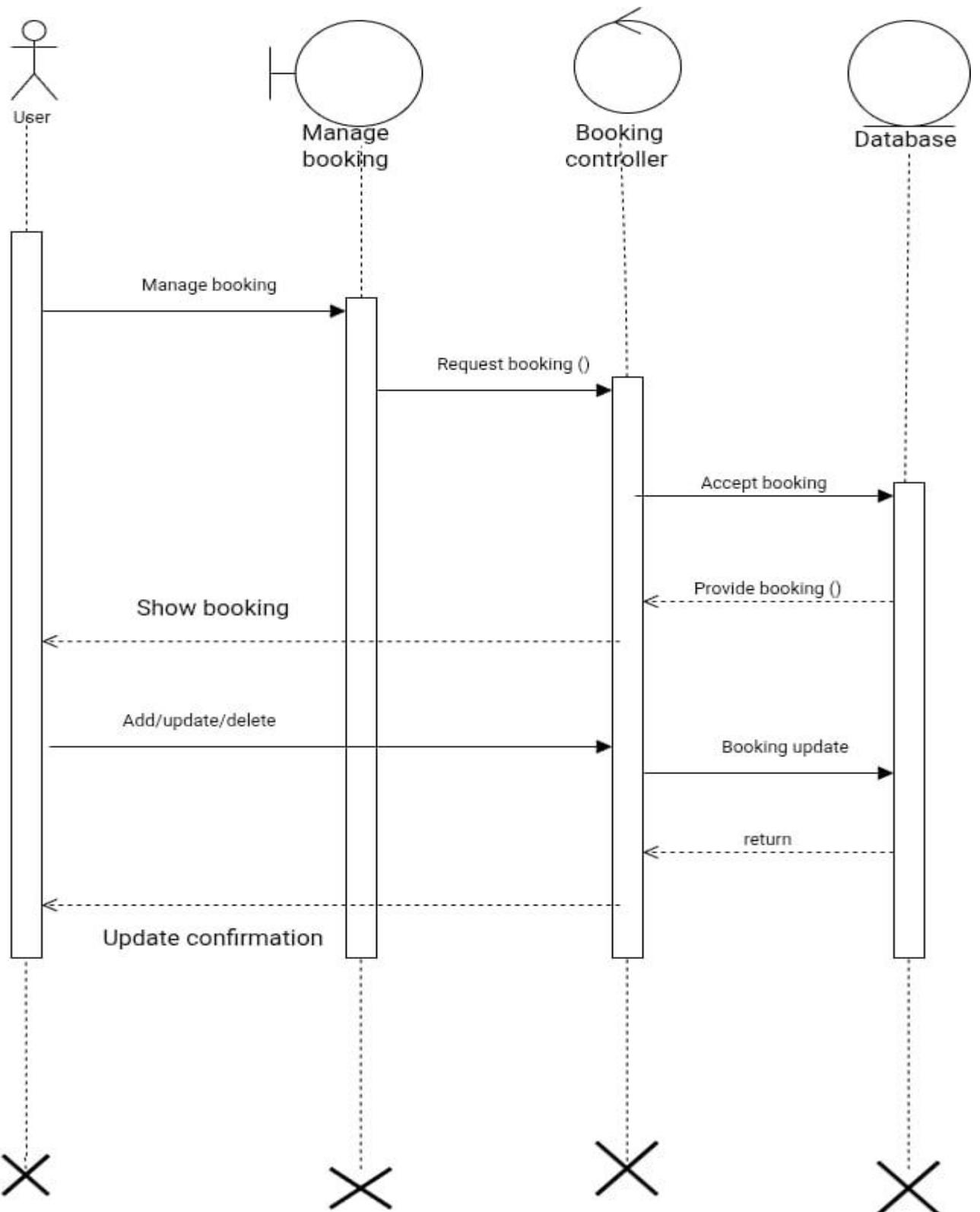
Login
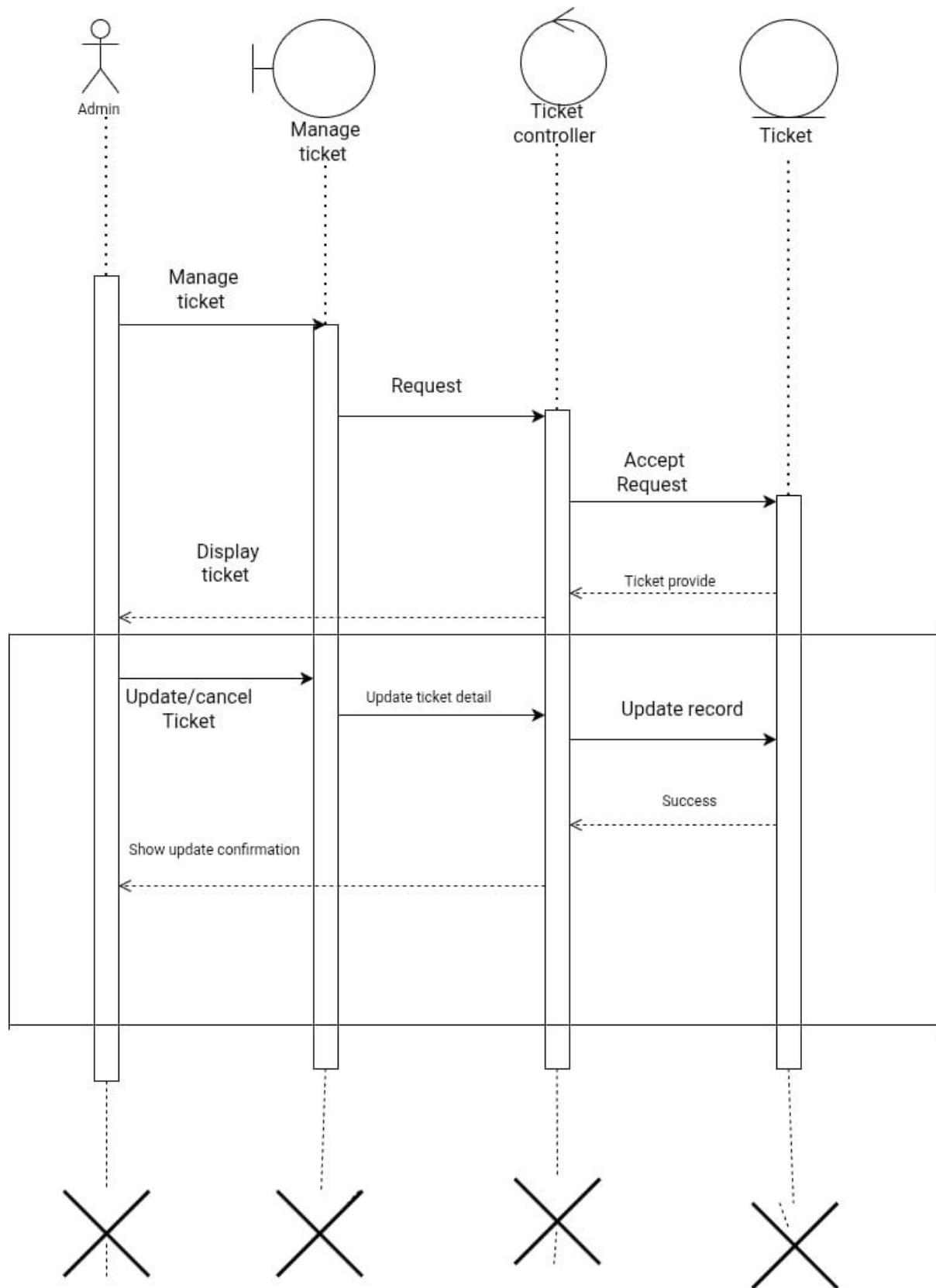
