National University of Computer & Emerging Sciences

“Software Design Specifications”

Version: *1.0*

“*Flight Management System (Flighty)*”

Instructor:

Miss Fizza Mansoor

Group Members:

ARSH (23K-0078)
MIR AHMED (23I-0142)

Table of Contents

1 Introduction:

1.1 Purpose of Document

This document provides a detailed software design for the Flight Management System. It includes architectural design, data structures, application workflows, and interface specifications to guide the implementation phase.

1.2 Intended Audience

- Project supervisors and evaluators
- Development team
- Quality assurance team
- Database administrators

1.3 Document Convention

1. Font: Times New Roman
2. Font Size: 18 pt (Normal text), 20 pt (Headings)
3. Headings use bold titles
4. Diagrams and examples are labeled with appropriate titles
5. Code: Highlighted in monospace font

1.4 Project Overview

A web-based flight management system built with Flask (Python) and MySQL that provides flight booking and management, user registration/authentication, admin dashboard for flight/airline management and payment processing

1.5 Scope

The Covers system design including:

- Database schema
- Application workflows
- Security considerations
- Interface specifications

Excludes:

- Hardware infrastructure details
- Advanced analytics modules
- Mobile application components

2 Design Considerations:

2.1 Assumptions and Dependencies

- Users have internet access
- MySQL server is available
- Python 3.7+ environment
- Flask framework dependencies
- Modern web browsers (Chrome/Firefox/Edge)

2.2 Risks and Volatile Areas

- SQL injection vulnerabilities
- Session hijacking risks
- Payment processing security
- Database connection failures
- Security and privacy risks due to handling sensitive
- Performance under high load

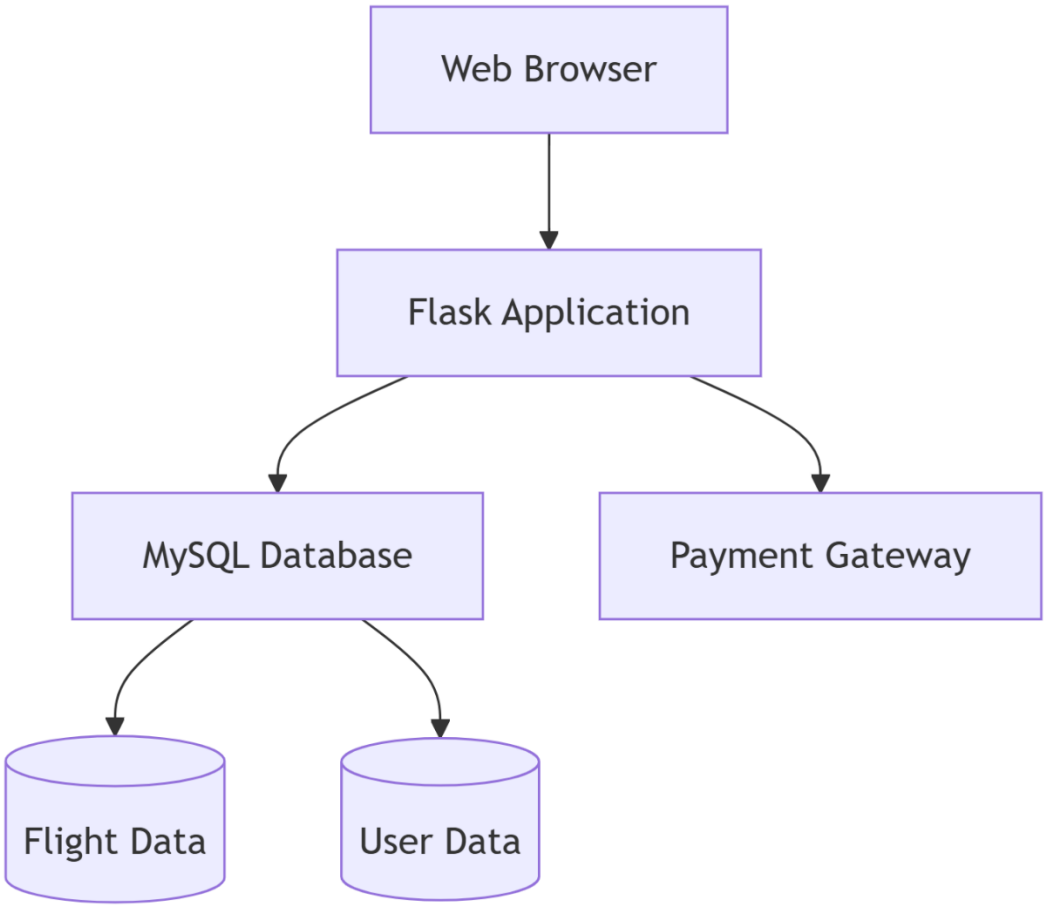
3 System Architecture:

3.1 System Level Architecture

System Decomposition:

- User Authentication
- Flight Management
- Booking System
- Admin Dashboard

Relationship Between Elements:



External Interfaces:

- MySQL Database
- Web Browser (HTTP/HTTPS)
- Payment Gateway (Simulated)

Major Physical Design:

- Flask Application runs through a web browser.
- Backend services are hosted via python integrated through MySQL.

Global Design Strategies:

- Exception handling at API and UI levels.
- Session management and secure login.

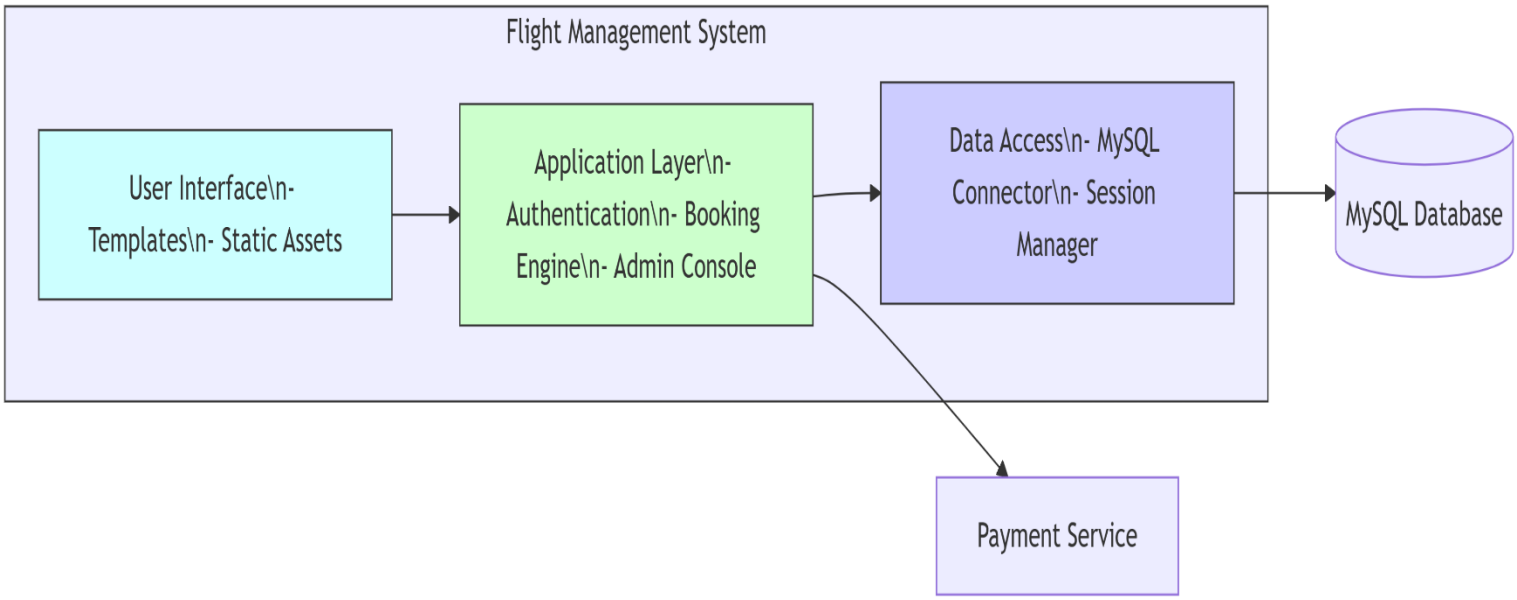
3.2 Software Architecture

Layered architecture:

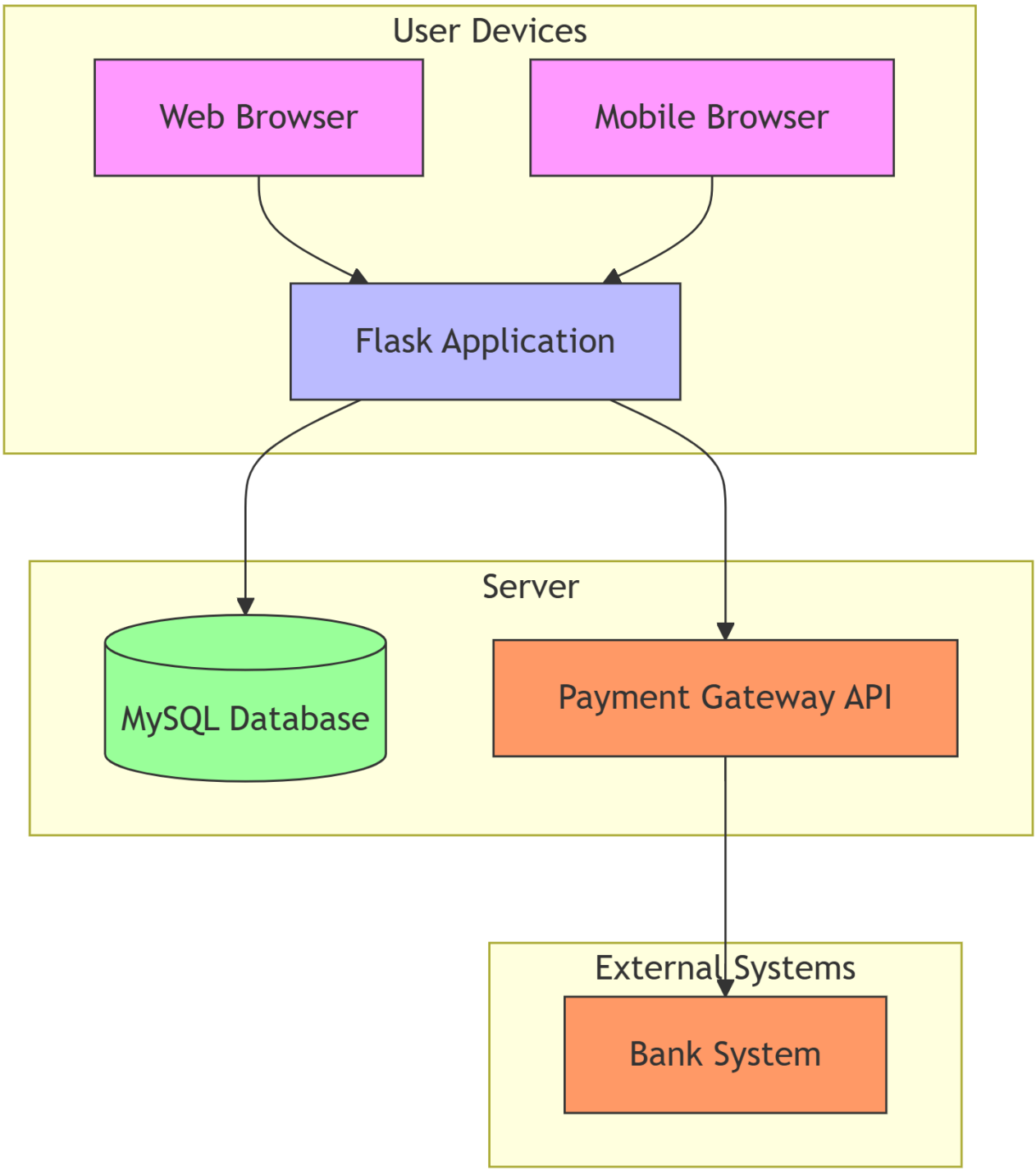
- ***User Interface Layer:*** Flask templates (HTML/CSS/JS) responsive web views.
- ***Business Logic Layer:*** Flask routes and business logic / Provider patterns.
- ***Service Layer:*** API communication and Workbench interaction.
- ***Data Access Layer:*** Direct communication with MySQL connector

System Architecture Diagrams:

- ***Component Diagram***



- ***Deployment Diagram***



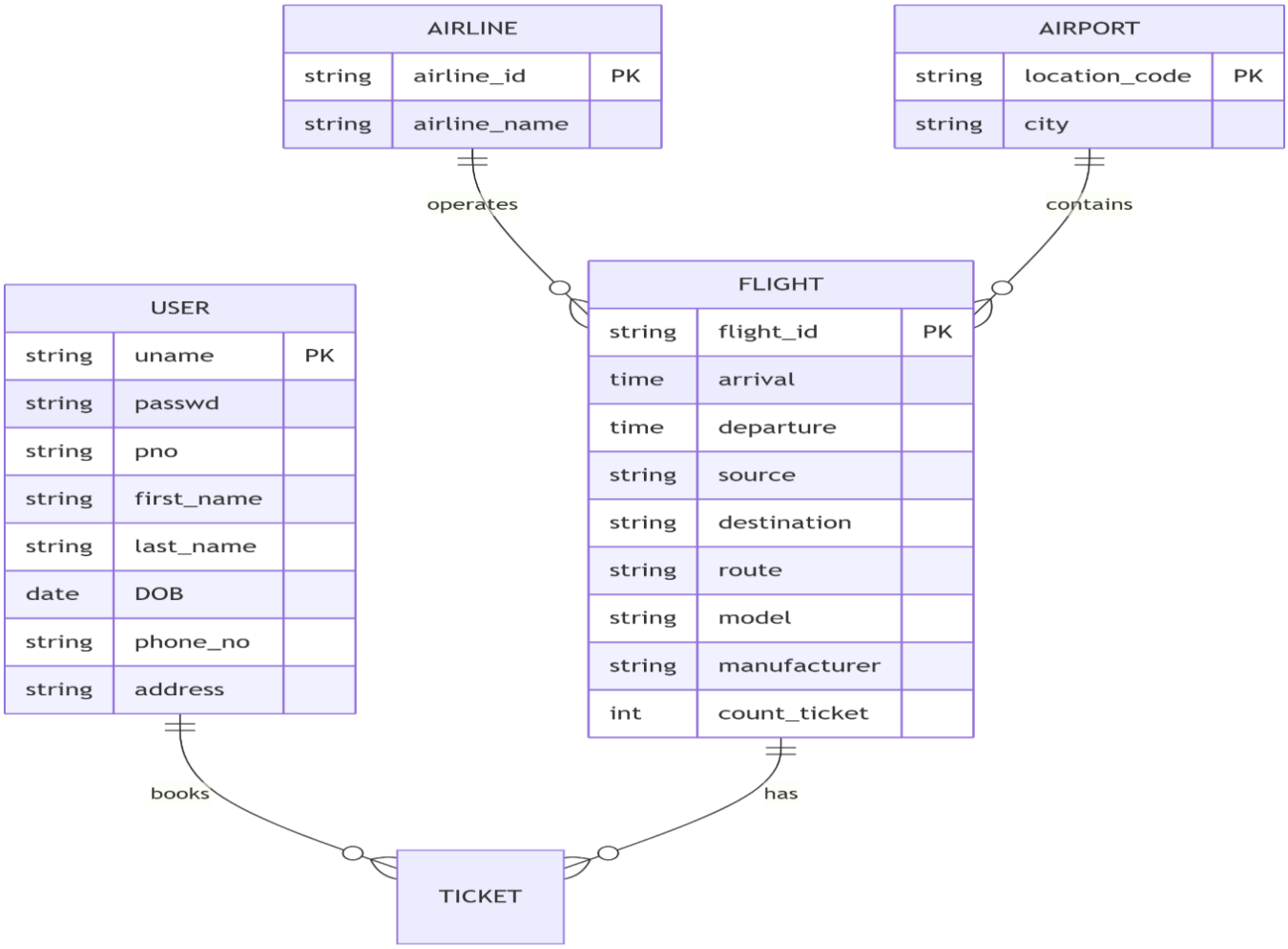
4 Design Strategy:

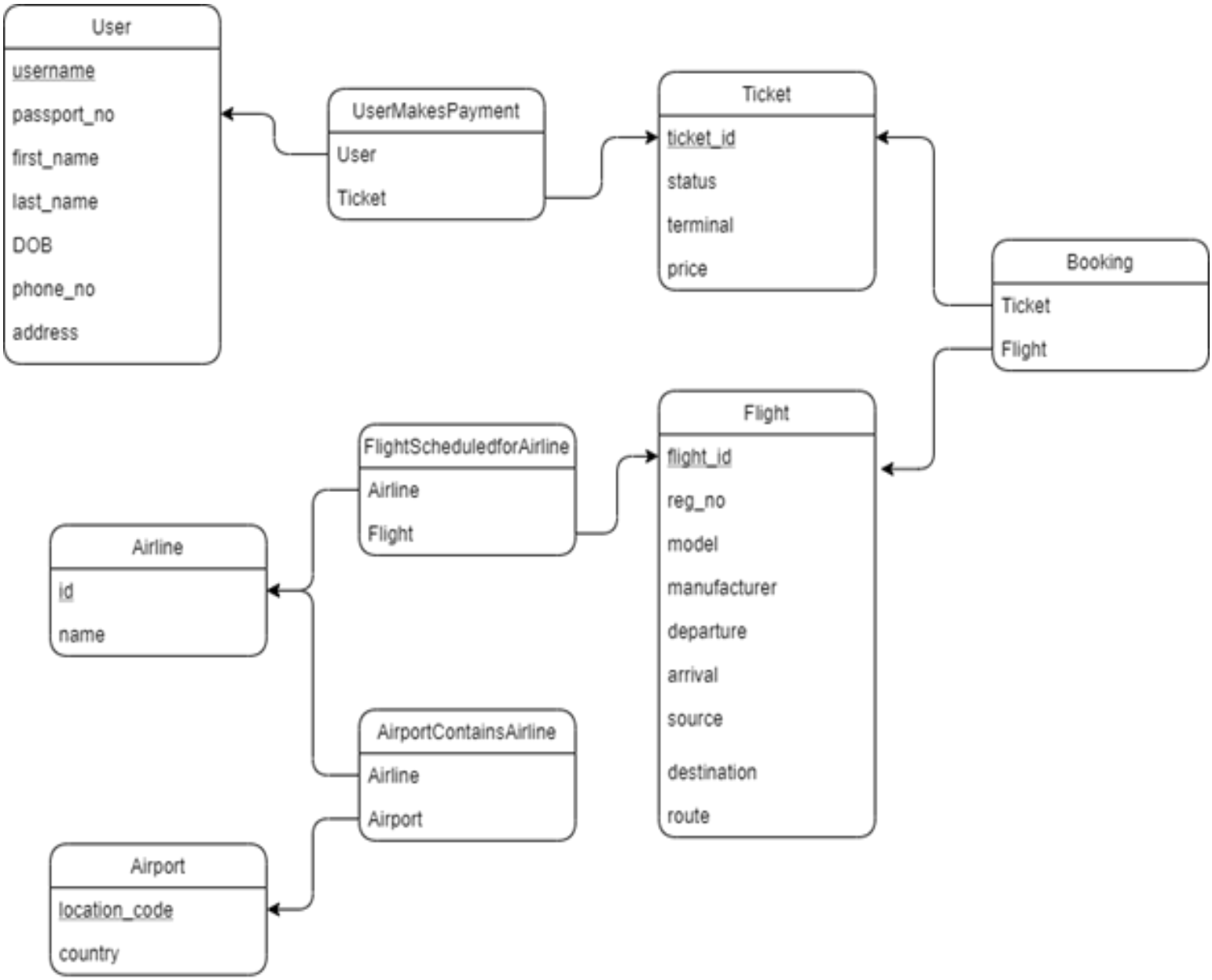
- **Modularity**: Separated by functional components
- **Security**: Password hashing, session management
- **Error Handling**: Comprehensive exception handling
- **Scalability**: Stateless design where possible

