Software Requirement Specification (SRS)

Project Title: Airline Reservation System

Developer's Info:

ID	Name
0322320105101009	Md Ashiqur Rahman
0322320105101023	Pollob Kumar
0322320105101048	Md Tanzid Mondol

1. Introduction

1.1 Purpose and Scope:

Purpose:
The Airline Reservation System is designed to simplify the flight ticket booking and management process. It is intended for airline staff and administrators to efficiently manage flight schedules, seat reservations, and customer data through a CLI-based interface.

Scope:
This project provides a terminal-based system to manage:

- Flight information
- User (Passenger) registration
- Booking and cancelation
- Ticket management
- Admin functions for adding/updating flights

This system does not support graphical UI. It is lightweight and designed for educational and small-scale operational purposes.

1.2 Definitions, Acronyms, and Abbreviations:

Term	**Definition**
CLI	Command Line Interface
SRC	Software Requirements Specification
SQL	Structured Query Language
DBMS	Database Management System (MySQL used here)
GCC	GNU Compiler Collection (used for compiling C programs)
Admin	Person with access to manage flights and booking data

2. General Description

2.1 Project Perspective:

This system is a standalone CLI-based application that interacts with a MySQL database. It does not require third-party services or APIs. It fits into the local computing environment, ideal for small airline systems or academic purposes.

2.2 Project Functions:

- Admin login/logout
- Add/update/delete flights
- View all available flights
- Search flights by source/destination/date
- Passenger registration
- Booking tickets
- Canceling bookings
- Generating booking receipts

2.3 User Classes and Characteristics:

User Class	Description
Admin	Manages flights, views bookings, edits data
Passenger	Books, cancels tickets, and views reservations

2.4 Operating Environment:

- OS: Windows/Linux (with terminal support)
- Language: C (compiled using GCC)
- Database: MySQL 8.x
- RAM: Minimum 2 GB
- Storage: Minimum 100 MB for records
- Connectivity: Localhost connection to MySQL server

2.5 Design and Implementation Constraints:

- CLI-only interface (no GUI)
- Fixed schema for flight and user tables
- Passwords stored as plain text or basic hashing
- No concurrent booking control

2.6 Assumptions and Dependencies:

- MySQL server must be running locally
- Users will have basic terminal knowledge
- Only one user will access the system at a time (no concurrency control)

3. Study of Existing Work and Comparison with Our Projects

Projects Name	Used Technology	Functional Features	Limitation	Our Contribution
AirBooker	Python, Flask, SQLite	Web-based flight search, user login, booking history	High memory usage, web server dependency	Lightweight CLI version, more suitable for low-end or academic

				systems
FlightCLI	Java, JDBC, MySQL	CLI admin module, add/update flights, basic ticket booking	No user registration, no cancelation option	Enhanced with user management, cancelation system, and structured database
MyFlightApp	C#, .NET, SQL Server	GUI for booking management, encrypted login, flight dashboard	Windows-only, .NET dependency, heavy resource usage	Cross-platform CLI in C, minimal resources needed
AirlineLite	C++, File I/O	Basic reservation with flat file storage	No database, risk of data corruption, poor scalability	Switched to MySQL backend, scalable, secure, persistent
SkyReserve	PHP, MySQL, HTML/CSS	Online booking form, dynamic search, confirmation via email	Requires hosting, not offline operable	Offline CLI tool for terminal-based management
FlightEase	JavaScript (Node.js), MongoDB	RESTful APIs, JWT authentication, multi-user environment	No frontend, not CLI-friendly	Pure CLI-based interface, direct control, simpler design for academic use
TerminalAir	C++, MySQL	CLI interface for seat booking and flight display	Poor error handling, weak input validation	Added strong validation, secure DB transactions, and better exception handling
BasicAirRes	Python, CSV File Storage	Booking via command line, seat assignment, data stored in CSV	No database support, no concurrency, vulnerable to data loss	Upgraded to structured database with MySQL, ensured persistent reliable data
FlightDeck	Kotlin, Firebase	Android app with real-time database sync, booking notifications	Mobile-only, cloud dependency, limited offline features	CLI system with full offline capability, no third-party cloud dependencies

4. Specific Requirements

4.1 Functional Requirements:

- [FR1] System shall allow admin to login with credentials.
- [FR2] Admin shall be able to add/update/delete flight schedules.
- [FR3] Passengers shall be able to register using name, age, contact info.
- [FR4] Passengers shall be able to search for flights.
- [FR5] Booking shall reduce the number of available seats.
- [FR6] Cancellations shall increase available seats.
- [FR7] Booked tickets shall be saved in the database.
- [FR8] System shall print booking receipt on confirmation.

4.2 Non-Functional Requirements:

- [NFR1] System should respond within 2 seconds for any user query.
- [NFR2] Must be operable in low-resource environments.
- [NFR3] Passwords should be minimally encrypted.
- [NFR4] System should not crash on invalid input.

4.3 External Interface Requirements:

- **MySQL Database:** For all persistent storage
- **Terminal Input/Output:** User input and display
- **C MySQL Connector:** To interact with MySQL database

4.4 Data Requirements:

- Flight table: Flight_ID, source, destination, date, time, seat_capacity, fare
- User table: User_ID, name, contact, password
- Booking table: Booking_ID, User_ID, Flight_ID, Seat_no, Status

5. System Architecture

User (CLI Terminal) → C Program Logic(Validation, Flow Control) → MySQL Connector (C API) → MySQL Database(Flights, Users, Bookings)

6. System Features

Admin Login System

Provides secure access for airline staff to manage flight-related operations using predefined credentials.

