
Software Requirements Specification for Airline Management System(AMS)

v1.0

Prepared by

Srikantam Sai Srinivas

Amballa Saketh

Jadav Uday Kiran

Gullolla Vamshinath

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Project Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Features	2
2.2.1 Ticket Booking	2
2.2.2 Cancelling Ticket	2
2.2.3 Wallet	3
2.2.4 Query Information of Shows	3
2.2.5 Balance Sheet	3
2.3 User Classes and Characteristics	3
1) Show Manager	3
2) Account Clerk	3
3) Sales Persons	3
4) Spectators	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	4
2.7 Assumptions and Dependencies	5
3. System Features	5
3.1 Ticket Booking	5
3.2 Cancelling Ticket	5
3.3 Wallet Management	6
3.4 Show Information Query	6
3.5 Balance Sheet Generation	7
3.6 Expenditure Management	7
4. External Interface Requirements	8
4.1 User Interfaces	8
4.1.1 Login Interface	8
4.1.2 Dashboard Interface	8
4.1.3 Ticket Booking Interface	8
4.1.4 Wallet Management Interface	9
4.1.5 Show Information Query Interface	9
4.1.6 Balance Sheet Generation Interface	9
4.1.7 Bookings Overview	10

4.1.8 Ticket Details

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

6. Other Requirements

10

1

0

1

0

1

0

1

1

11

1

1

1

1

1

Revision History

Name	Date	Reason For Changes	Version
SAMS v1.0	April 2, 2024	Initial version of the document	v1.0
			1

1. Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for the Airline Management System (AMS) version 1.0. It outlines the functionalities required for managing airline operations, including scheduling flights, seat reservations, cancellations with penalties, fare management, and financial reporting.

1.2 Document Conventions

CHMS: Abbreviation for Concert Hall Management System

AC: Abbreviation for Account Coordinator

CM: Abbreviation for Concert Manager

SP: Abbreviation for Sales Personnel

1.3 Intended Audience and Reading Suggestions

This document caters to various stakeholders including developers, project managers, testers, and users.

- *Developers: Refer to Sections 2 (Overview) and 3 (Detailed Requirements) for comprehensive understanding of system functionalities and technical specifications.*
- *Project Managers: Utilize Sections 2 and 3 with an overview of Section 1 (Introduction) to gain context on project scope, timelines, and resource allocation.*
- *Testers: Focus on Section 3 for functional requirements and Section 4 for non-functional requirements related to performance or usability.*
- *Users (Concert Managers, Spectators, Sales Personnel, Account Coordinators): Refer to Section 2 for an overview of each user role and respective functionalities.*

1.4 Project Scope

CHMS endeavors to optimize concert management in the university concert hall by automating tasks, providing online ticketing and cancellations, implementing user roles and access control, facilitating basic financial reporting, and managing concert expenses. The primary objective of this software is to enhance operational efficiency, ensure financial transparency, and focus on core functionalities such as concert scheduling, ticketing, user roles, financial reporting, and expense management.

1.5 References

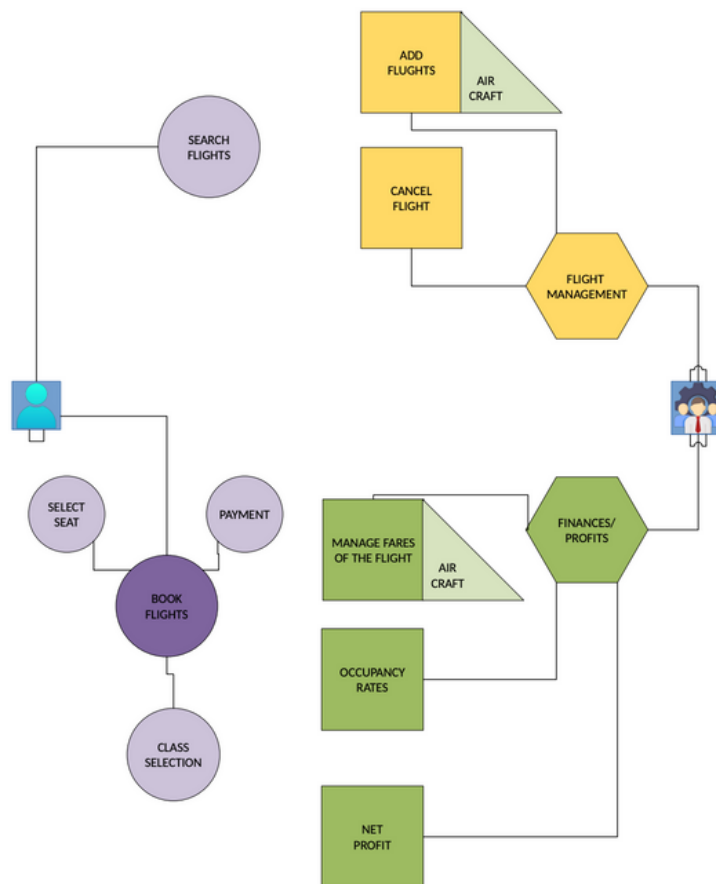
- *Software Engineering Lecture on System Analysis and Design (NPTEL course by Partha Pratim Das)*
- *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.*

