SOFTWARE REQUIREMENT SPECIFICATIONS

1 INTRODUCTION

Requirements specification is the first and crucial step in the software development process. It forms the foundation for all further development and validation activities. Software Requirements Specification (SRS) defines the client’s needs in detail, enabling effective communication between stakeholders including the client, users, and developers. The complexity arises because these three parties often view the system differently. Hence, a clear SRS helps align their goals and ensures that the product meets expectations.

2 PURPOSE

The need for Gusto Magis arose from the demand to digitize and simplify cafeteria operations within educational institutions. The system is developed to meet the needs of: the client (college administration),the users (students, teachers, and cafeteria staff), and the developer.

The primary aim of this SRS is to clearly document all expectations from the software system to avoid confusion and miscommunication. Benefits of a good SRS include:

- Acts as a contract between client and developer.
- Forms the basis for product validation.
- Reduces cost by minimizing redesign.
- Helps clarify the real needs of the users.

3 SCOPE

3.1 SYSTEM STATEMENT OF SCOPE

The GustoMagis system is an adaptive web-based cafeteria management platform built for Marian College. It offers role-based dashboards for students, teachers, and administrators to simplify and enhance the food ordering and dining experience. Users can browse menus, place food orders, make reservations, pre-book in bulk orders. Administrators can manage inventory, update menus, monitor orders

4 TECHNICAL OVERVIEW

4.1 USER CHARACTERISTICS

There are three user types:

Students/Teachers (Users): Can register, log in, view menu, place orders, reserve food .

Admin: Can manage food items, reservations, view analytics, and approve bulk orders.

All interfaces are built to be user-friendly and responsive, requiring basic digital literacy.

5 FUNCTIONAL REQUIREMENTS

1. **User Authentication**
   - Registration for students and teachers with role-based input
   - Email and password validation with proper input constraints
   - Secure login with session management
   - Logout functionality that destroys session and redirects user
2. **Menu Management**
   - Admin can add, edit, or delete food items
   - Food categorized under Breakfast, Lunch, Snacks, and Beverages
   - Items can have images, descriptions, and pricing
   - Option to mark items as "out of stock"
3. **Order Management**
   - Users (students/teachers) can place individual orders
   - System maintains quantity, price, and order timestamp
   - Orders are trackable by status: Pending, Preparing, Delivered
   - Admin dashboard for updating delivery status and viewing daily sales
4. **Bulk Order & Pre-Booking**
   - Teachers can place bulk orders for events or groups
   - Specify required items, quantity, and date/time
   - Admin panel for approval, modification, or rejection
   - Notifications to teachers on status updates
5. **Reservation System**
   - Table or meal reservations for specific time slots
   - Booking confirmation system with auto-cancellation of expired slots
   - Admin can view, approve, and manage reservations
   - Display of available reservation slots

6. **Purchase History**
    - Users can view order history including:
        - Item list
        - Total cost
        - Date & time
    - Useful for re-ordering and tracking spending
7. **Admin Dashboard**
    - Real-time display of new orders and reservations
    - Summary of total earnings, peak order times
    - Controls for menu management and user management
    - Approval interface for bulk orders and pre-bookings
8. **Notification System** *(Add-on)*
    - Email or in-platform alerts for order confirmation, status changes, or reservation updates
9. **Role-Based Access**
    - Student: Place orders, view history, reserve tables
    - Teacher: Same as student + bulk order access
    - Admin: Full access including content and user management

6 NON-FUNCTIONAL REQUIREMENTS

1. **Usability**
    - Clean, modern UI with consistent navigation
    - Responsive design for desktops, tablets, and smartphones
    - Proper contrast and font sizes for accessibility compliance
2. **Reliability**
    - System uptime target: ≥ 95% monthly
    - Automatic fallback pages for minor service interruptions

3. **Availability**
    - 24/7 access to platform for users and admins
    - Scheduled backup system to prevent data loss
4. **Security**
    - Password encryption using hashing algorithms
    - Role-based access controls to protect admin functionalities
    - SQL injection and CSRF protection
    - Secure session handling and logout protocols
5. **Performance**
    - Page load time: Under 2 seconds for standard views
    - Optimized SQL queries and use of indexes for speed
    - Lazy loading of images and minimal resource usage
6. **Scalability**
    - Database and UI designed to handle increased load
    - Support for adding more food categories or user roles
7. **Maintainability**
    - Modular folder structure (MVC pattern recommended)
    - Detailed code comments and documentation
    - Separate configuration files for environment variables
8. **Portability** *(Add-on)*
    - Platform can be deployed on Linux or Windows servers
    - Mobile-responsive for future app migration

7 STATED REQUIREMENTS

7.1 GENERAL REQUIREMENTS

Login

➢ Email and password required.

➢ Different dashboards based on user type (admin, student, teacher).

Sign Up

➢ Required fields: Name, Email, Phone, Password, Role (student/teacher).

➢ Password rules enforced (min 8 chars, mixed case, symbols).

Admin Panel

➢ Menu management, order monitoring, bulk order approvals.

Order & Reservation

➢ Users can place and cancel orders or reservations.

➢ Track current and past bookings.

Purchase History

➢ View completed and cancelled orders with full details.

