

**Software Requirements Specification**  
**MealMate: A Web-Based Hall canteen Management System**

**CSE-3112: Software Engineering**

**Submitted By:**

Anika Sanzida Upoma(02)  
Ishrat Jahan Mim(52)  
Mst.Tasmia Sultana Sumi(54)  
Sara Faria Sundra(58)

**Submitted to:**

Dr. Saifuddin Md. Tareeq  
Professor, CSEDU  
and  
Dr. Sarker Tanveer Ahmed Rume  
Associate Professor, CSEDU

**Date:** Nov 12, 2025

**Revision History**

Date	Version	Description	Author
Nov 10, 2025	1.0	Initial draft of SRS	Anika, Ishrat, Tasmia, Sara
Jan 01,2026	2.0	Added specific functional and non functional requirements	Anika, Ishrat, Tasmia, Sara

## **Contents**

### **1. Introduction**

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms and Abbreviations
- 1.4 References
- 1.5 Overview

### **2. Overall Description**

- 2.1 Use-Case Model Survey
- 2.2 Assumptions and Dependencies

### **3. Specific Requirements**

- 3.1 Use-Case Reports
- 3.2 Supplementary Requirements

## **1.Introduction**

The Canteen Management System (MealMate) Software Requirements Specification (SRS) provides a comprehensive description of the requirements for a digital solution to manage food ordering, billing and inventory in an institutional canteen.

This document serves as the foundation for the design, development and testing of the CMS application.

### **1.1 Purpose**

This Software Requirements Specification fully describes the external behavior of the Canteen Management System.

It outlines functional and non-functional requirements, design constraints and performance expectations necessary for a complete understanding of the system.

The SRS is intended for use by:

- Developers responsible for implementing the system
- Testers verifying the functional correctness
- Project managers overseeing development
- Stakeholders such as canteen staff, students and administration

### **1.2 Scope**

The Canteen Management System is a web-based or hybrid application that automates all major activities within a hall or institutional canteen. It aims to replace manual processes with an online platform where users can:

- View available food items and menus
- Place and manage food orders
- Make secure online payments
- Receive real-time order status updates

- Provide feedback and view order history

Administrators can:

- Manage menus and food categories
- Monitor orders
- Track daily sales and inventory levels
- Analyze customer feedback and sales reports

This system enhances operational efficiency, reduces waiting time, prevents order errors and provides transparent financial tracking.

### **3 Specific Requirements**

This section contains all software requirements to a level of detail sufficient to enable designers to design the system and testers to verify it. When using use-case modeling, these requirements are reflected through Use-Case Reports and Supplementary Specifications.

#### **3.1 Use-Case Reports**

Each use case defines a specific interaction between users and the system. The major use cases for the Canteen Management System are described below.

##### **3.1.1 Use Case 1: User Registration and Login**

Actors: Student, Manager

Priority: Optional

Trigger: User Clicks the “Login” to login if already have an account and clicks “register” to open an account.

Description: Enables users (students or managers) to log in securely using their registered credentials and access their respective dashboards.

Preconditions: Connection to the server must be established. Users must already have a registered account and valid verified email address and registration id.

**Main Flow:**

1. The user opens the home page.
2. Users enter their account credentials (email, registration ID and password etc) to complete registration.
3. The user clicks the Login button.
4. The user enters their verified email password and registration number(student) .
5. Email verification is sent to validate the information.
6. The system (via Supabase) validates the credentials.
7. If the credentials are valid, the user is redirected to their respective dashboard (Student or Manager).

**Alternate Flow:**

1. Login fails due to incorrect email or password.
2. The system displays an error message: *"Incorrect email or password."*
3. Users are prompted to try again or click Forgot Password.

Postconditions: User's profile and dashboard information are displayed successfully.

**Exceptions:**

- Network connection error.
- Supabase authentication service unavailable.
- The user attempts to log in with an unregistered email.

**3.1.2 Use Case 2: Meal Selection & Take Token**

Actors: Student, Server

Priority: High

Trigger: Student clicks on "Get Token" from the sidebar navigation.

Description: Allows students to select their desired meal (Breakfast, Lunch or Dinner) for a specific date and generate a unique OTP for meal collection.

Preconditions: Users must be logged in and authenticated. Supabase database connection must be active.

**Main Flow:**

1. Students open their dashboard and click “Get Token.”
2. The system displays available meal types: Breakfast, Lunch, Dinner.
3. The student selects a meal type and chooses a date.
4. The system displays all menu items for that meal and date, including prices and availability.
5. Students select desired items and confirm their selection.
6. The system calculates the total cost and asks for final confirmation.
7. Upon confirmation, the system generates a unique OTP and sends it to the student’s registered email or mobile number.