Manage user
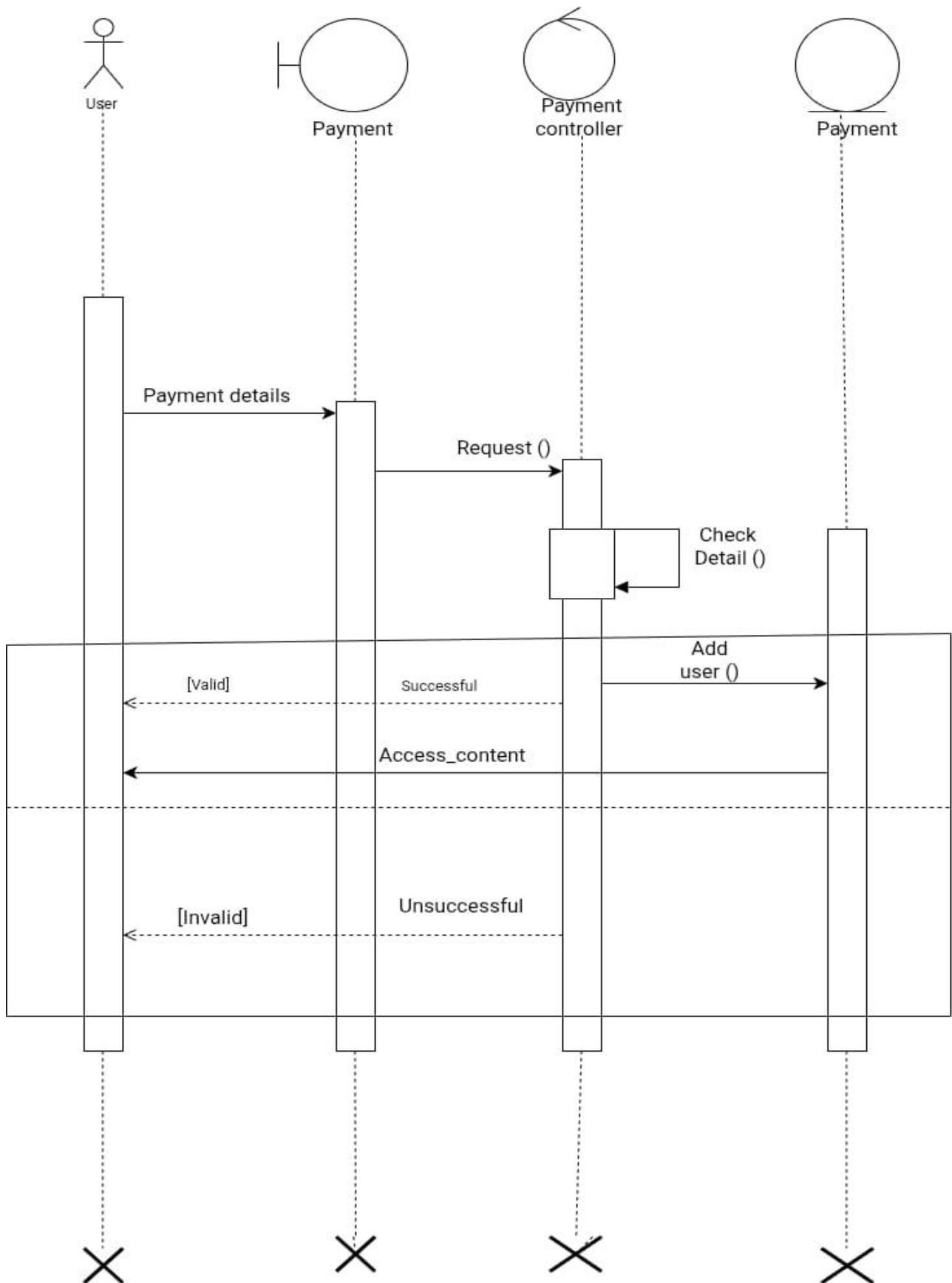
# Manage booking

