Software Requirements Specification

**for**

**<Central Mess Management System>**

**<Sep 30, 2017>**

**<version 1.0>**

**Team Members:**

**Varnika Jain - 2015268**

**Ankita Makker - 2015034**

**Pratibha Singh - 2015189**

**Kakulavaram Venkateshwar - 2015117**

**Table of Contents**

**Table of Contents**

**1. Introduction [1](#_heading=h.qsh70q)**

1.1 Purpose [1](#_heading=h.3as4poj)

1.2 Product Scope [1](#_heading=h.1pxezwc)

1.3 References [1](#_heading=h.49x2ik5)

**2. Overall Description [2](#_heading=h.2p2csry)**

2.1 Product Perspective [2](#_heading=h.147n2zr)

2.2 Product Functions [2](#_heading=h.3o7alnk)

2.3 User Classes and Characteristics [2](#_heading=h.23ckvvd)

**3. System Features(Functional Requirements) 3**

**4. Other Nonfunctional Requirements [4](#_heading=h.ihv636)**

4.1 User Characteristics [4](#_heading=h.32hioqz)

4.2 Security Requirements 4

4.3 Business Rules 4

# 5. List of open issues (unresolved issues) with the module 5

# 6. Some of the issues, problems you faced, lessons you learned during the project, 6 suggestions and your guanine feedback (for future).

**Appendix A: To Be Determined List**

**7**

## Introduction

* 1. **Purpose**

This document specifies various requirements for the “Central Mess Management System” of the Institute ERP. The purpose of our software is to automate the Central Mess Functionality and provide both the users and the administrator a smart platform to interact with each other.

It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and what kind of system interactions will take place.

* 1. **Module Scope**

The users of this module will be the registered students of the Institute (PDPM IIITDM Jabalpur), Mess Manager, Mess Warden and the members of the Mess Committee. The module will be used by the students.  
This software system will be a web based mess management system to be used by the above mentioned concerned people.  
Interface will enable students to register for veg/non-veg mess, login, to keep a track of their attendance, view mess menu, submit leave applications, apply for special food or non vegetarian items, give feedbacks and view notices. The admin will record the mess bill of students and update mess menu, check the stock details, budget details and the guest entries, and respond to student applications. This interface will help the mess committee and the mess warden to coordinate and review the meetings. Help users to access all the functionalities of the mess without having a visit to the mess physically.  
 A relational database will be used for handling all the data such as menu items, bills, student IDs, consumption statistics, student feedback, etc.

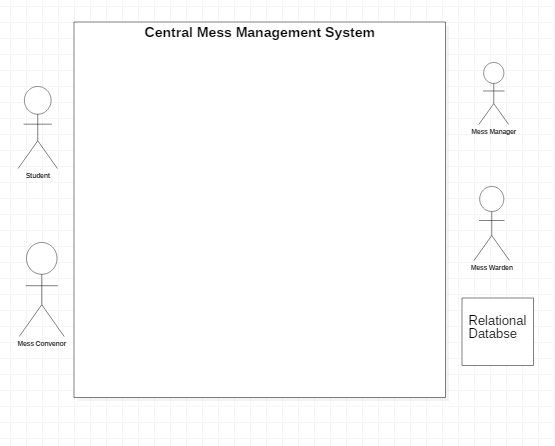
* 1. **References**
* Mess Committee Members
* Mess Manager
* Dr. Atul Gupta   
  Associate Professor, Computer Sc. & Engineering, Indian Institute of Information Technology Design and Manufacturing, Jabalpur.   
  https://sites.google.com/site/atulkg/courses/software-engineering-semester-i-2017-18

# Overall Description

## Product Perspective

This module is one of the twelve modules being developed for institute ERP. This module will be available as a feature for all the students, Mess Convener, Mess Warden and Manager of the institute. This functionality will be exercised through the dashboard interface of the concerned people. A relational database is used to store all the data such as feedback, consumption statistics, list of registered students, list of faculty members and regular menus with items and special dish menu with items and prices etc.

