Software Requirements Specification

for

Smart Mess

Prepared by:
Mahi Agrawal & Yashika Nagdev
Computer Science And Engineering
Manipal University Jaipur

August 29, 2025

Version 1.0 approved

Contents

1 Introduction **1**
1.1 Purpose . . . 1
1.2 Document Conventions . . . 1
1.3 Intended Audience and Reading Suggestions . . . 1
1.4 Product Scope . . . 1
1.5 References . . . 2

2 Overall Description **2**
2.1 Product Perspective . . . 2
2.2 Product Functions . . . 2
2.3 User Classes and Characteristics . . . 2
2.4 Operating Environment . . . 3
2.5 Design and Implementation Constraints . . . 3
2.6 User Documentation . . . 3
2.7 Assumptions and Dependencies . . . 3

3 External Interface Requirements **4**
3.1 User Interfaces . . . 4
3.2 Hardware Interfaces . . . 4
3.3 Software Interfaces . . . 4
3.4 Communications Interfaces . . . 4

4 System Features **4**
4.1 System Feature 1: User Account Management . . . 4
4.1.1 Description and Priority . . . 4
4.1.2 Stimulus/Response Sequences . . . 5
4.1.3 Functional Requirements . . . 5
4.2 System Feature 2: Menu Display & Booking . . . 5
4.2.1 Description and Priority . . . 5
4.2.2 Stimulus/Response Sequences . . . 5
4.2.3 Functional Requirements . . . 5
4.3 System Feature 3: Feedback System . . . 6
4.3.1 Description and Priority . . . 6
4.3.2 Stimulus/Response Sequences . . . 6
4.3.3 Functional Requirements . . . 6
4.4 System Feature 4: Waste Tracking & Analytics . . . 6
4.4.1 Description and Priority . . . 6
4.4.2 Stimulus/Response Sequences . . . 6
4.4.3 Functional Requirements . . . 6

5 Other Nonfunctional Requirements **7**
5.1 Performance Requirements . . . 7
5.2 Software Quality Attributes . . . 7

6 Other Requirements **7**

A Glossary **8**

B Analysis Models **8**
B.1 UML Diagrams . . . 8
B.2 Data Flow Diagrams (DFDs) . . . 10

C To Be Determined List **11**

Revision History

Name	Date	Reason For Changes	Version
Student 1, Student 2	07/11/2025	Version 1 functionalities including login, weekly menu display, scanner for booking/cancellation, and AI nutrition tracker	1.0

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for the **Smart Mess**, version 1.0. The primary purpose of the SmartMess System is to address and mitigate the significant issue of food waste in the hostel mess. By introducing a digital platform, the system empowers students to pre-book their meals, allowing the mess administration to accurately forecast demand and prepare only the required quantity of food. Additionally, the platform enhances meal quality and transparency through features like daily feedback and ratings, menu suggestions from students, and waste tracking analytics. Ultimately, SmartMess aims to foster a sustainable and student-centric dining experience by improving communication, reducing waste, and ensuring the food served is both preferred and appreciated.

1.2 Document Conventions

This document adheres to the IEEE 830-1998 standard, uniquely identifying all functional requirements as FR-X.Y and non-functional requirements as NFR-X.Y. The term ”The system” refers to the entire Smart Mess System, while ”Student” and ”Admin” designate the user roles.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- **Developers (Project Team):** To understand what needs to be built.
- **Project Manager (Course Instructor):** To oversee the project's progress and ensure it aligns with the course objectives.
- **Testers:** To create test cases based on the specified requirements.
- **University Administration (Stakeholders):** To review the proposed functionality.

It is recommended to read the document sequentially, starting with the Introduction and Overall Description to get a high-level overview, followed by the specific System Features and Nonfunctional Requirements for detailed understanding.