5 Detailed System Design:

5.1 Database Design

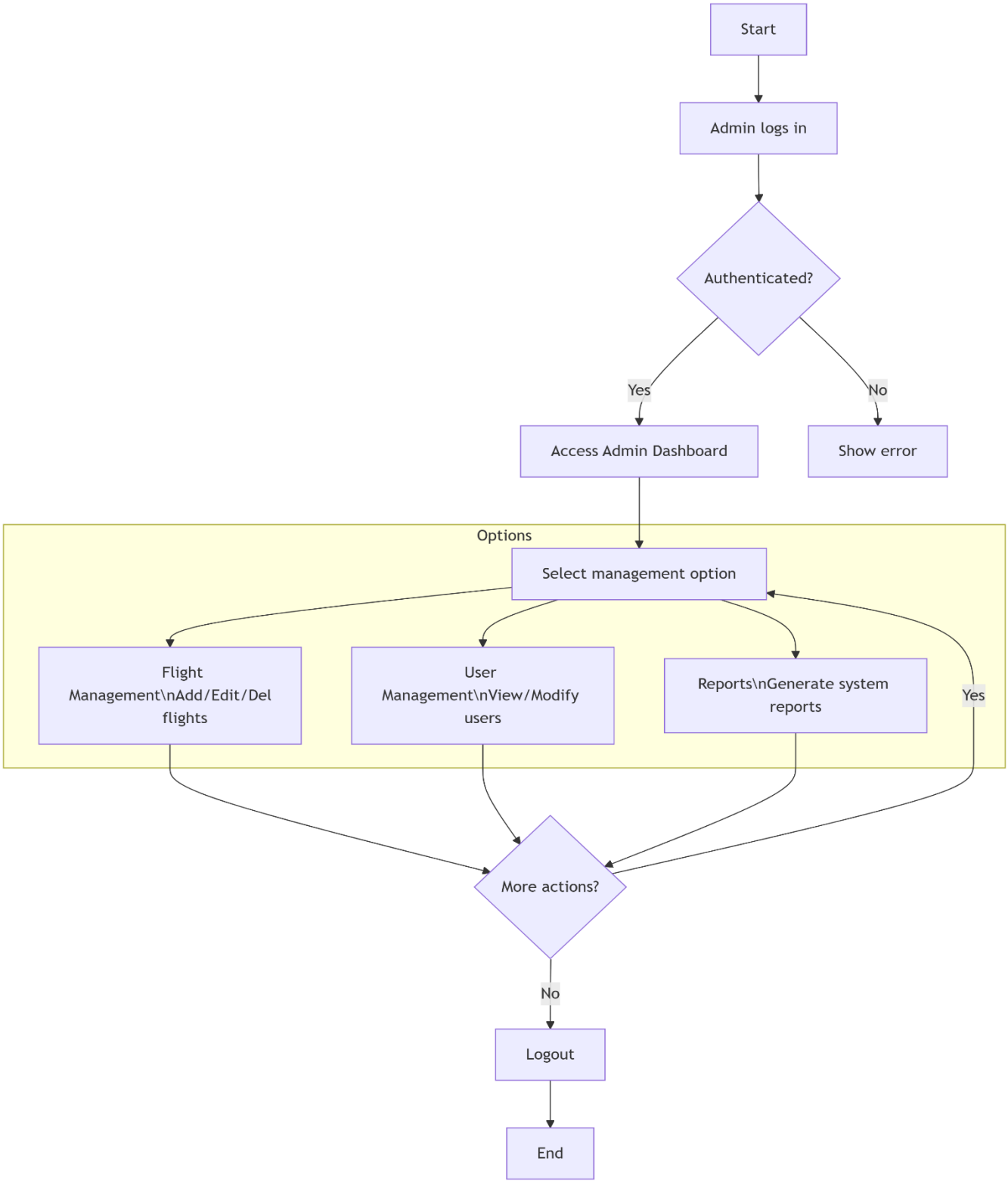
5.1.1 ER Diagram



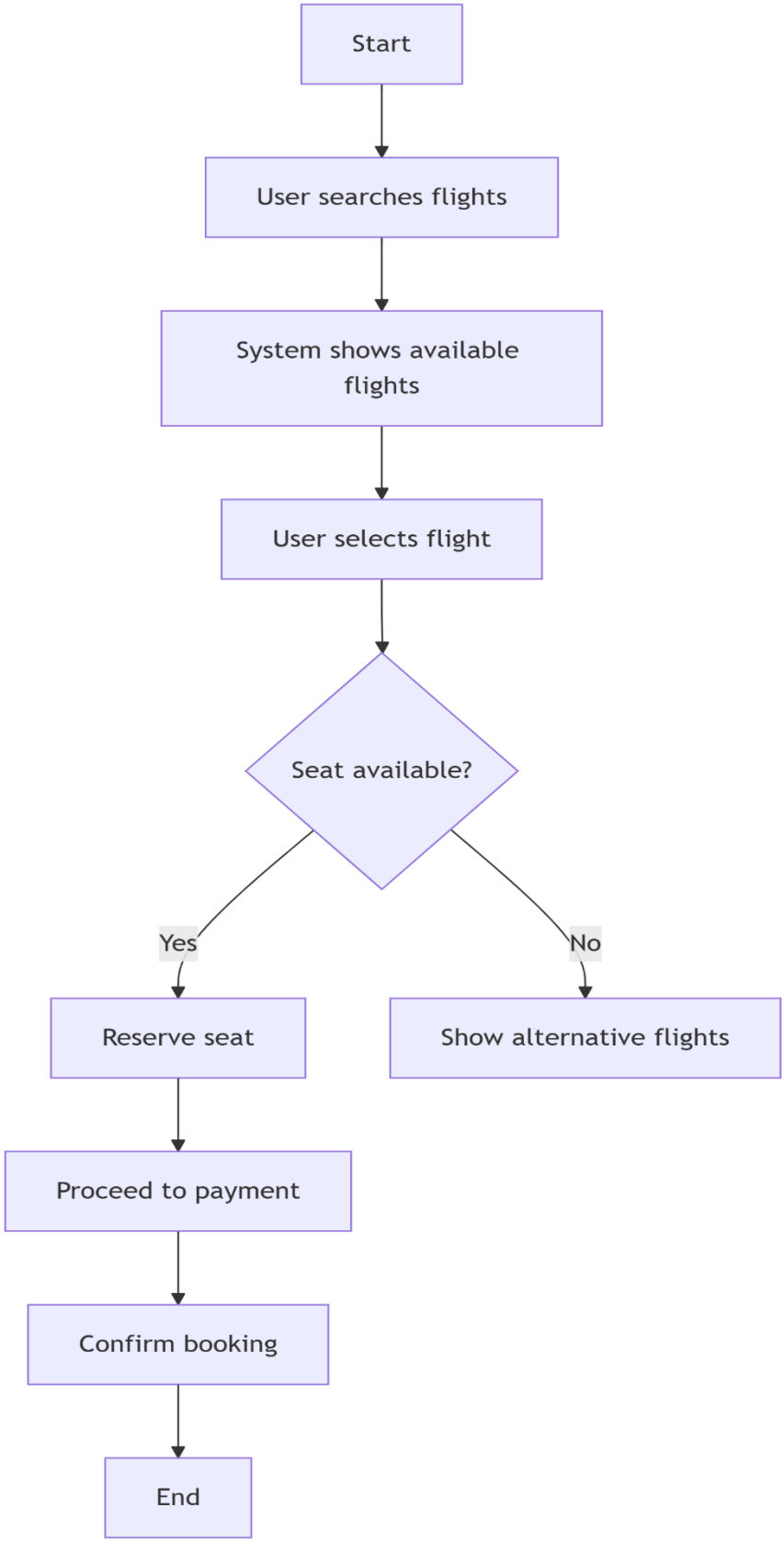

User
username
passport_no
first_name
last_name
DOB
phone_no
address
UserMakesPayment
User
Ticket
Ticket
ticket_id
status
terminal
price
Booking
Ticket
Flight
Flight
flight_id
reg_no
model
manufacturer
departure
arrival
source
destination
route
FlightScheduledforAirline
Airline
Flight
Airline
id
name
AirportContainsAirline
Airline
Airport
Airport
location_code
country