**System Environment**

****

## 

## 

## 2.2 Functional Requirements Specifications

This section outlines the use cases for each of the actors separately. The Students have a different set of use cases (which are related to availing the mess facilities, provide feedbacks and apply for leave or special food items) than the Mess Manager (which are related to managing the various aspects of the mess such as special orders, respond to requests for changing the mess, updation of mess menu, occasional mess booking, managing stock details, etc.) Mess convener (member of the mess committee) have different functions, It organizes polls, inform members about the meetings, publish minutes of the meetings, requests Mess Manager for changes in the mess menu, provide budget requests to the Mess Warden, etc. Mess warden can review the meetings and respond to budget requests accordingly. A student can register for a specific mess only during the registration time of the semester.

## 2.2.1 User Classes and Characteristics

Following are the users of the module

Student: A student of the Institute (PDPM-IIITDM Jabalpur) should have a roll number (for unique identification) and an official institute email id, which should be used for accessing the system.

Mess Manager: A person In Charge for all the activities of the mess.(unique email and pf number)

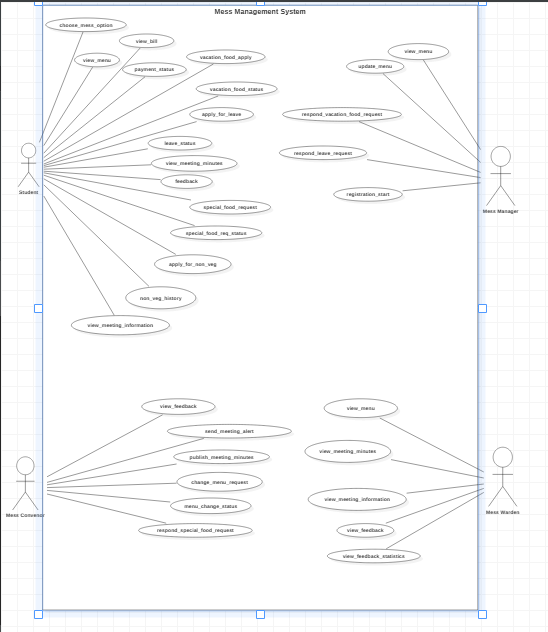
Mess Convener: A member of the mess committee has a unique roll number and email to access the system.

Mess Warden: Mess Warden is the Faculty Incharge who supervises the Mess Meetings and processes the Budget requests. (Unique email id and pf number).

# System Features

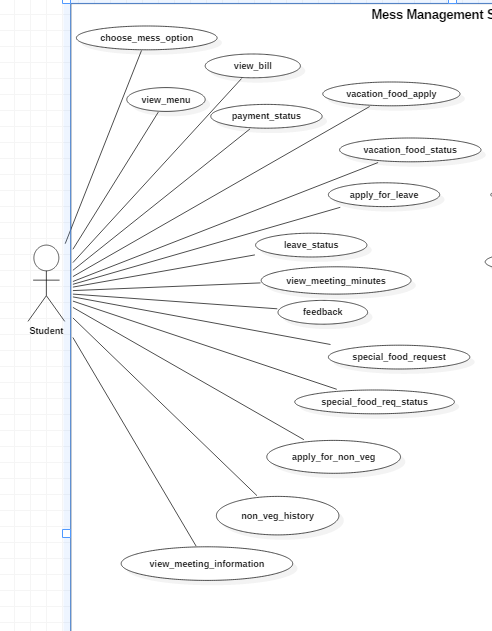
We specify the functional requirements for the module using use cases. The figure represents the use case diagram for the module features.