1.4 Product Scope

The Hostel Mess Feedback and Meal Booking System is a web-based platform designed to improve communication and efficiency between hostel students and mess administrators. The main objective of this project is to minimize food wastage, enhance meal quality, and simplify the mess management process. The system allows students to log in, view the weekly menu, and pre-book their meals in advance, enabling the mess staff to plan food quantities accurately and reduce waste. It also provides a feedback module where students can rate and comment on meals, allowing the management to monitor satisfaction levels and make data-driven improvements. Mess administrators can update menus, track bookings, and analyze food wastage trends through an analytics dashboard. The platform will focus on usability, security, and performance to ensure smooth functioning and accessibility for all users. It will not include online payment integration or mobile application development in the current version. Overall, the system aims to create a transparent, efficient, and environmentally responsible mess management process that benefits both students and staff.

1.5 References

1. *CS-3230 Software Engineering Lab Course Handout, JAN-MAY 2022, Manipal University Jaipur.*
2. *Wiegers, Karl E. Software Requirements Specification Template, 1999.*
3. *Google Material Design Guidelines* (for UI/UX inspiration).

2 Overall Description

2.1 Product Perspective

The Mess Food Feedback and Waste Reduction System is a standalone web application for hostel campuses. It allows students to view menus, pre-book meals, and give feedback, while staff can track bookings, monitor wastage, and analyze trends. The system complements existing mess operations by improving efficiency, reducing food waste, and enhancing student satisfaction.

2.2 Product Functions

The major functions of the system are:

- User Management – Students, staff, and admins can register, login, and manage accounts.
- Menu Management – Staff/Admin can publish and update weekly meal menus.
- Meal Booking – Students can pre-book, modify, or cancel meals.
- Feedback System – Students can rate meals and provide comments.
- Booking Dashboard – Staff can view the total number of pre-booked meals.
- Waste Tracking – Staff can record daily leftovers.
- Analytics Reports – Admin can view graphs of bookings, feedback, and wastage trends.
- Menu Suggestions – Students can suggest new dishes; staff can review popularity.

2.3 User Classes and Characteristics

User Class	Characteristics
General User (Student)	Primary users of the system.Can view menus, pre-book meals, give feedback, and suggest dishes.Expected to have basic computer/mobile literacy.Access via web browser or mobile-friendly interface..
Administrator	Responsible for managing the weekly menu, viewing pre-booked counts, and recording food wastage.Can view feedback and generate reports for analysis.Moderate technical knowledge; the system should be simple, intuitive, and require minimal training

2.4 Operating Environment

- **Client-Side:** The system shall be accessible through modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. It will support both desktop and mobile devices, ensuring a responsive and user-friendly interface.
- **Server-Side:** The application will be developed using the MERN stack (MongoDB, Express.js, React, Node.js). It will be deployed on a cloud platform (e.g., Heroku or Vercel) for scalability and accessibility. The backend will handle authentication, booking logic, feedback collection, and data analytics, while the frontend will provide an interactive user experience.

2.5 Design and Implementation Constraints

- The project must be developed using open-source technologies to avoid licensing costs.
- The system must be developed following the Agile/Scrum framework as outlined in the course plan.
- All development must be version-controlled using Git and hosted on a shared GitHub repository.
- The user interface must be responsive and accessible on both desktop and mobile devices.
- The project must be completed within the 12-week semester timeline.

2.6 User Documentation

The following user documentation will be provided:

- An online FAQ page addressing common user questions.
- A `README.md` file in the GitHub repository detailing the project setup and deployment instructions.

2.7 Assumptions and Dependencies

- It is assumed that students will provide accurate meal bookings and honest feedback on food quality.
- It is assumed that mess staff/administrators will regularly update menus and record food wastage for proper analytics.
- The system depends on a stable internet connection for students and staff to access the application.
- The system depends on a cloud hosting provider (e.g., Heroku, Vercel) for deployment and availability.
- It is assumed that the university will support the adoption of this system by encouraging students to pre-book meals and participate in feedback.
- The system may depend on a third-party notification service (e.g., SendGrid API, Twilio) for sending reminders or alerts.

