Hostel Mess Management System
Software Requirements Specification

Project: DEMO

Document: SRS

Author: Group 1

Published on: 2024-03-18

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Group 1	March 18, 2024	Requirement Analysis	1

1. Introduction

This document outlines the Software Requirements Specification (SRS) for a Hostel Mess Management System. The system aims to automate and streamline the management of hostel mess operations, improving efficiency and communication for residents, wardens, and mess managers.

1.1. Purpose

The primary purpose of the Hostel Mess Management System is to:

- Eliminate manual mess management processes.
- Enhance transparency and communication between residents, wardens, and mess managers.
- Reduce food wastage and optimise mess budgets.
- Facilitate efficient inventory management and meal planning.
- Provide residents with flexible meal subscription options and feedback mechanisms.

1.2. Scope

The system encompasses the functionalities related to:

- Menu management with descriptions and nutritional information.

- Resident meal subscription and unsubscription with deadlines.
- Resident feedback mechanism for meal rating and comments.
- Mess budget and expense tracking for the warden.
- Management of mess timings and special dietary needs.
- Resident feedback reports for the warden.
- Inventory management and purchase order generation for groceries.
- Recording of daily meal consumption by the mess manager.
- Generation of reports on daily, weekly, and monthly meal consumption.
- Menu creation and management with customisation options.
- Tracking of resident attendance for each meal.

1.3. Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification
- Mess: Hostel dining facility
- API: Application Programming Interface
- UI: User Interface
- UX: User Experience
- UAT: User Acceptance Testing

1.4. References

- Project Management Institute (PMI): https://www.pmi.org/ (for project management practices)
- Web Content Accessibility Guidelines (WCAG): https://www.w3.org/WAI/ (for accessibility considerations)
- OWASP Top 10: https://owasp.org/www-project-top-ten/ (or similar security framework)

1.5. Overview

The Hostel Mess Management System will be a web-based application accessible through a web browser on various devices (computer, mobile phone). The system will utilise a secure database to store user information, meal data, and financial records.

2. Overall Description

2.1. Product Perspective

The Hostel Mess Management System is intended for use in residential hostels, particularly those with a centralised mess facility. The system will be beneficial for:

- Residents: Simplifying meal planning, providing feedback, and managing mess bills.
- Wardens: Monitoring mess budgets, addressing resident concerns, and managing mess operations effectively.
- Mess Managers: Streamlining inventory management, menu creation, and meal tracking.

2.2. Product Functions

The core functionalities of the system include:

- **Resident Functions:**
  - View daily menu with descriptions and nutritional information.
  - Subscribe or unsubscribe for meals within specified deadlines before each meal.
  - Provide feedback on meals (rating and comments).
  - Access monthly mess bills with a detailed breakdown of charges.

- **Warden Functions:**
  - Monitor mess budget and expenses through a dashboard.
  - View resident feedback reports to address concerns.
  - Manage mess timings (breakfast, lunch, dinner) and announce changes through the system or other designated channels (e.g., notice boards).
  - Manage special dietary needs of residents (e.g., vegetarian, allergies).

- **Mess Manager Functions:**
  - Manage inventory with stock level tracking and purchase order generation.
  - Record daily meal consumption for each dish prepared.
  - Generate reports on daily, weekly, and monthly meal consumption to analyse food trends and wastage.

- Create and manage menus with customisation options based on resident preferences and dietary needs.
- Track resident attendance for each meal to calculate meal costs and plan accordingly.

2.3. User Characteristics & Stakeholders

The system will cater to three distinct user groups:

- **Residents:** Students residing in the hostel who utilise the mess facility.
- **Warden:** The hostel warden responsible for overseeing all hostel operations, including the mess.
- **Mess Manager:** The individual responsible for managing the day-to-day operations of the mess, including meal preparation and inventory control.

2.4. General Constraints

- The system should be user-friendly and accessible for users with varying technical expertise.
- Security measures should be implemented to protect user data (login credentials, dietary information).
- The system should be scalable to accommodate a growing number of residents.

3. Specific Requirements

3.1. Functional Requirements

3.1.1. Resident Requirements

[FR-1] The system shall allow residents to view the daily menu with descriptions and nutritional information.

[FR-2] The system shall allow residents to subscribe or unsubscribe for meals with a specified deadline before each meal (e.g., 24 hours in advance).

[FR-3] The system shall provide a feedback mechanism for residents to rate meals (star rating) and leave comments.

[FR-4] The system shall allow residents to access their monthly mess bills with a detailed breakdown of charges (including meal costs, taxes, etc.).