7.2 INPUTS

- **User Registration**
    - ➢ Name
    - ➢ Email
    - ➢ Password
    - ➢ Phone
    - ➢ Role
- **Login**
    - ➢ Email
    - ➢ Password
- **Order Placement**
    - ➢ Selected Items(s)
    - ➢ Quantity
- **Reservation/Booking**
    - ➢ Date
    - ➢ Time
    - ➢ Food Category
- **Bulk Order**
    - ➢ List of Items

  - Quantity
  - Delivery Date

7.3 PROCESSING

Validation

- Ensures correct email format, password strength, and no empty fields during user inputs
- Prevents SQL injection and enforces required field constraints

Order Management

- Deducts stock quantity upon successful order placement
- Calculates item-wise subtotal and final billing amount (including taxes, if any)

Reservation Logic

- Checks for slot availability before confirming reservations
- Confirms bookings and prevents double reservations for the same time slot

Admin Actions

- Allows admin to add, update, or delete menu items
- Approves or rejects bulk orders and table reservations
- Edits user roles and reviews real-time order activity

7.4 OUTPUTS

Order Confirmation

- Displays receipt with food items, quantity, price per item, total amount, and expected delivery/reservation time

Reservation Details

- Shows confirmation including date, time, reserved items or seats, and user details

Bulk Order Status

- Indicates whether the order is *Approved*, *Rejected*, or *Pending*
- Admin can provide comments for transparency

Purchase History

- Tabular report with:
    - Date of transaction
    - List of items
    - Total amount
    - Order/reservation status

8 EXTERNAL INTERFACE REQUIREMENTS

8.1 USER INTERFACES

User Dashboard

- Access to daily menus, order placement, reservation booking, and purchase history

Admin Panel

- Tools to manage food listings, monitor real-time orders, handle reservations, and bulk orders

Design

- Clean, modern interface with a consistent color scheme
- Use of icons and labeled buttons for intuitive navigation

Responsiveness

- Fully optimized for:
    - Desktop (Chrome, Firefox, Edge)
    - Tablets (iPad, Android)
    - Mobile Devices (iOS, Android)

8.2 HARDWARE INTERFACES

The system can be accessed on any smart device that meets the following minimum specs:

Component	**Specification**
Processor	Intel Pentium or higher

Component	Specification
RAM	512 MB or higher
Disk Space	Minimum 100 MB
Network	Stable internet connection

8.3 SOFTWARE INTERFACES

Component	Technology Used
Operating System	Windows 10/11, Linux
Backend	PHP (v7.4 or above)
Frontend	HTML5, CSS3, JavaScript
Database	MySQL
Web Server	Apache (XAMPP or LAMP stack)

UML DIAGRAMS

USE CASES

1. STUDENT LOGIN

Use Case Id:	GM_UC_01
Use Case Name:	Student Login
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows students to log into the GustoMagis cafeteria system to access menu and ordering features.
Primary actor:	Student
Secondary actor:	None
Precondition:	Student has a valid account and navigates to the login page.
Postcondition:	Student is successfully logged in and redirected to the main dashboard.
Main flow:	1. Student navigates to the login page. 2. Student enters email and password. 3. Student submits the login form. 4. System validates credentials. 5. If valid, student is redirected to dashboard. 6. Use case ends.

2. FACULTY LOGIN

Use Case Id:	GM_UC_02
Use Case Name:	Faculty Login
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows faculty members to log into the GustoMagis system to access enhanced features including pre-booking and bulk orders.
Primary actor:	Faculty
Secondary actor:	None
Precondition:	Faculty member has a valid account with appropriate role permissions.

Postcondition:	Faculty is logged in and redirected to the faculty dashboard with enhanced features.
Main flow:	1. Faculty navigates to the login page. 2. Faculty enters email and password. 3. Faculty submits the login form. 4. System validates credentials and role. 5. If valid, faculty is redirected to faculty dashboard. 6. Use case ends.

3. ADMIN LOGIN

Use Case Id:	GM_UC_03
Use Case Name:	Admin Login
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows administrators to access the admin panel for complete system management and control.
Primary actor:	Admin
Secondary actor:	None
Precondition:	Admin has valid credentials with administrative privileges.
Postcondition:	Admin is logged in and has access to all administrative functions.
Main flow:	1. Admin navigates to the admin login page. 2. Admin enters administrative credentials. 3. Admin submits the login form. 4. System validates admin privileges. 5. If valid, admin is redirected to admin dashboard. 6. Use case ends.

4. VIEW DAILY MENU

Use Case Id:	GM_UC_04

Use Case Name:	View Daily Menu
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows users to browse the daily menu with all available food items, prices, and descriptions.
Primary actor:	Student/Faculty
Secondary actor:	None
Precondition:	User must be logged into the system.
Postcondition:	Daily menu is displayed with current availability status.
Main flow:	1. User navigates to the menu section. 2. System displays daily menu with categories. 3. User can view item details, prices, and images. 4. System shows availability status for each item. 5. Use case ends.

5. SORT FOOD BY CATEGORY

Use Case Id:	GM_UC_05
Use Case Name:	Sort Food by Category
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows users to filter and sort food items by different categories such as vegetarian, non-vegetarian, beverages, snacks, etc.
Primary actor:	Student/Faculty
Secondary actor:	None
Precondition:	User must be logged in and viewing the menu.
Postcondition:	Menu items are filtered and displayed according to selected category.