3 External Interface Requirements

3.1 User Interfaces

The system will feature a clean, modern, and intuitive graphical user interface (GUI). Key screens will include:

- **Homepage:** Displays the weekly menu, quick meal booking option, and announcements.
- **Login/Registration Pages:** For students and mess staff/administrators to authenticate securely.
- **Meal Booking Page:** A user-friendly interface for students to pre-book meals, modify bookings, or cancel before the cutoff.
- **Feedback Form Page:** Allows students to rate meals, add comments, and submit feedback easily.
- **Waste Tracking Page (Staff/Admin):** Enables staff to record leftover quantities after each meal'.
- **Analytics Reports Page (Staff/Admin)**: Provides charts/graphs showing booking trends, feedback ratings, and wastage statistics.
- **Menu Suggestion Page:** Lets students suggest new dishes and view popular requests.

3.2 Hardware Interfaces

No direct hardware interfaces are required for this web-based application.

3.3 Software Interfaces

The system will use MongoDB as its database, accessed through Node.js/Express APIs. The frontend, built with React.js, will run on modern web browsers and communicate with the backend via REST APIs. Authentication will be handled using JWT for secure role-based access. Optional integration with third-party services like SendGrid (emails) or Twilio (SMS) may be used for notifications. The entire application will be deployed on a cloud platform such as Heroku or Vercel for accessibility and scalability.

3.4 Communications Interfaces

All communication between the client (browser) and the server will use the standard HTTPS protocol to ensure data security.

4 System Features

4.1 System Feature 1: User Account Management

4.1.1 Description and Priority

Allows users to create and manage their accounts. Priority: **High**.

4.1.2 Stimulus/Response Sequences

- *Stimulus:* Student enters email address and Name and clicks Register.
- *Response:* The system creates an account and stores details in the database.
- *Stimulus:* Student enters email address and clicks Login.
- *Response:* The system authentication by email address and opens the student dashboard.
- *Stimulus:* Staff enters email address and clicks Login.
- *Response:* The system authenticates the staff account and redirects to the Staff Dashboard.
- *Stimulus:* Admin enters email address and clicks Login.
- *Response:* The system authenticates the admin account and redirects to the Admin Dashboard..
- *Stimulus:* Admin updates user roles (e.g., upgrade staff, revoke access).
- *Response:* The system modifies account permissions and updates them in the database..

4.1.3 Functional Requirements

`REQ-1` Register: Students enter email address and Name to create an account.

4.2 System Feature 2: Menu Display & Booking

4.2.1 Description and Priority

This feature allows students to view the weekly menu and pre-book meals in advance. The bookings help mess staff prepare accurate quantities of food, reducing wastage. Priority: **High**.

4.2.2 Stimulus/Response Sequences

- *Stimulus:* Student navigates to the menu page.
- *Response:* The system displays the weekly menu with meal details.
- *Stimulus:* Student selects meals for the week and clicks *Book*.
- *Response:* The system stores the bookings in the database and updates the total count for staff.
- *Stimulus:* Student modifies a meal booking before the cutoff time.
- *Response:* The system updates the booking details in the database.
- *Stimulus:* Staff requests a summary of pre-booked meals.
- *Response:* The system generates a report of meal counts for each day.

4.2.3 Functional Requirements

`REQ-1` Menu Display: Students must be able to view the weekly mess menu.

`REQ-2` Meal Booking: Students must be able to pre-book meals for the week/day.

4.3 System Feature 3: Feedback System

4.3.1 Description and Priority

This feature allows students to provide feedback on meals, including ratings and comments. Staff and admin can view aggregated feedback to improve food quality and service. matches. Priority: **High**.

4.3.2 Stimulus/Response Sequences

- *Stimulus:* Student submits a rating and optional comment for a meal.
- *Response:* The system stores the feedback in the database.
- *Stimulus:* Staff requests to view feedback summary.
- *Response:* The system displays average ratings and student comments.
- *Stimulus:* Admin views detailed reports of meal feedback.
- *Response:* The system provides charts showing rating distribution and common issues.

