Software Requirements Specification

for

Students’ Auditorium Management Software(SAMS)

v1.0

Prepared by

Sai Deepak Reddy Mara

Dhanunjay Gurram

Eswara Adithya

Table of Contents

Revision History

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

1. Introduction

1.1 Purpose

The document details the software requirements for Students' Auditorium Management System (SAMS) v1.0. It outlines functionalities for managing show schedules, allocating and booking balcony & ordinary seats, handling cancellations with penalties, managing authorised sales personnel and their transactions, recording show expenses, and generating various balance sheets.

1.2 Document Conventions

SAMS : SAMS is used for our Auditorium Management System.
AC : Abbreviation of Account Clerk
SM : Abbreviation of Show Manager
SP : Abbreviation of Sales Person

1.3 Intended Audience and Reading Suggestions

This document caters to a diverse audience of developers, project managers, testers and users.

Developers can refer to Sections 2 (Overall Description) and 3 (Specific Requirements) to provide a comprehensive understanding of the system's functionalities and technical specifications.
Project Managers can refer to Section 2, and 3 with overview of Sections 1 (Introduction) to equip with sufficient context on project scope, timelines, and resource allocation.
Testers require understanding of the functionalities outlined in Section 3 , along with any specific non-functional requirements related to performance or usability (Section 4)as primary function is to identify and report potential issues.
Users (Show Manager, Spectators, Sales Personnel, Accounts Clerk) require reference to the Section 2 to equip context on each user role and functionalities.

1.4 Project Scope

SAMS aims to streamline show management in the college auditorium by automating tasks, offering online ticketing and cancellations, facilitating user roles and access control, enabling basic financial reporting, and managing show expenses. The main goal of this software is efficient resource utilisation, improved financial transparency, focusing on core functionalities like show creation, ticketing, user roles, basic financial reports, and expense management.

1.5 References

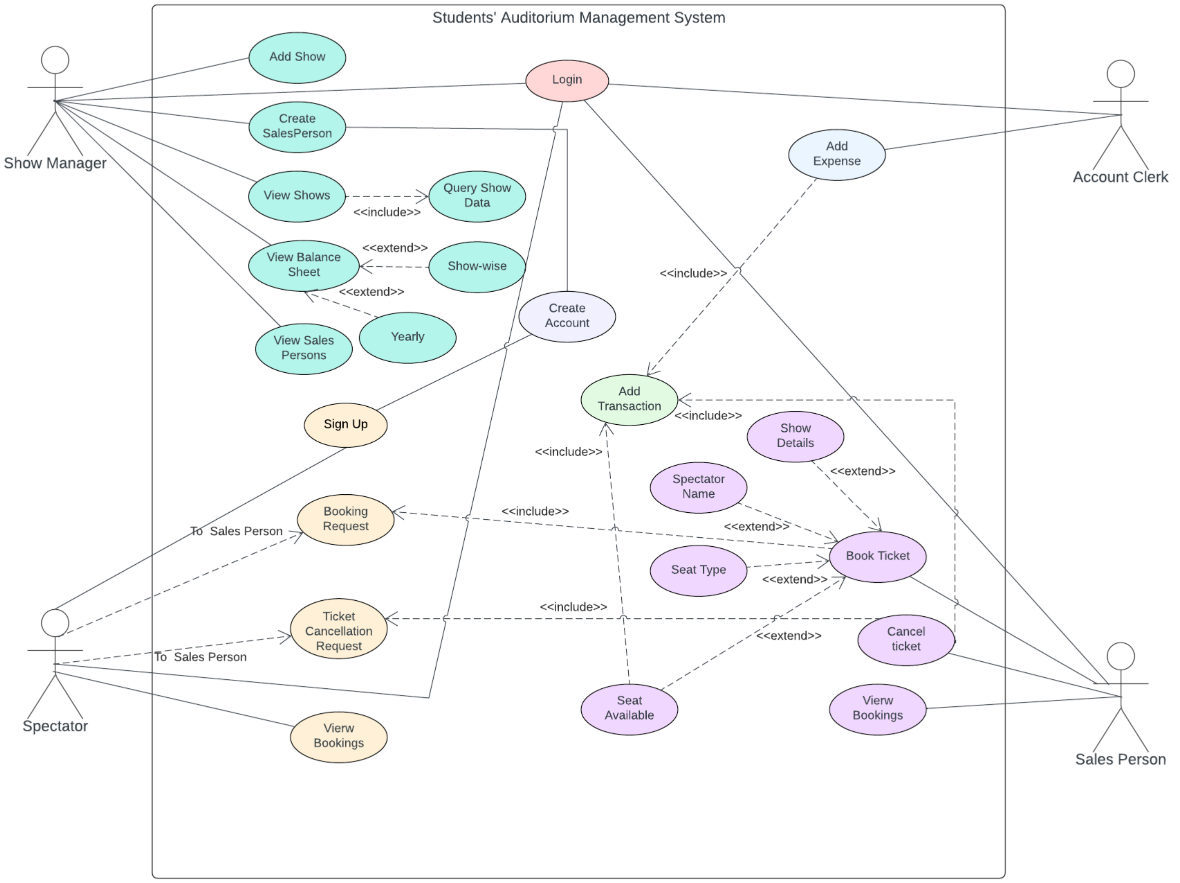
- *Software Engineering Lecture on SASD(NPTEL course by Partha Pratim Das)*
- *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.*