**Complete System**

****

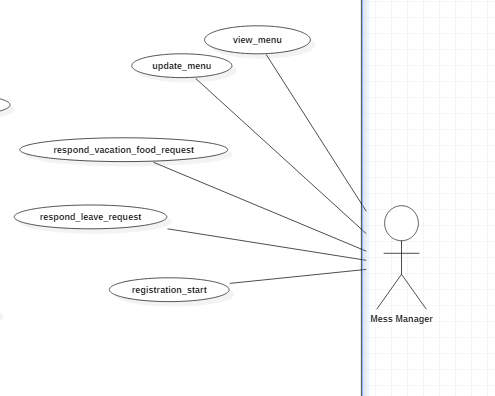
# Subsystem

## Student

****

## 

## Mess Manager

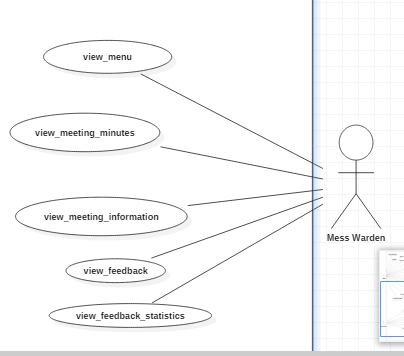
****

## 

## Mess Convener

****

## Mess Warden

****

## Functional Requirements

* 1. **Use Case #1**

| **UC ID** | UC#1 | |
| --- | --- | --- |
| **Use case Name** | **login** | |
| **Description** | The user gets to login and avail the facilities of the web based mess management system. | |
| **Actor** | Student , Mess Manager, Mess Warden, Mess Convener | |
| **Precondition** | NIL | |
| **Main Flow** | 1 | The actor clicks the ‘login’ button. |
| 2 | The actor fills the roll number and password at the login page. |
| 3 | The user clicks at the submit button.[A1] |
| **Post conditions** | Session starts for the logged in user. | |
| **Alternate Flow** | A1 | The student is prompted to  Enter the login details again. (mismatch credentials) |
| A2 | The user returns to initial screen (cancel) |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to the main screen. |

## Use Case #2

| **UC ID** | UC#2 | |
| --- | --- | --- |
| **Use case Name** | **logout** | |
| **Description** | The user gets to logout whenever he wants. | |
| **Actor** | Student , Mess Manager, Mess Warden, Mess Convener | |
| **Precondition** | The actor must be logged-in. | |
| **Main Flow** | 1 | The actor clicks the ‘logout’ button. |
| 2 | The actor is redirected to the login page. |
| 3 | Current session is destroyed from the system. |
| **Post conditions** | The actor is logged out of the system. | |
| **Alternate Flow** | NIL | |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to the main screen. |

## Use Case #3

| **UC ID** | UC#3 | |
| --- | --- | --- |
| **Use case Name** | **change\_password** | |
| **Description** | The actor can change the password anytime. | |
| **Actor** | Student , Mess Manager, Mess Warden, Mess Convener | |
| **Precondition** | The actor should be logged-in. | |
| **Main Flow** | 1 | The actor clicks the ‘change password’ button. |
| 2 | The actor enters the old password and the new required password. |
| 3 | After validation of passwords the password is changed.[A1] |
| **Post conditions** | The user’s password is updated in the database. | |
| **Alternate Flow** | A1 | The student can click on the forgot password button. A link will be sent to the user’s email address to change password. |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to the main screen. |

## Use Case #4

| **UC ID** | UC#4 | |
| --- | --- | --- |
| **Use case Name** | **choose\_mess-option** | |
| **Description** | The student gets to choose the mess in the starting of every semester. | |
| **Actor** | Student | |
| **Precondition** | Selecting mess feature is available limited period of time. | |
| **Main Flow** | 1 | The student has two options in the form of radio button to choose from. |
| 2 | Once the user has selected an option, he clicks the submit button where the server receives this data. |
| **Post conditions** | The student’s choice for the following semester is updated in the database. | |
| **Alternate Flow** | A1 | If the student clicks submit button without selecting an option, the system will prompt for an option. |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to the main screen. |

## Use Case #5

| **UC ID** | UC#5 | |
| --- | --- | --- |
| **Use case Name** | **view-menu** | |
| **Description** | The student gets to view the menu for the week. The query from the student is sent to the database where the data is read and is displayed on the screen. | |
| **Actor** | Student | |
| **Precondition** | The student 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** | GA1 | The student returns to the main screen. |

## Use Case #6

| **UC ID** | UC#6 | |
| --- | --- | --- |
| **Use case Name** | **view-bill** | |
| **Description** | The student gets to view the bill for the month. If a student applies for the leave then he can also see his rebate money, rebate days and that rebate money will be deducted from his monthly bill.If the student is a part of non-veg mess, he/she could see the non-veg bill details as well. | |
| **Actor** | Student | |
| **Precondition** | The actor must be logged in. | |
| **Main Flow** | 1 | The actor clicks on the ‘view bill’ button. |
| 2 | A list is displayed on the screen. |
| 3 | The actor can view any month bill. |
| 4 | The bill details and rebate details(if any) of the selected month will be displayed. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The actor returns to the main screen. |