3.1.2. Warden Requirements

[FR-5] The system shall provide a dashboard for the warden to monitor the mess budget and expenses (total spending, category-wise breakdown).

[FR-6] The system shall display reports summarising resident feedback, including average meal ratings, common comments, and any recurring issues.

[FR-7] The system shall allow the warden to manage mess timings (breakfast, lunch, dinner) by setting specific times and announcing changes through the system notifications or designated channels (e.g., notice boards).

[FR-8] The system shall enable the warden to manage special dietary needs of residents (e.g., vegetarian, vegan, gluten-free) by allowing residents to register their preferences and enabling menu planning accordingly.

3.1.3. Mess Manager Requirements

[FR-9] The system shall provide an inventory management module to track stock levels of groceries (rice, lentils, vegetables, etc.). The system should generate automatic purchase orders when stock falls below a predefined threshold to maintain sufficient supplies.

[FR-10] The system shall allow the mess manager to record daily meal consumption for each dish prepared, specifying the quantity used.

[FR-11] The system shall generate reports on daily, weekly, and monthly meal consumption, analysing food trends, identifying potential wastage areas, and enabling informed menu planning.

[FR-12] The system shall provide tools for the mess manager to create and manage menus with customisation options. This may include offering alternative dishes based

on dietary preferences or allowing residents to choose from a selection for specific meals.

[FR-13] The system shall enable the mess manager to track resident attendance for each meal using methods like swiping ID cards or manual attendance registers. This data will be used to calculate meal costs per resident and plan accordingly.

3.2. Non-Functional Requirements

3.2.1. Usability

- The system interface should be intuitive and user-friendly, catering to users with varying technical expertise. A clear navigation menu and informative labels will ensure ease of use.
- The system should be responsive and accessible on various devices (computer, mobile phone, tablet) with adjustments to the layout for optimal viewing on different screen sizes.

3.2.2. Reliability

- The system should be highly available with minimal downtime to ensure seamless operation of the mess and meal service for residents.
- The system should implement data backup and recovery procedures to protect against data loss in case of system failures or hardware malfunctions. Regular

backups should be performed, and a disaster recovery plan should be established.

3.2.3. Security

- The system should enforce secure user authentication mechanisms (e.g., username and password with strong password policies) to restrict unauthorised access to resident information, financial data, and system functionalities.
- The system should encrypt sensitive user data (login credentials, dietary information) to ensure confidentiality even if there's a data breach. Encryption protects data at rest and in transit.

3.2.4. Performance

- The system response time should be fast and efficient to handle concurrent user requests without delays. Residents should be able to access menus, subscribe for meals, and provide feedback quickly.
- The system should be scalable to accommodate a growing number of residents and data volume without compromising performance. The system architecture should be designed to handle increased usage efficiently.

4. Design Constraints

- The specific technologies and programming languages used for development will be determined during the design phase. The chosen technologies should be suitable for web application development and meet the system requirements.
- The system should be designed for potential integration with existing payment systems (optional) to enable online mess fee payments. This would require defining APIs or data exchange mechanisms with the payment gateway.

5. Quality Attributes

- The system should be well-documented to facilitate future maintenance, upgrades, and troubleshooting. Clear and concise documentation will improve understanding for developers and support personnel.
- The system should be modular in design to allow for easy modification and addition of new features. A modular design enables future enhancements and customisation based on evolving needs.
- The system should undergo thorough testing to ensure functionality, usability, and security. This will involve various testing methodologies like unit testing, integration testing, system testing, and user acceptance testing (UAT) to identify and rectify any issues before deployment.

6. Verification and Validation

- **Verification:** The verification process will ensure the SRS accurately reflects the needs and expectations of stakeholders. This can be achieved through:

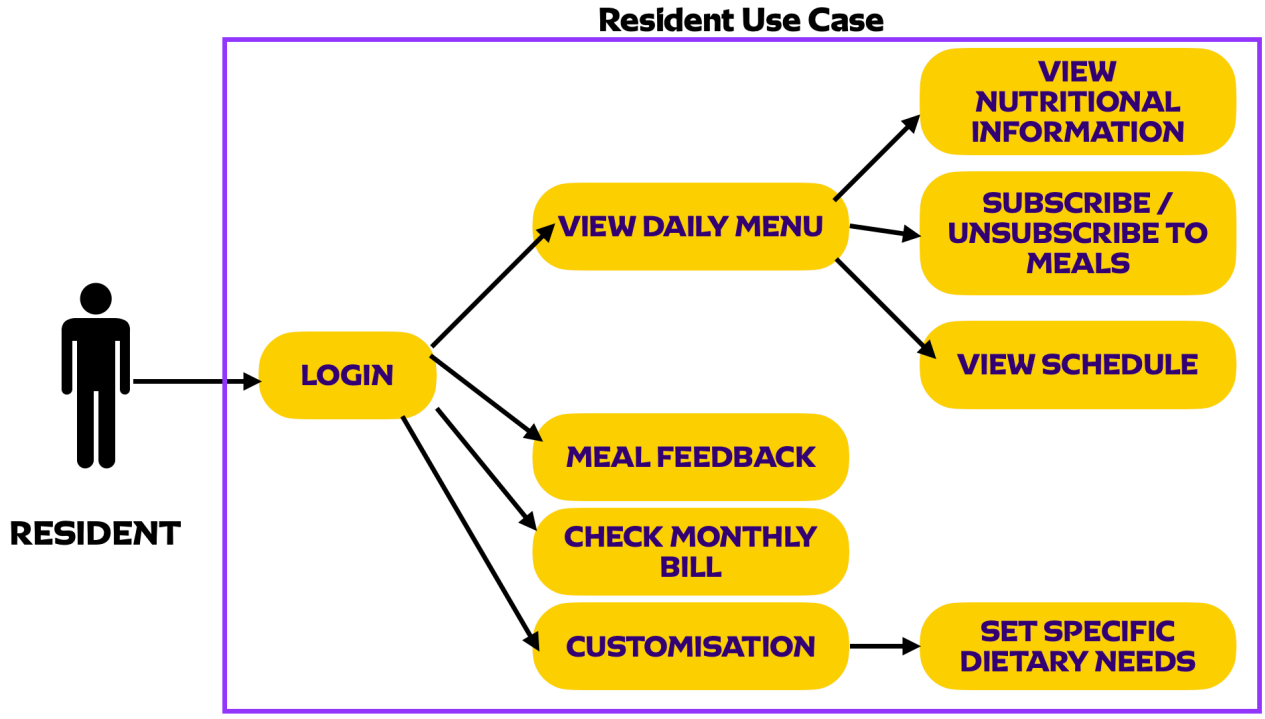
  - **Reviews:** Conducting formal and informal reviews with residents, wardens, and mess managers to obtain feedback on the functionalities outlined in the SRS.
  - **Use Case Diagrams:** Developing use case diagrams that depict user interactions with the system for specific tasks (e.g., viewing menu, managing inventory). These diagrams will be reviewed by stakeholders to verify if they capture the intended functionalities. (**See Appendix A for Example Use Case Diagrams**)
  - **Traceability Matrix:** Creating a matrix that maps each functional requirement (FR) to specific design elements and test cases. This ensures all functionalities are addressed during design and testing phases.

- **Validation:** The validation process will confirm that the developed system meets the requirements outlined in the SRS. This can be achieved through:

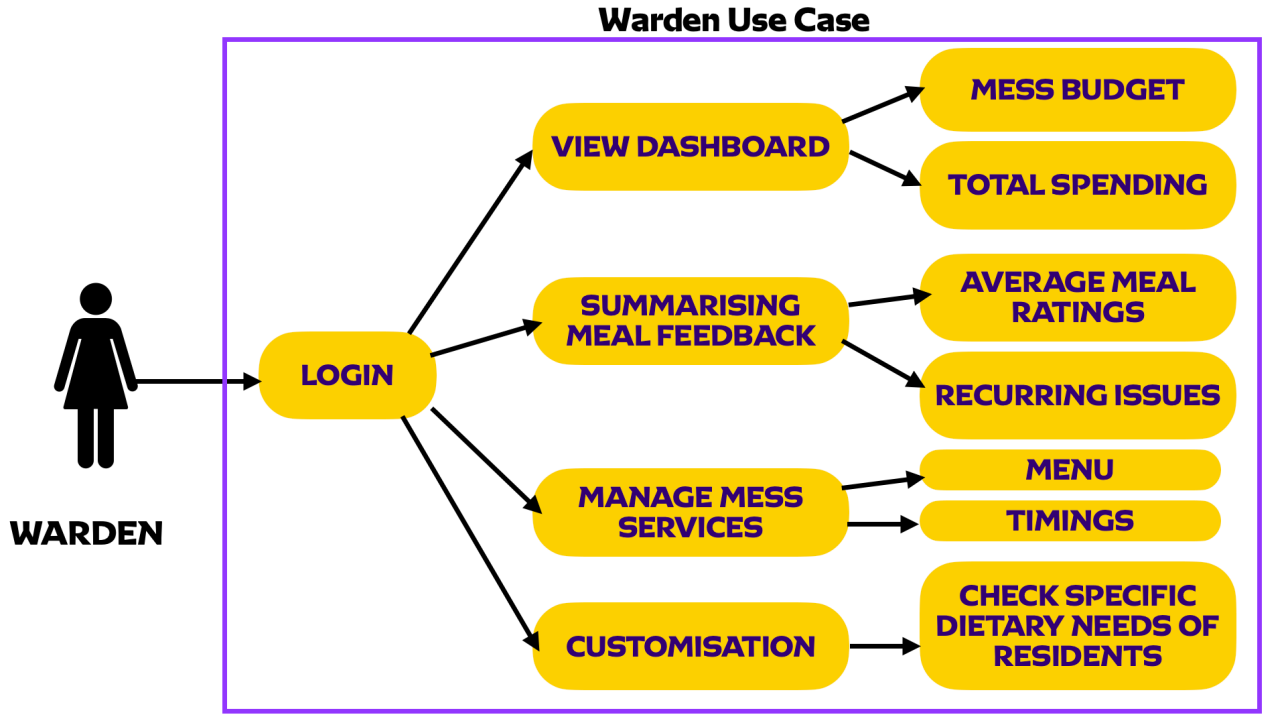
  - **User Acceptance Testing (UAT):** Developing UAT plans based on the SRS to test the system functionalities with actual users. UAT helps identify usability issues and ensure the system meets user expectations.