4.3.3 Functional Requirements

`REQ-1` Submit Feedback:Students must be able to submit ratings and comments for meals.

`REQ-3` Feedback Report: Admin must be able to view detailed reports with charts and trends.

4.4 System Feature 4: Waste Tracking & Analytics

4.4.1 Description and Priority

This feature allows mess staff to record leftover food quantities daily. The admin can analyze this data using reports and charts to reduce food wastage in the future. **Priority: High**

4.4.2 Stimulus/Response Sequences

- *Stimulus:* Staff enters daily leftover food quantity.
- *Response:* The system records the wastage data in the database.
- *Stimulus:* Admin requests wastage analytics.
- *Response:* The system generates graphs showing wastage trends over days/weeks.
- *Stimulus:* Admin compares meal bookings with actual wastage.
- *Response:* The system highlights patterns (e.g., meals with consistently high wastage).

4.4.3 Functional Requirements

`REQ-3` Booking-Waste Comaprison: The system must provide comparisons between booked meals and leftover food.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- Web pages shall have a load time of under 3 seconds on a standard campus Wi-Fi connection.
- The system should support at least 200 concurrent users without performance degradation.

5.2 Software Quality Attributes

- **Usability:** The interface must be intuitive and easy to navigate for non-technical users.
- **Maintainability:** The code must be well-structured, commented, and modular to facilitate future updates.

6 Other Requirements

There are no other requirements at this time.

A Glossary

Admin: A privileged user responsible for managing the system, including user roles, menu updates, and report generation.

Booking: The process by which a student reserves a meal in advance for a specific date and time.

Feedback: A rating or comment submitted by a student after consuming a meal, used to evaluate food quality and service.

Menu: A list of meals available for students to view and book within a given week.

Mess Staff: Personnel responsible for food preparation, menu management, and recording daily leftovers.

Pre-booking Cutoff: The time limit before a meal after which students can no longer book or modify their orders.

SmartMess: The proposed Mess Food Quality and Waste Reduction System that enables pre-booking, feedback collection, and data analytics for hostel mess management.

Student: The end-user of the system who can view menus, pre-book meals, and provide feedback.

Waste Report: A summary showing leftover quantities and food wastage trends, entered by mess staff and viewed by the admin.

SRS: Software Requirements Specification, a document describing the functional and non-functional requirements of the system.

B Analysis Models

This appendix presents the analysis models for the SmartMess system. These models illustrate the logical flow of data and the interaction between system components.

B.1 UML Diagrams

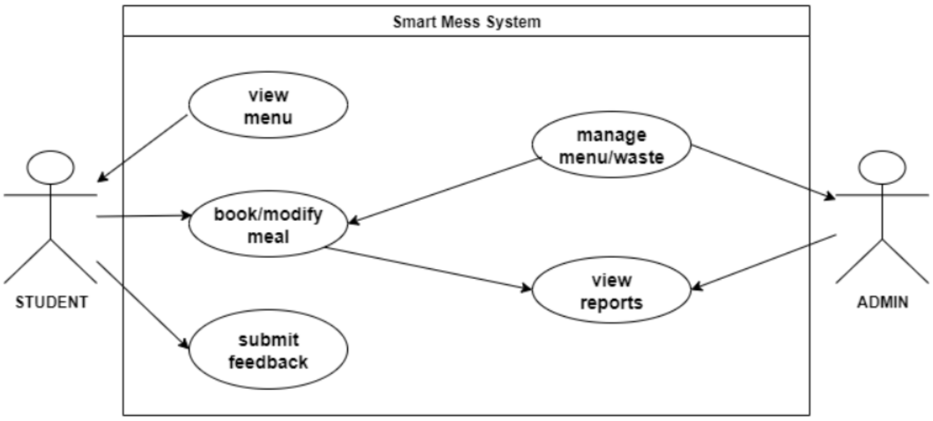

Figure 1: Use Case Diagram for Smart Mess.