Main flow:	1. User accesses the menu page. 2. User selects a category filter option. 3. System filters and displays items from selected category. 4. User can browse filtered items. 5. User can change or clear filters as needed. 6. Use case ends.

6. VIEW DAILY SPECIALS

Use Case Id:	GM_UC_06
Use Case Name:	View Daily Specials
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows students to explore special dishes or offers available for the current day.
Primary actor:	Student
Secondary actor:	None
Precondition:	Student must be logged in and daily specials must be available.
Postcondition:	Daily specials are displayed with special prices and descriptions.
Main flow:	1. Student navigates to daily specials section. 2. System displays current day's special items. 3. Student can view special pricing and descriptions. 4. Student can add specials to cart if interested. 5. Use case ends.

7. MAKE TABLE RESERVATION

Use Case Id:	GM_UC_07
Use Case Name:	Make Table Reservation
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows students to reserve tables in the cafeteria for dining.

Primary actor:	Student
Secondary actor:	None
Precondition:	Student must be logged in and tables must be available.
Postcondition:	Table reservation is confirmed and recorded in the system.
Main flow:	1. Student navigates to table reservation section. 2. System displays available tables and time slots. 3. Student selects preferred table and time. 4. Student confirms the reservation. 5. System records the reservation and sends confirmation. 6. Use case ends.

8. WEEKLY/MONTHLY PRE-BOOKING

Use Case Id:	GM_UC_08
Use Case Name:	Weekly/Monthly Pre-booking
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows faculty to pre-book meals for weekly or monthly periods with advance planning.
Primary actor:	Faculty
Secondary actor:	None
Precondition:	Faculty member must be logged in and have appropriate permissions.
Postcondition:	Pre-booking is confirmed for the selected period with meal preferences saved.

Main flow:	1. Faculty navigates to pre-booking section. 2. Faculty selects weekly or monthly option. 3. Faculty chooses date range and meal preferences. 4. Faculty confirms the pre-booking. 5. System schedules the meals and sends confirmation. 6. Use case ends.

9. STAFFROOM DELIVERY

Use Case Id:	GM_UC_09
Use Case Name:	Staffroom Delivery
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows faculty to request food delivery directly to their staffroom.
Primary actor:	Faculty
Secondary actor:	Delivery Staff
Precondition:	Faculty must be logged in and have items in cart.
Postcondition:	Delivery request is processed and delivery is scheduled.
Main flow:	1. Faculty adds items to cart. 2. Faculty selects delivery option during checkout. 3. Faculty provides staffroom location details. 4. Faculty confirms the delivery order. 5. System assigns delivery staff and schedules delivery. 6. Faculty receives delivery confirmation. 7. Use case ends.

10. BULK ORDER FOR EVENTS

Use Case Id:	GM_UC_10

Use Case Name:	Bulk Order for Events
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows faculty to place large quantity orders for departmental events or meetings.
Primary actor:	Faculty
Secondary actor:	Admin
Precondition:	Faculty must be logged in and event details must be provided.
Postcondition:	Bulk order is placed and approved by admin for event fulfillment.
Main flow:	1. Faculty navigates to bulk order section. 2. Faculty enters event details (date, attendees, location). 3. Faculty selects menu items and quantities. 4. Faculty submits bulk order request. 5. System notifies admin for approval. 6. Admin reviews and approves the order. 7. Faculty receives confirmation. 8. Use case ends.

11. ADD TO CART

Use Case Id:	GM_UC_11
Use Case Name:	Add to Cart
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows users to add selected food items to their shopping cart for later checkout.
Primary actor:	Student/Faculty
Secondary actor:	None
Precondition:	User must be logged in and browsing available menu items.
Postcondition:	Selected items are added to user's cart with quantities and total price updated.

Main flow:	1. User browses menu items. 2. User selects desired food item. 3. User specifies quantity. 4. User clicks "Add to Cart" button. 5. System adds item to cart and updates total. 6. System displays cart confirmation. 7. Use case ends.

12. CHECKOUT

Use Case Id:	GM_UC_12
Use Case Name:	Checkout
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows users to review their cart, provide payment information, and complete their food order.
Primary actor:	Student/Faculty
Secondary actor:	Payment System
Precondition:	User must be logged in and have items in their cart.
Postcondition:	Order is successfully placed and confirmation is provided to the user.
Main flow:	1. User reviews items in cart. 2. User proceeds to checkout. 3. User selects pickup/delivery options. 4. User enters any special instructions. 5. User selects payment method. 6. User confirms the order and payment. 7. System processes payment and creates order. 8. User receives order confirmation with details. 9. Use case ends.

13. RATE AND REVIEW FOOD

Use Case Id:	GM_UC_13
Use Case Name:	Rate and Review Food
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows students and faculty to rate food items and write reviews after consumption.
Primary actor:	Student/Faculty
Secondary actor:	None
Precondition:	User must be logged in and have completed an order containing the food item.
Postcondition:	Rating and review are saved and displayed for other users to see.
Main flow:	1. User navigates to order history. 2. User selects a completed order to review. 3. User selects specific food items to rate. 4. User provides rating (1-5 stars) and written review. 5. User submits the review. 6. System saves the review and updates item ratings. 7. Review becomes visible to other users. 8. Use case ends.