5.1.2 Activity Diagram

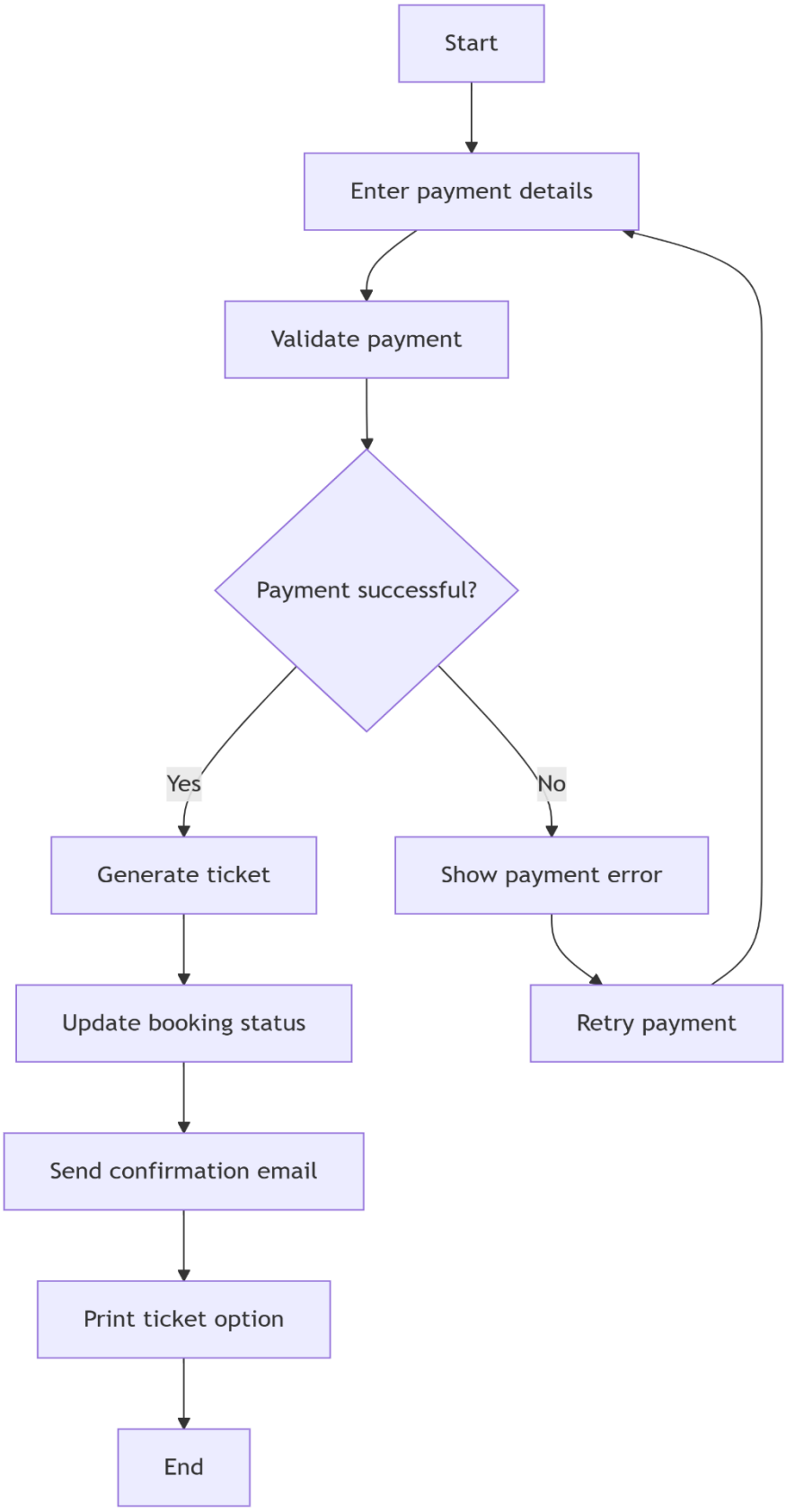
5.1.2.1 Administration



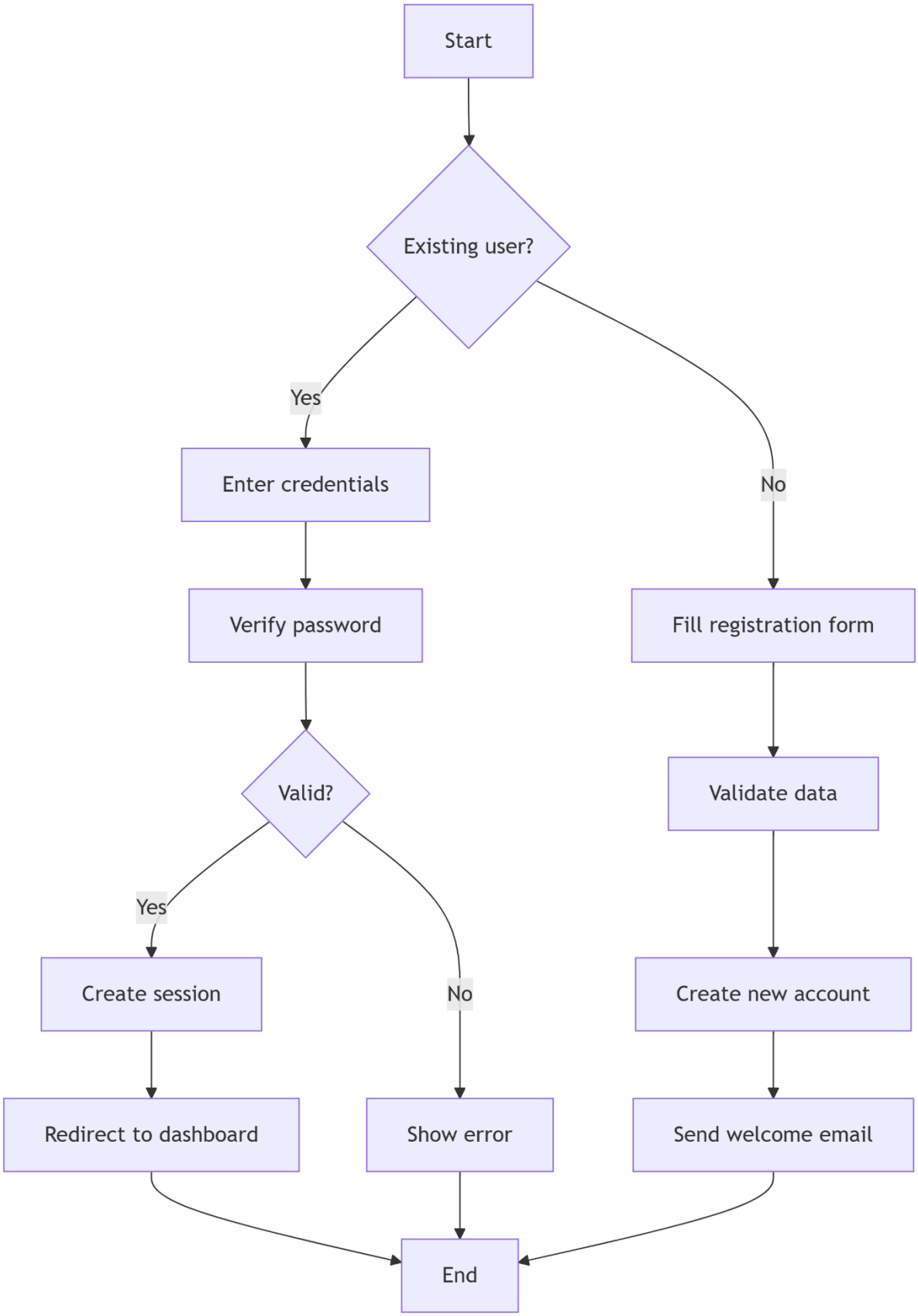
5.1.2.2 Book Ticket



5.1.2.3 Payment and Ticket Print



5.1.2.4 Login and Signup



5.1.3 Data Dictionary

5.1.3.1 Customer

username, passport no, phone no, first name, last name

Customer	
Name	Customer
Alias	Passenger, User
Where-used/how-used	Booking, Authentication, Profile Management
Content description	Customer = ID + Name + Contact + Booking History