2. Overall Description

2.1 Product Perspective

The SAMS is a new system to replace the current manual process of ticket booking for a show and generation of tickets. The context Diagram illustrates the external entities and interface of the system for this version. The system is expected to evolve over several releases.

2.2 Product Features



The set of features that are performed by the software are as follows:

2.2.1 Ticket Booking

2.2.1 Ticket Booking Input: Show details, seat type and sales ID **Process:** The ticket is booked and generated if a seat is available in that show. Tickets can be either booked by a salesperson or under a sales person by a spectator.

2.2.2 CancellingTicket

Input: Transaction ID of the ticket to be cancelled. Process: The ticket with that transaction ID is looked for in the database and is cancelled and the refund amount is credited as per the initial terms set. The ticket information of that booking is removed from the database and the refund transaction is added to the balance sheet.

2.2.3 Search Flight

This allows users to query available flights based on criteria like departure and arrival locations and travel dates. It retrieves and displays relevant flight information, including schedules and ticket prices, enabling users to select suitable travel options efficiently within the system.

2.2.4 Financial Reporting

Reporting Input:

Input: Revenue and expense data, transaction records, and financial parameters. Process: The system generates various financial reports, including balance sheets, to provide insights into the airline's financial performance. Show Managers can query balance sheets annually or by specific shows to analyze revenue generated and expenses incurred.

2.2.5 Query Flight Information

Authorized users, such as the management team and sales personnel, can query flight information, including seat availability, occupancy rates, and revenue generated for specific flights or routes.

2.3 User Classes and Characteristics

The classes used in the software are:

User:

Characteristics: Users can search for available flights based on criteria and book tickets for their travel.

Responsibilities: Booking flights, managing their bookings, accessing flight information.

Access Privileges: Limited to actions related to booking and managing personal bookings.

Interactions: Interacts primarily with the booking and flight search functionalities.

Admin:

Characteristics: Admins have access to administrative functionalities for managing flights and overseeing system operations.

Responsibilities: Adding new flights, deleting existing flights, managing profits and financial data.

Access Privileges: Full access to administrative functions, including flight management and profit monitoring.

Interactions: Engages with administrative tools and functionalities to maintain and optimize system operations.

2.4 Operating Environment

The software is developed in Django running on MacOS. While the front end was developed using HTML and CSS. The software is compatible on any OS with internet connection available.

2.5 Design and Implementation Constraints

- 1. There are fixed number of balcony seats and ordinary seats in the auditorium at any given time.*
- 2. An ordinary seat cannot be overpriced than a balcony seat.*
- 3. The Show Manager and Account Clerk cannot be fired.*
- 4. Tickets cannot be transferred over to another spectator.*
- 5.No particular design constraints are observed in this software.*

3. System Features

3.1 Ticket Booking

3.1.1 Description and Priority

This feature allows users to book tickets for flights within the Airline Management System (AMS). It is of high priority as it forms the core functionality of the system.

3.1.2 Stimulus/Response Sequences

- *Sales personnel or spectators log in to the system and select a desired flight.*
- *They specify the seat type and provide necessary details, including sales ID.*
- *The system checks seat availability and processes the booking request.*
- *Upon successful booking, the system generates a ticket and updates the database with relevant information.*

3.1.3 Functional Requirements

1. *User authentication is mandatory for booking and cancelling tickets.*
2. *The system should display available flights and seat options to users.*
3. *Upon booking, the system should update the database with ticket information and deduct the appropriate amount from the spectator's wallet, adding it to the salesperson's wallet if applicable.*
4. *Each ticket booking should generate a unique transaction ID for tracking purposes.*

3.2 Cancelling Ticket

3.2.1 Description and Priority This feature allows users to cancel their booked tickets. It is of high priority to ensure user flexibility and convenience.