## Use Case #7

| **UC ID** | UC#7 | |
| --- | --- | --- |
| **Use case Name** | **payment\_status** | |
| **Description** | The user can view the payment status of his previous and current semester. | |
| **Actor** | Student | |
| **Precondition** | The actor must be logged in. | |
| **Main Flow** | 1 | The actor clicks on the ‘payment status button’. |
| 2 | The student can see the payment status of the seemsters till date. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The actor returns to the main screen. |

**3.8 Use Case #8**

| **UC ID** | UC#8 | |
| --- | --- | --- |
| **Use case Name** | **Vacation\_food\_apply** | |
| **Description** | The student can apply for the food in the mess during vacation. | |
| **Actor** | Student | |
| **Precondition** | The student must be logged in. | |
| **Main Flow** | 1 | The student clicks the “vacation food” button. |
| 2 | The vacation food application is displayed |
| 3 | The student fills in the required details and submits it.[A1] |
| **Post conditions** | The vacation food details are updated in the database. | |
| **Alternate Flow** | A1 | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to initial screen. |

**3.9 Use Case#9**

| **UC ID** | UC#9 | |
| --- | --- | --- |
| **Use case Name** | **Vacation\_food\_status** | |
| **Description** | This request is made when the student is staying back in the college during vacations. | |
| **Actor** | Student | |
| **Precondition** | The student must be logged in. | |
| **Main Flow** | 1 | The student clicks on ‘vacation food status’ button. |
| 2 | A list of status is displayed. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to initial screen. |

## Use Case #10

| **UC ID** | UC#10 | |
| --- | --- | --- |
| **Use case Name** | **apply\_for\_leave** | |
| **Description** | The student can fill a leave application form stating the duration and purpose of leave which will be approved/rejected by the mess manager. | |
| **Actor** | Student | |
| **Precondition** | The actor must be logged in. | |
| **Main Flow** | 1 | The student clicks on the ‘apply for leave’ button |
| 2 | A leave application form is displayed. |
| 3 | The actor fills in the necessary details |
| 4 | He then clicks on the submit button.[A1] |
| **Post conditions** | The leave request details are added in the database and is notified to the mess ma  The student can view the leave status through leave status button. | |
| **Alternate Flow** | A1 | NIL |
|  |  | **Post-condition**: returns to initial screen |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The actor returns to the main screen abandoning any intermediate operation. |

## Use Case #11

| **UC ID** | UC#11 | |
| --- | --- | --- |
| **Use case Name** | **leave\_status** | |
| **Description** | The student gets to view the status of his leave i.e whether the leave has been approved, rejected or is still in process. | |
| **Actor** | Student | |
| **Precondition** | The student must be logged in. | |
| **Main Flow** | 1 | The student clicks the ‘view leave 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** | GA1 | The actor returns to the main screen abandoning any intermediate operation. |

## Use Case #12

| **UC ID** | UC#12 | |
| --- | --- | --- |
| **Use case Name** | **view\_meeting\_minutes** | |
| **Description** | The student gets to view minutes of previous meetings. | |
| **Actor** | The actor must be logged-in. | |
| **Precondition** | The student must be logged in. | |
| **Main Flow** | 1 | The actor clicks on ‘meeting minutes’ button. |
| 2 | The list of all meetings is displayed. |
| 3 | He could then view the details of the meeting selected. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The actor returns to the main screen. |

## Use Case #13

| **UC ID** | UC#13 | |
| --- | --- | --- |
| **Use case Name** | **feedback** | |
| **Description** | The student can give feedback for the services provided by the mess. Also reviewing recent feedbacks. | |
| **Actor** | Student | |
| **Precondition** | The student must be logged in. | |
| **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 mess convenor. | |
| **Alternate Flow** |  | NIL |
|  |  | **post-condition**: returns to initial screen |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The actor returns to main screen abandoning any intermediate operation. |

## Use Case #14

| **UC ID** | UC#14 | |
| --- | --- | --- |
| **Use case Name** | **special\_food\_request** | |
| **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 ‘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 convener. | |
| **Alternate Flow** | A1 | The process can be terminated at any time. |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to initial screen. |