Manage tickets

Manage
ticket

Request

Accept
Request

Display
ticket

Ticket provide

Update/cancel
Ticket

Update ticket detail

Update record

Success

Show update confirmation

Admin

Manage
ticket

Ticket
controller

Ticket

Payment

User — Payment — Payment controller — Payment

Payment details

Request ()

Check Detail ()

Add user ()

[Valid]   Successful

Access_content

[Invalid]   Unsuccessful

ACTIVITY DIAGRAM

What is Activity Diagram?

Activity Diagram is one of the important diagrams of UML. It represents the flow from one activity to another activity which can be shown as operations of system.

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

It represents what activities can be done in parallel and whether there are alternative paths through the workflow.

• Benefits of Activity Diagram:

Shows the progress of workflow amongst the users, and the system. It provides the visual representation of logic of a particular algorithm. It represents the relationship between one activity to another activity. It provides a better understanding of how the system works. We can use Activity diagrams to depict concurrent activities easily.

**Symbols of Activity diagram**

❖ Start / Initial point
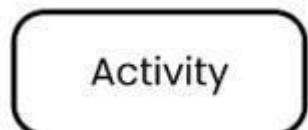
Represents the beginning of the process.

❖ End point

Represents the completion of all the flow and end of process.

❖ Activity
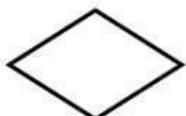
Indicates the main activity with small description within the shape.