3.2.2 Stimulus/Response Sequences

- Sales personnel or spectators provide the transaction ID of the ticket to be cancelled.
- The system verifies the transaction ID and retrieves the corresponding ticket details.
- Upon cancellation, the system refunds the appropriate amount as per the cancellation policy.
- The cancelled ticket information is removed from the database.

1. Users must authenticate before cancelling tickets.
2. The system should validate the provided transaction ID.
3. Upon cancellation, the system should update the wallet balance and database records accordingly.

3.3 Flight Management

3.3.1 Description and Priority This feature enables spectators and salespersons to manage their wallet balances within the AMS. It is of medium priority to facilitate payment processing.

3.4 Show Information Query

3.4.1 Description and Priority This feature allows users to query information about flights, such as seat availability and revenue generated. It is of medium priority to provide users with relevant flight details.

3.4.2 Stimulus/Response Sequences

- Show manager queries seat availability and revenue for a specific flight.
- Salespersons and spectators query seat availability for a specific flight.

3.4.3 Functional Requirements

1. Users must authenticate to access flight information.
2. The system should provide real-time updates on seat availability and revenue.
3. Flight information queries should be efficient and responsive.

- 2 The system should provide real-time updates on seat availability and revenue.
- . Show information queries should be efficient and responsive.
- 3
- .

3.5 Balance Sheet Generation

3.5.1 Description and Priority

This feature enables the management team to generate balance sheets to track financial performance within the AMS. It is of high priority for financial transparency and reporting.

3.5.2 Stimulus/Response Sequences

- *Show manager requests a balance sheet for a specific year or show.*
- *The system retrieves relevant financial data and generates the balance sheet.*
- *The balance sheet is presented to the show manager for review.*

3.5.3 Functional Requirements

1. *Only authorised users (management team) can access balance sheet generation.*
2. *The system should accurately calculate income and expenses for the specified period.*
3. *Generated balance sheets should be available for viewing and exporting.*

3.6 Expenditure Management

3.6.1 Description and Priority

3.6.1 Description and Priority This feature enables the account clerk to add and manage expenditures for each flight, ensuring accurate financial tracking within the AMS. It is of high priority for maintaining financial records and transparency.

3.6.2 Stimulus/Response Sequences

- *Account clerk logs in to the system with appropriate credentials.*
 - *Account clerk selects the option to add expenditures.*
 - *The system prompts the account clerk to input details such as flight ID, expenditure category, and amount.*
 - *Upon submission, the system updates the expenditure records in the database and generates a confirmation message*
1. *Only authorised users (account clerk) can access expenditure management features.*
 2. *The system should validate input data to ensure accuracy and consistency.*
 3. *Account clerk can view the list of added expenditures for each flight.*
 4. *Expenditure records should be easily searchable and editable by the account clerk.*

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Login Interface

● Description

● Components

- The login interface allows users to authenticate themselves before accessing the Airline Management System (AMS).
- Components:
 - Username field: Users enter their username.
 - Password field: Users enter their password.
 - Login button: Users click this button to authenticate or Signup

4.1.2 Dashboard Interface

● Description

- Description: The dashboard interface serves as the central hub for users after logging in, providing access to various functionalities based on user roles.
- Components:
 - Navigation menu: Provides links to different sections such as Flight Booking, Show Management, Wallet Management, etc., depending on user roles.
- Visual Design: The dashboard features a clean layout with intuitive navigation.

4.1.3 Ticket Booking Interface

● Description

The ticket booking interface allows users to select shows, choose seat types, and book tickets.

● Components

Show selection dropdown: Users can choose from available shows.

Seat type selection: Users can select between balcony and ordinary seats.

Salesperson selection: For spectators, the option to choose a salesperson may be available.

Book button: Users confirm their booking by clicking this button.

● Visual Design

The ticket booking interface is designed to be user-friendly, with clear options for show selection, seat type, and booking confirmation.

4.1.5 Show Information Query Interface

- Description: The show information query interface allows users to inquire about flight details such as seat availability and revenue generated.
- Components:
 - Query options: Users can select parameters such as flight ID, date, etc., to retrieve specific information.
 - Query results display: Shows relevant information based on the user's query.
- Visual Design: The show information query interface features intuitive controls for selecting query parameters and displays query results in a structured format

4.1.6 Balance Sheet Generation Interface

- The balance sheet generation interface enables authorized users to generate balance sheets for financial reporting.
- Components:
 - Date range selection: Allows users to specify the period for which the balance sheet should be generated.
 - Show selection: Allows users to pick the flight for which the balance sheet should be generated.
 - Generate button: Initiates the process of generating the balance sheet.
- Visual Design: The balance sheet generation interface provides options for selecting the date range or flight and a button to trigger the generation process.

