EPOKA UNIVERSITY

INTRODUCTION TO SOFTWARE ENGINEERING

[Flight Reservation System]

Team Members:

Name Surname [Alvi Hysa]

Name Surname [Denaldo Selala]

Name Surname [Arlind Alliu]

Name Surname [Alesio Rajta]

[Flight Reservation System] Requirements Specification

[1] - "User Interface: The Flight Reservation System should have a user-friendly interface that allows users to search for flights, and book tickets.

[2] - Flight Information Management: The software should allow users to view flight schedules, availability, and prices for different airlines.

[3] - Seat Selection: The software should allow users to select seats on the flight, and it should show them which seats are available and which are already booked.

[4] - Booking Management: The software should allow users to book

[5] - Payment Processing: The software should have a secure payment processing system that accepts different payment methods such as credit cards, PayPal, etc.

[6] - Flight Confirmation: The software should confirm the user's flight booking and send them an email or text message with their itinerary details.

[7] - Flight Cancellation: The software should allow users to cancel their flight bookings and get a refund, subject to the airline's cancellation policy.

[8] - Error Handling: The software should handle errors gracefully and display error messages to the user when necessary.

[9] - Security: The software should have appropriate security measures, such as user authentication and authorization, to prevent unauthorized access to sensitive information.

[1 0] - Code Quality: The software should be well-organized and documented with comments for easy maintenance and understanding. It should also follow good coding practices such as modular programming and error checking."

Table of Contents

1. Executive Summary

1.1 Project Overview

The Flight Reservation System is a software application designed to streamline the process of searching, booking, and managing flight reservations. It provides an intuitive and user-friendly interface that allows users to access flight information, select seats, make payments, receive confirmation, and manage bookings efficiently. The system incorporates various functionalities such as flight information management, seat selection, booking management, payment processing, flight confirmation, flight cancellation, error handling, security, and code quality.

The primary audience for this project are travelers who need to book flights for personal or business purposes.

1.2 Purpose and Scope of this Specification

In scope

- User Interface: The Flight Reservation System should have a user-friendly interface that allows users to search for flights, and book tickets.
- Flight Information Management: The software should allow users to view flight schedules, availability, and prices for different airlines.
- Booking Management: The software should allow users to book
- Flight Cancellation: The software should allow users to cancel their flight bookings and get a refund, subject to the airline's cancellation policy.
- Error Handling: The software should handle errors gracefully and display error messages to the user when necessary.
- Code Quality: The software should be well-organized and documented with comments for easy maintenance and understanding. It should also follow good coding practices such as modular programming and error checking.

- Flight Confirmation: The software should confirm the user's flight booking
- Security: The software should have appropriate security measures, such as user authentication and authorization, to prevent unauthorized access to sensitive information. (applies only for admin)

Out of Scope

- Flight Confirmation: send customers an email or text message with their itinerary details.
- Seat Selection: The software should allow users to select seats on the flight, and it should show them which seats are available and which are already booked.
- Payment Processing: The software should have a secure payment processing system that accepts different payment methods such as credit cards, PayPal, etc.
- Security: The software should have appropriate security measures, such as user authentication and authorization, to prevent unauthorized access to sensitive information.

2. Product/Service Description

This Product/Service provides:

User Interface:

The system features a user-friendly interface that enables users to search for flights, view flight schedules, check availability, and prices across different airlines. It provides an intuitive and easy-to-navigate design to enhance the user experience.

Flight Information Management:

The software allows users to access comprehensive flight information, including schedules, availability, and pricing details from multiple airlines. It ensures that users can make informed decisions regarding their travel plans.

Booking Management:

The software facilitates the booking process by allowing users to select flights and make payments for their tickets. It streamlines the reservation process and ensures a smooth and efficient booking experience.

Payment Processing:

To ensure secure transactions, the system incorporates a reliable payment processing system. It accepts various payment methods such as credit cards, PayPal, and other relevant options, providing flexibility for users to complete their ticket purchases.

