

FUSION ERP

SOFTWARE REQUIREMENTS SPECIFICATION

SA-2 MESS MANAGEMENT SYSTEM (WEB)

STUDENT MENTOR: [HIMANSHU RANJAN](#) 21BCS101

FACULTY MENTOR: DR. VIJAYPAL SINGH
RATHOR

TEAM MEMBERS:

- | | |
|---------------------|----------|
| • HARSHIT TIWARI | 21BCS096 |
| • HARSHUL CHOUDHARY | 21BCS098 |
| • HARSHUL NAWANI | 21BCS099 |
| • HEMANT SINGH | 21BCS100 |
| • HRITIK YADAV | 21BCS103 |

1. Introduction

1.1 Introduction about the Fusion – A brief Description

Fusion is the ERP portal for IIITDM Jabalpur. It is a platform to integrate and elevate the day to day operations of the institute. This system integrates essential modules such as Academics, Mess Management, Hostel Management, Complaint system etc to create a centralized digital ecosystem. Fusion optimizes administrative processes improving accessibility and efficiency of operations, streamlined mess operations, and efficient hostel management.

The ERP portal serves as a platform for collaboration between students, faculty and staff members of the institute providing a user-friendly interface for a digitally enhanced IIITDM Jabalpur experience. It not only simplifies the day to day tasks but also acts as a stage for an overall connected experience. Fusion's role is pivotal in the institute's future as a dynamic and evolving platform that adapts to the changing needs of IIITDM Jabalpur.

1.2 Module Purpose

The Central Mess Management System discusses and looks towards the day to day functioning of mess and how these functions can be automated so that all the actors and activities involved can be streamlined with minimal amount of errors. It aims to modernize and streamline mess operations within Fusion, the institute's ERP portal. The software's goal is to streamline the administrative tasks, enhance the user experience and foster transparency. It works at various levels, and try to achieve above goals by :

- **Automating essential mess functionalities:** Replace manual processes with digital solutions for tasks like menu planning, meal ordering, billing, and feedback management.
- **Facilitating seamless user interaction:** There are three major actors, it tries to provide users with intuitive and interactive interfaces for mess registration, tracking expenses, booking meals, creating vacation rebate requests, etc.
- **Offering Centralized Administration:** ERP portals' goal is to unify the administration tasks and to automate them so that efficiency can be achieved and redundancy and mistakes can be avoided. It aims to empower staff with administrative features like bill management, student mess allotment, complaint management, and feedback response management.

1.3 Scope of Module

The users of this module will be the registered students of the Institute (PDPM IIITDM Jabalpur), Mess Warden and Mess Caretaker of the Mess Committee. The module will be used by the students, mess warden and mess caretaker.

This software system will be a Web and App based mess management system to be used by the above mentioned concerned people.

Interface will enable students to register for mess, login, view mess menu, respond to vacation food request, apply for rebate, request special food, make payment, give feedback, view announcement and request for deregistration.

Mess Caretaker will manage registrations, make announcements, create vacation food requests, respond to vacation food requests, view feedback, manage bills of students and update the mess menu.

Mess Warden can make and view announcements. Apart from this he/she can also search or generate reports.

This interface will help the mess caretaker and the mess warden to coordinate, control and help users to view past records and access all the functionalities of the mess without having a visit to the mess physically.

2.User/Actor Description(characteristics)

Module users include:

1. Student:

- Individuals enrolled at the Institute (PDPM-IIITDM Jabalpur), and registered with the mess and have a roll number for distinctive identification.
- Access to the system is granted using an official institute email id and with the respective password.
- He needs to be registered with the mess, to access inner mess functionalities.

Functionalities:

- Students can register and deregister for the mess.
- Students can view the menu, mess bill and other details.
- Students can make bill payments, ask for rebates, request special food, and provide feedback for various aspects of mess.

2. Mess Caretaker:

- The designated authority overseeing all mess activities including registrations, making announcements etc.
- Authentication through a unique email and phone number.
- The person can be changed, but the caretaker functionalities remain the same.