14. MANAGE FOOD ITEMS

Use Case Id:	GM_UC_14
Use Case Name:	Manage Food Items
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows admin to add, edit, delete, and manage all food items in the cafeteria system.
Primary actor:	Admin
Secondary actor:	None
Precondition:	Admin must be logged into the system with appropriate permissions.
Postcondition:	Food item database is updated with changes reflected in the menu.

Main flow:	1. Admin navigates to food management section. 2. Admin selects add, edit, or delete option. 3. Admin enters or modifies item details (name, price, description, image). 4. Admin sets availability status and category. 5. Admin submits the changes. 6. System updates the food database. 7. Changes are reflected in user menus. 8. Use case ends.

15. UPDATE FOOD AVAILABILITY

Use Case Id:	GM_UC_15
Use Case Name:	Update Food Availability
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows admin to quickly update the availability status of food items based on stock levels.
Primary actor:	Admin
Secondary actor:	None
Precondition:	Admin must be logged into the system with appropriate permissions.
Postcondition:	Food availability status is updated and reflected to all users.
Main flow:	1.Admin navigates to food management section. 2. Admin selects add, edit, or delete option. 3. Admin enters or modifies item details (name, price, description, image). 4. Admin sets availability status and category. 5. Admin submits the changes.

	6. System updates the food database. 7. Changes are reflected in user menus. 8. Use case ends.

16. VIEW INVENTORY STATUS

Use Case Id:	GM_UC_16
Use Case Name:	View Inventory Status
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows admin to monitor current inventory levels and stock status of all food items.
Primary actor:	Admin
Secondary actor:	None
Precondition:	Admin must be logged in with inventory access permissions.
Postcondition:	Current inventory status is displayed with alerts for low stock items.
Main flow:	1. Admin navigates to inventory section. 2. System displays current stock levels for all items. 3. Admin can view detailed inventory reports. 4. System highlights low stock or out-of-stock items. 5. Admin can update stock quantities if needed. 6. Use case ends.

17. VIEW ORDER STATUS

Use Case Id:	GM_UC_17
Use Case Name:	View Order Status
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025

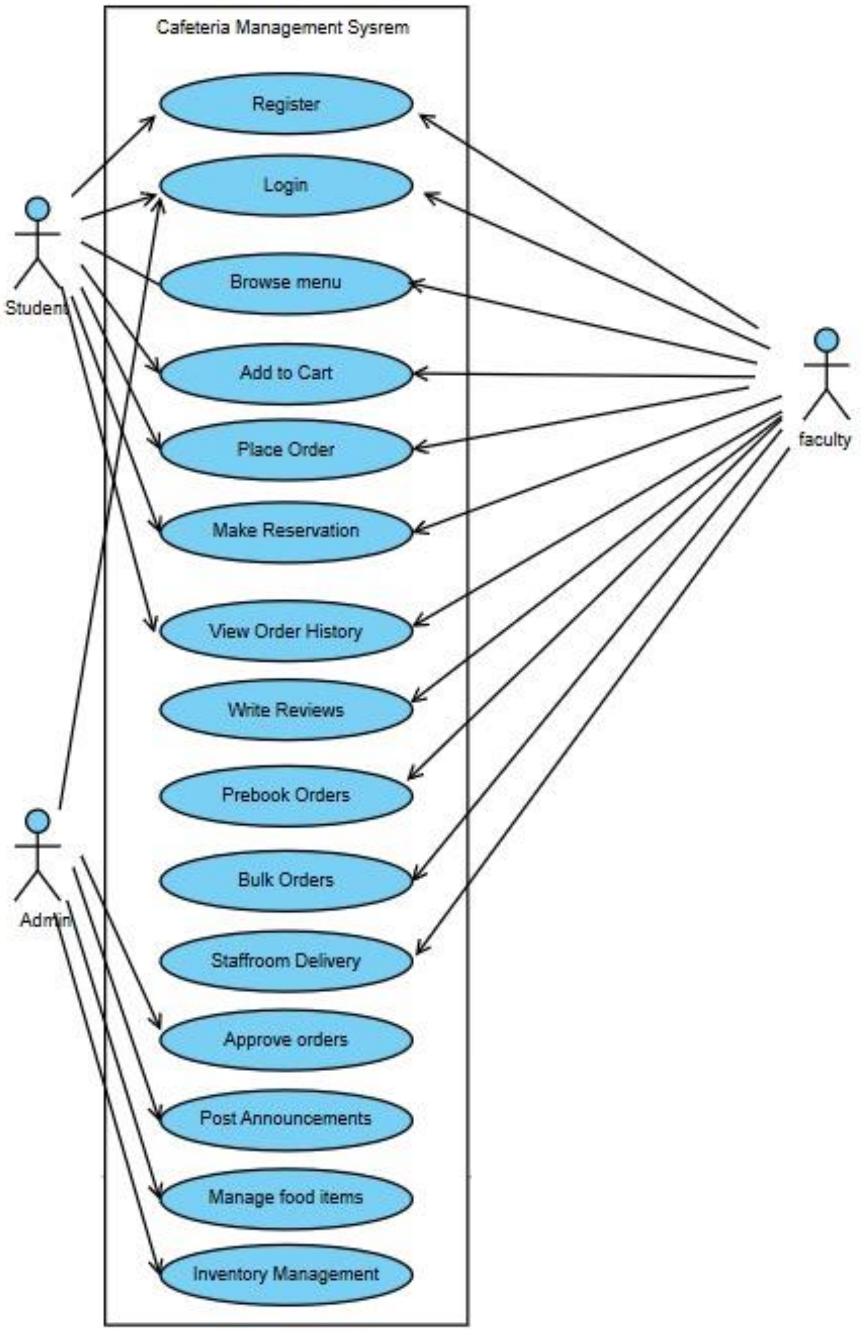
Description:	Allows admin to monitor all orders in the system and track their progress from placement to completion
Primary actor:	Admin
Secondary actor:	Ktichen Staff
Precondition:	Admin must be logged in with order management permissions.
Postcondition:	Current status of all orders is displayed with ability to update order progress.
Main flow:	1. Admin navigates to order management section. 2. System displays all active orders with current status. 3. Admin can filter orders by status, date, or user type. 4. Admin can view detailed order information. 5. Admin can update order status (preparing, ready, delivered). 6. System notifies users of status changes. 7. Use case ends.