Add Flight Schedule

Allows admin users to add new flight details such as flight number, source, destination, date, time, fare, and seat capacity.

Update/Delete Flights

Enables modification or removal of existing flight schedules for operational flexibility.

Passenger Registration

New users (passengers) can register by providing personal details like name, age, and contact number. This information is stored securely in the MySQL database.

Flight Search

Passengers can search for available flights based on source, destination, and travel date to find suitable options.

Ticket Booking

Registered passengers can book available seats on a selected flight. Booking details are stored in the database and the available seat count is updated in real-time.

Cancel Reservation

Passengers can cancel previously booked tickets. This action will mark the booking as canceled and free up the corresponding seat.

View Booking History

Passengers and admins can view the history of bookings for a particular user or flight, useful for tracking and reporting.

Receipt Generation

After booking confirmation, the system generates a booking receipt containing ticket details like flight number, passenger name, seat number, and fare.

Data Persistence via MySQL

All flight, user, and booking data are stored and managed through a MySQL database, ensuring consistency, reliability, and data persistence.

Input Validation & Error Handling
Includes checks for invalid inputs (e.g., invalid flight ID, date format) and appropriate error messages for better user experience.

Command Line Interface (CLI)
Entire system operates through the command line, making it lightweight and suitable for systems with limited resources.

7. Testing and Acceptance

7.1 Testing Requirements:

- Unit testing for all input fields (e.g., dates, IDs)
- Integration testing for DB queries
- Boundary testing for seat limits and invalid inputs
- Manual testing for end-to-end booking flow

7.2 Acceptance Criteria:

- All core features (login, booking, canceling, data persistence) must work.
- Database must update accurately with each transaction.
- System must handle invalid inputs gracefully.

8. Other Supporting Information

8.1 Project Timeline:

Week	Date	Milestone
Week 1	2 Aug to 8 Aug 2025	Requirements gathering
Week 2	9 Aug to 15 Aug 2025	Database schema and setup
Week 3	16 Aug to 22 Aug 2025	C-MySQL connection & Admin module
Week 4	23 Aug to 29 Aug 2025	Passenger registration + booking
Week 5	30 Aug to 5 Sep 2025	Cancelation + receipt
Week 6	6 Sep to 12 Sep 2025	Testing and refinement
Week 7	13 Sep to 19 Sep 2025	Final documentation and submission

8.2 Stakeholder List:

Role	Name
Developers	Ashiqur, Pollob, Tanzid
Supervisor	Mst. Sahela Rahman
Users	Airline staff and admin

6.3 User Documentation:

- Start terminal
- Run compiled binary: `./airline_reservation`
- Choose admin or passenger mode
- Follow on-screen prompts for each operation
- Use credentials set during registration

9. References - Citations

[1] MdRubelRana. (n.d.). GitHub - MdRubelRana/Airline-Ticket-Booking-System: Airline Ticket Booking System. This Project is developed in C programming language. GitHub. https://github.com/MdRubelRana/Airline-Ticket-Booking-System/ [Accessed: July 25, 2025].

[2] AgyeyaMishra. (n.d.). GitHub - AgyeyaMishra/airline-reservation-system: A basic implementation of Airline Reservation System in C language using the concept of Structures and Data File Handling. GitHub. https://github.com/AgyeyaMishra/airline-reservation-system/ [Accessed: July 25, 2025].

[3] Airline booking project using C Language - StudyTonight. (n.d.). https://www.studytonight.com/c-projects/airline-booking-project-using-c-language [Accessed: July 25, 2025].

[4] Codeprojects. (2024, June 29). Airline booking in C with source code. Source Code & Projects. https://code-projects.org/airline-booking-in-c-with-source-code/ [Accessed: July 25, 2025].

[5] Amlendra. (2021c, September 11). Airline Ticket Booking System Project in C with

Source code.
Aticleworld.https://aticleworld.com/airline-ticket-booking-system-project-in-c/ [Accessed: July 25, 2025].

[6] Yunyeng. (n.d.). Algorithms/airline.c at master · yunyeng/Algorithms. GitHub. https://github.com/yunyeng/Algorithms/blob/master/airline.c/ [Accessed: July 25, 2025].

[7] Pragnyashriprakhya. (n.d.). AIRLINE RESERVATION SYSTEM (1). Scribd. https://www.scribd.com/document/682905951/AIRLINE-RESERVATION-SYSTEM-1 [Accessed: July 25, 2025].

[8] Mehulbhardwaj. (n.d.). GitHub
mehulbhardwaj1609/AIRLINE-RESERVATION-C-CODE-Open-Ended-Project-GitHub.ht
tps://github.com/mehulbhardwaj1609/AIRLINE-RESERVATION-C-CODE-Open-Ended-Project- [Accessed: July 25, 2025].

Prepared By:

Pollob Kumar

Md Tanzid Mondol

Md Ashiqur Rahman

Supervisor Name: Mst.Sahela Rahman

Department: Computer Science and Engineering (cse)

Institution: Pundra University of Science & Technology.Gokul, Bogura.