IT314 Software Engineering

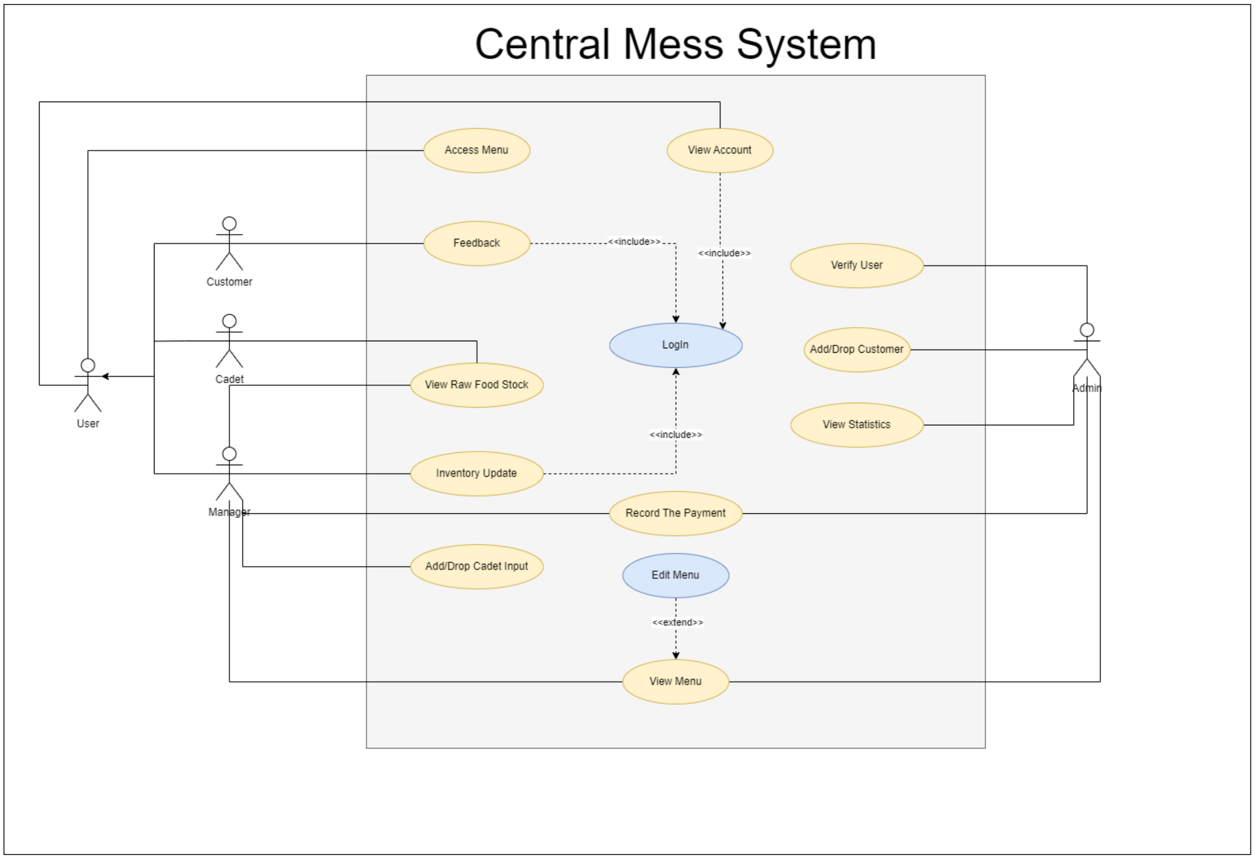
Lab-3 Report

Central Mess Management Software
Group: G32

Members:

202001444	PARMAR GAURANG GAJENDRABHAI
202001454	KANANI DEEP MUKESHBHAI
202001075	RAJPUT VRUND ASHWINKUMAR
202001418	RAHIL HARDIK SHUKLA
202001053	KRISH GULABANI
202001443	LIMBACHIYA OM
202001232	PRAJAPATI PARTH KANUBHAI
202001427	MOHAL RAJYAGURU
202001435	MEHTA NEELKUMAR HIRENKUMAR
202001451	VIVEK GODHASARA
202001417	DHANANJAY A VORA

Use-case diagram:



Actor:

1. Customer
2. Cadet
3. Manager
4. Admin

Use-Case:

1. Login
2. Feedback
3. View Inventory
4. Inventory Updation
5. Add/drop cadet information
6. View menu
7. Edit menu
8. Verify user
9. Record payment
10. View statistics
11. Add/drop customer
12. Authenticate user

User Stories:

Customer:

1. **Front of the card (User Story):**
   As a customer, I want to be able to login to the software, so that I can view and manage my account.