18. SEND UPDATES

Use Case Id:	GM_UC_18
Use Case Name:	Send Updates
Created by:	Sreebhavya M & Rose Mary Daris
Date Created:	30-06-2025
Description:	Allows admin to send announcements, notifications, and updates to all system users.
Primary actor:	Admin
Secondary actor:	Students/Faculty
Precondition:	Admin must be logged in with communication permissions.
Postcondition:	Updates are sent to all users and displayed in their dashboards.

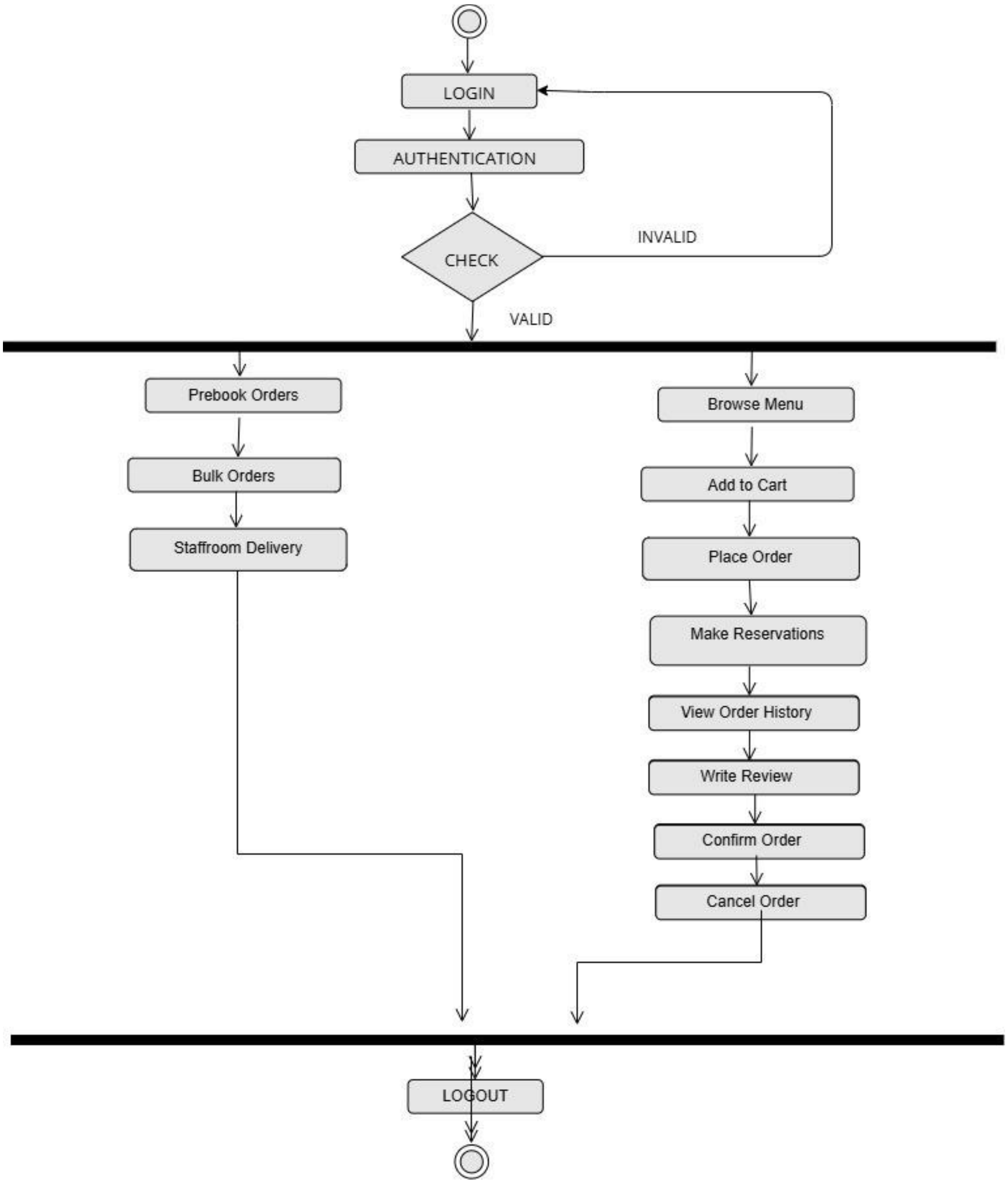
Main flow:	1. Admin navigates to communication section. 2. Admin selects update type (announcement, menu change, etc.). 3. Admin composes the message content. 4. Admin selects target audience (all users, students only, faculty only). 5. Admin schedules or sends the update immediately. 6. System delivers notifications to selected users. 7. Use case ends.