Functionalities:

- Caretaker can respond to the registration/deregistration requests of students
- Caretaker can update the mess menu, create requests for vacation food and respond to students' rebate requests.
- Caretaker can view the feedback, make announcements and manage the monthly bill of the mess.

3. Mess Warden:

- Faculty member overseeing Mess Meetings and handling Budget requests.
- He also oversees the overall mess functionalities and his decisions are final. Most of his decisions are directed by caretaker functionalities.
- Authentication via a unique email id and phone number.

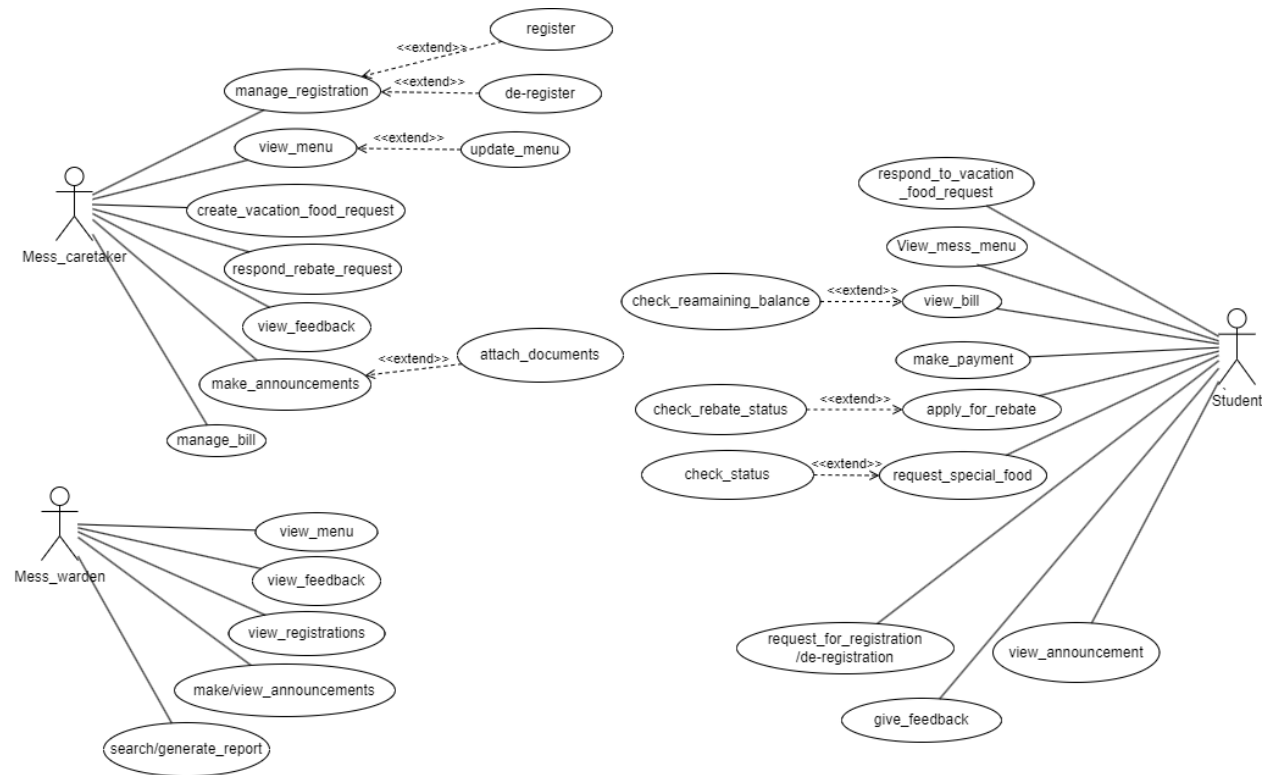
Functionalities:

- Warden can check the menu, the feedback and registrations of the students.
- Warden can make announcements for the registered students.
- Warden can generate different reports with the available mess data.