7. Appendix - A: Use Case Diagrams

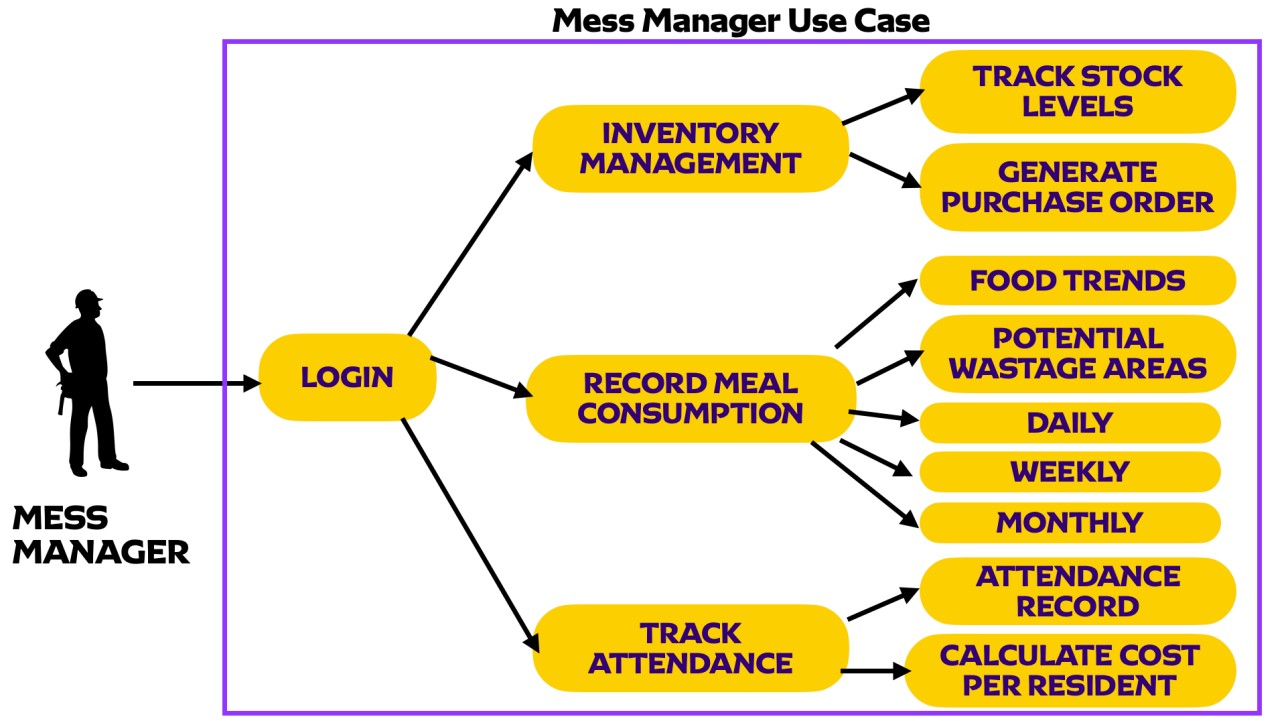
7.1. Resident Use Case Diagram



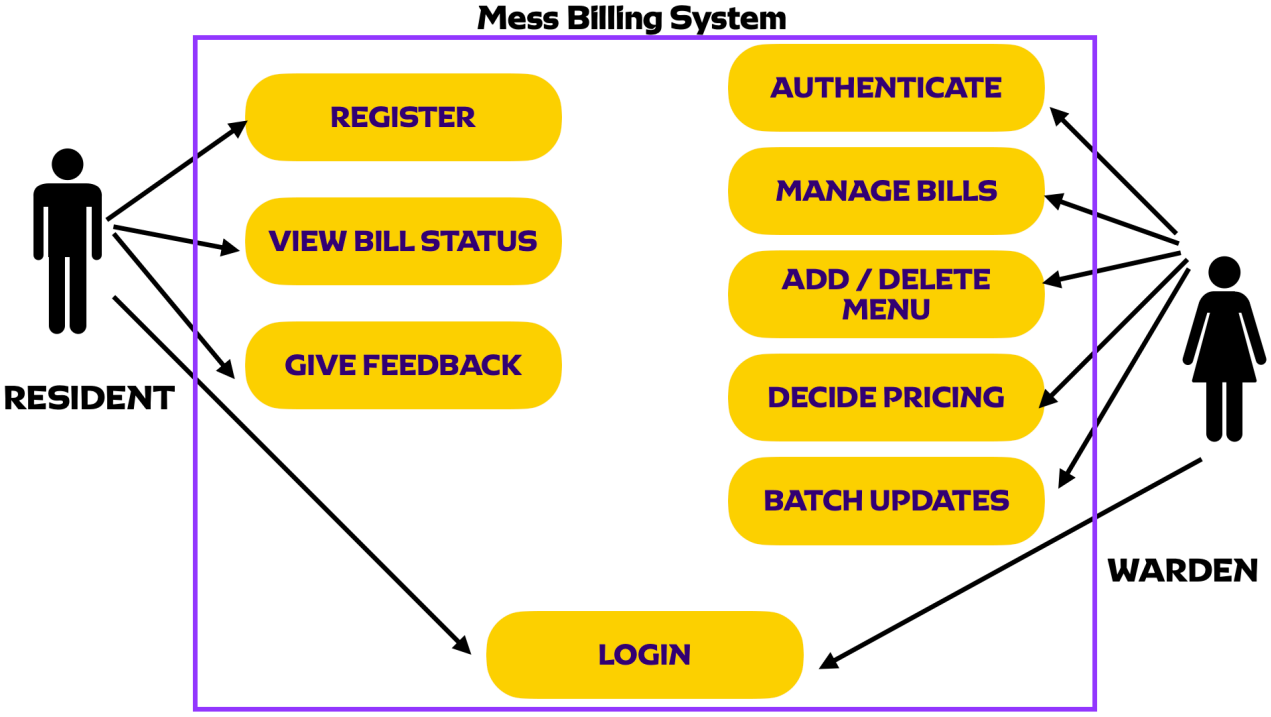
7.2. Warden Use Case Diagram



7.3. Mess Manager Use Case Diagram



7.4. Mess Billing System Use Case Diagram



8. Appendix - B: Data Dictionary

Data Element Name	Data Type	Description	Example
Resident Name	String	Unique Identifier for a Resident	“Virat Kohli”
Dish Name	String	Name of the dish offered in the menu	“Chicken Curry”
Meal Date	Date	Date for which the meal is served	“2024-03-18”
Dietary Restriction	String	Specific dietary restrictions of a resident	“Vegan”

9. Appendix - C: List of Interfaces

Interface Name	Description	Interface Type	Standards / Protocols
Payment Gateway Integration	Enables online payment for mess fees	API	RESTful APIs, HTTPS
SMS Gateway Integration	Sends notification SMS to residents about menu changes or announcements	API	Twilio API

10. Conclusion

This Software Requirements Specification (SRS) provides a comprehensive overview of the Hostel Mess Management System's functionalities and requirements. It outlines the use-case diagrams and functional requirements for each stakeholder. It serves as a foundation for the development and design phases, ensuring clarity and understanding among both clients and designers. The system aims to enhance efficiency and transparency in hostel mess operations.