❖ Activity flow

Shows the directional flow of the activity.
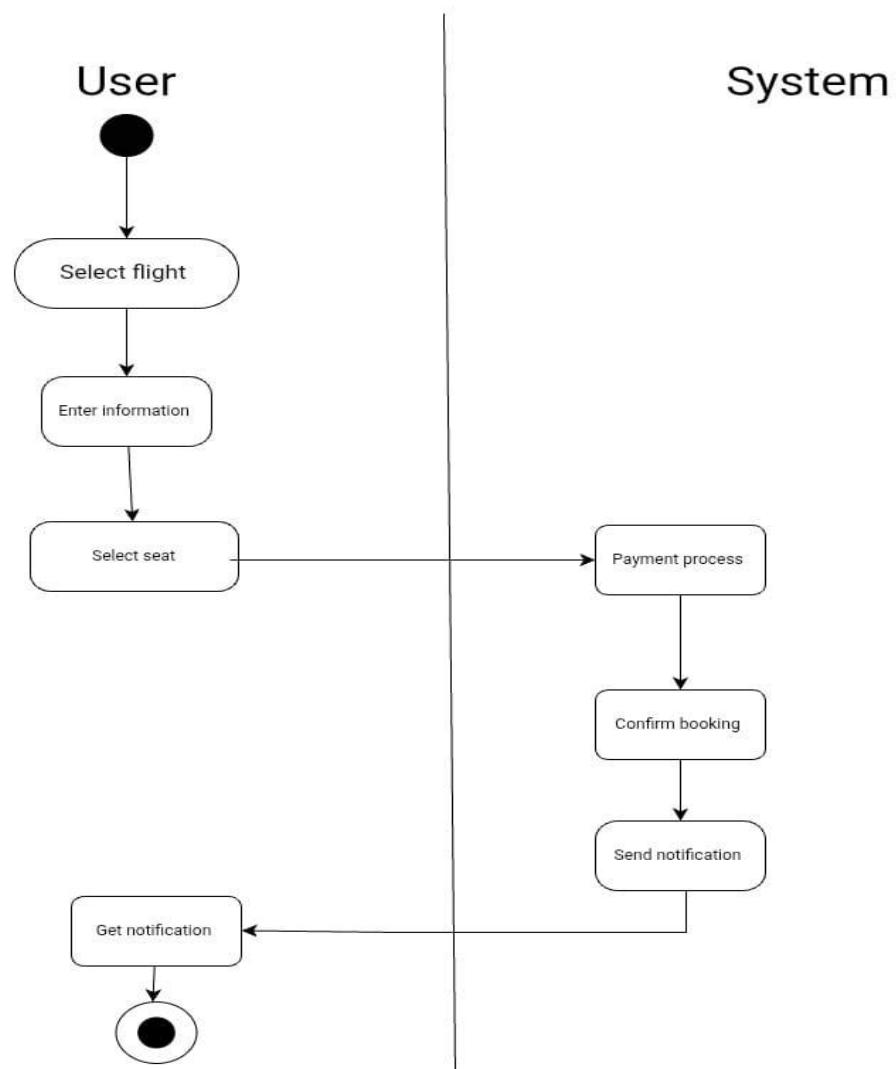
❖ Decision symbol
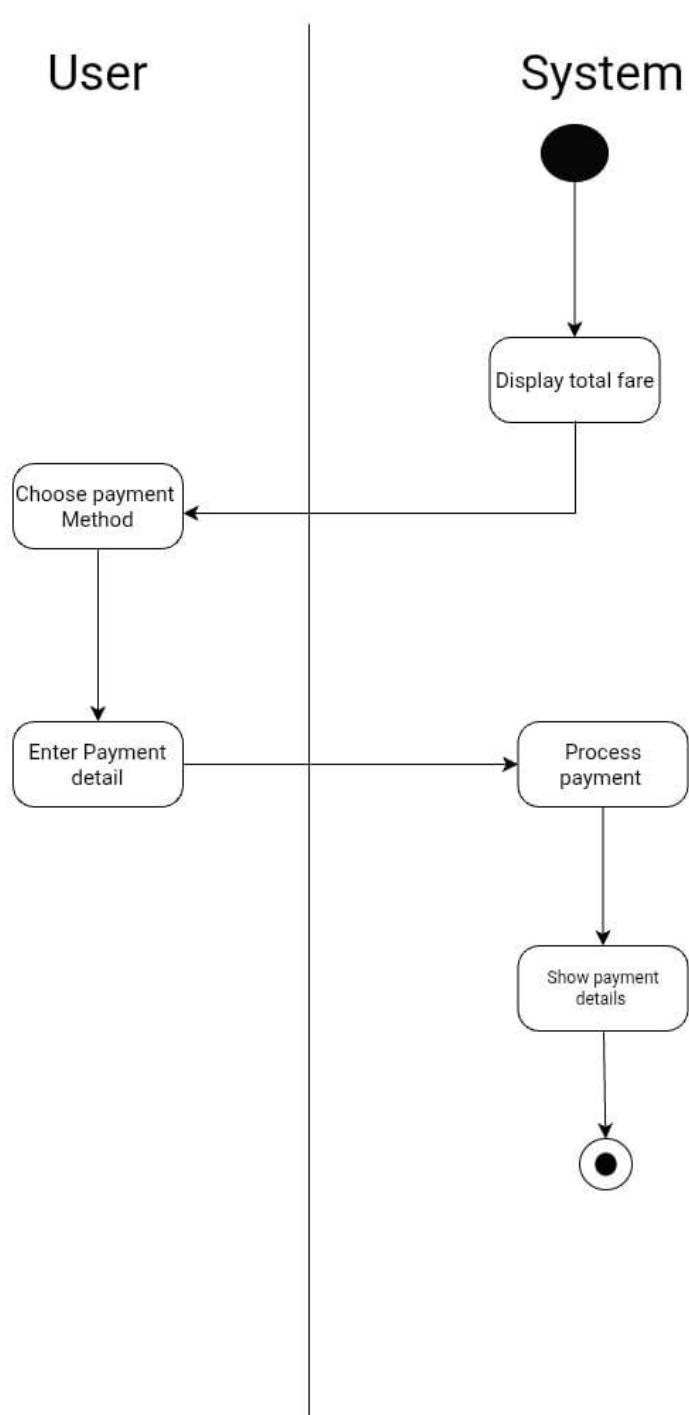
Represents decision making with one input and multiple output