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

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 Cancelling Ticket

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 Wallet

Spectators and sales persons have wallet in the software in which spectators can add money and book tickets and salespersons can either add or withdraw the amount from the wallet. This process is to ensure the ease of payment processing.

2.2.4 Query Information of Shows

The Show Manager can query the empty seat percentage, amount collected for a particular show. Salespersons and spectators can query about the availability of the seats for a particular show.

2.2.5 Balance Sheet

The Show Manager can query about the balance sheet yearly or show-wise to enquire about the amount collected and amount spent on each show/year and is only available to the show manager. The Balance sheet is generated from the ticket income by selling tickets to the show and the expenditures added by the account clerk and also the commision payable to the sales person.

2.3 User Classes and Characteristics

The classes used in the software are:

1) **Show Manager**

   The auditorium is managed by Show Manager who is responsible for managing shows and supervising the sales persons, account-clerk working under him. He grants permission to his employees to use the software. SM can add shows, view balance sheets(yearly/show-wise).

2) **Account Clerk**

   AC is responsible for generating the balance sheets by reporting the expenditure for each show (including artist payments, or any other expenditures).

3) **Sales Persons**

They are responsible for booking the tickets on demand of the spectators and also cancelling the tickets on demand. He also receives a commission for the tickets he has sold.

4) **Spectators**

They can query for the seat availability and can book tickets under a sales person by choosing the sales person. They can also cancel the ticket under the sales person they booked before.

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.

2.6 User Documentation

The following components will be delivered along with the software:

- *User Manuals*

Show Manager Manual: This manual is tailored for Show Manager and provides detailed instructions on managing shows, supervising personnel, generating balance sheets, and other managerial tasks within SAMS.

Salesperson Manual: Designed for Sales Persons, this manual offers step-by-step guidance on booking tickets,and utilising other relevant features.

- *Release Notes*

Detailed release notes will accompany each software update, highlighting new features, enhancements, bug fixes, and any changes to existing functionality. This documentation will ensure users are informed about the latest improvements and modifications to the system.

2.7 Assumptions and Dependencies

This software has been designed to operate on Windows, macOS and Linux systems using the Django framework, with MySQL as the chosen database management system. It relies on an online MySQL database accessible from your local PC. An active internet connection is necessary for accessing and interacting with the database. As Django is a Python-based framework, it ensures platform independence, enabling seamless operation across environments.

3. System Features

3.1 Ticket Booking

3.1.1 Description and Priority

This feature enables users to book tickets for auditorium shows. It is of high priority as it is fundamental to the system's functionality.

3.1.2 Stimulus/Response Sequences

- *Sales person/Spectator logs in and selects a show.*
- *They select the seat type and sales ID to book a ticket.*
- *System checks availability and processes the booking.*
- *Upon successful booking, the system generates a ticket and updates the database.*

3.1.3 Functional Requirements

1. *User authentication is required for booking and cancelling tickets.*
2. *The system should display available shows and seat options.*
3. *Upon booking, the system should update the database with the ticket information and deduct the appropriate amount from the spectator's wallet and add it to the SP wallet.*
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.

3.2.2 Stimulus/Response Sequences

- Sales person/Spectator provides the transaction ID of the ticket to be cancelled.
- System verifies the transaction ID and retrieves the ticket details.
- Upon cancellation, the system refunds the appropriate amount as per the cancellation policy.

- The cancelled ticket information is removed from the database.

3.2.3 Functional Requirements

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 Wallet Management

3.3.1 Description and Priority

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

3.3.2 Stimulus/Response Sequences

- Spectators can add money to their wallet for ticket booking.
- Salespersons can add or withdraw money from their wallet.
- The system updates the wallet balance based on user actions.