Flight Confirmation:

Upon successful booking, the software confirms the user's flight reservation. It sends an email or text message to the user containing their itinerary details, including flight details, seat assignment, and other relevant information.

Flight Cancellation:

In case users need to cancel their flight bookings, the system allows them to do so. It follows the airline's cancellation policy and initiates the refund process if applicable, providing a convenient cancellation experience for users.

2.1 Product Context

The flight management system being developed is designed to be independent and self-contained, operating without relying on external dependencies or third-party software. While it may be a standalone system, it does have the ability to interface with various related systems to fulfill specific requirements. These systems could include aircraft avionics for flight control, communication, and navigation; air traffic control for adherence to regulations and guidance; weather data providers for real-time weather updates; flight databases or reservation systems for flight information and status updates; and maintenance systems for aircraft maintenance and updates.

2.2 User Characteristics

Administrator:

- Add Flight
- Remove Flights
- Print Flights
- Add Airport
- Remove Airport
- Print Airport
- Quit

Costumer:

- Search Flights
- Search Flights By From Airport
- Search Flights by ID
- Book Flights
- Cancel Booking
- My Booking
- Quit

2.3 Assumptions

The operator is assumed to be trained and not malicious.

2.4 Constraints

The main constraint is that the database is linear and transactional by default. What this means is that every operation is stored as a new line, and no previous operations are deleted. This is helpful because it leaves an audit trail of every action performed, however it is not good for horizontal scaling or parallel computing.

2.5 Dependencies

- No dependencies outside of C standard library.

3. Requirements

- CLEAN and Scalable code. The code should be written in a CLEAN manner such that adding new additional features is simple and straightforward. Code duplication must be kept to a minimum.
- Ease of use. Every user input should have clear labels that do not leave room for interpretation.
- Graceful error handling. If the user provides invalid input such as a number when a string is required or vice versa, the program should handle the errors gracefully.
- Transactional database. Meaning that each operation is stored separately (including delete operations) to protect against accidental deletions, updates ECT.

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
BR_LR_05	The Flight Reservation System should have a user-friendly interface that allows users to search for flights, select seats, and book tickets.	User Interface			
BR_LR_08	The software should allow users to view flight schedules, availability, and prices for different airlines.	Flight Information Management			

BR_LR _10	The software should allow users to select seats on the flight, and it should show them which seats are available and which are already booked.	Seat Selection			
BR_LR _16	The software should allow users to book flights and make payments for their tickets.	Booking Management			
	The software should have a secure payment processing system that accepts different payment methods such as credit cards, PayPal, etc.	Payment Processing			
	The software should confirm the user's flight booking and send them an email or text message with their itinerary details.	Flight confirmation			
	The software should allow users to cancel their flight bookings and get a refund, subject to the airline's cancellation policy.	Flight Cancellation			
	The software should handle errors gracefully and display error messages to the user when necessary.	Error Handling			
	The software should have appropriate security measures, such as user authentication and authorization, to prevent unauthorized access to sensitive information.	Security			

	The software should be well-organized and documented with comments for easy maintenance and understanding. It should also follow good coding practices such as modular programming and error checking.	Code Quality:			

3.2 Non-Functional Requirements

- Product Requirements:
  - The application should be reliable.
  - The application should be clear and concise
  - The application should not crash
- Organizational Requirements
  - Each pull request should be tested by quality control.
  - Each programmer must name their branches after themselves.
- External Requirements
  - The program should be able to interact with other programs via APIs.

3.2.1 User Interface Requirements

The user interface shall be a TUI (terminal user interface) with added terminal colors for added clarity and menus that only show operations based on the specific role.

3.2.2 Manageability/Maintainability

3.2.1.1 Monitoring

When compiled the DEBUG macro extra logs and monitoring info is logged to stdout.

3.2.1.2 Maintenance

Modular code design which allows simple maintenance.