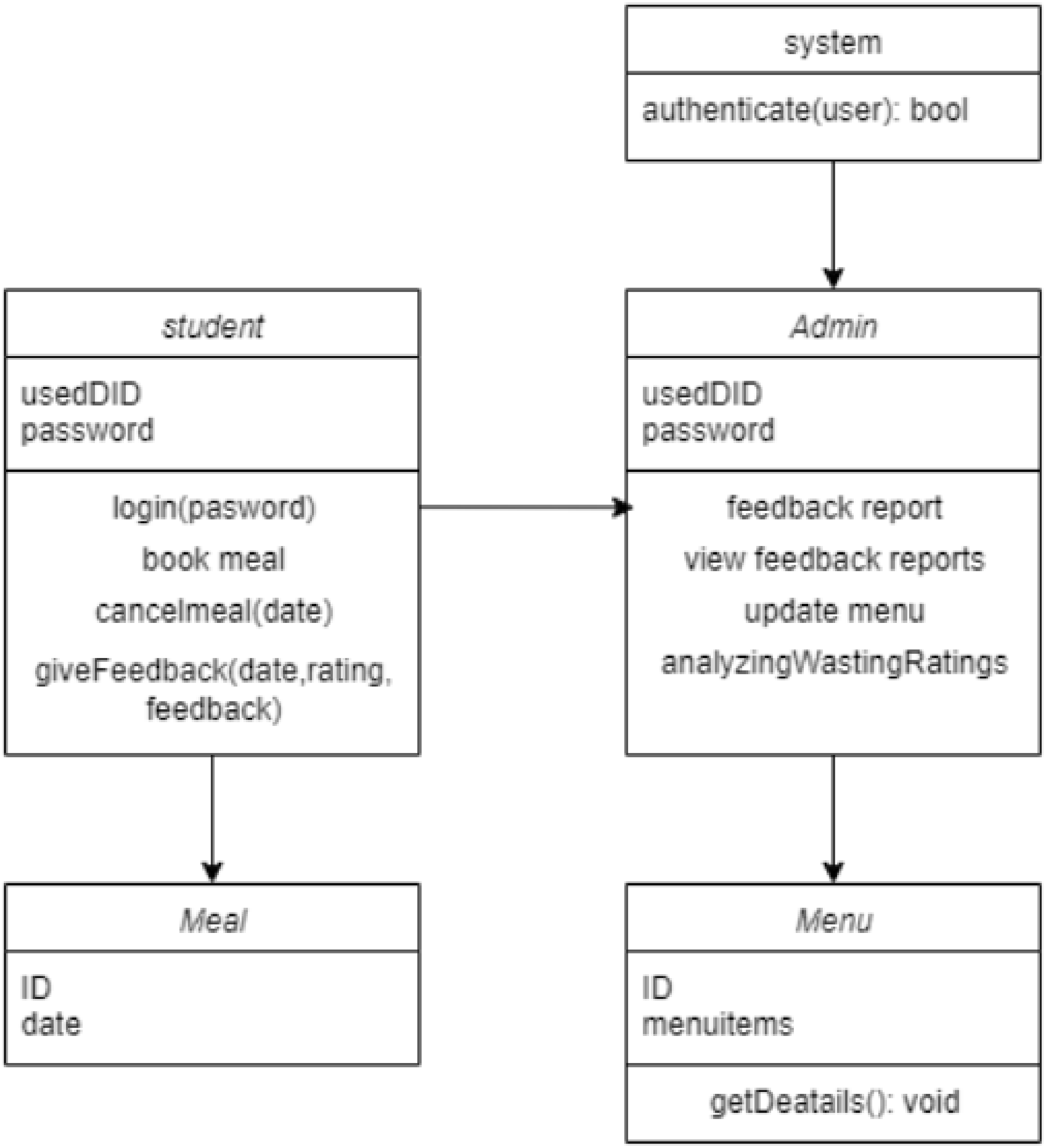

Figure 2: Class Diagram showing the main entities and their relationships.