3. System Features

We specify the functional requirements for the module using use cases. The figure represents the use case diagram for the module features.

Complete System

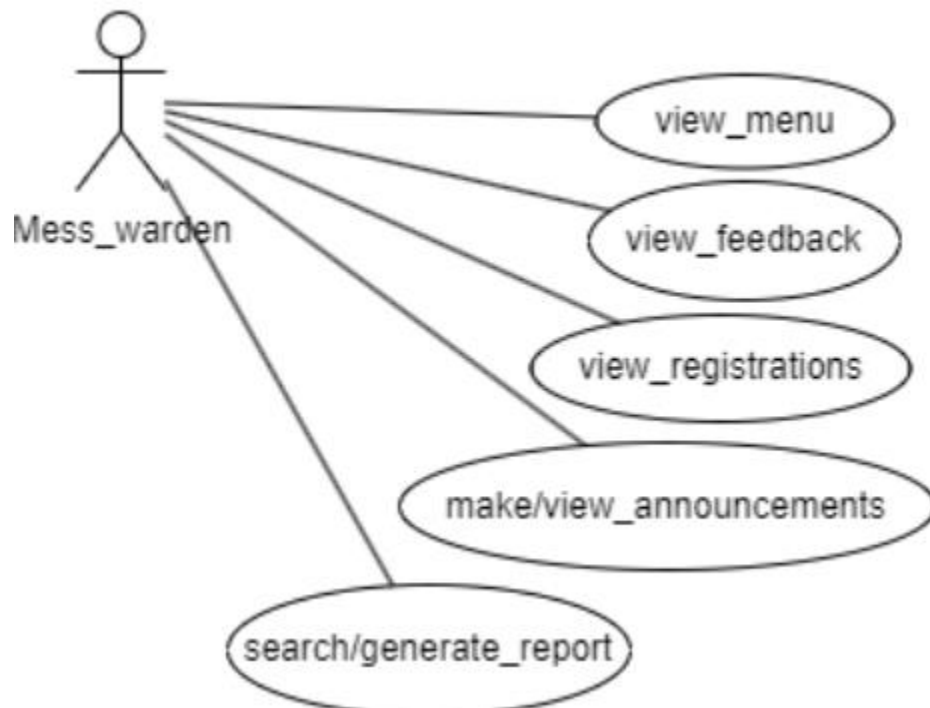


Subsystem

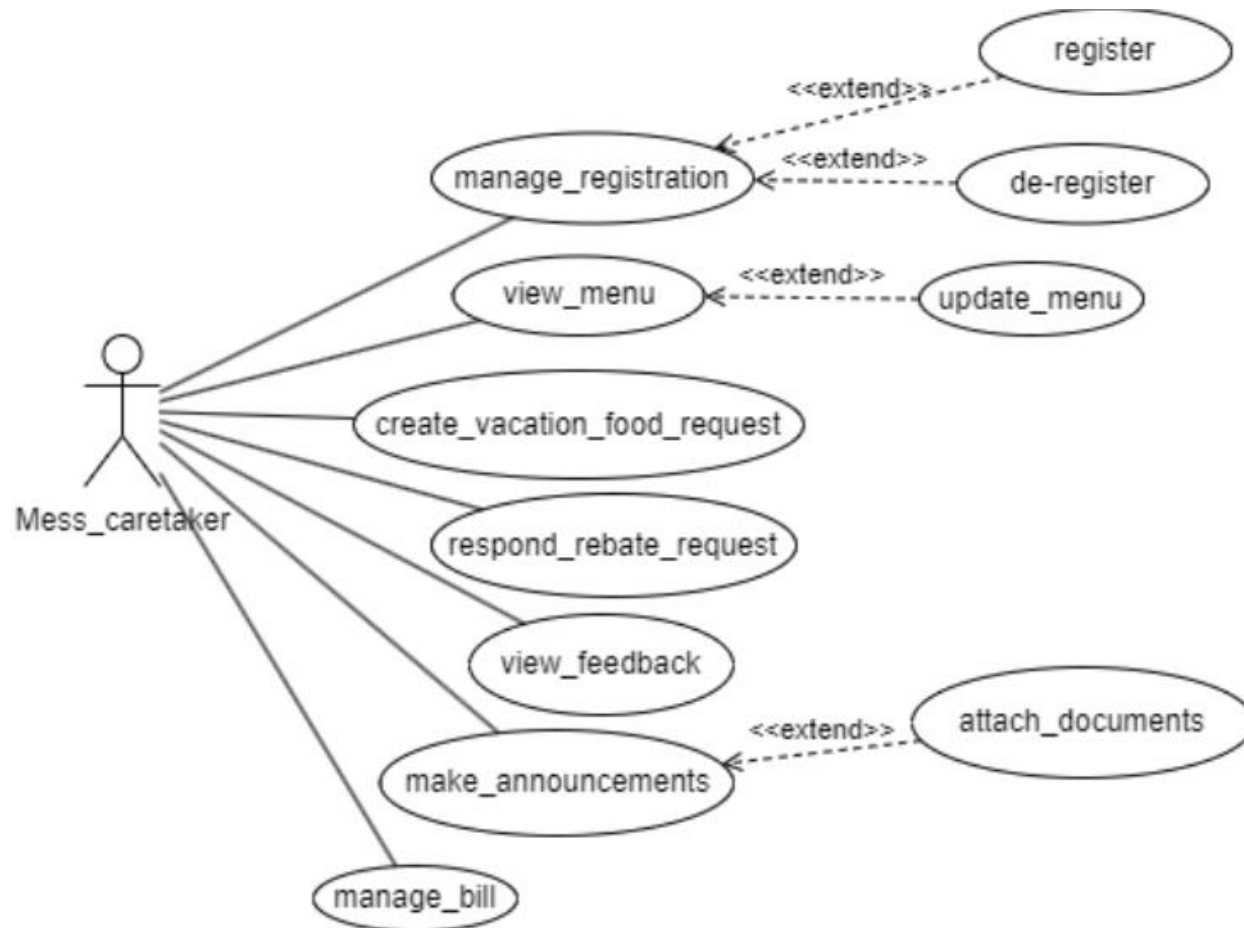
Student



Mess Warden



Mess Caretaker



3.1 Functional Requirements

1.1 Use Case #1

| | | |
|------------------------------|---|--|
| UC ID | UC#1 | |
| Use case Name | request_for_registration/de-registration | |
| Description | The student will post a request for the registration which will be viewed and approved/rejected by the Mess-Caretaker and can be viewed by Mess-Warden. | |
| Actor | Student | |
| Precondition | The student must be logged in. | |
| Main Flow | 1 | The actor clicks the 'Request Registration' button. |
| | 2 | The actor gets a screen of the 'Request is Posted'. |
| Post conditions | Student gets notification of his registration and deregistration. | |
| Alternate Flow | | NIL |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | The student returns to the main screen, on pressing the cancel button. |

1.2 Use Case #2

| | | |
|------------------------------|---|--|
| UC ID | UC#2 | |
| Use case Name | view-menu | |
| Description | The actor gets to view the menu for the week. The query from the actor is sent to the database where the data is read and is displayed on the screen. | |
| Actor | Student , Mess Warden, Mess Caretaker | |
| Precondition | The actor must be logged in.. | |
| Main Flow | 1 | The actor clicks the 'view menu' button. |
| | 2 | The actor gets a screen of the current menu. |
| Post conditions | NIL | |
| Alternate Flow | | NIL |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | The student returns to the main screen. |