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
uname	*Unique username*	*VARCHAR*	*50*	*No*		*PRIMARY*
pno	*Passport number*	*VARCHAR*	*20*	*No*		*UNIQUE*
phone_no	*Contact number*	*VARCHAR*	*15*	*No*		*UNIQUE*
first_name	*Legal first name*	*VARCHAR*	*50*	*No*		
last_name	*Legal last name*	*VARCHAR*	*50*	*No*		

5.1.3.2 Book Ticket

Ticket_id, flight_id, uname, status

Book Ticket	
Name	Ticket
Alias	Booking, Reservation
Where-used/how-used	Flight Booking System
Content description	Ticket = ID + Flight + Passenger + Status

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
Id	*Unique Pid*	*String*	*36*	*No*	*Auto-*	
ticket_id	*Unique booking ID*	*VARCHAR*	*20*	*No*		PRIMARY
flight_id	*Associated flight*	*VARCHAR*	*10*	*No*		FOREIGN
uname	*Passenger username*	*VARCHAR*	*50*	*No*		FOREIGN
Status	*Available or not*	*String*	*20*	*Yes*	*Booked*	

5.1.3.3 Login User

Uname, passwd

Login						
Name	User Login					
Alias	Passenger, user					
Where-used/how-used	Authentication system					
Content description	Login = Credentials + Personal Info + Validation					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
uname	*Username*	*String*	*20 char*	*No*		
passwd	*Password*	*String*	*Max 20 char*	*no*		

5.1.3.4 Admin

Admin id , admin passwd

Admin						
Name	Admin Login					
Alias	System Manager					
Where-used/how-used	Backend Management					
Content description	Admin = Elevated Privileges + System Access					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
uname	*Username*	*String*	*fixed*	*No*		
passwd	*Password*	*String*	*fixed*	*no*		

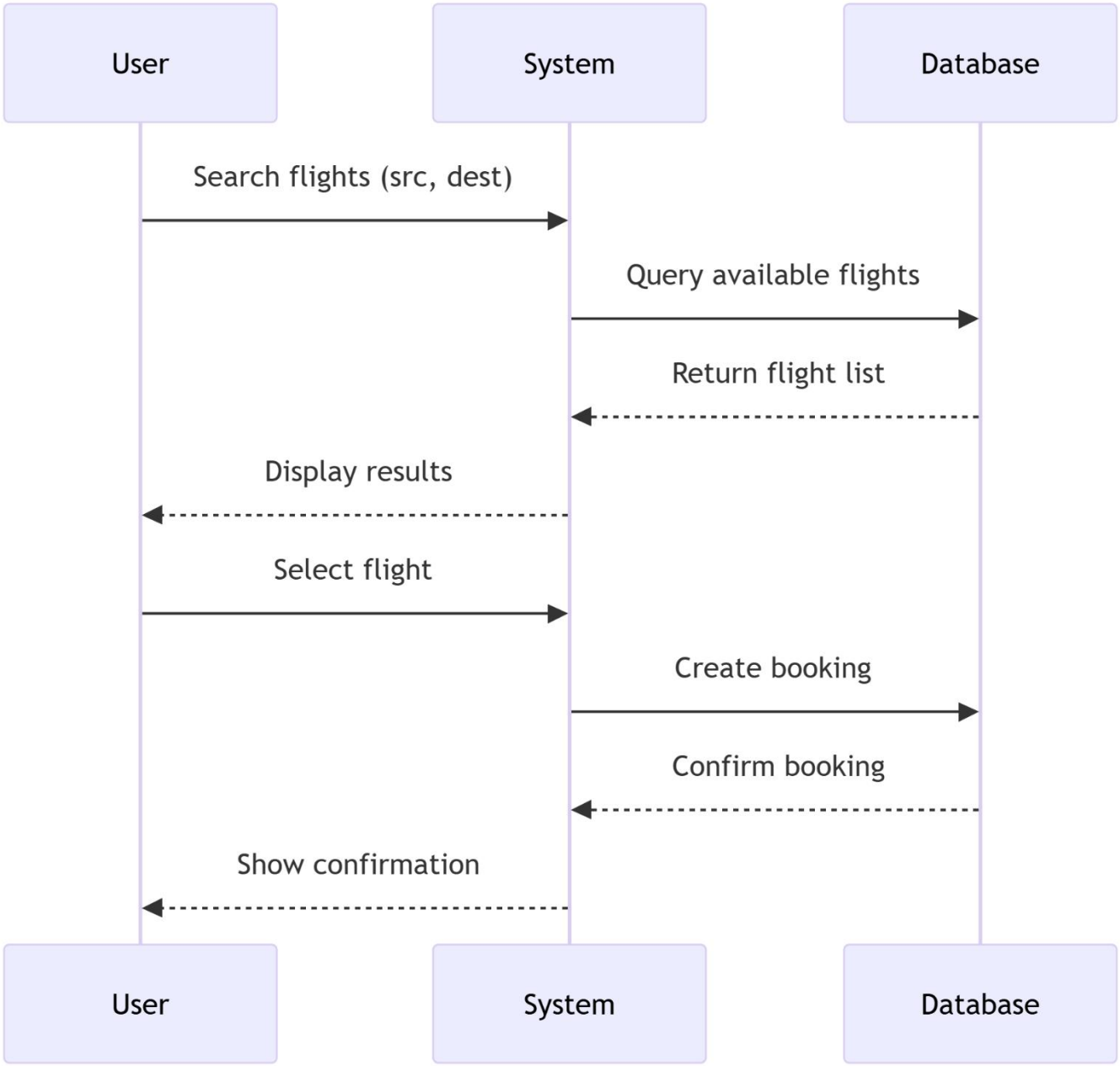
The notation to develop content description is given below:

Data construct	***Notation***	***Meaning***
	=	*is composed of*
Sequence	+	*And*
Selection	*[|]*	*either-or*
Repetition	*{}n*	*n repetitions of*
	( )	*optional data*
	* ... *	*delimits comments*