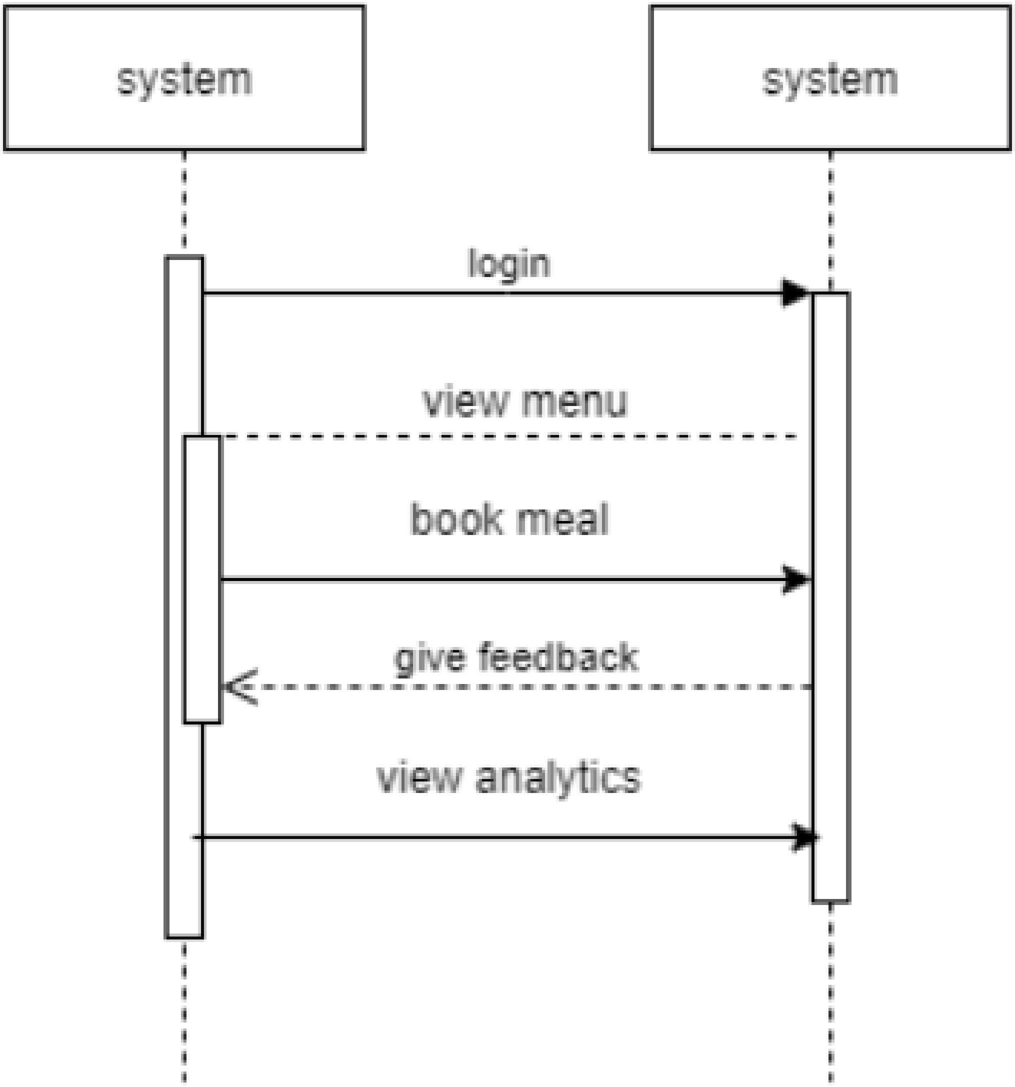

Figure 3: Sequence Diagram for the User and the system.