## Use Case #15

| **UC ID** | UC#15 | |
| --- | --- | --- |
| **Use case Name** | **apply\_for\_non\_veg** | |
| **Description** | The students who belong to the non-veg mess can only avail this facility. He will apply for the non-veg items and the dates when this service will be prevailed and accordingly the database will get updated. | |
| **Actor** | Student | |
| **Precondition** | The student must be logged-in as a member of non veg mess. | |
|  | 2 | .A non veg application form is displayed. |
| 3 | The student fills in the form and submits it.[S1] |
| **Post conditions** | On processing, the database will get updated | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** | S1 | The bill will get updated as per the chosen items. |
| **Global Alternate Flow** | GA1 | The user returns to initial screen. |

## Use Case #16

| **UC ID** | UC#16 | |
| --- | --- | --- |
| **Use case Name** | **special\_food\_request\_status** | |
| **Description** | The student gets to view the status of his special food request i.e whether the request has been approved, rejected or is still in process. | |
| **Actor** | Student | |
| **Precondition** | The student must be logged in. | |
| **Main Flow** | 1 | The student clicks on ‘special food request status’ button. |
| 2 | The list of all the applied request are displayed along with status i.e. approved,rejected or processing. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to initial screen. |

**3.17 Use Case #17**

| **UC ID** | UC#17 | |
| --- | --- | --- |
| **Use case Name** | **Non\_veg\_history** | |
| **Description** | The student can view his non-veg food history. The data is fetched from the database. | |
| **Actor** | Student | |
| **Precondition** | The student must be logged in. | |
| **Main Flow** | 1 | The student clicks on ‘non veg history’ button. |
| 2 | Non veg food history will be displayed on the screen. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to initial screen. |

## 3.18 Use Case #18

| **UC ID** | UC#18 | |
| --- | --- | --- |
| **Use case Name** | **view\_feedback** | |
| **Description** | The convenor can view the feedbacks filled by the students. | |
| **Actor** | Mess Convenor | |
| **Precondition** | The user must be logged in as a mess convenor | |
|  | 2 | List of feedbacks would be displayed on the screen. |
| 3 | Convenor can select any feedback and review it. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to the Dashboard, procedure aborted(initial screen). |

## Use Case #19

| **UC ID** | UC#19 | |
| --- | --- | --- |
| **Use case Name** | **send\_meeting\_alert** | |
| **Description** | The mess convenor sends the information about the meeting to the respective mess committee members. | |
| **Actor** | Mess Convenor | |
| **Precondition** | The user must be logged in as a mess convenor | |
| **Main Flow** | 1 | Convenor clicks on ‘send meeting alert’. |
| 2 | A form is displayed on the screen. |
| 3 | Convenor fills the agenda, description, date and time of the meeting. |
| 4 | Convenor clicks on ‘alert’.[A1] |
| **Post conditions** | The information of the meeting is sent to all the mess committee members. | |
| **Alternate Flow** | A1 | The convenor chooses not to alert. |
|  |  | **Post condition** : convenor returns to initial screen. |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The system stops the procedure and returns to the initial screen. |

## Use Case #20

| **UC ID** | UC#20 | |
| --- | --- | --- |
| **Use case Name** | **publish\_meeting\_minutes** | |
| **Description** | The minutes of the meeting are published by the convenor. | |
| **Actor** | Mess Convenor | |
| **Precondition** | The user must be logged in as a mess convenor | |
| **Main Flow** | 1 | Convenor will click on ‘publish minutes’. |
| 2 | A form would appear on the screen asking for meeting agenda, description and minutes. |
| 3 | Convenor will fill the form. |
| 4 | Convenor will submit the form. [A1] |
| **Post conditions** | The minutes for the meeting will get published and all the student and the mess warden can access it. | |
| **Alternate Flow** | A1 | The user chooses not to publish |
|  |  | **Post condition**: convenor returns to initial screen. |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The system aborts the procedure, return to Dashboard. |

## Use Case #21

| **UC ID** | UC#21 | |
| --- | --- | --- |
| **Use case Name** | **change\_menu\_request** | |
| **Description** | Mess Convenor can request the mess manager for change in the mess menu after analysing the feedbacks, meetings or polling results. | |
| **Actor** | Mess Convenor | |
| **Precondition** | The user must be logged in as a mess convenor | |
| **Main Flow** | 1 | Mess Convenor clicks on ‘change menu request’ |
| 2 | Application is displayed on the screen. |
| 3 | Mess Convenor inputs the new menu (items along with time)[A1] |
| 4 | The application is submitted. [A1] |
| **Post conditions** | The application is forwarded to the Mess Manager for processing. | |
| **Alternate Flow** | A1 | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The procedure is aborted, return to initial screen. |