4.1.7 Bookings Overview

- Description:
 - The bookings overview section displays a list of all the tickets booked by the user, including relevant details such as flight ID, seat type, booking ID, and booking date.
- Components:
 - Ticket list: A table or list format displaying all booked tickets.
 - Ticket details: Each ticket entry includes information such as flight ID, seat type, transaction ID, booking date, and options to view or cancel the ticket.
- Visual Design: The booking overview presents ticket information in a clear and organized manner, making it easy for users to review their bookings at a glance.

4.1.8 Ticket Details

- Description:
 - Clicking on a specific ticket entry expands the view to show detailed information about that ticket.
- Components:
 - Flight details: Information about the booked flight, including flight ID, date, time, and venue.
 - Seat information: Details about the booked seat, such as seat number and type.
- Visual Design: The ticket details section provides users with comprehensive information about each booked ticket, allowing them to verify details and track their bookings effectively.

4.2 Hardware Interfaces

The Airline Management System (AMS) requires standard hardware equipment including:

- *PC with a monitor, keyboard, and mouse: These components are necessary for users to interact with the AMS software.*
- *Printer: A printer is required to print tickets generated by the system.*

4.3 Software Interfaces

The AMS is a standalone system developed using the Django framework on macOS. It operates independently without reliance on any additional software except Django. Key software interfaces include:

- *Django Framework: The system is built using the Django framework, which provides the necessary infrastructure for web application development.*
- *Operating System: The system is developed and deployed on macOS.*
- *Database: The AMS interacts with an online database, which requires an internet connection for communication.*
- *GUI (Graphical User Interface): The GUI for the AMS is created using Django's built-in templating system, incorporating HTML, CSS, and JavaScript for frontend design and functionality.*

4.4 Communications Interfaces

● WebBrowserInterface

The system shall be accessible through standard web browsers such as Google Chrome, Mozilla Firefox, and Safari.

HTTP (Hypertext Transfer Protocol) shall be used for communication between the web server and clients.

● **ElectronicForms:**

The system shall utilise electronic forms for user input, such as ticket booking forms and registration forms. Form submissions shall be processed securely over HTTP to protect sensitive user data during transmission.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- *Response Time: The AMS must respond within 2 seconds under normal load conditions.*
- *Scalability: The system must be capable of handling 100 simultaneous transactions without experiencing performance degradation.*

5.2 Safety Requirements

- *Data Integrity: The AMS must ensure the integrity and consistency of data stored within the database, preventing unauthorized access, modification, or corruption.*

5.3 Security Requirements

User Authentication: Users must authenticate themselves before accessing the AMS.

Data Encryption: Passwords stored in the database must be encrypted using hashing algorithms to ensure data security.

5.4 Software Quality Attributes

Software Quality Attributes: Ensuring intuitive interface with minimum 80% usability score and maintaining code complexity below 20 for ease of maintenanc

6. Other Requirements

- *Legal Requirements: The software shall comply with relevant laws and regulations governing data privacy, security, and intellectual property rights. This includes but is not limited to GDPR (General Data Protection Regulation) and any applicable local regulations.*

- Django==3.2.12
django_autoslug==1.9.8
django_email_verification==0.3.1
django_envirion==0.10.0
python-decouple==3.8
python_dateutil==2.8.1

Appendix A: Glossary

SAMS : SAMS is used for our Auditorium Management System.

AC: Abbreviation of Account Clerk

SM : Abbreviation of Show Manager

SP : Abbreviation of Sales Person

Transaction ID : Every ticket booked has its own transaction ID. This is a unique number and is used during cancellation and also to set it against the sales person ID so that he gets the commission for that.

Balance Sheet : A balance sheet contains the incomes and expenditures. Income is generated from only tickets. Expenditures include artists' payments, logistics etc.

Sales ID : Every sales person has a unique sales ID which helps in accessing the software and also to get commission for the tickets booked under the sales ID.

Appendix B: Issues List

- 1. Finalize user interface design for ticket booking.*
- 2. Clarify procedure for handling last-minute ticket cancellations.*
- 3. Determine whether to support exporting balance sheets in PDF format.*
- 4. Define maximum number of tickets allowed per user.*
- 5. Review options for integrating with payment gateway services.*