USECASE DIAGRAM

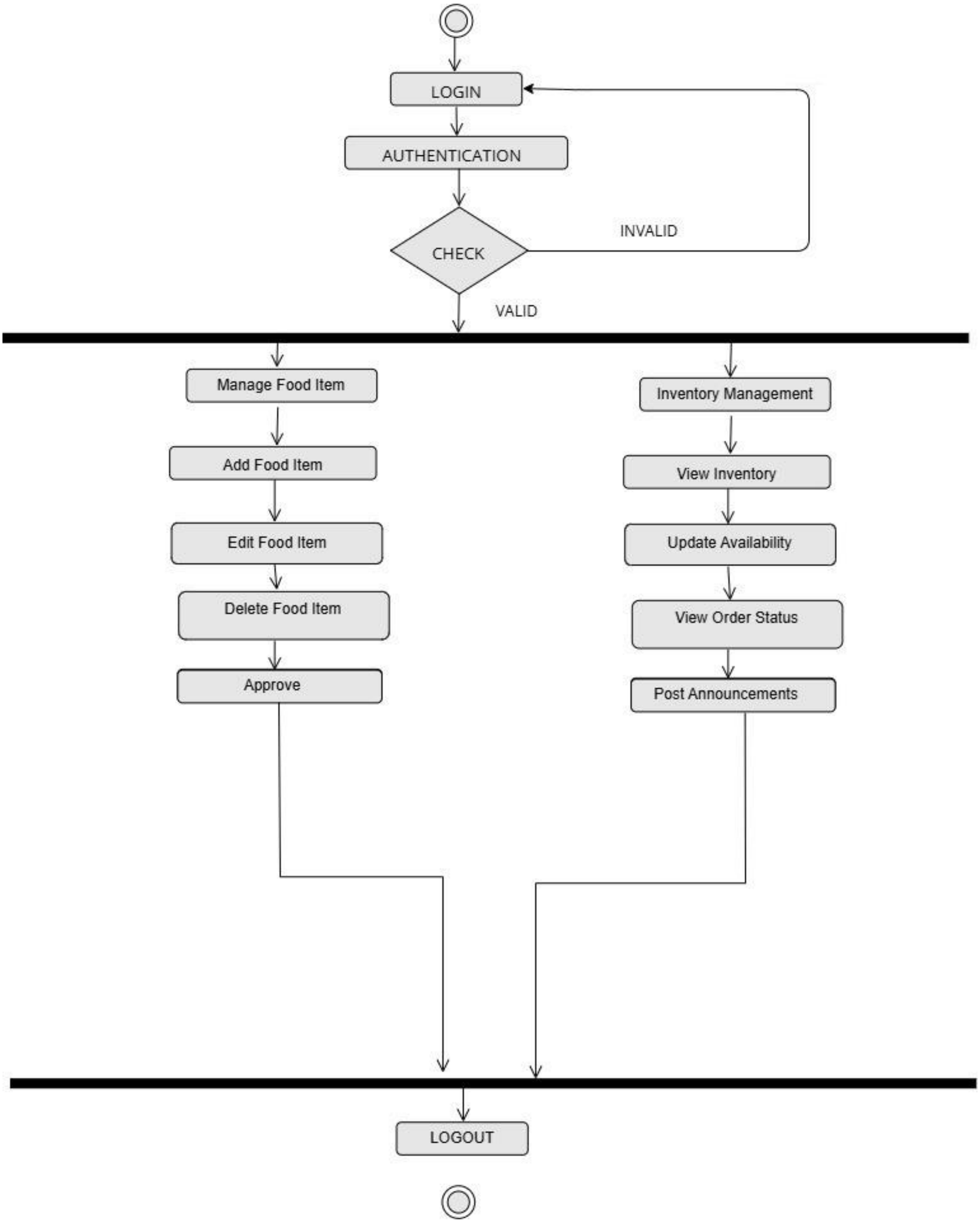


ACTIVITY DIAGRAMS

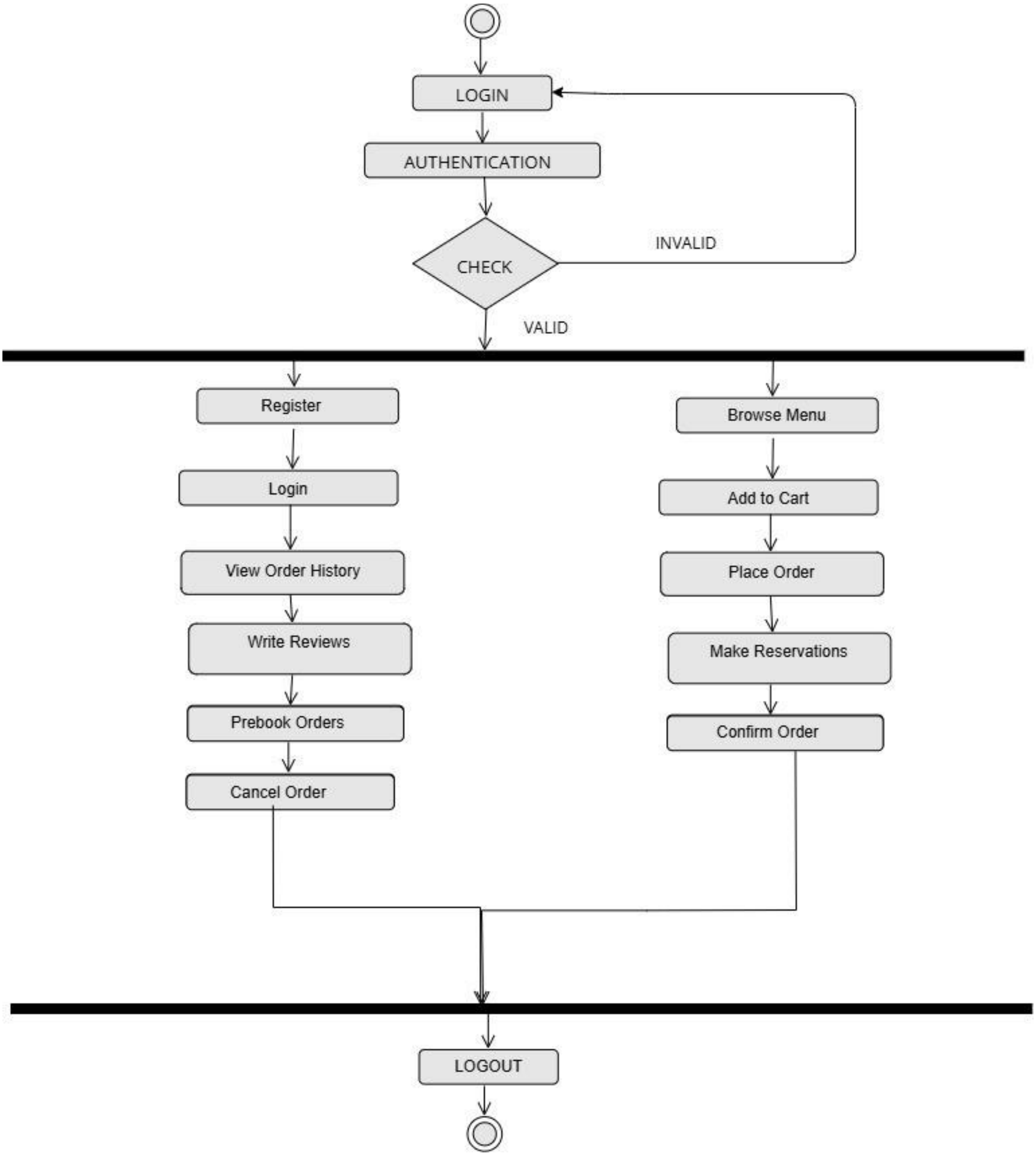
1. Faculty



2. Admin


LOGIN
AUTHENTICATION
CHECK
INVALID
VALID
Manage Food Item
Add Food Item
Edit Food Item
Delete Food Item
Approve
Inventory Management
View Inventory
Update Availability
View Order Status
Post Announcements
LOGOUT

3. Student


LOGIN
AUTHENTICATION
CHECK
INVALID
VALID
Register
Login
View Order History
Write Reviews
Prebook Orders
Cancel Order
Browse Menu
Add to Cart
Place Order
Make Reservations
Confirm Order
LOGOUT

DATABASE DESIGN

1. USERS

Field	**Type**	**Constraints**	**Description**
user_id	INT	PK, AUTO_INCREMENT	Unique user identifier
username	VARCHAR(50)	UNIQUE, NOT NULL	Login username
email	VARCHAR(100)	UNIQUE, NOT NULL	Email address
password_hash	VARCHAR(255)	NOT NULL	Hashed password
role	ENUM(‘student’,‘faculty’,‘admin’)	NOT NULL	User role
full_name	VARCHAR(100)	NOT NULL	Full name
phone	VARCHAR(15)	NULL	Phone number
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Timestamp of creation

2. **FACULTY_PROFILES**

Field	**Type**	**Constraints**	**Description**
faculty_id	INT	PK, AUTO_INCREMENT	Unique faculty profile ID
user_id	INT	FK → users(user_id), UNIQUE, NOT NULL	Linked user
department	VARCHAR(100)	NOT NULL	Department name
designation	VARCHAR(100)	NULL	Job title
staffroom_number	VARCHAR(20)	NULL	Room number
employee_id	VARCHAR(50)	UNIQUE, NULL	Employee ID

3. **ACCOUNCEMENTS**

Field	**Type**	**Constraints**	**Description**
announcement_id	INT	PK, AUTO_INCREMENT	Announcement ID
admin_id	INT	FK → users(user_id), NOT NULL	Issuing admin
title	VARCHAR(200)	NOT NULL	Title