## Use Case #22

| **UC ID** | UC#22 | |
| --- | --- | --- |
| **Use case Name** | **menu\_change\_status** | |
| **Description** | Change in menu if accepted is reflected in the menu table. | |
| **Actor** | Mess Convenor,Student,Mess Manager,Mess Warden | |
| **Precondition** | Any actor logged in could see the change in menu through ‘view menu’ button. | |
| **Main Flow** | 1 | User on ‘view menu’ button. |
| 2 | The updated menu(if any) is displayed on the screen. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** |  | NIL |

## Use Case #23

| **UC ID** | UC#23 | |
| --- | --- | --- |
| **Use case Name** | **respond\_special\_food\_request** | |
| **Description** | The mess convenor can respond to the applications filed by the students for special food during festivals or in cases of sickness. | |
| **Actor** | Mess Convenor | |
| **Precondition** | The user is logged in as Mess Convenor and there are requests for special food to respond to. | |
| **Main Flow** | 1 | Convenor will click on ‘applications for special food’. |
| 2 | List of applications will be displayed on the screen. |
| 3 | Convenor can select any of the application, review its purpose and duration and special food preferences and accordingly respond to the request (accept or reject).[S1] [A1] |
| **Post conditions** | The application is processed, database is updated. | |
| **Alternate Flow** | A1 | NIL |
| **Sub Flow** | S1 | The status of the application is updated and the student is notified. |
| **Global Alternate Flow** | G1 | The procedure will be aborted, system return to initial screen (dashboard). |

## Use Case #24

| **UC ID** | UC#24 | |
| --- | --- | --- |
| **Use case Name** | **view\_meeting\_info** | |
| **Description** | The use case facilitates viewing of various meetings and their information. | |
| **Actor** | Mess Warden | |
| **Precondition** | The user should be logged in as mess warden. | |
| **Main Flow** | 1 | Mess Warden clicks on ‘view meeting info’. |
| 2 | Lists of meetings scheduled along with their agenda and description is displayed. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** |  | NIL |

## Use Case #25

| **UC ID** | UC#25 | |
| --- | --- | --- |
| **Use case Name** | **view\_meeting\_minutes** | |
| **Description** | The minutes of the meetings (between mess committee members are published) are accessible by the mess Warden for supervising. | |
| **Actor** | Mess Warden,Mess Convener,Mess Committee | |
| **Precondition** | The user should be logged in as mess warden. | |
| **Main Flow** | 1 | Mess Warden clicks on ‘view meeting minutes’. |
| 2 | Lists of meetings scheduled along with their complete information and minutes is displayed |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** |  | NIL |

* 1. **Use Case #26**

| **UC ID** | UC#26 | |
| --- | --- | --- |
| **Use case Name** | **view-menu** | |
| **Description** | The actor gets to view the menu of both the mess. The query from the actor is sent to the database where the data is read and is displayed on the screen | |
| **Actor** | Mess Warden , Mess Manager | |
| **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** | GA1 | The actor returns to the main screen. |

* 1. **Use Case #27**

| **UC ID** | UC#27 | |
| --- | --- | --- |
| **Use case Name** | **view\_feedback** | |
| **Description** | The Mess warden can view the feedbacks of both the mess filled by the students. | |
| **Actor** | Mess warden | |
| **Precondition** | The user must be logged in as a mess warden | |
| **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** | GA1 | The user returns to the Dashboard, procedure aborted(initial screen). |

* 1. **Use Case #28**

| **UC ID** | UC#28 | |
| --- | --- | --- |
| **Use case Name** | **view\_feedback\_statistics** | |
| **Description** | The mess warden can view the feedback statistics of the mess that he wants to view. | |
| **Actor** | Mess warden | |
| **Precondition** | The user must be logged in as a mess warden. | |
| **Main Flow** | 1 | Mess warden will click on the feedback statistics of the mess that he want to view. |
| 2 | A list of feedback will be displayed on the screen and then he can click on the feedback statistics from the list that he want to review. |
| **Post conditions** | NIL | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The user returns to the Dashboard, procedure aborted(initial screen). |