   Back of the card (Functional Requirements):

   The system should provide a login form for customers to enter their username and password.
   The system should validate the customer's user ID and password and allow access to the customer's account.
   The customer should be able to view their previous billing history.
   The customer should be able to update their account information (name, email, password, etc.).

   Back of the card (Non-Functional Requirements):

   The login form should be secure and protected against unauthorized access.
   The system should have a strong password policy to ensure the security of customer accounts.
   The system should be available 24/7 for customers to access their accounts.
   The system should provide a quick response to customer login requests.

2. **Front of the card (User Story):**
   As a customer, I want to be able to provide feedback about the food quality and service in the canteen, so that the management can improve the quality of food and service.

   Back of the card (Functional Requirements):
   The system should allow customers to provide feedback through a feedback form.
   The feedback form should contain fields for the Food review, Cleanliness review, overall experience and comments.
   The system should store the feedback in a database for future reference.

   Back of the card (Non-Functional Requirements):
   The feedback form should be easy to use and user-friendly.
   The system should ensure the privacy and security of customer information.
   The system should be available 24/7 for customers to provide feedback.
   The system should provide a quick response to customer feedback.

Cadet:

1. **Front of the card (User Story):**
   As a cadet, I want to be able to view the raw food stock, so that I can monitor the inventory levels and make informed decisions about ordering and menu planning.

   Back of the card (Functional Requirements):
   The system should display a list of all raw food items in stock, along with their quantity and expiry dates.

   Back of the card (Non-Functional Requirements):
   The system should be easy to use and intuitive for the cadet.
   The system should have a fast and responsive interface to allow for quick searches.
   The system should ensure the privacy and security of inventory information.
   The system should be available 24/7 for the cadet to access inventory information.

Manager:

1. **Front of the card (User Story):**
   As a manager, I want to be able to view the raw food stock, so that I can monitor the inventory levels and make informed decisions about ordering and menu planning.

   Back of the card (Functional Requirements):
   The system should display a list of all raw food items in stock, along with their quantity and expiry dates.

   Back of the card (Non-Functional Requirements):
   The system should be easy to use and intuitive for the manager.
   The system should have a fast and responsive interface to allow for quick searches.
   The system should ensure the privacy and security of inventory information.
   The system should be available 24/7 for the manager to access inventory information.

2. **Front of the card (User Story):**
   As a manager, I want to be able to update the inventory, so that the stock levels are accurate and up-to-date.

   Back of the card (Functional Requirements):
   The system should allow the manager to add new ingredients to the inventory, along with their quantity and expiry date.
   The system should allow the manager to update the quantity of existing ingredients as they are used or restocked.
   The system should update the raw food stock list in real-time as new ingredients are added or used up.

Back of the card (Non-Functional Requirements):
The inventory update interface should be easy to use and intuitive for the manager.
The system should have a fast and responsive interface to allow for quick updates.
The system should ensure the privacy and security of inventory information.
The system should be available 24/7 for the manager to update inventory information.

3. **Front of the card (User Story):**
   As a manager, I want to be able to record payments for the central mess, so that I can keep track of the revenue and expenses.

   Back of the card (Functional Requirements):
   The system should provide a form for the manager to record payments.
   The system should calculate the total revenue for each day, week, and month.

   Back of the card (Non-Functional Requirements):
   The payment recording interface should be easy to use and intuitive for the manager.
   The system should ensure the accuracy and security of financial information.
   The system should be available 24/7 for the manager to record payments.

4. **Front of the card (User Story):**
   As a manager, I want to be able to add/drop cadet information, so that the system can keep track of the cadets seeking training in the PTC.

   Back of the card (Functional Requirements):
   The system should allow the manager to enter new cadet information (name, age, gender, etc.).
   The system should allow the manager to remove cadet information when necessary.
   The system should validate that all required information is provided before adding a new cadet.

   Back of the card (Non-Functional Requirements):
   The cadet information form should be easy to use and intuitive for the manager.
   The system should ensure the privacy and security of cadet information.
   The system should be available 24/7 for the manager to add or remove cadet information.

5. **Front of the card (User Story):**
   As a manager, I want to be able to view the menu, and edit it if required, so that I can make changes as necessary.

   Back of the card (Functional Requirements):
   The system should display the current menu, including all available food items and their prices.
   The system should allow the manager to edit the menu, adding or removing items and changing prices as necessary.
   The system should update the menu in real-time as changes are made.