3.2.1.3 Operations

Easy database backups and audits. (Due to the database infrastructure)

3.2.3 Security

3.2.3.1 Protection

- The administrator requires a password for the purpose of authentication.
- There are currently no implemented security measures in place for customers.

3.2.4 Standards Compliance

Highly scalable codebase developed using a component-based architecture, characterized by its modular design and well-defined boundaries. The system demonstrates excellent vertical and horizontal scalability, making it effortless to increase both its computational capacity and the number of instances deployed.

4. Design thinking methodologies +

Provide all the Used Design Muscles in Software product

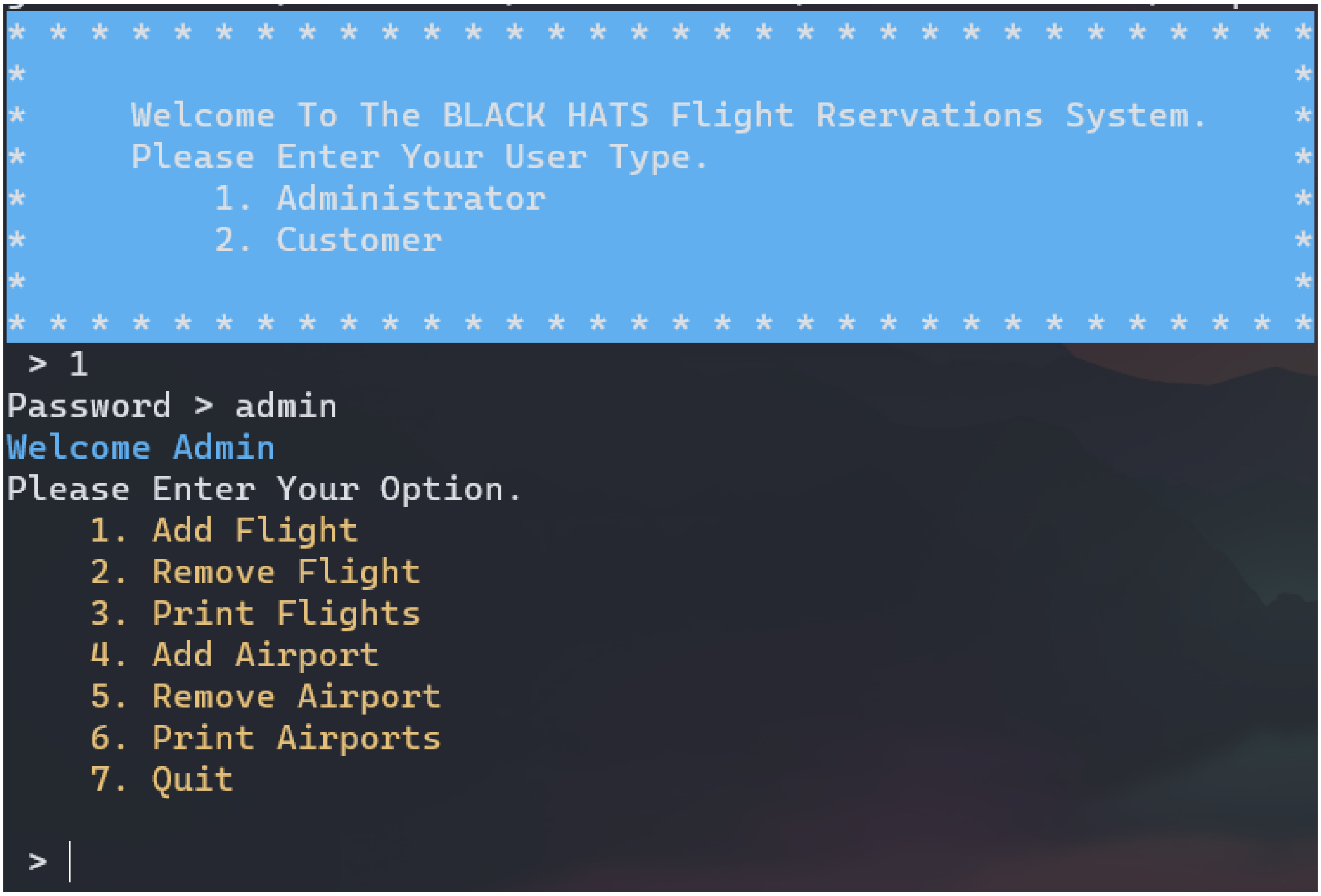
4.1 Empathy: *We first looked at other flight management systems and saw how they were approaching problems.*