3.3.3 Functional Requirements

1. Wallet transactions should be securely processed.
2. Spectators and salespersons must authenticate before accessing wallet functions.
3. The system should maintain accurate wallet balances and transaction history.

3.4 Show Information Query

3.4.1 Description and Priority

This feature allows users to query information about shows, such as seat availability and revenue generated. It is of medium priority to provide users with relevant show details.

3.4.2 Stimulus/Response Sequences

- Show manager queries empty seat percentage and revenue for a particular show.
- Salespersons and spectators query seat availability for a specific show.

3.4.3 Functional Requirements

1. Users must be authenticated to access show information.

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

3.5 Balance Sheet Generation

3.5.1 Description and Priority

This feature enables the show manager to generate balance sheets to track show finances. 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 (show manager) 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

This feature empowers the account clerk to add and manage expenditures for each show, ensuring accurate financial tracking. 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 show ID, expenditure category, and amount.*
- *Upon submission, the system updates the expenditure records in the database and generates a confirmation message.*

3.6.3 Functional Requirements

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 show.
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**
  The login interface allows users to authenticate themselves before accessing the system.
- **Components**
  Username field: Users enter their username.
  Password field: Users enter their password.
  Login button: Users click this button to authenticate.
  Sign Up button: Redirects Spectators to sign up.
- **Visual Design**
  The login interface follows standard design conventions with input fields for username and password, and a login button.

4.1.2 Dashboard Interface

- **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 Ticket Booking, Show Management, Wallet Management, etc. depending on different users.
- **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.

4.1.4 Wallet Management Interface

- **Description**
  The wallet management interface enables users to add funds to their wallet (for spectators) or perform transactions (for salespersons).
- **Components**
  Wallet balance display: Shows the current balance in the user's wallet.
  Add funds/Withdraw funds buttons: Users can add or withdraw funds based on their role.
- **Visual Design**
  The wallet management interface provides a clear overview of the user's wallet balance and options for adding or withdrawing funds.

4.1.5 Show Information Query Interface

- **Description**
  The show information query interface allows users to inquire about show details such as seat availability and revenue generated.
- **Components**
  Query options: Users can select parameters such as show 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

- **Description**
  The balance sheet generation interface enables the show manager to generate balance sheets for financial reporting.
- **Components**
  Date range selection: Allows the SM to specify the period for which the balance sheet should be generated.
  Show selection : Allows SM to pick the show for which 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 show and a button to trigger the generation process.

4.1.7 Bookings Overview

- **Description**
  The booking overview section displays a list of all the tickets booked by the user, including relevant details such as show ID, seat type, Booking ID, booking date.
- **Components**
  Ticket list: A table or list format displaying all booked tickets.
  Ticket details: Each ticket entry includes information such as show ID, seat type (balcony or ordinary), transaction ID, booking date, and option to view ticket or cancel it.
- **Visual Design**
  The booking overview presents ticket information in a clear and organised 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**
  Show details: Information about the booked show, including show ID, date, time, and venue.
  Seat information: Details about the booked seat, such as seat number and type (balcony or ordinary).
- **Visual Design**
  The ticket details section provides users with comprehensive information about each booked ticket, allowing them to verify details, print ticket and track their bookings effectively.

4.2 Hardware Interfaces

The hardware equipment required is a PC with a monitor, a keyboard and a mouse as well as a printer to print the ticket generated.

4.3 Software Interfaces

This software is a standalone system developed using the Django framework on macOS. It does not depend on any other software except for Django. It requires an internet connection to interact with an online database. The GUI for the software will be created using Django's built-in templating system(ie, HTML, CSS, JavaScript).

4.4 Communications Interfaces

- ***Web Browser Interface***

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.

- ***Electronic Forms****:*

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****: System must respond within 2 seconds under normal load.*
- ***Scalability****: System must handle 100 simultaneous transactions without performance degradation.*

5.2 Safety Requirements

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

5.3 Security Requirements

- ***User Authentication****: Users must authenticate before accessing the system.*
- ***Data Encryption****: Encrypt password while storing in the database by hashing.*

5.4 Software Quality Attributes

- ***Usability****: Maintaining an intuitive interface with an 80% usability score.*
- ***Maintainability****: Keeping the code complexity below 20 for ease of maintenance.*

6. Other Requirements

- ***Database Requirements:*** *The system shall utilise a relational database management system (RDBMS) for data storage and retrieval. MySQL version 8.0 or later is recommended.*
- ***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.*

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 commision for the tickets booked under the sales ID.*

Appendix B: Issues List

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