**Alternate Flow:**

1. Menu is unavailable for the selected date.
  - 1.1. The system displays a message: “*Menu not available.*”
  - 1.2. The student can select a different date or meal type.

Postconditions: A unique OTP is generated, stored in the database and sent to the student for meal collection.

**Exceptions:**

- Supabase database connection failure.
- Email/SMS service not responding.
- OTP generation failure or timeout.

**3.1.3: Use Case 3: Add and Edit Meal**

Actors: Manager, Server

Priority: High

Trigger: Manager clicks on “Update Menu” from the sidebar and selects a meal type (Breakfast, Lunch or Dinner).

Description: Enables the mess manager to add, edit or delete meals for a selected date and meal type, ensuring that students see the updated menu when selecting meals.

Preconditions: Manager must be logged in and authenticated. Supabase database connection must be active.

Main Flow:

1. Manager navigates to “Update Menu” from the sidebar.
2. Manager selects a date and a meal type (e.g., Lunch on 12-Nov-2025).
3. The system displays existing menu items for that date and meal type.
4. Manager can perform the following actions:
  - Add New Item: Enter item name, price and availability, then click Save.
  - Edit Item: Update item details such as name, price or stock status, then click Update.
  - Delete Item: Remove an existing item from the menu.
5. The system validates input and updates the Supabase database.
6. A confirmation message is displayed: *“Menu updated successfully.”*

Alternate Flow:

1. The manager tries to add a meal item that already exists for the selected date and meal type.
  - 1.1. The system displays: *“Item already exists. Try editing instead.”*
  - 1.2. The manager can choose to edit the existing item or cancel the operation.

Postconditions: The updated meal menu (added, edited or deleted items) is stored in the Supabase database and reflected in the student dashboard for meal selection.

Exceptions:

- Database connection failure (changes not saved).
- Input validation error (missing or invalid item details).
- Server timeout during menu update request.

### 3.1.4: Use Case 4: View Bill and Automatic Payment Generation (Student)

**Actors:** Student, Server

**Priority:** High

**Trigger:** Student clicks on “View Bill” or “Payment” from the sidebar menu.

**Description:** Allows students to view their daily and monthly meal consumption and billing information. Total payment/bill amount is automatically generated once the manager approves the student’s meal token.

**Preconditions:**

Students must be logged in and authenticated.  
Meal token must be approved by the manager.  
Supabase database connection must be active.

**Main Flow:**

1. Students log in and open the “**View Bill**” or “**Payment**” section from the dashboard.
2. Students select a month from the available options.
3. The system retrieves all approved meal records for the selected month.
4. The system automatically calculates daily meal costs and the total monthly bill based on approved tokens.
5. The system displays:
  - Daily meal breakdown
  - Total monthly amount
  - Payment status (Paid / Unpaid)
6. If the bill is unpaid, the student can proceed to the payment option.

**Alternate Flow:**

1. If no approved meals exist for the selected month:
  - 1.1. The system displays: “*No billing data available.*”
  - 1.2. Students can select another month.

**Postconditions:**

The student can view the automatically generated total bill and payment status based on approved meal tokens.

**Exceptions:**

- Database connection failure.
- Billing calculation error.
- Server timeout while retrieving payment data.

**3.1.5: Use Case 5: View Token and Payment Overview (Manager)**

**Actors:** Manager, Server

**Priority:** High

**Trigger:** Manager clicks on “**View Token**” or “**View Payment**” from the sidebar menu.

**Description:** Enables the manager to view student meal tokens, approve them, and monitor automatically generated payment records for each student.

**Preconditions:**

The manager must be logged in and authenticated for his/her particular hall.

Student meal tokens must exist in the system.

Supabase database connection must be active.

**Main Flow:**

1. The manager opens the **“View Token”** section from the dashboard.
2. The manager selects a meal type (Breakfast, Lunch, or Dinner) and a date.
3. The system displays a list of students along with their generated tokens and selected meal items.
4. The manager verifies the token and clicks **“Approve.”**
5. Upon approval, the system marks the token as approved and updates meal consumption records.
6. Manager navigates to the **“View Payment”** section.
7. The manager selects a month.
8. The system displays each student’s total payment amount, generated automatically from approved tokens, along with payment status (Paid / Unpaid).

**Alternate Flow:**

1. If no tokens are found for the selected date or meal type:
  - 1.1. The system displays: *“No tokens available.”*
  - 1.2. The manager can select another date or meal type.

**Postconditions:**

Approved tokens generate valid billing records, and the manager can successfully view all student payment summaries.

**Exceptions:**

- Token verification failure.
- Database update error after approval.
- Server timeout while loading token or payment data.