4.2 Define: *We found that most of them were very complicated when the user wanted to just simply make a reservation.*

4.3 Ideate: *Our solution was to make the menus as simple as possible and not over-bloat the program with useless features.*

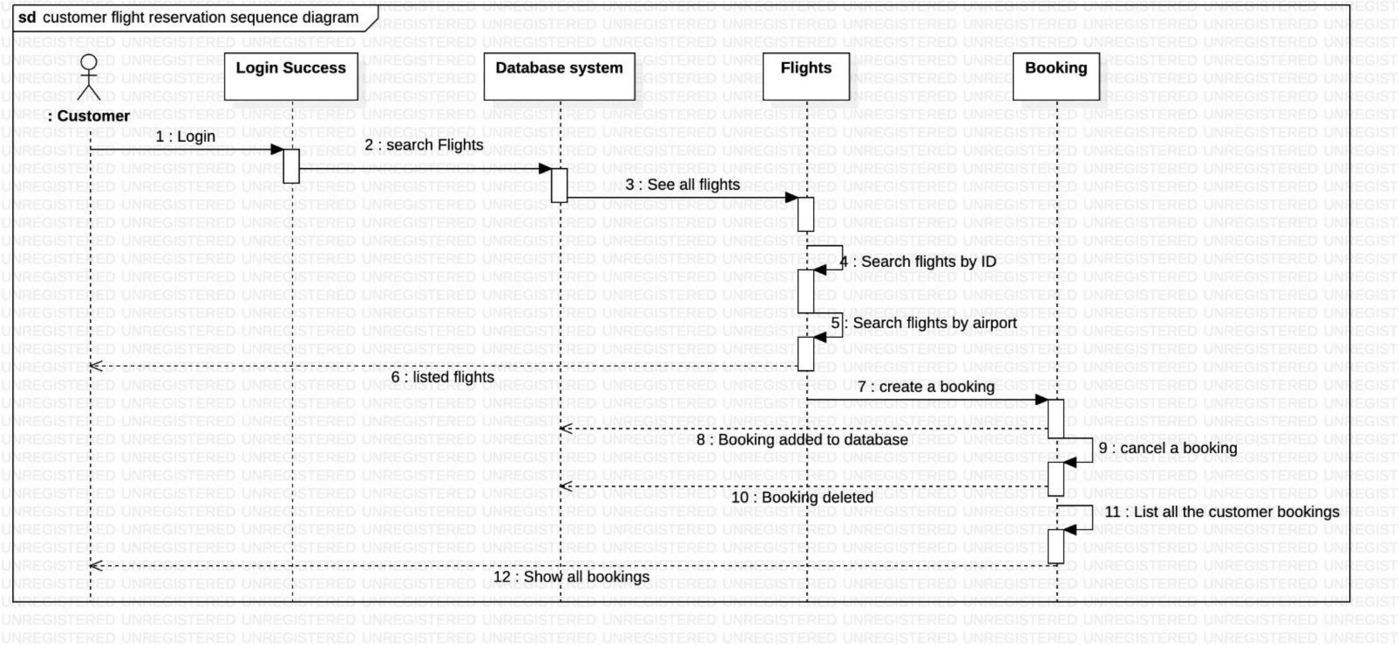
4.4 Test: *After each new feature we tested it thoroughly with a fresh pair of eyes to make sure the UX was acceptable and the code was not buggy.*