Field	Type	Constraints	Description
content	TEXT	NOT NULL	Message content
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Creation timestamp
is_active	TINYINT(1)	DEFAULT 1	Status flag

4. **FOOD_CATEGORIES**

Field	Type	Constraints	Description
category_id	INT	PK, AUTO_INCREMENT	Category ID
category_name	VARCHAR(50)	UNIQUE, NOT NULL	Category title
description	TEXT	NULL	Description
is_active	TINYINT(1)	DEFAULT 1	Status flag

5. **FOOD_ITEMS**

Field	Type	Constraints	Description
item_id	INT	PK, AUTO_INCREMENT	Food item ID
category_id	INT	FK → food_categories(category_id)	Category
item_name	VARCHAR(100)	NOT NULL	Name
description	TEXT	NULL	Description
price	DECIMAL(8,2)	NOT NULL	Price
image_url	VARCHAR(255)	NULL	Image path
is_available	TINYINT(1)	DEFAULT 1	Availability
is_special	TINYINT(1)	DEFAULT 0	Special flag
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Created timestamp
updated_at	TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Last updated

6. **INVENTORY**

Field	Type	Constraints	Description
inventory_id	INT	PK, AUTO_INCREMENT	Inventory ID
item_id	INT	FK → food_items(item_id)	Food item

Field	Type	Constraints	Description
stock_quantity	INT	DEFAULT 0	Quantity in stock
last_updated	TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP	Last update

7. **ORDERS**

Field	Type	Constraints	Description
order_id	INT	PK, AUTO_INCREMENT	Order ID
user_id	INT	FK → users(user_id)	Who ordered
order_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Order time
total_amount	DECIMAL(10,2)	NOT NULL	Total bill
status	ENUM(…)	DEFAULT ‘pending’	Order status
order_type	ENUM(‘regular’,‘prebook’,‘bulk’)	DEFAULT ‘regular’	Type
delivery_location	VARCHAR(100)	NULL	Location
special_instructions	TEXT	NULL	Notes

8. **ORDER_ITEMS**

Field	Type	Constraints	Description
order_item_id	INT	PK, AUTO_INCREMENT	Order item ID
order_id	INT	FK → orders(order_id)	Linked order
item_id	INT	FK → food_items(item_id)	Food item
quantity	INT	DEFAULT 1	Quantity ordered
unit_price	DECIMAL(8,2)	NOT NULL	Per item price
subtotal	DECIMAL(10,2)	NOT NULL	Line total

9. **PREBOOK_ORDERS**

Field	Type	Constraints	Description
prebook_id	INT	PK, AUTO_INCREMENT	Prebook ID
faculty_id	INT	FK → users(user_id)	Ordering faculty
item_id	INT	FK → food_items(item_id)	Item
quantity	INT	DEFAULT 1	Quantity
prebook_date	DATE	NOT NULL	Booking date
delivery_time	TIME	NULL	Delivery time
status	ENUM(…)	DEFAULT 'scheduled'	Booking status
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time placed

10. **BULK_ORDERS**

Field	Type	Constraints	Description
bulk_order_id	INT	PK, AUTO_INCREMENT	Bulk order ID
faculty_id	INT	FK → users(user_id)	Requester
event_name	VARCHAR(255)	NULL	Event name
event_date	DATE	NOT NULL	Date of event
total_guests	INT	NOT NULL	Guest count
special_menu	TEXT	NULL	Notes
status	ENUM(…)	DEFAULT 'requested'	Status
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time created

11. **BULK_ORDER_ITEMS**

Field	Type	Constraints	Description
bulk_order_item_id	INT	PK, AUTO_INCREMENT	Line ID
bulk_order_id	INT	FK → bulk_orders(bulk_order_id)	Parent order
item_id	INT	FK → food_items(item_id)	Food item
quantity	INT	NOT NULL	Quantity needed

12. **DELIVERY_REQUESTS**

Field	Type	Constraints	Description
delivery_id	INT	PK, AUTO_INCREMENT	Delivery ID

Field	Type	Constraints	Description
faculty_id	INT	FK → users(user_id)	Requested by
order_id	INT	FK → orders(order_id)	Linked order
delivery_location	VARCHAR(100)	NOT NULL	Address
preferred_time	TIME	NULL	Time slot
status	ENUM(…)	DEFAULT ‘requested’	Status
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time created

13. **RESERVATIONS**

Field	Type	Constraints	Description
reservation_id	INT	PK, AUTO_INCREMENT	Booking ID
user_id	INT	FK → users(user_id)	Booker
table_number	INT	NOT NULL	Table assigned
reservation_date	DATE	NOT NULL	Date
reservation_time	TIME	NOT NULL	Time
party_size	INT	DEFAULT 1	People count
status	ENUM(…)	DEFAULT ‘pending’	Status
special_requests	TEXT	NULL	Notes
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time created

14. **REVIEWS**

Field	Type	Constraints	Description
review_id	INT	PK, AUTO_INCREMENT	Review ID
user_id	INT	FK → users(user_id)	Reviewer
item_id	INT	FK → food_items(item_id)	Food item
rating	INT	CHECK (1–5)	Rating
review_text	TEXT	NULL	Optional feedback
review_date	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Time given