5.2 Application Design

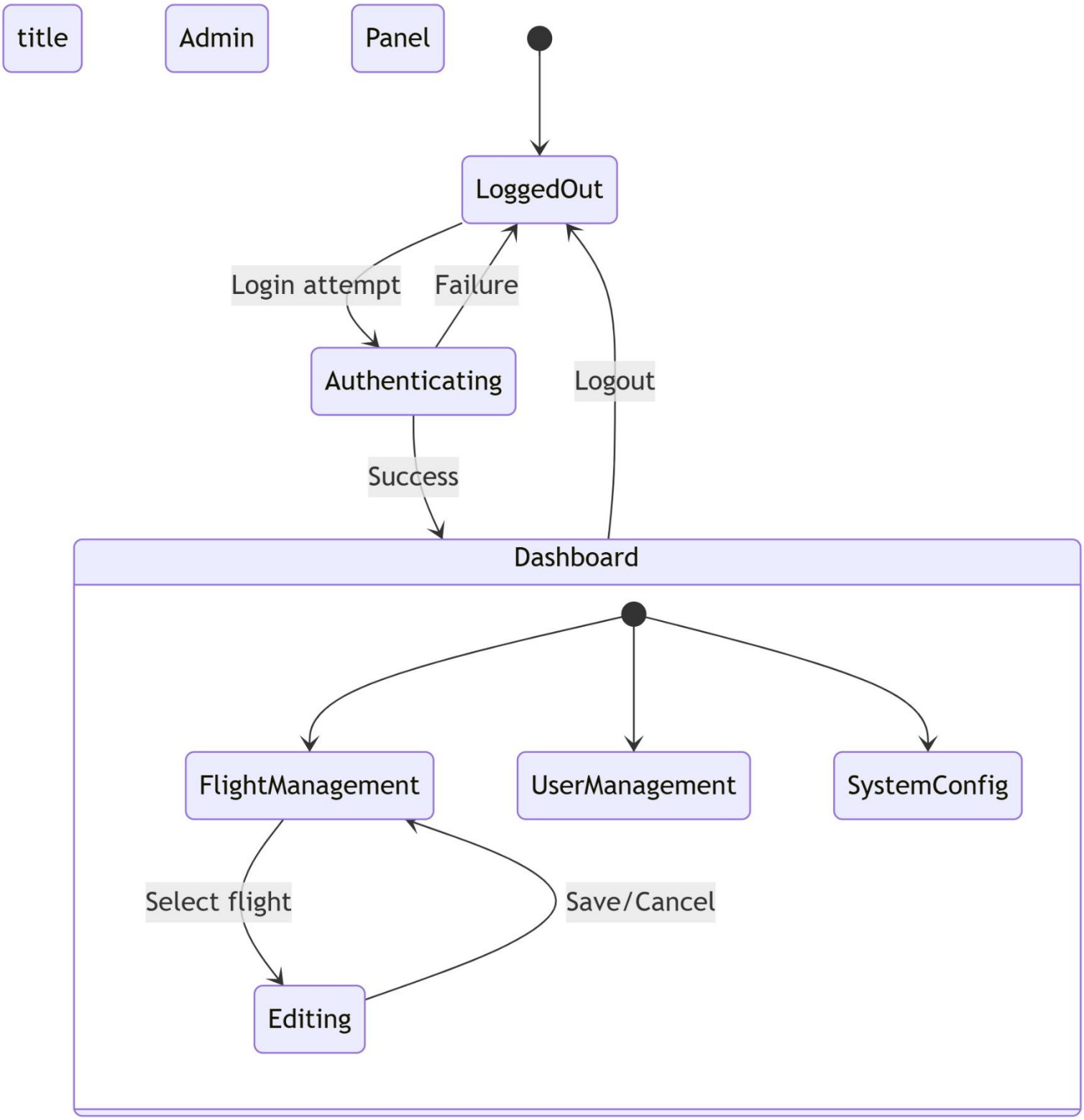
5.2.1 Sequence Diagram

5.2.1.1 Flight Booking

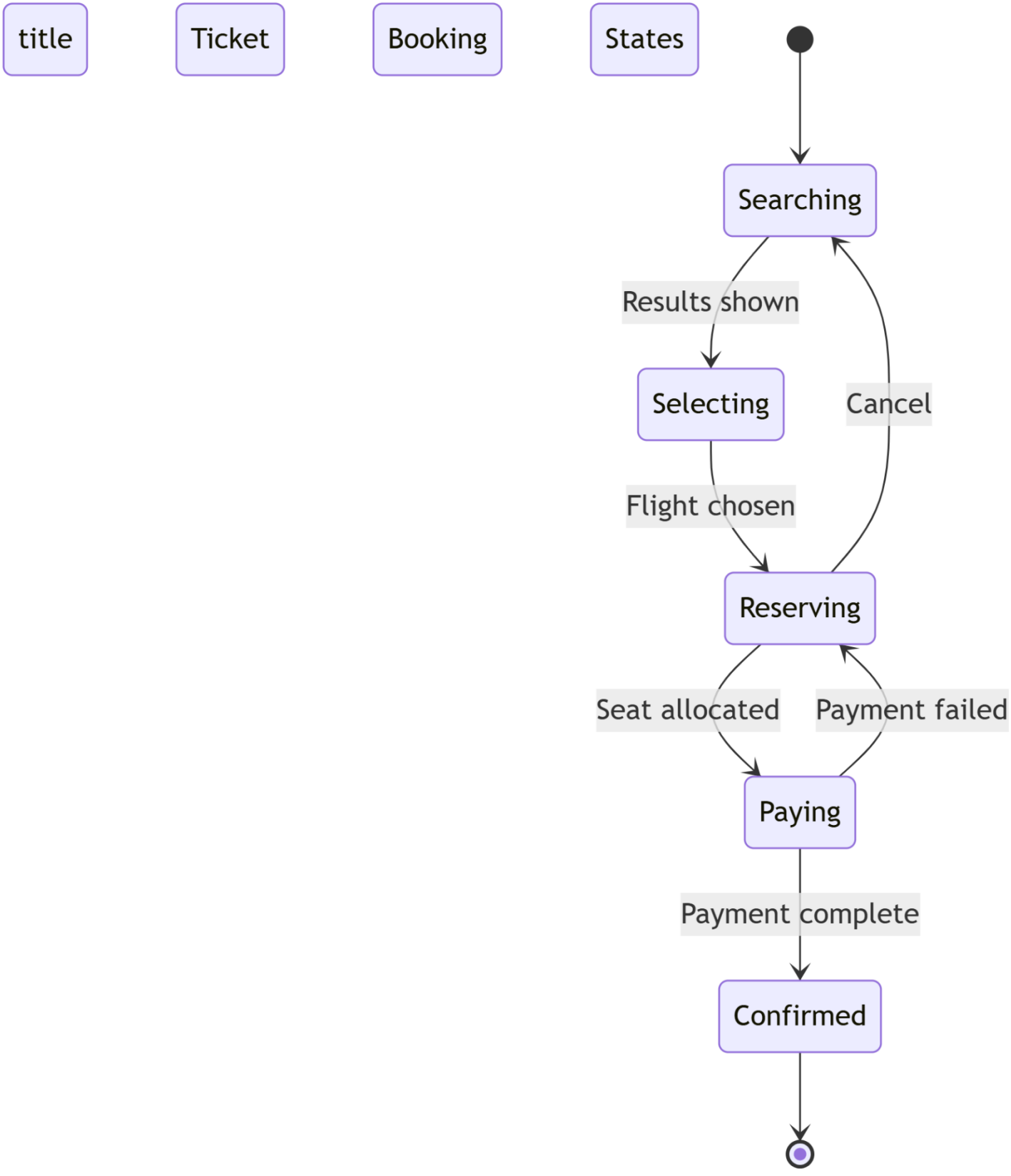


5.2.2 State Diagram

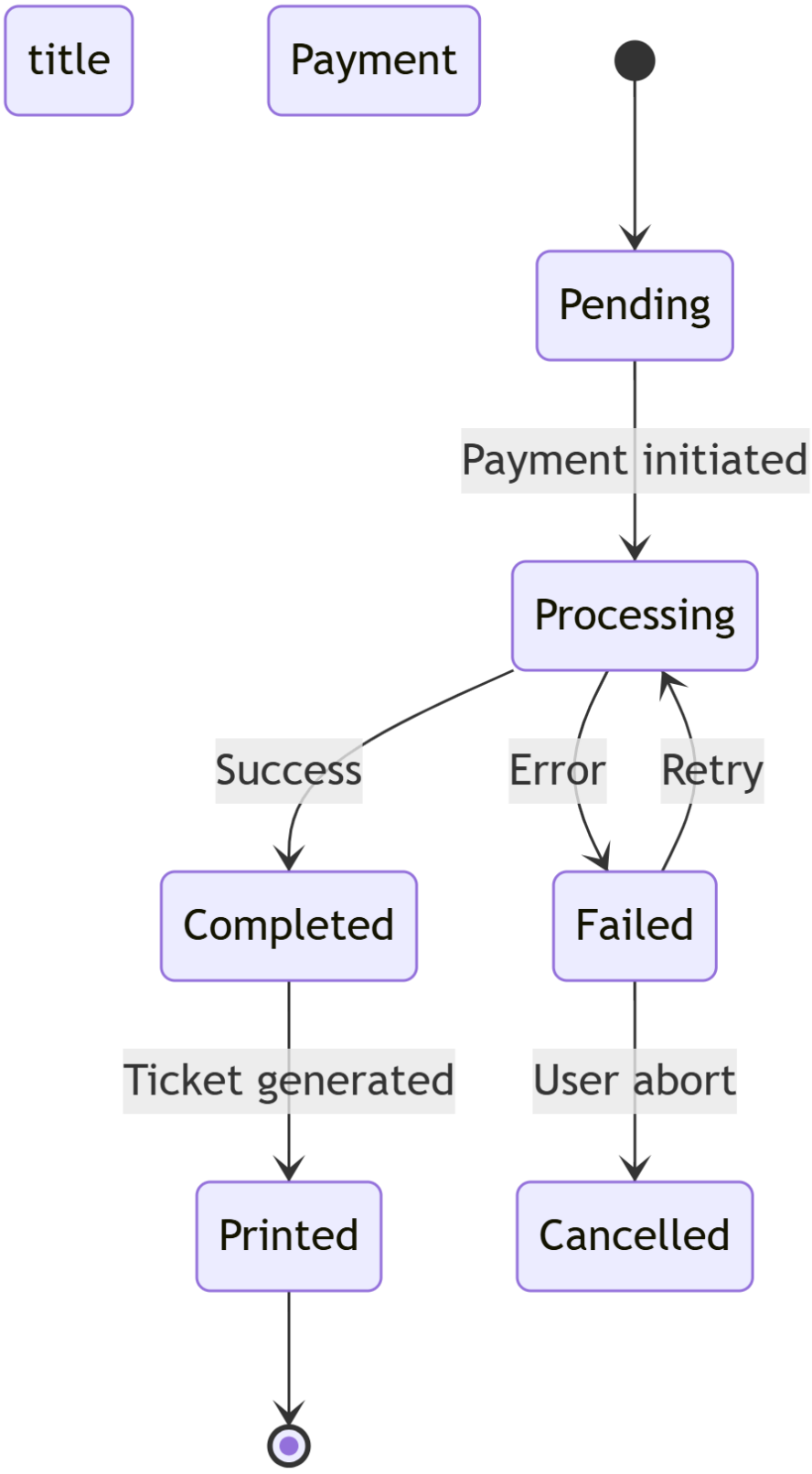