Activity diagram
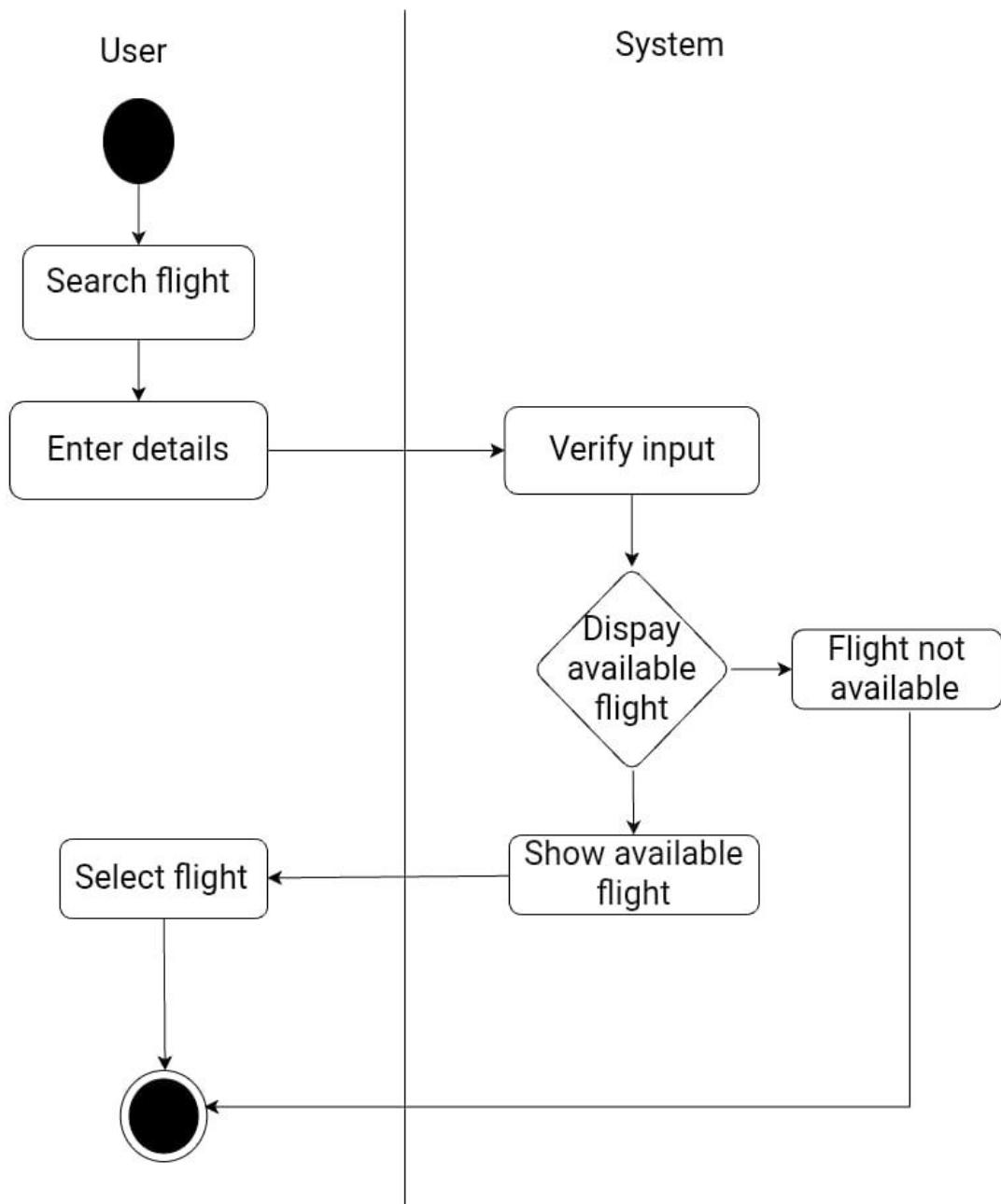
Booking

Payment

Search flight

User

System

Search flight

Enter details → Verify input

Dispay available flight → Flight not available

Show available flight → Select flight

Seat selection

User

System

Select flight

Enter information

Select seat → Payment process

Payment process → Confirm booking

Confirm booking → Send notification
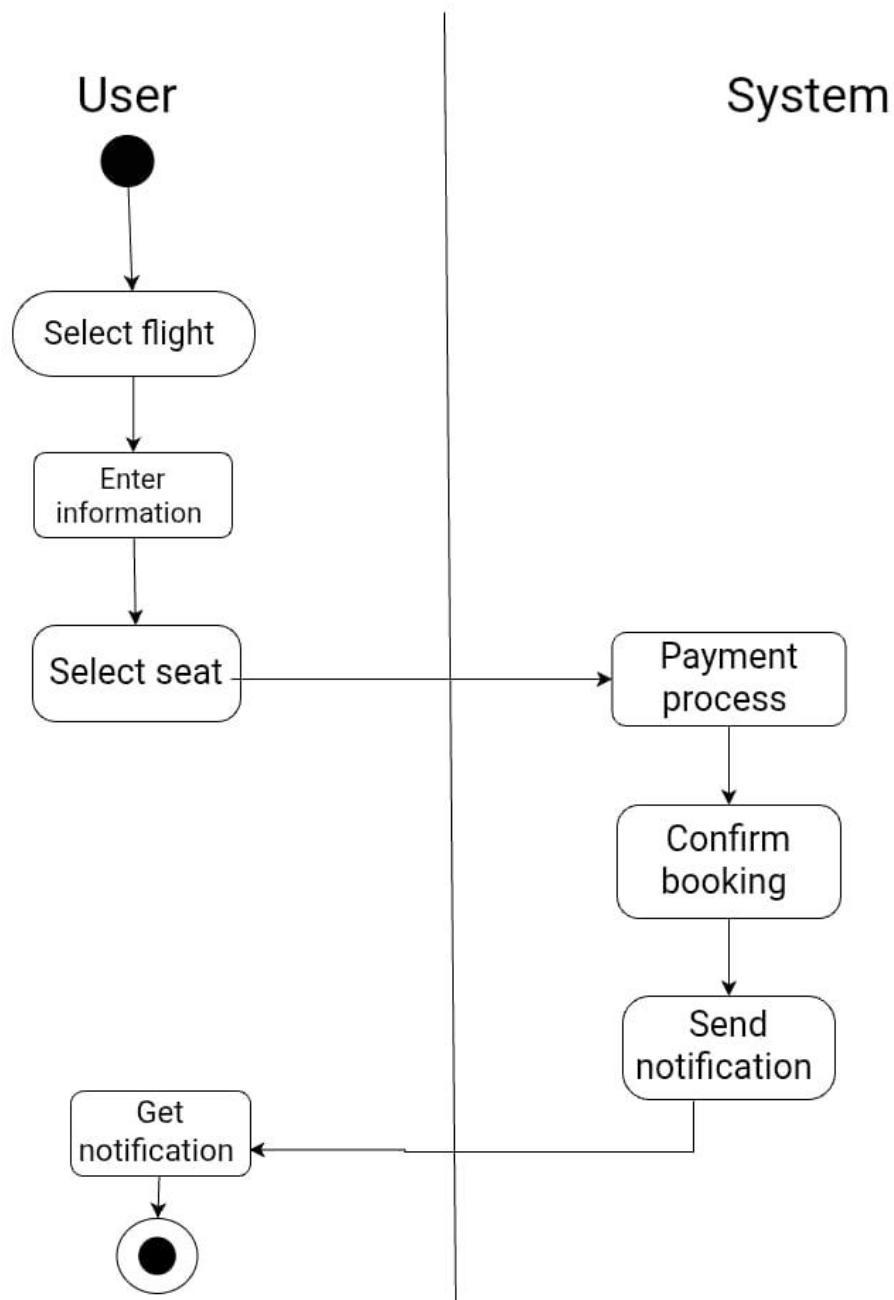
Get notification

Class Diagram

- ❖ Class diagram are one of the most useful types of diagram in UML as they clearly map out the structure of a particular system by modelling its classes, attributes , operations, and relationships between objects.