1.3 Use Case #3

| | |
|--------------|------|
| UC ID | UC#3 |
|--------------|------|

| | | |
|------------------------------|--|---|
| Use case Name | view-bill | |
| Description | The student gets to view the bill and can check the remaining balance. | |
| Actor | Student | |
| Precondition | The actor must be logged in. | |
| Main Flow | 1 | The actor clicks on the 'view bill' button. |
| | 2 | A list of monthly bills of mess is displayed on the screen. |
| | 3 | The actor can view any monthly bill. |
| Post conditions | NIL | |
| Alternate Flow | [A1] | Students can check the remaining balance. |
| Sub Flow | | NIL |
| Global Alternate Flow | GA1 | The actor returns to the main screen. |

1.4 Use Case #4

| | | |
|------------------------------|--|---|
| UC ID | UC#4 | |
| Use case Name | give_feedback | |
| Description | The student can give feedback for the services and can also view recent feedbacks. | |
| Actor | Student | |
| Precondition | The student must be logged in and must be registered with the mess. | |
| Main Flow | 1 | The student clicks on the 'feedback' button. |
| | 2 | The feedback form is opened. |
| | 3 | The student fills in the feedback form and submits it.[A1] |
| Post conditions | The feedback details are added in the database and are notified to the mess caretaker. | |
| Alternate Flow | A1 | User will again be prompted to the same form with the message"some error occurred". |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | The actor returns to the main screen abandoning any intermediate operation. |

1.5 Use Case #5

| | |
|--------------|------|
| UC ID | UC#5 |
|--------------|------|

| | | |
|------------------------------|---|--|
| Use case Name | respond_to_vacation_food_request | |
| Description | The student can respond to the request generated by the Mess-Caretaker regarding registration for food during vacation. | |
| Actor | Student | |
| Precondition | The student must be logged in and must be registered with the mess. | |
| Main Flow | 1 | The student responds to the request generated by the Mess-Caretaker button("Yes" or "No").[A1][A2] |
| | 2 | The vacation food application is displayed |
| | 3 | The student fills in the required details and submits it. |
| Post conditions | The vacation food details are updated in the database. | |
| Alternate Flow | A1 | If a student selects the "Yes" option then he/she will need to fill the form. |
| | A2 | If a student selects "No" then the student will not be registered for vacation food. |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | The user returns to the initial screen. |

1.6 Use Case #6

| | | |
|------------------------------|---|---|
| UC ID | UC#6 | |
| Use case Name | apply_for_rebate | |
| Description | The student can fill a rebate application form stating the duration and purpose of leave which will be approved/rejected by the mess caretaker. | |
| Actor | Student | |
| Precondition | The student must be logged in. | |
| Main Flow | 1 | The student clicks on the 'apply for rebate' button |
| | 2 | A rebate application form is displayed. |
| | 3 | The actor fills in the necessary details |
| | 4 | Student then clicks on the submit button. |
| Post conditions | The leave request details are added in the database and are notified to the mess caretaker. The student can view the rebate status through the "check rebate status" button. | |
| Alternate Flow | | NIL |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | The actor returns to the main screen abandoning any intermediate operation. |

1.7 Use Case #7

| | | |
|----------------------|----------------------------|--|
| UC ID | UC#7 | |
| Use case Name | check_rebate_status | |

| | | |
|------------------------------|--|---|
| Description | The student gets to view the status of his rebate i.e whether the rebate has been approved, rejected or is still in process. | |
| Actor | Student | |
| Precondition | The student must be logged in and registered in the mess. | |
| Main Flow | 1 | The student clicks the 'check rebate status' button. |
| | 2 | The list of all the applied leaves are displayed along with status i.e. approved, rejected or processing. |
| Post conditions | NIL | |
| Alternate Flow | NIL | |
| Sub Flow | NIL | |
| Global Alternate Flow | GA 1 | The actor returns to the main screen abandoning any intermediate operation. |