## Use Case #29

| **UC ID** | UC#29 | |
| --- | --- | --- |
| **Use case Name** | **update\_menu** | |
| **Description** | The menu can be updated by the mess Manager on the basis of the requests by mess convenor. | |
| **Actor** | Mess Manager | |
| **Precondition** | The manager must be logged in. | |
| **Main Flow** | 1 | The manager gets current menu on the screen. |
| 2 | Manager can either add new items or replace a food item in the menu. |
| **Post conditions** | The changes made in the menu are updated in the database. | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** | GA1 | The mess manager returns to the main screen abandoning any intermediate process. |

## Use Case #30

| **UC ID** | UC#25 | |
| --- | --- | --- |
| **Use case Name** | **registration\_start** | |
| **Description** | Mess Manager decides the starting and ending date of registration of mess for the coming semester. | |
| **Actor** | Mess Manager | |
| **Precondition** | The mess manager must be logged in.. | |
| **Main Flow** | 1 | Mess Manager clicks on ‘registration start’ button. |
| 2 | A form will be displayed on the screen comprises of start date and end date for the registration of mess. Mess manager will fill up the form. |
| **Post conditions** | The database will be updated. | |
| **Alternate Flow** |  | NIL |
| **Sub Flow** |  | NIL |
| **Global Alternate Flow** |  | NIL |

## Use Case #31

| **UC ID** | UC#31 | |
| --- | --- | --- |
| **Use case Name** | **respond\_leave\_request** | |
| **Description** | The mess manager can respond to the leave requests made by the students. | |
| **Actor** | Mess Manager | |
| **Precondition** | The mess manager must be logged in. | |
| **Main Flow** | 1 | The mess manager gets the screen with a list of all the applied requests for leaves. |
| 2 | It can select any application and review it. |
| 3 | Manager can either approve or reject the request on the basis of the purpose and duration.[S1][A1] |
| **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** | GA1 | The mess manager returns to the main screen abandoning any intermediate process. |

## Use case#32

| **UC ID** | UC#32 | |
| --- | --- | --- |
| **Use case Name** | **respond\_vacation\_food\_request** | |
| **Description** | The mess manager can respond to the vacation food requests made by the students. | |
| **Actor** | Mess Manager | |
| **Precondition** | The mess manager must be logged in. | |
| **Main Flow** | 1 | The mess manager gets the screen with a list of all the applied requests for vacation food. |
| 2 | It can select any application and review it. |
| 3 | Manager can either approve or reject the request on the basis of the purpose and duration.[S1][A1] |
| **Post conditions** | The 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** | GA1 | The mess manager returns to the main screen abandoning any intermediate process. |

# Other Nonfunctional Requirements

## Security Requirements

Following are the security-related issues to be taken care of:

1. Only authorized users (Institute students, Mess manager, Mess Warden) can use the system and no others.
2. The privileges to the mess warden, mess convener and mess manager must not be accessed by the students.
3. An automatic logout system (session expired) will logout a student after certain amount of time (may be 10 minutes). In case a password is forgotten, a new one will be emailed to that user's email ID after requesting new password by user.

## User Characteristics

1. To operate this software the user (student/faculty/guest) must have a knowledge of computer. The user will have visual interfaces to carry each operations, the user does not necessarily have to possess all the knowledge of the database.

2. The interface should be user friendly and all the features must be easily accessible.

## Business Rules

1.The rebate details will be carry forward to the next month.

2.The bill of the current month will be generated on the first day of next month.

# List of open issues (unresolved issues) with the module

| **S. No** | **Issue details** | **Category** | **How it can be resolved?** | **Any other relevant information** |
| --- | --- | --- | --- | --- |
| 1. | Feedback | Ajax | Already tried much of the ways but didn’t find any solution | NA |
| 2. | Calendar Problems(Dates displaying in text format) | Ajax | Didn’t found any solution | NA |
| 3. | Notification | Unimplemented Functionality | NA | NA |

# Some of the issues, problems you faced, lessons you learned during the project, suggestions and your guanine feedback (for future).

**Issues**

* More functionality could be added like updation of the convener and committee member, monthly bill cost.

**Problem faced**

* Ajax

**Lessons learned**

* Team work
* Working under pressure

# Appendix C: To Be Determined List

*References for the guest’s entries, staff and stock details.*