B.2 Data Flow Diagrams (DFDs)

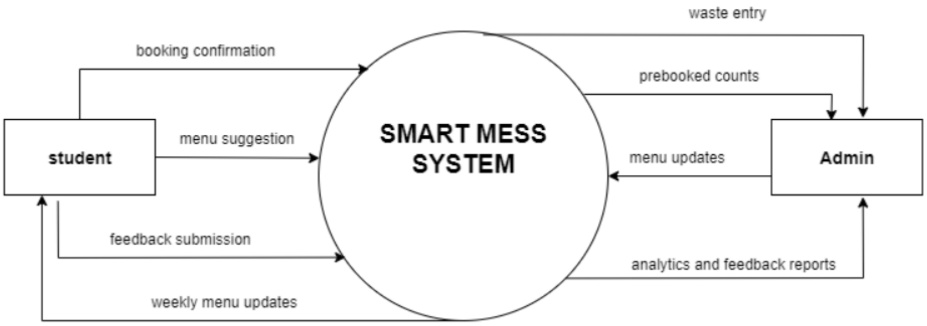

Figure 4: DFD Level 0 (Context Diagram) for the System.

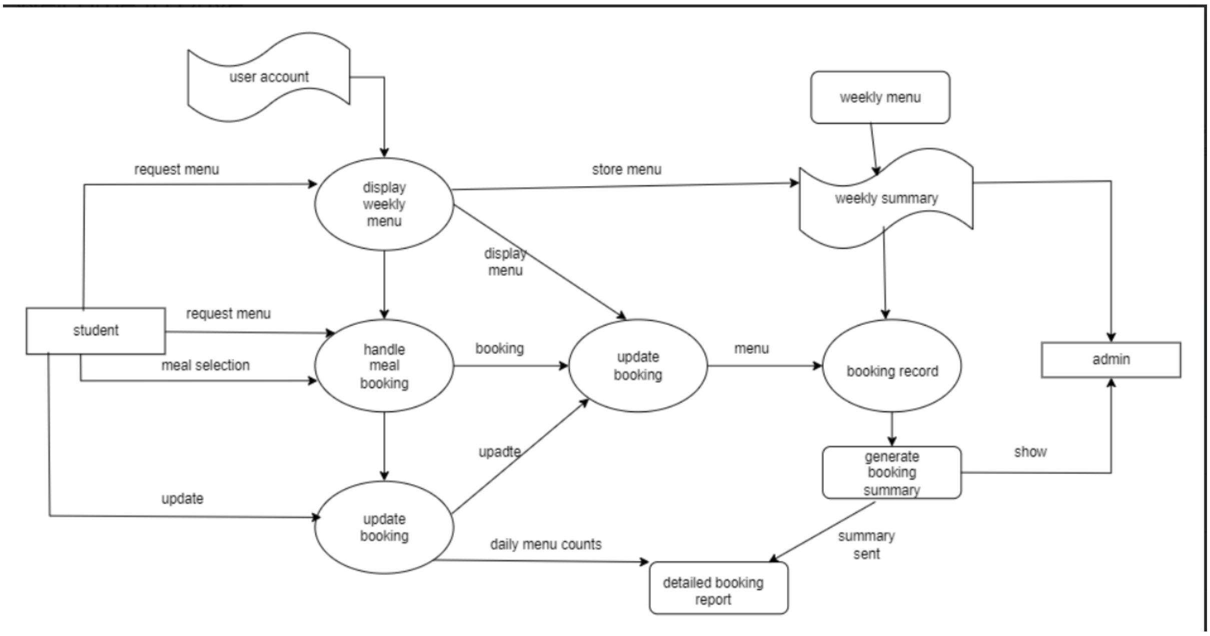

Figure 5: DFD Level 1 for the system.

C To Be Determined List

1. **TBD-1:** The exact process and criteria for verifying and granting Administrator and Mess Staff privileges (e.g., through college database, manual verification, or admin approval).

2. **TBD-2:** The cutoff time for daily meal pre-booking (e.g., 2 hours or 4 hours before meal time).

