Software Requirements Specification for

Mess Management System

Indian Institute of Information Technology Guwahati

Prepared by:

- **Abhishek Kumar (2101012)**
- **Aman Soni (2101029)**
- **Aman Verma (2101031)**

29th Jan, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
v1	21.01.2024	Initial Version of SRS	1.0
v2	29.01.2024	Updated SRS with Life Cycle Model	2.0

1. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Mess Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and what kind of system interactions take place. It outlines the requirements, features, and functionality expected from the system.

1.2. Document Conventions

When you read this manual, certain words are represented in different fonts, typefaces, sizes, and weights. This highlighting is systematic, different words are represented in the same style to indicate their inclusion in a specific category.

1.3. Product Scope

The Mess Management System is a comprehensive solution tailored for students, Mess Committee members, and Faculty to enhance the management of mess-related processes within the institution. It aims to streamline communication, improve complaint resolution, and simplify menu management. Students can create and access their accounts using a username and password to view mess-related information, including menus, complaint statuses, notices, fines, and mess duties.

The system also allows students to submit complaints online and track their status. By consolidating all information in one accessible platform, the system aims to simplify the mess management process, making it more efficient and transparent for all stakeholders, while reducing the manual effort of maintaining handwritten lists.

This system aims to ease the workload for the mess staff and provide transparency to students. Centralizing all information in one accessible platform makes it simpler for everyone to view and understand mess-related details.

1.4. References

IEEE Software Requirement Specification Template.

2. Overall Description

2.1. Product Perspective

The Mess Management System is a comprehensive platform serving the entire campus community, including all students, mess committee and faculty in charge of mess. This system is designed to facilitate efficient communication, streamline operations, and enhance the overall dining experience for everyone involved. It will interact with user interfaces for both administrators and end-users.

2.2. Product Features

- **QR-Based Complaint Registration:**
    - Allow users to register complaints by scanning QR codes placed in the mess.
    - Include details such as complaint type, description, and date.
- **Mess Menu Displaying:**
    - Show the daily or weekly mess menu to the users, including breakfast, lunch, and dinner. It also displays if there is any modification on the current day.
- **Track Complaints:**
    - Users can track their complaint status via a unique reference number.
- **Concerned Authority for Problems:**
    - Display contact information for the relevant authority to address any issues or concerns related to the mess.

2.3.User Classes and Characteristics

- Administrator (Faculty-in-Charge):
    - The faculty should be able to login with their username and password.
    - Responsible for managing meeting details, member information, and system settings.
- Mess Committee Member
    - The mess manager should be able to login with his/her username and password.
    - Access to meeting details and manage mess-related issues/complaints.

- Students (End-Users)
    - A menu list is available to all the students and timings.
    - Information about their preference, monthly mess charge, mess fines and mess duties.
    - A complaint portal for the students including complaint registration and its tracking.

2.4.Operating Environment

The software will be a web based system in which a server will store all the information of mess committee members, students and other data which will be customizable by the respective users. The user end will be a graphical interface.

2.5. Design and Implementation Constraints

- **Capacity limit of database:**

The system is constrained by the capacity limit of the database.The system's scalability and ability to handle a large volume of data are limited by the capacity of the underlying database.

- **Internet Speed and Connectivity Issues:**

Varying internet speeds and connectivity issues may affect the user experience. Users in regions with poor internet connectivity may experience slower performance, affecting the overall user experience.

2.6.Assumptions and Dependencies

It is assumed that the software developed will be compatible with the hardware in use and they will be using this through web browsers.

3. Functional Requirements

3.1. User Management

- ## User Registration and Authentication:

  Functional Requirement:
  - ***Description***: Users shall be able to register with the system. User authentication shall be secure, employing two-factor authentication (2FA) for system access.
  - ***Inputs***: User-provided registration details. User credentials and 2FA verification.
  - ***Outputs***: Confirmation of successful registration. Access granted upon successful authentication.

- ## Role-based Access Control:

  Functional Requirement:
  - ***Description***: The system shall implement role-based access control, assigning specific permissions to administrators, mess committee members, and end-users.
  - ***Inputs***: Role assignment by administrators.
  - ***Outputs***: Users can access functionalities based on their assigned roles.

3.2. Mess Meeting Management

- ## Upcoming Mess Meeting Details::

  Functional Requirement:
  - ***Description***: Display information about the date, time, and agenda of upcoming mess committee meetings.
  - ***Inputs***: Meeting details entered by mess committee members.
  - ***Outputs***: Visible upcoming meeting details for all relevant users.

- **Responsible Faculty In Charge:**

  Functional Requirement:
  - ***Description***: Identify and display the faculty member responsible for overseeing mess management.
  - ***Inputs***: Assignment of the faculty member in charge by administrators.
  - ***Outputs***: Display of the responsible faculty member's information.

3.3. Complaint Portal

- **QR-Based Complaint Registration:**

  Functional Requirement:
  - ***Description***: Users can register complaints by scanning QR codes in the mess, providing details such as type, description, and date.
  - ***Inputs***: Scanned QR code and complaint details.
  - ***Outputs***: Confirmation of complaint registration with a unique identifier.

- **Complaint Tracking:**

  Functional Requirement:
  - ***Description***: Users can track the status and resolution progress of their registered complaints through the complaint tracking system.
  - ***Inputs***: Unique identifier of the registered complaint.
  - ***Outputs***: Real-time status updates, including current status, actions taken.