5.2.2.1 Administration



5.2.2.2 Book Ticket



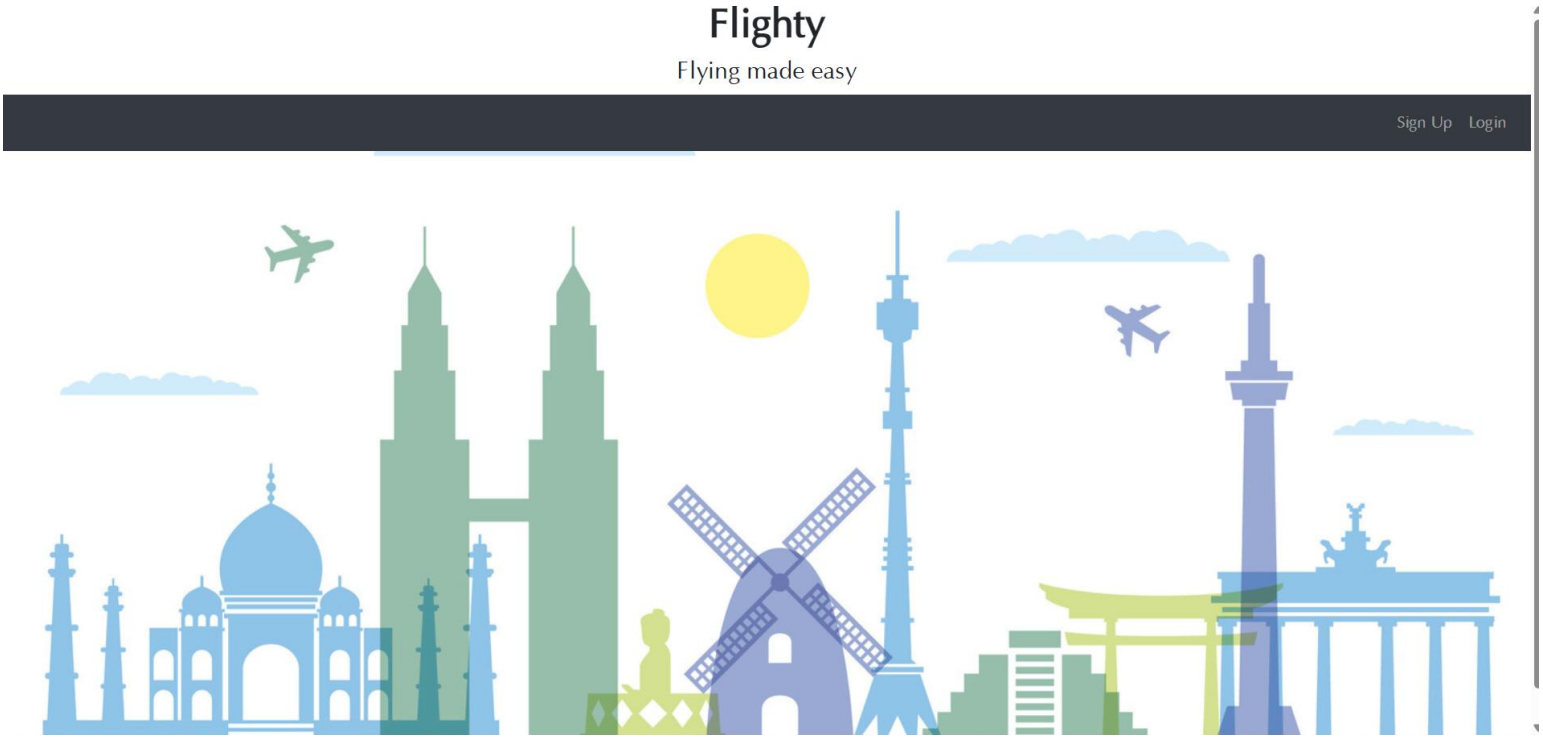
5.2.2.3 Payement and Ticket Print

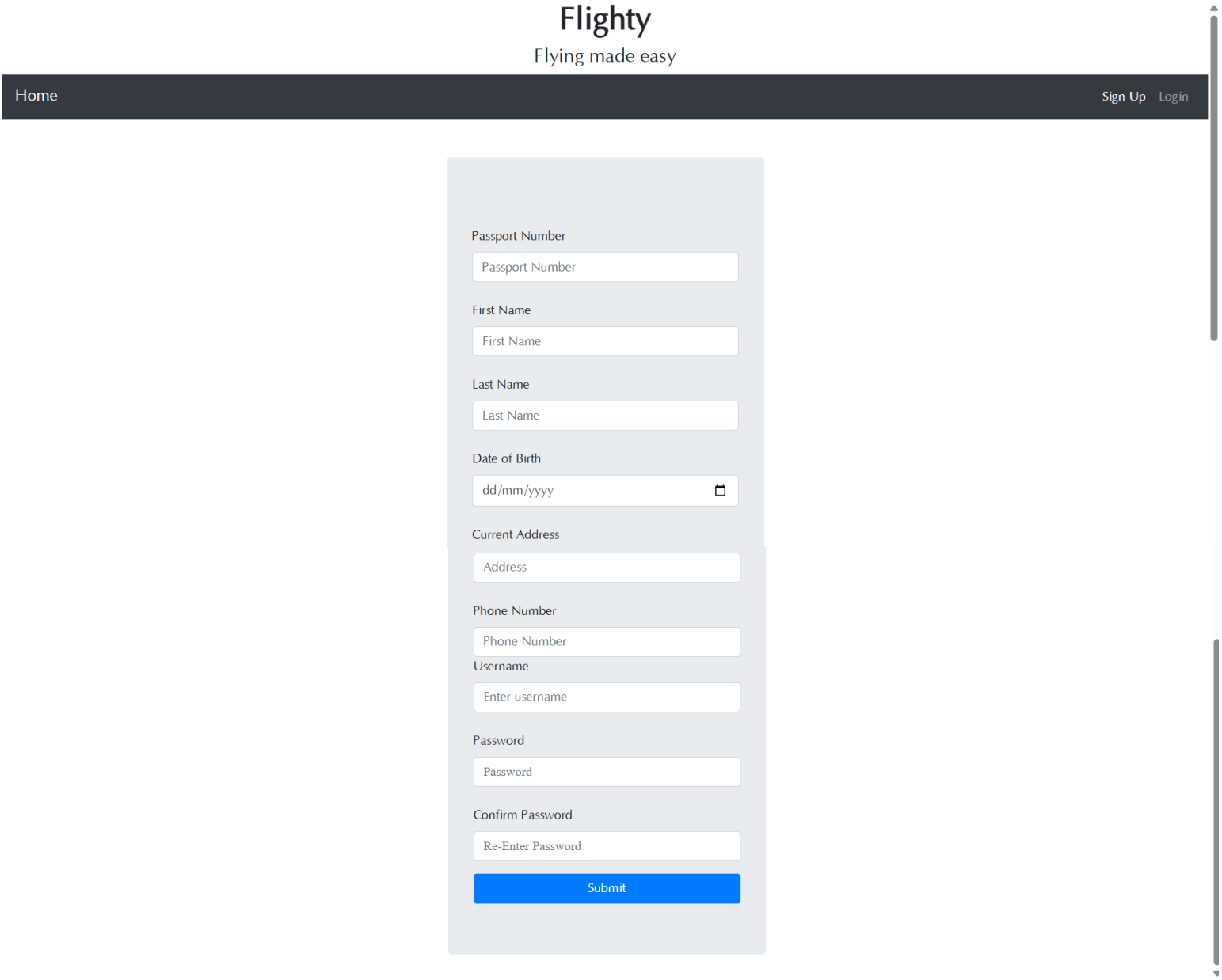


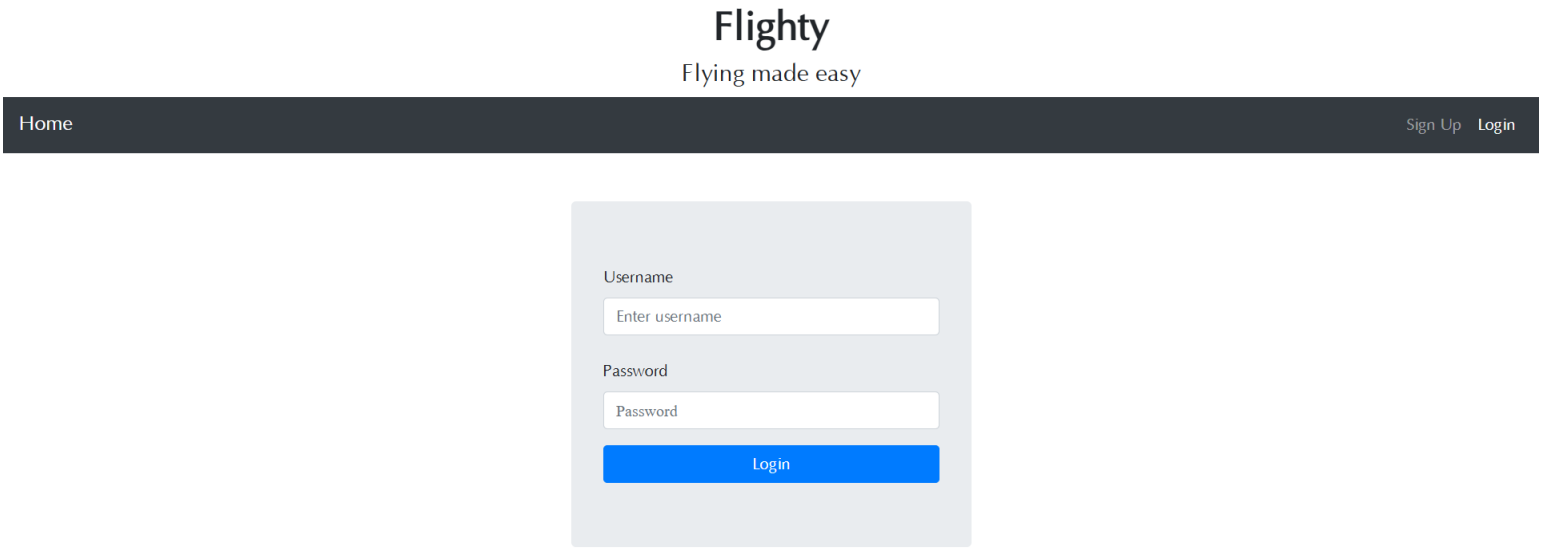
6 References:

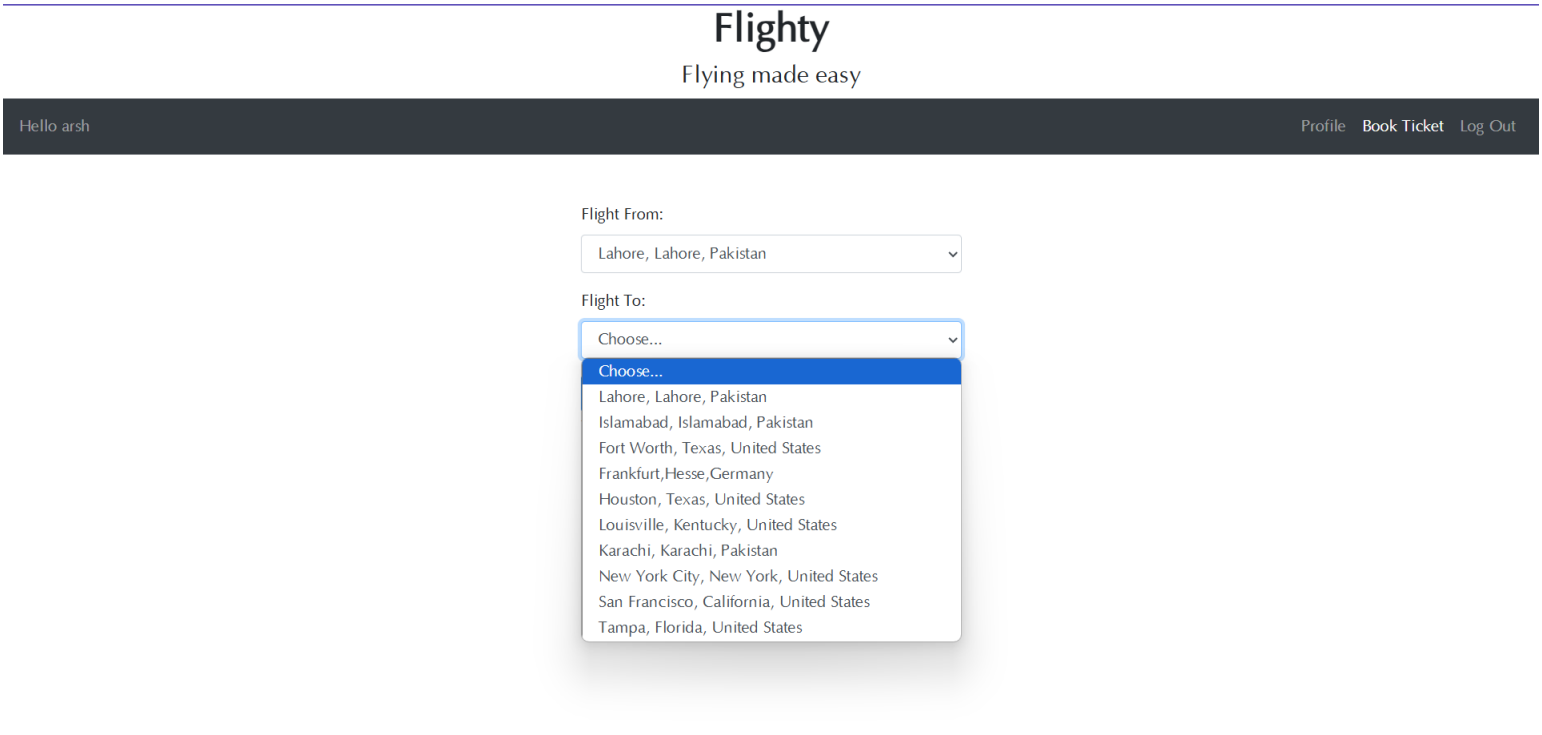
- Flask Documentation: https://flask.palletsprojects.com/
- MySQL Connector: https://dev.mysql.com/doc/connector-python/en/
- Werkzeug Security: https://werkzeug.palletsprojects.com/en/2.3.x/utils/

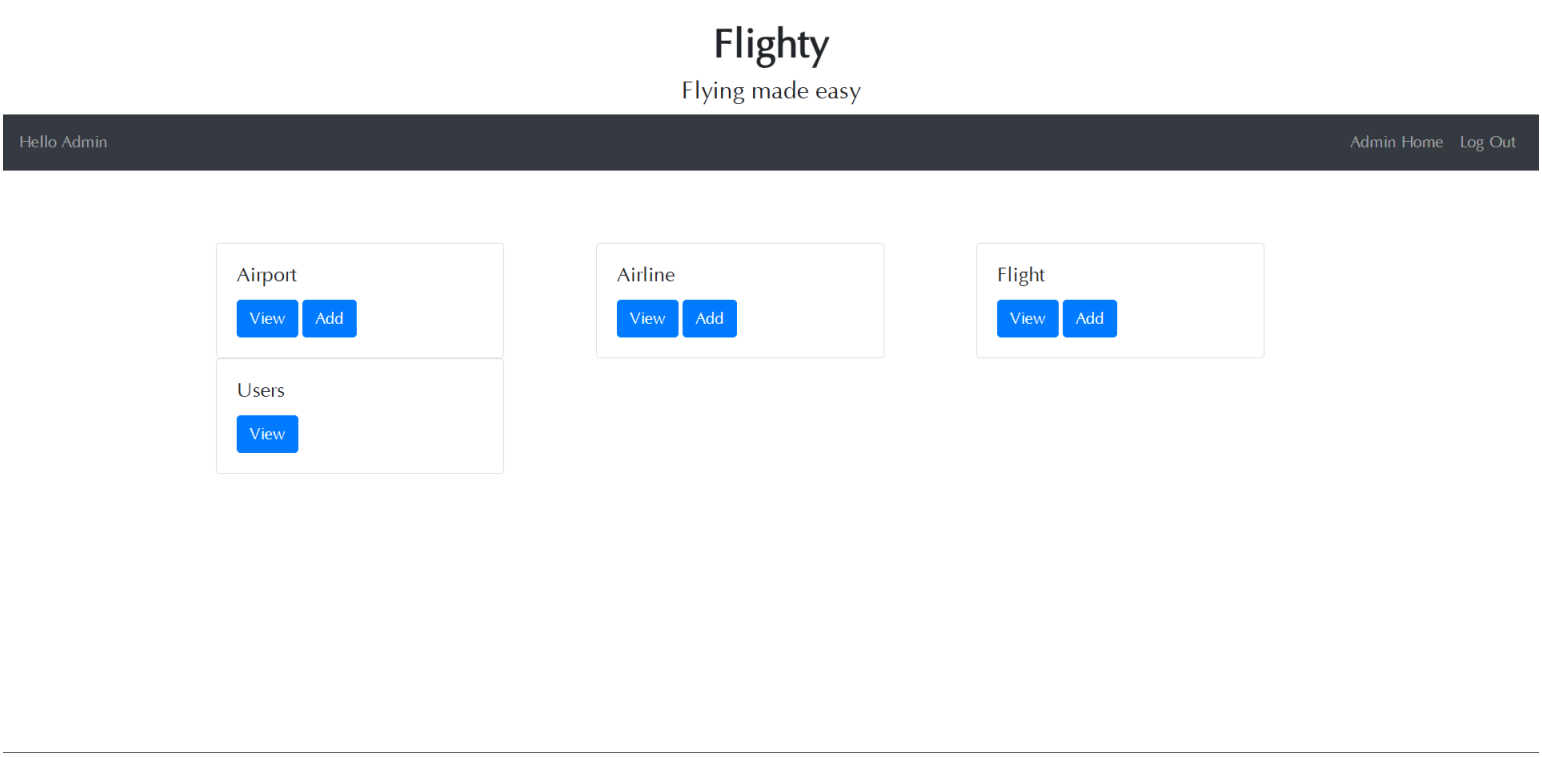
7 Appendices:




Flighty
Flying made easy
Home
Sign Up
Login
Passport Number
Passport Number
First Name
First Name
Last Name
Last Name
Date of Birth
dd/mm/yyyy
Current Address
Address
Phone Number
Phone Number
Username
Enter username
Password
Password
Confirm Password
Re-Enter Password
Submit


Flighty
Flying made easy
Home
Sign Up
Login
Username
Enter username
Password
Password
Login


Flighty
Flying made easy
Hello arsh
Profile
Book Ticket
Log Out
Flight From:
Lahore, Lahore, Pakistan
Flight To:
Choose...
Choose...
Lahore, Lahore, Pakistan
Islamabad, Islamabad, Pakistan
Fort Worth, Texas, United States
Frankfurt,Hesse,Germany
Houston, Texas, United States
Louisville, Kentucky, United States
Karachi, Karachi, Pakistan
New York City, New York, United States
San Francisco, California, United States
Tampa, Florida, United States


Flighty
Flying made easy
Hello Admin
Admin Home
Log Out
Airport
View
Add
Airline
View
Add
Flight
View
Add
Users
View