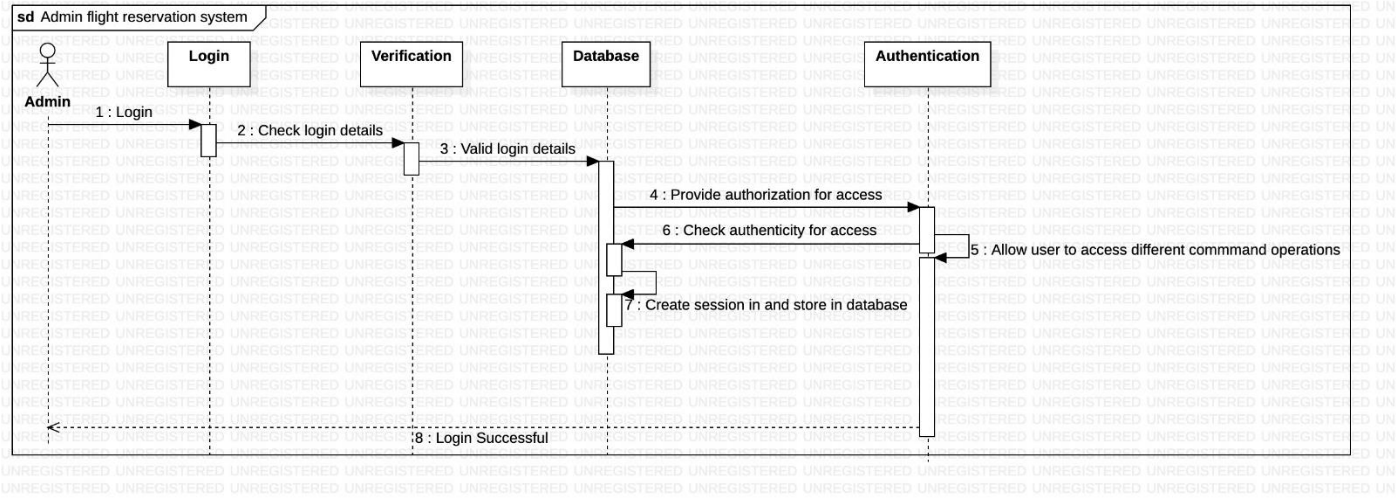
4.5 GUI (Screenshots)