- ❖ Class diagram contain icons representing classes , interfaces and their relationships.
- ❖ Since classes are the building block of objects, class diagrams are the building blocks of UML.
- ❖ The various components in a class diagram can represent the classes that will be programmed, the main objects,or the iteractions between classes and objects
- ❖ The class shape itself consists of a rectangle with three rows. The top row contains the name of the class, the middle row contains the attributes of the class , and the bottom section expresses the methods or operations that the class may use.
- ❖ Classes and subclasses are grouped together to show the static relationship between each object.
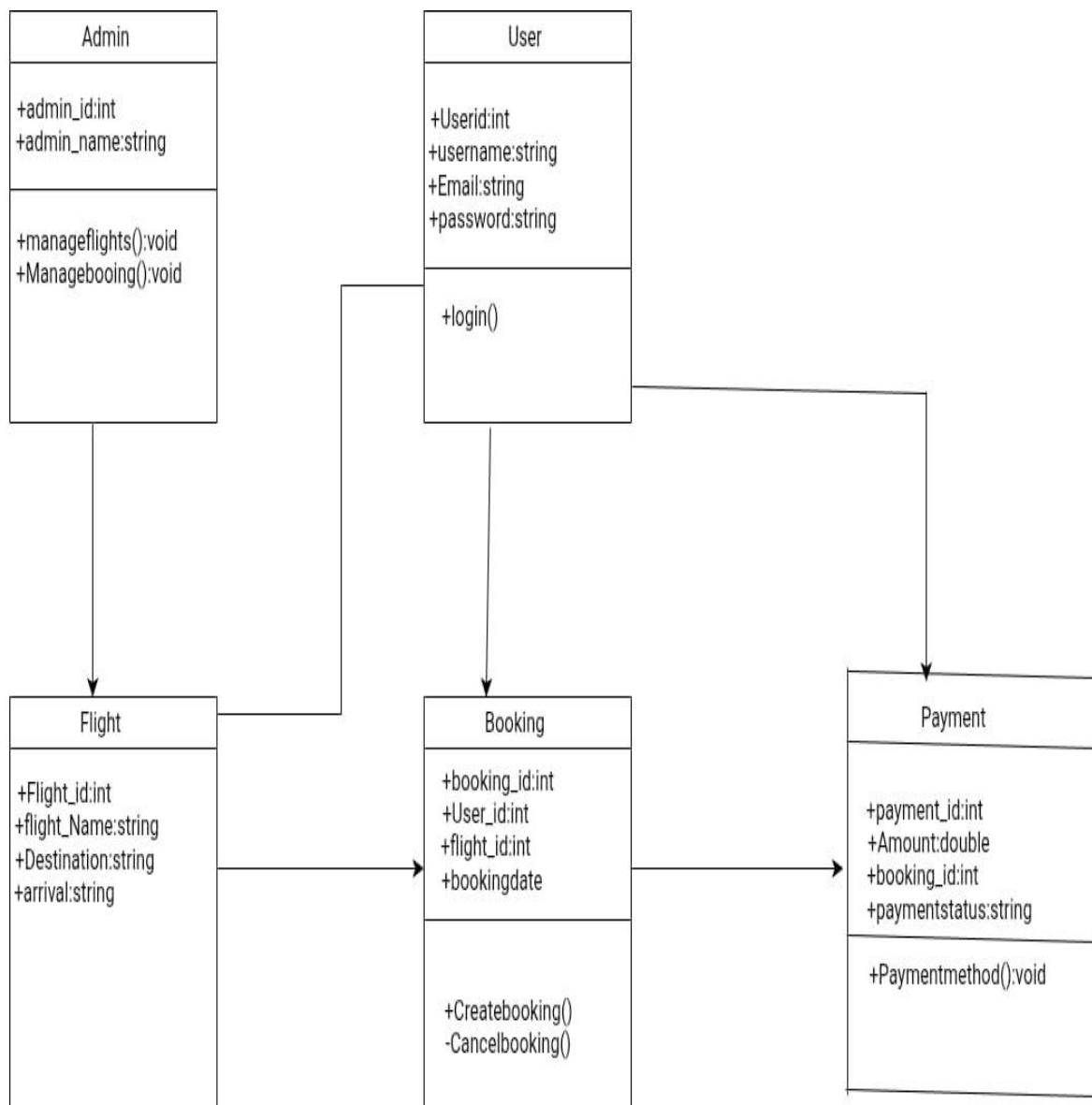
BASIC COMPONENTS OF CLASS DIAGRAM