3.4. Mess Information Display

- **Mess Menu Displaying::**

  Functional Requirement:
  - ***Description***: Show the daily or weekly mess menu, including breakfast, lunch, and dinner.
  - ***Inputs***: Menu details updated by mess committee members.
  - ***Outputs***: Visible and up-to-date menu for users.

- ## Daily Mess Menu & Timings:

  Functional Requirement:
  - ***Description***: Clearly state the operational timings of the mess for each meal.
  - ***Inputs***: Timings configured by administrators.
  - ***Outputs***: Display of operational timings for users.

3.5. System Administration

- ## System Configuration and Settings:

  Functional Requirement:
  - ***Description***: Allow administrators to configure system settings and customize mess-related parameters.
  - ***Inputs***: Configuration changes made by administrators.
  - ***Outputs***: Updated system settings.

- ## Log and Audit Trails:

  Functional Requirement:
  - ***Description***: Maintain detailed logs and audit trails for system activities.
  - ***Inputs***: Automated logging of system activities.
  - ***Outputs***: Accessible logs for administrators to track changes and review system activity.

  All options will be displayed in a menu based format provided with a dashboard. It will be specifically designed with their users in mind.

3.6.Hardware Interfaces

A web server will be required so that the students and the mess admin can connect to it to exchange information. The server will have a database to store all the data entries. The Server will have to have a high speed ethernet connection to the college's local network.
For servers and other technical things we will be using AWS services.

4. Nonfunctional Requirements

4.1. Performance Requirements

Performance requirements define acceptable response times for system functionality

- The load time for the user interface shall not take long, especially while viewing the mess menu by the students.
- The login information shall be verified fast.
- Queries such as menu information, complaint information shall return results quickly.

4.2. Safety Requirements

The information will be saved in two different locations so that we could recover from it when there is extensive damage to the database.

4.3. Security Requirements

The server on which the Mess Management System resides will have its own security to prevent unauthorized write or delete access. There will be no restriction on read access.

In case a password is forgotten, the student and the mess workers can approach the admin (Mess Manager).

4.4. Scalability

The system should be able to handle more users and information without slowing down or breaking. It should grow smoothly if more people start using it.

4.5. Software Usability

The system should be easy for everyone to use. Buttons and options should make sense, and it shouldn't be confusing to find things like mess menu or complaint history.

4.6. Business Rules

The user must have an email ID provided by IIIT Guwahati in order to register in this software.

5.System Requirements

● **Operating System:** The Mess Management System is compatible with Windows, macOS, and Linux.

● **Web Browser:** Developers should use the latest versions of major browsers.

● **Database Management System:** The system is built on MySQL, serving as the primary database system.

● **Frontend Development:** HTML, CSS, JavaScript, React.

● **Server Requirements:** Node.js and Express.js are required components for the server-side implementation, ensuring an efficient backend infrastructure.

● **Programming Language:** JavaScript is the core programming language, with React.js utilized for the frontend and Node.js for the backend development.

● **Framework/Library Dependencies:** Developers should utilize npm packages for Express.js middleware and utility libraries, ensuring efficient development practices.

● **Security Protocols:** HTTPS must be implemented for secure data transmission.

● **Authentication Mechanism:** User authentication is implemented using JWT (JSON Web Token) or Google Auth.

6. Future Implementation:

6.1.Transaction Integration:

Integrate a secure transaction system to facilitate payments within the Mess Management System. This feature would enable users to seamlessly make transactions for meal orders in normal mess or night canteen through the platform.

6.2.Automated Notifications:

Implement a notification system to alert users about upcoming events, announcements, and important updates by the mess committee.

6.3.Feedback Form/Food Review:

Introduce a comprehensive feedback form and food review system within the Mess Management System. This feature allows users to provide detailed feedback on their dining experiences, including dining cleanliness, meal satisfaction, menu suggestions, mess hygiene, and overall service quality.

APPENDIX A

Analysis Model

Evolutionary Model Overview for Mess Management System

- **Software Concept:** The Mess Management System streamlines mess-related processes for students and faculty. Key features include menu display, complaint registration, and meeting management.

- **Preliminary Requirement Analysis:** The SRS document outlines system purpose, users, and requirements, serving as a foundation for development.

- **Design of Architecture and System Core:** MMS is a web-based system, utilizing AWS services with compatibility for various browsers. High-speed connectivity and database capacity are key considerations.

- **Develop a Version:** Identify and develop core features (user registration, complaint portal) in specific versions, mirroring the Evolutionary Model.

- **Deliver a Version:** Incremental delivery ensures users interact with evolving features at the end of each development cycle.

- **Elicit Customer Feedback:** Iterative feedback from students, faculty, and mess committee members guides continuous improvement.

- **Incorporate Customer Feedback:** User insights drive refinement, addressing concerns and aligning the system with practical needs.

- **Deliver Final Version:** Culmination of iterative cycles, integrating refined features and enhancements for a robust Mess Management System.

Additional Insights:

- Iterative Development: Accommodates dynamic mess management needs.
- Stakeholder Involvement: Encourages continuous feedback.
- Adaptability: Responds to evolving requirements.
- Feedback-Driven Refinement: Ensures alignment with user expectations.
- Reduced Planning Overhead: Promotes agility in development.

Refer link to see the Mapping:
Software Development Life-Cycle Model for Mess Management System