   Back of the card (Non-Functional Requirements):
   The menu editing interface should be easy to use and intuitive for the manager.
   The system should have a fast and responsive interface to allow for quick updates.
   The system should ensure the privacy and security of menu information.
   The system should be available 24/7 for the manager to update the menu.

6. **Front of the card (User Story):**
   As a manager, I want to be able to authenticate users, so that only authorized individuals can access sensitive information and perform certain actions.

   Back of the card (Functional Requirements):
   The system should require users to provide valid login credentials (e.g. username and password) to access the system.
   The system should have different levels of access and permissions for different users.

   Back of the card (Non-Functional Requirements):
   The authentication process should be secure and protect against unauthorized access.
   The system should be available 24/7 for users to login and access the system.
   The login interface should be user-friendly and intuitive for users to use.

Admin:

1. **Front of the card (User Story):**
   As an admin, I want to be able to verify users, so that only valid managers can access the system.

   Back of the card (Functional Requirements):

   The system must allow admins to view and manage user accounts.
   The system must have a user registration process where user provide their personal information and create a username and password.

The system must allow admins to review and approve manager accounts before they are granted access to the system.
The system must have password policies in place to ensure secure manager authentication, such as requiring managers to create a strong password and resetting passwords periodically.
The system must allow admins to revoke manager access if needed, such as in cases of security breaches or policy viola
tions.
The system must maintain a log of manager authentication and access attempts for auditing and monitoring purposes

Back of the card (Non-Functional Requirements):

The account details should be in the proper user friendly interface.
The system should ensure the security of the data.
The system must be able to handle a large number of manager accounts.

2. **Front of the card (User Story):**
   As an admin, I want to be able to record payments for the central mess, so that I can keep track of the revenue and expenses.

   Back of the card (Functional Requirements):

   The system should provide a form for the admin to record payments.
   The system should calculate the total revenue for each day, week, and month.

   Back of the card (Non-Functional Requirements):

   The payment recording interface should be easy to use and intuitive for the admin.
   The system should ensure the accuracy and security of financial information.
   The system should be available 24/7 for the admin to record payments.

3. **Front of the card (User Story):**
   As an admin, I want to be able to view the menu, and edit it if required, so that I can make changes as necessary.

   Back of the card (Functional Requirements):
   The system should display the current menu, including all available food items and their prices.
   The system should allow the manager to edit the menu, adding or removing items and changing prices as necessary.
   The system should update the menu in real-time as changes are made.

Back of the card (Non-Functional Requirements):
The menu editing interface should be easy to use and intuitive for the manager.
The system should have a fast and responsive interface to allow for quick updates.
The system should ensure the privacy and security of menu information.
The system should be available 24/7 for the manager to update the menu.

4. **Front of the card (User Story):**
   As an admin, I want to be able to view statistics, so that I can monitor the performance of the canteen/mess.

   Back of the card (Functional Requirements):
   The system should provide a dashboard that displays key performance indicators (KPIs) for the canteen/mess (e.g. revenue, expenses, etc.).
   The system should allow the admin to filter data by date, time period, etc.

   Back of the card (Non-Functional Requirements):
   The statistics dashboard should be easy to use and intuitive for the admin.
   The system should ensure the accuracy and security of data.
   The system should be available 24/7 for the admin to view statistics.

5. **Front of the card (User Story):**
   As an admin, I want to be able to add/drop customers (manager/Users), so that I can maintain an accurate customer database.

   Back of the card (Functional Requirements):
   The system should provide a form for the admin to add new customers (e.g. name, contact information, etc.).
   The system should allow the admin to remove customers when necessary.
   The system should validate that all required information is provided before adding a new customer.

   Back of the card (Non-Functional Requirements):
   The customer management interface should be easy to use and intuitive for the admin.
   The system should ensure the privacy and security of customer information.
   The system should be available 24/7 for the admin to add or remove customers.

Product backlog sprints:

Sprint 1:

- Learn the technology stack
- Learn a basic understanding of the requirements
- Set up development environment

Sprint 2:

- Basic user interface design
- Design Database schema

Sprint 3:

- Designing home page
- Designing login page
- Designing user account details page

Sprint 4:

- Designing feedback form
- Designing inventory page
- Designing cadet add/drop feature
- Designing customer add/drop feature

Sprint 5:

- Designing statistics page
- Designing menu page
- Designing User authentication feature
- Designing record payment feature

Sprint 6:

- Implementation of database
- Designing backend
- Testing and debugging