- ❖ Upper Section
- ❖ Middle Section
- ❖ Bottom Section
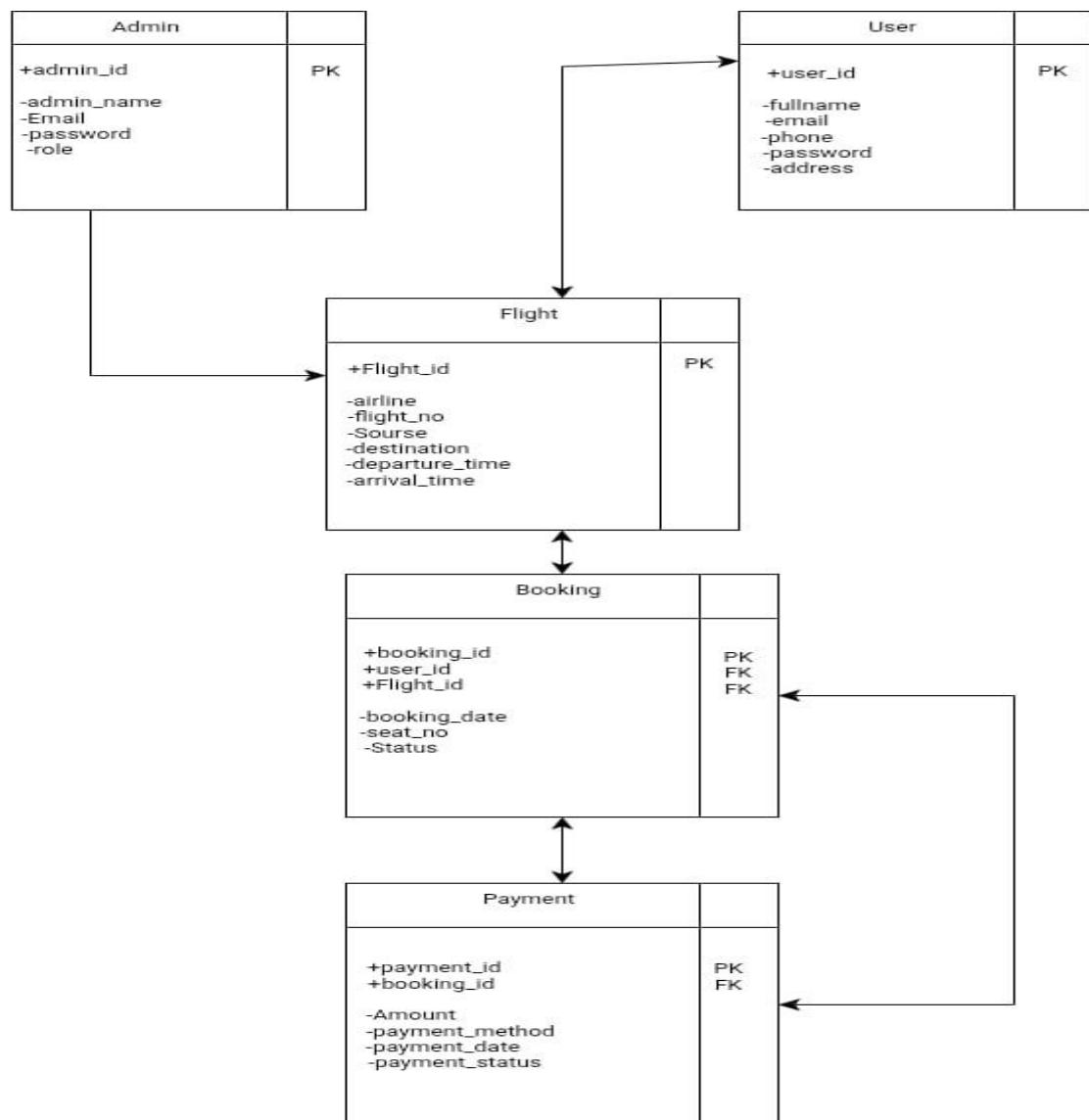
MEMBER ACCESS MODIFIERS