1.8 Use Case #8

| | | |
|------------------------------|--|---|
| UC ID | UC#8 | |
| Use case Name | request_special_food | |
| Description | This request is made when either people are sick or there is some festival. The student can fill an application stating the requested food, with the date and duration for which it is to be made available. | |
| Actor | Student | |
| Precondition | The student must be logged in. | |
| Main Flow | 1 | The student clicks on the 'special food' button. |
| | 2 | A special food request form is displayed. |
| | 3 | The student fills in the required details and submits it.[A1] |
| Post conditions | The request is notified to the mess caretaker. | |
| Alternate Flow | A1 | The process can be terminated at any time by clicking on the cancel button. |
| Sub Flow | NIL | |
| Global Alternate Flow | GA 1 | The user returns to the initial screen. |

1.9 Use Case #9

| | | |
|----------------------|----------------------|--|
| UC ID | UC#9 | |
| Use case Name | view_feedback | |

| | | |
|------------------------------|---|---|
| Description | The Mess warden, caretaker can view the feedbacks of both the mess given by the students. | |
| Actor | Mess warden, Mess Caretaker | |
| Precondition | The user must be logged in as a mess warden or mess caretaker. | |
| Main Flow | 1 | List of feedbacks would be displayed on the screen. |
| | 2 | Warden can select any feedback and review it. |
| Post conditions | NIL | |
| Alternate Flow | NIL | |
| Sub Flow | NIL | |
| Global Alternate Flow | GA 1 | The user returns to the Dashboard, procedure aborted(initial screen). |

1.10 Use Case #10

| | | |
|------------------------------|--|---|
| UC ID | UC#10 | |
| Use case Name | update_menu | |
| Description | The menu as per requirement is updated by mess caretaker.. | |
| Actor | Mess Caretaker | |
| Precondition | The caretaker must be logged in. | |
| Main Flow | 1 | The caretaker clicks on the update menu button. |
| | 2 | The caretaker sees the current menu, every cell of the table is editable. |
| | 2 | Caretaker can add/delete/replace a food item in the menu. He then presses submit. |
| Post conditions | The changes made in the menu are updated in the database. | |
| Alternate Flow | NIL | |
| Sub Flow | NIL | |
| Global Alternate Flow | GA 1 | The mess caretaker returns to the main screen abandoning any intermediate process. No changes are applied and original menu is rolled back. |

1.11 Use Case #11

| | | |
|----------------------|--|---|
| UC ID | UC#11 | |
| Use case Name | respond_rebate_request | |
| Description | The mess caretaker can respond to the rebate requests made by the students. | |
| Actor | Mess caretaker | |
| Precondition | The mess caretaker must be logged in. There must be a pending rebate request. | |
| Main Flow | 1 | The mess caretaker clicks on the field of view rebates. |

| | | |
|------------------------------|---|--|
| | 2 | It sees all the rebate requests made by the students. |
| | 3 | Caretaker can either approve or reject the request on the basis of the purpose and duration.[S1] |
| Post conditions | The rebate details are updated in the database. | |
| Alternate Flow | | NIL |
| Sub Flow | S1 | The student gets notified about the status of its application. |
| Global Alternate Flow | GA 1 | The mess caretaker returns to the main screen abandoning any intermediate process. |

1.12 Use case#12

| | | |
|------------------------------|---|---|
| UC ID | UC#12 | |
| Use case Name | create_vacation_food_request | |
| Description | The mess caretaker will create a “vacation food request” for students. Those students who want to apply for vacation food can respond to it by “Yes” and others no need to respond. | |
| Actor | Mess Caretaker | |
| Precondition | The mess caretaker must be logged in. | |
| Main Flow | 1 | The mess caretaker clicks on create vacation food request button . |
| | 2 | Notification will be sent to the students. They will respond to it. |
| | 3 | Caretaker will get the list of students who have responded “Yes”. |
| | 4 | Caretaker will check their balance and request for payment if sufficient amount is not available.[S1] |
| Post conditions | The details are updated in the database. | |
| Alternate Flow | | NIL |
| Sub Flow | S1 | Student gets notification about the amount for the payment to be made in vacation. |
| Global Alternate Flow | GA 1 | The mess caretaker returns to the main screen abandoning any intermediate process. |