3. **TBD-3:** The specific formula or logic for calculating reward points for consistent meal pre-booking.

4. **TBD-4:** The format and frequency of analytics reports to be generated for mess management (daily, weekly, or monthly).

5. **TBD-5:** The exact scale (e.g., 1–5 or 1–10) and weighting system for feedback ratings related to food quality, hygiene, and quantity.

6. **TBD-6:** The method of data backup and recovery (e.g., automated daily backups or manual database export).

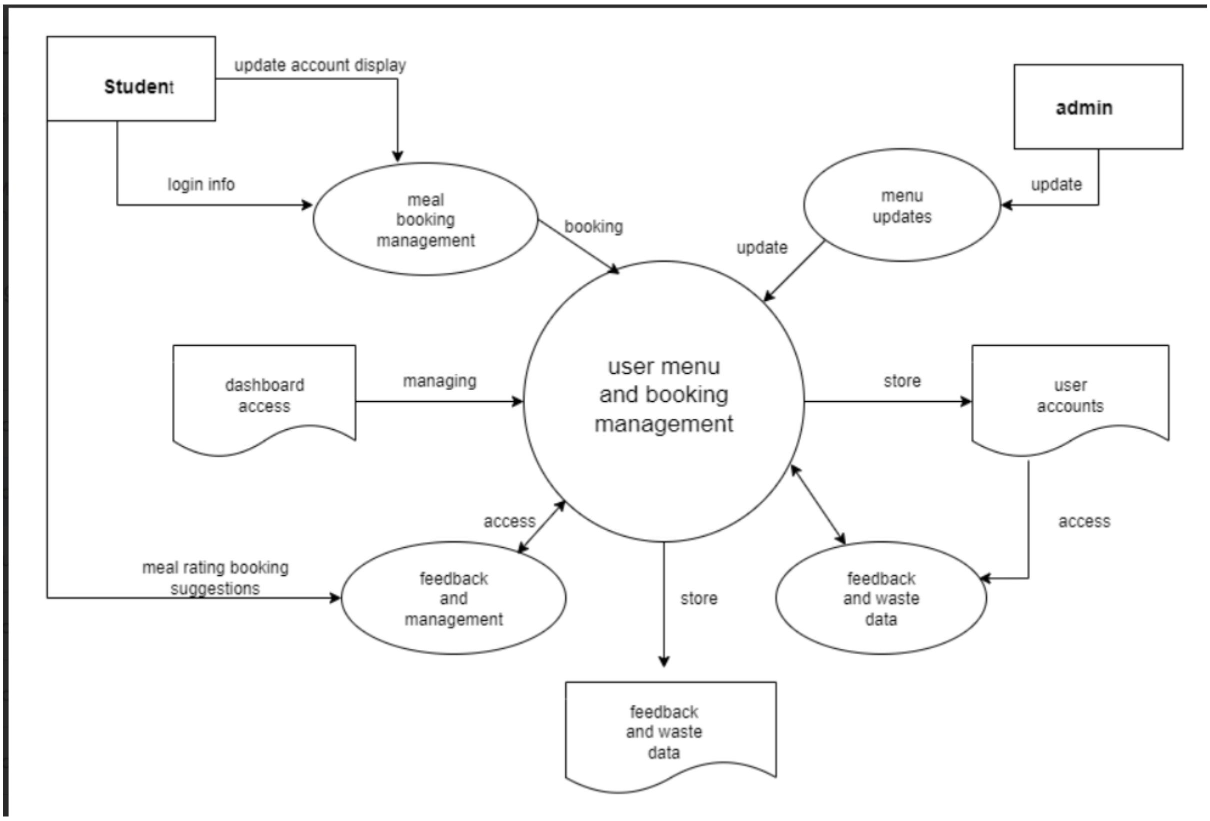

Figure 6: DFD Level 2 for Process 2.0 (booking items).

7. **TBD-7:** The future integration with payment or attendance systems (if extended to a full campus mess automation platform).