- ❖ All classes have different access levels depending on the access modifier(visibility). Here are the access levels with their corresponding symbols:
- ❖ Public(+)
- ❖ Private(-)
- ❖ Protected(#)
- ❖ Derived(/)
- ❖ Static (underlined)

Class Diagram

Admin

+admin_id:int
+admin_name:string

+manageflights():void
+Managebooing():void

User

+Userid:int
+username:string
+Email:string
+password:string

+login()

Flight

+Flight_id:int
+flight_Name:string
+Destination:string
+arrival:string

Booking

+booking_id:int
+User_id:int
+flight_id:int
+bookingdate

+Createbooking()
-Cancelbooking()

Payment

+payment_id:int
+Amount:double
+booking_id:int
+paymentstatus:string

+Paymentmethod():void

E-R DIAGRAM (ENTITY RELATIONSHIP DIAGRAM)

Admin	
+admin_id	PK
-admin_name -Email -password -role	

User	
+user_id	PK
-fullname -email -phone -password -address	

Flight	
+Flight_id	PK
-airline -flight_no -Sourse -destination -departure_time -arrival_time	

Booking	
+booking_id +user_id +Flight_id	PK FK FK
-booking_date -seat_no -Status	

Payment	
+payment_id +booking_id	PK FK
-Amount -payment_method -payment_date -payment_status	

DATA DICTIONARY

1. Admin Table

Field Name	Data type	Size	Constraints	Description
Admin_id	INT	10	PK,Auto incremet	Unique ID for admin
Admin_name	VARCHAR	50	Not Null	Admin full name
email	VARCHAR	100	Unique	Login email
password	VARCHAR	255	Not Null	Login password
Role	VARCHAR	20	Default:staff	Type of admin

2. flight table

Field Name	Data type	Size	Constraints	Description
Flight_id	INT	10	PK,Auto increment	Unique ID of flight
Airline_id	INT	10	FK-Airlines	Airlines reference
Flight_number	VARCHAR	20	unique	Uniqueflight number
Source	VARCHAR	50	Not null	From location
Destination	VARCHAR	50	Not null	To location
Departure_detetime	DATETIME	-	Not null	Departure time
Arrival_datetime	DATETIME	-	Not null	Arrival time
Total_seats	INT	5	Not null	Total seats
Available_seats	INT	5	Not null	Seats available

3. user table

Field name	Data type	Size	Constraints	Description
User_id	INT	10	PK,Auto increment	Unique user ID
Full_name	VARCHAR	100	Not null	Passenger name
Email	VARCHAR	100	Unique	User email
Phone	VARCHAR	15	Not null	User phone number
Password	VARCAHR	255	Not null	Hashed password
Address	VARCHAR	255	Null allowed	User address

4. booking table

Field Name	Data type	Size	Consatraints	Description
Booking_id	INT	10	PK, Auto increment	Unique booking ID
User_id	INT	10	Fk-user(user_id)	User reference
Flight_id	INT	10	FK-flight(flight_id)	Flight reference
Booking_date	DATETIME		Not null	Date of booking
Seat_count	INT	3	Not null	Number of seats booked
status	VARCHAR	20	Defalt:confirmed	Booking status(confirmed/cancelled)

5. payment table

Field Name	Data type	Size	Consatraints	Description
Payment-id	INT	10	PK , Auto incremenr	Unique payment ID
Booking_id	INT	10	Fk-Booking(booking_id)	Booking reference
Amount	DECIMAL	10,2	Not null	Ticket amount
Payment_method	VARCHAR	50	Not null	UPI/Card/NetBanking
Payment_date	DATETIME	-	Not null	Date of payment
Payment_status	VARCHAR	20	Dafault:Success	Success/Failed/pending