1.13 Use case#13

| | | |
|----------------------|---|--|
| UC ID | UC#13 | |
| Use case Name | make_announcements | |
| Description | Mess Caretaker or Mess Warden can make important announcements and every actor involved in the mess module can view it. | |
| Actor | Mess Caretaker, Mess Warden | |
| Precondition | The mess caretaker must be logged in. | |

| | | |
|------------------------------|--|--|
| Main Flow | 1 | The actor clicks on the Make Announcement button. |
| | 2 | Actor then types the message and clicks on the submit button. |
| Post conditions | Announcements are made and registered Students are notified. | |
| Alternate Flow | | NIL |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | The mess caretaker returns to the main screen abandoning any intermediate process. |

1.14 Use case#14

| | | |
|------------------------------|---|---|
| UC ID | UC#14 | |
| Use case Name | view_announcements | |
| Description | Students , Warden and caretaker can view announcements made by different entities.. | |
| Actor | Students, Mess Warden, Mess Caretaker | |
| Precondition | Students, Mess warden, Mess caretaker must be logged in. | |
| Main Flow | 1 | The actor clicks on the View Announcement button. |
| | 2 | Actor can view the announcements made by the caretaker/warden. |
| Post conditions | NIL | |
| Alternate Flow | | NIL |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | Students, Mess warden returns to the main screen abandoning any intermediate process. |

1.15 Use case#15

| | | |
|----------------------|---|--|
| UC ID | UC#15 | |
| Use case Name | manage_bill | |
| Description | The mess caretaker manages the monthly bill of every student. | |
| Actor | Mess Caretaker | |
| Precondition | The user must be logged in as a mess caretaker. | |
| Main Flow | 1 | Mess Caretaker clicks on generate bill button. |
| | 2 | Mess Caretaker then generates a bill on the basis of rebate history. |

| | | |
|------------------------------|--|---|
| Post conditions | Data is then stored in Database and Students are notified. | |
| Alternate Flow | A1 | NIL |
| | | Post condition : Mess Caretaker returns to initial screen. |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | The system stops the procedure and returns to the initial screen. |

1.16 Use case#16

| | | |
|------------------------------|--|---|
| UC ID | UC#16 | |
| Use case Name | view_registrations | |
| Description | Mess Warden can view a list of students registered for Mess. | |
| Actor | Mess Warden | |
| Precondition | Mess warden must be logged in. | |
| Main Flow | 1 | The actor clicks on the View Registrations button. |
| | 2 | Actor can see the list of registered students. |
| Post conditions | NIL | |
| Alternate Flow | | NIL |
| Sub Flow | | NIL |
| Global Alternate Flow | GA 1 | Mess warden returns to the main screen abandoning any intermediate process. |

1.17 Use case#17

| | | |
|----------------------|-----------------------------|--|
| UC ID | UC#17 | |
| Use case Name | manage_registrations | |

| | | |
|------------------------------|---|--|
| Description | Mess Caretaker can view a list of students applied for registration and can register and de register any particular student . | |
| Actor | Mess Caretaker | |
| Precondition | Mess Caretaker must be logged in. | |
| Main Flow | 1 | The actor clicks on the View Registrations button. |
| | 2 | List of Students with register and deregister options become available.[A1][A2] |
| Post conditions | The selected student is registered or deregistered from the mess. | |
| Alternate Flow | A1 | Mess Caretaker clicks on register button available from the list of students available |
| | A2 | Mess Caretaker clicks on deregister button to de register a student. |
| Sub Flow | NIL | |
| Global Alternate Flow | GA 1 | Mess Caretaker returns to the main screen abandoning any intermediate process. |