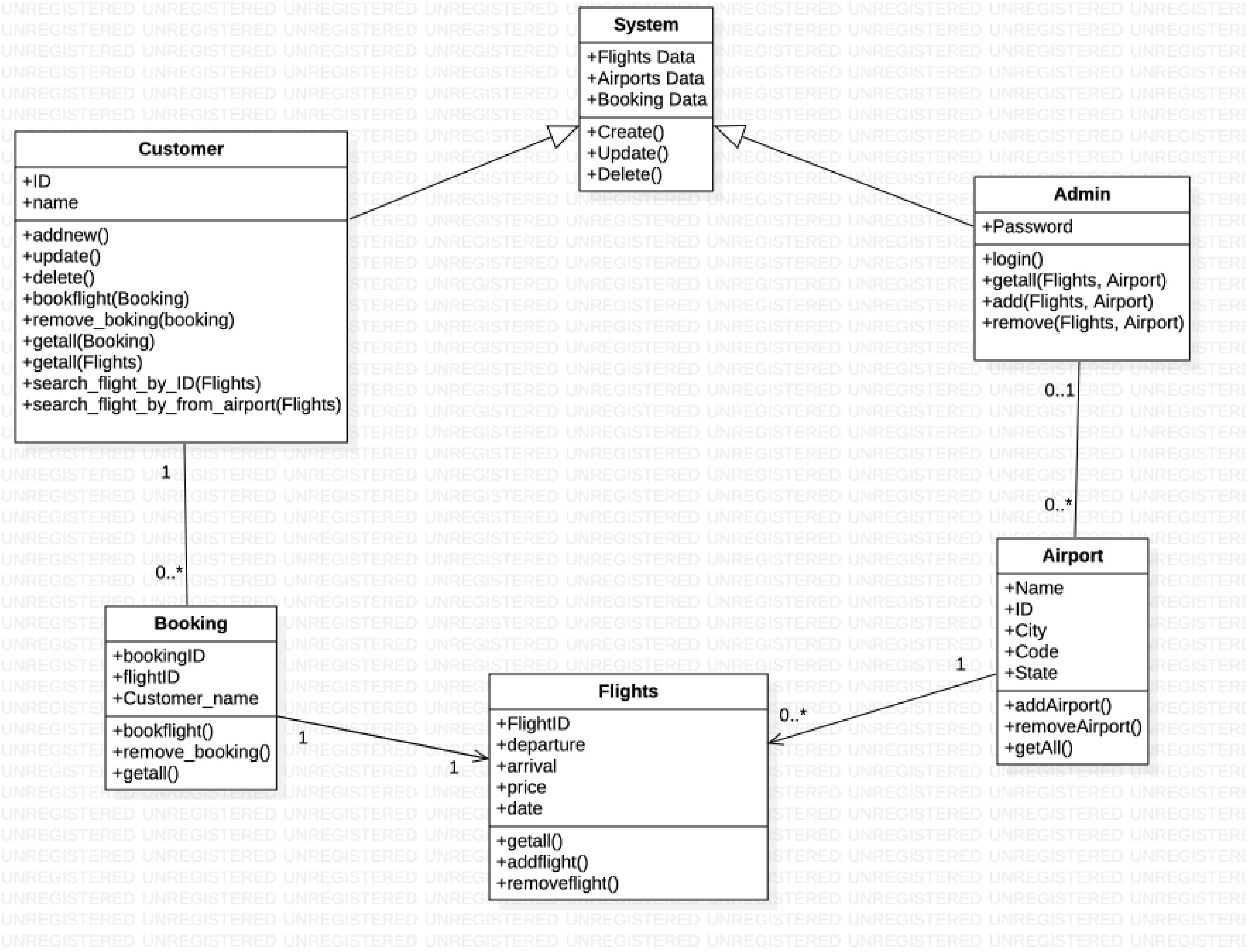
```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                             *
*     Welcome To The BLACK HATS Flight Rservations System.    *
*     Please Enter Your User Type.                            *
*         1. Administrator                                    *
*         2. Customer                                         *
*                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 > 1
Password > admin
Welcome Admin
Please Enter Your Option.
    1. Add Flight
    2. Remove Flight
    3. Print Flights
    4. Add Airport
    5. Remove Airport
    6. Print Airports
    7. Quit

 > 
```

5. Software Design




sd Admin flight reservation system
Admin
Login
Verification
Database
Authentication
1 : Login
2 : Check login details
3 : Valid login details
4 : Provide authorization for access
5 : Allow user to access different commmand operations
6 : Check authenticity for access
7 : Create session in and store in database
8 : Login Successful

[YourProject] Requirements


System
+Flights Data
+Airports Data
+Booking Data
+Create()
+Update()
+Delete()
Customer
+ID
+name
+addnew()
+update()
+delete()
+bookflight(Booking)
+remove_boking(booking)
+getall(Booking)
+getall(Flights)
+search_flight_by_ID(Flights)
+search_flight_by_from_airport(Flights)
Admin
+Password
+login()
+getall(Flights, Airport)
+add(Flights, Airport)
+remove(Flights, Airport)
0..1
1
0..*
0..*
Airport
+Name
+ID
+City
+Code
+State
+addAirport()
+removeAirport()
+getAll()
1
Booking
+bookingID
+flightID
+Customer_name
+bookflight()
+remove_booking()
+getall()
1
1
Flights
+FlightID
+departure
+arrival
+price
+date
+getall()
+addflight()
+removeflight()
0..*