1.18 Use case#18

| | | |
|------------------------------|---|---|
| UC ID | UC#18 | |
| Use case Name | search/generate report | |
| Description | Mess Warden can search for any student record and generate reports. | |
| Actor | Mess Warden | |
| Precondition | Mess warden must be logged in. | |
| Main Flow | 1 | The actor can search in the search box. |
| | 2 | The Actor can click on the generate report button for requested data. |
| Post conditions | NIL | |
| Alternate Flow | NIL | |
| Sub Flow | NIL | |
| Global Alternate Flow | GA 1 | Mess warden returns to the main screen abandoning any intermediate process. |

3.3 Other Functional Requirements

- The module will make use of the notification module for sending notifications and alerts to various actors involved in the module suitably for various requests, confirmations, rejections, or modifications, etc.

- The module will make use of the payment gateway for enabling students to pay their mess fee.
- The Super admin of Fusion should be able to assign roles for mess caretaker and warden.

3.4 Other Constraints

3.4.1 User Interfaces

The user interface should comply with the color scheming and dashboard design of the FUSION portal. Users should be able to navigate from one menu to another with ease. Inter module navigation should be smooth. All the functionalities should be easy to use and no specific training should be required for the usage of the module.

3.4.2 Tech Stack Used

Django (python 3.8) with server side rendering.
Database used - postgresSQL

3.4.3 Business rules

- Only the registered students are allowed to use the functionalities of the module. Unregistered students can only view the mess menu and make a registration request.
- The bill of the current month will be generated on the first day of next month and will be calculated based on the rebate details and base amount set for the current month.

4. Non-Functional Requirements

4.1 Performance

- The system should respond quickly to user actions (loading pages, submitting forms, etc.).
- Response times should be within industry standards for web applications.
- The system should be available 24/7 with minimal downtime for scheduled maintenance.

4.2 Scalability

The system should be able to accommodate the growing needs of the institute, including increasing user base and data volume.

4.3 Security

- Secure login and access control mechanisms should be implemented to ensure only authorized users can access specific features.
- User data (personal information, meal bookings, financial transactions) should be encrypted and protected from unauthorized access.
- All user activities and system events should be logged for audit trails and accountability.
- Regular log reviews and incident response procedures should be established.

4.4 Usability

- The interface should be user-friendly, intuitive, and visually appealing for all user groups (students, staff, faculty).
- Navigation should be simple and consistent across modules.
- Responsive design for optimal experience on different devices.
- The design should be consistent and must be carried for the whole website.
- Contextual help documentation should be readily available within the module.
- Contact information for technical support should be easily accessible.

4.5 Maintainability

- The system should be designed in a modular way to facilitate updates, bug fixes, and future enhancements.
- Code should be well-documented and organized for easy maintenance.
- A version control system should be used to track changes and manage software releases.
- Automated testing routines should be implemented to ensure functionality and stability of the system.

5. Module Dependencies with Other Fusion Modules

5.1 Database Dependencies

This module uses the student database, to get all the students information and can store them. Additional fields are added to the database of students and stored accordingly creating a separate mess database in postgresql.

The major Relational tables used in this module is Students Table only.

5.2 Dependencies with Modules

5.2.1 Notification Module

The notification module will work closely with this module and handle all the notifications sent and generated by the caretaker and warden and will make sure that these notifications are properly shown on the dashboard and within the required time frame. Also other module notifications which are necessary and related to mess,

should be shown to the mess caretaker and mess warden.

5.2.2 Dashboard Module

This module will look for a transaction process for password change requests and other admin requests made by the mess Module. These requests power transfer when Mess Warden and Mess Caretaker changes. Also it will manage the directed routes.

5.2.3 Super Admin Functions

This will check roles and also assign various roles to the respective assigned people. It will also add new students to the students database and look for the removed students.

5.2.4 Payment Gateway Provider

This will work with the mess module to handle mess payments made by the students. The student will press the payment button and will be directed to the payment gateway where he will do the payment and after successful completion of the payment gateway will provide an acknowledgement for the same which will be processed by the mess module and will help in updating the student amounts details in the database.