Mess Management System

Software Requirements Specification

Version 2.0

Rishikesh Shukla

Contents

[1. Introduction 4](#_Toc350377374)

[2. Purpose 4](#_Toc350377375)

[3. Scope 4](#_Toc350377376)

[4. Acronyms and Abbreviations 5](#_Toc350377377)

[5. Overall Description 5](#_Toc350377378)

[6. External Interface Requirements 7](#_Toc350377379)

[7. Functional Requirements 8](#_Toc350377380)

[8. Non Functional Requirements 17](#_Toc350377381)

[8.1 Usability 17](#_Toc350377382)

[8.2 Security 17](#_Toc350377383)

[8.3 Performance 17](#_Toc350377384)

[8.4 Capacity 18](#_Toc350377385)

[8.5 Recovery 18](#_Toc350377386)

[8.6 Availability 18](#_Toc350377387)

[8.7 Reliability 18](#_Toc350377388)

[8.8 Maintainability 18](#_Toc350377389)

[8.9 Portability 19](#_Toc350377390)

[8.10 Privacy 19](#_Toc350377391)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Rishikesh Shukla | 21/2/13 | Introduction, purpose, scope, overall description updated | 1.0 |
| Rishikesh Shukla | 28/2/13 | Functional requirements Changes, non functional requirements updated. | 2.0 |

# Introduction

Mess facility is an integral part of the hostel, which provides homely and comfortable stay to the students. The Vellore Institute of Technology provides sumptuous food for all its resident students. The mess facility makes sure that students' health and nutrition is well taken care of. Mess serves breakfast, lunch, snacks and dinner every day. A choice of vegetarian or non-vegetarian food is provided to the students. They can also choose between North Indian or South Indian cuisine. The hostel management team manages hostel mess. Mess menu is planned and managed by the students in consultation with the caterer and the Management, who is responsible for managing the affairs of the hostel.  The Mess Committee has representation of the students.

The Mess Management System aims at bringing the students, mess and the hostel management on a common platform thereby providing transparency among them. It aims to reduce food wastage and help maintain its quality.

# Purpose

The purpose of this Software Requirements Specification (SRS) document is to describe the overall behavior of Mess Management System. This SRS defines and describes the operations, performance, and quality assurance requirements of the MMS that is to be developed. This document also describes the nonfunctional requirements. It also describes the design constraints and technologies that are to be considered when the system is to be designed, and other factors necessary to provide a complete description of the requirements for the system. This Software Requirements Specification (SRS) captures the complete software requirements for the system. Requirements described in this document will be used as guidelines to develop the Mess Management System

# Scope

The Mess Management System is supposed to have the following features:

* Caterer Registration
* Student Registration
* Daily Mess Menu Display
* Change of Mess
* Mess Leave
* Food Quality Review System
* Mess Bill Details(for both students and mess managers)

# Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Meaning** |
| MMS | Mess Management System |

# Overall Description

1. **Product Perspective**

The Mess Management System is an online web platform where students, mess managers and hostel management can interact hassle free regarding mess.

It provides the students with daily mess menu details directly integrated with facebook and a review system on the menu items they liked that day. The student may register for a change of mess before 25th of each month. The students can view there mess bill details online. They can tell beforehand if they would not be coming for next day meal, thereby letting the mess supervisor make necessary adjustments and avoid food wastage . The students registered for Food Mall can preorder the food online.

1. **Product Functions**

The following are the user requirements for the system

1. The Hostel Office can register the caterer by entering the following details into the system:
   1. Caterer name
   2. Mess\_Code
   3. Mess Manager name
   4. Types of food provided in the mess(Special or (North/South) Indian (Veg/Non-Veg) or Chinese or Food Mall
   5. Certification
   6. Location
2. The Hostel Management can register a student in the mess by entering his following details on the system:
   1. Name of the student
   2. Registration No.
   3. Mess name
   4. Block
   5. Room no.
   6. Type of Mess(Special or (North/South) Indian (Veg/Non-Veg) or Food mall
3. The student can apply for a change of mess by entering the following details into the system before 25th of each month:
   1. New Mess name
   2. Type of Mess(Special or (North/South) Indian (Veg/Non-Veg)
4. The Hostel management will put up the mess menu online which could be seen by both the students and the mess supervisors.
5. The mess supervisor can change the mess menu at his discretion, but the changed item will be highlighted to the students. The students can still see the old item.
6. The students can review about the food quality and complain if the item was unavailable in the mess during that day.
7. The student can notify to the mess manager if he would be absent the next day in the mess thereby allowing the manager to make necessary food quantity adjustment and the student could be awarded a dessert or any such thing on five such notifications in a month, thereby reducing the wastage of food and increasing the profit of the mess.
8. The students registered under ‘Food Mall’ can preorder their food online by entering the item details and the time they want it on. The food mall manager can then verify the request based upon the availability of the item or whether it is possible to make the request in provided time. The students can also see the no. of current orders already in the queue to be processed.

1. **User characteristics**

The proper user interface and online help must be sufficient to educate the users on how to use the system without any problems. Mess managers without any basic knowledge of computer systems will be able to adapt to the form based user interfaces offered by the system with sufficient documentation and end user training.

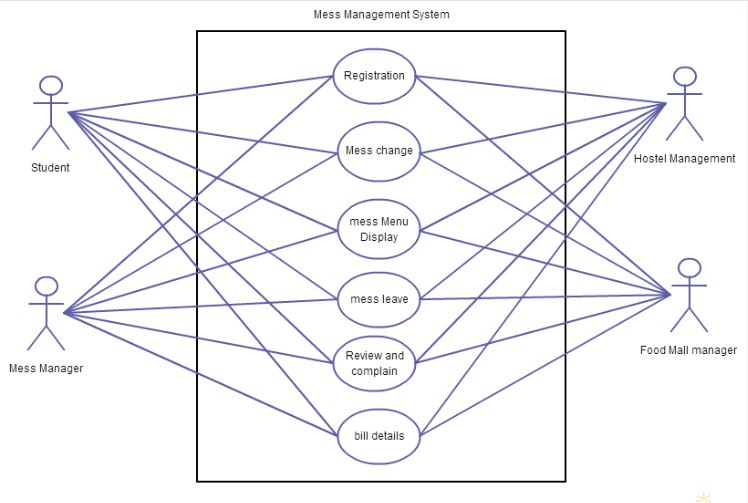
1. **Constraints** 
   1. Migration of all existing student data on paper to the database
   2. Training of all mess managers to get well acquainted with the UI of the software.
2. **Assumptions and dependencies**
3. Each mess is managed by a manager who represents a caterer.
4. All the mess and the students are enrolled by the hostel office.
5. The food menu is decided by the mess representative and the caterers.
6. The student cannot register for mess change after the first 25 days of each month.
7. The students cannot register for mess leave of less than 7 days.
8. The student and the mess information can be changed only by the hostel management.

# External Interface Requirements

1. **User Interfaces**
2. The system will provide GUI for the users.
3. The users will be able to access the system using their web browsers
4. **Software Interfaces**
5. The system will have an interface with the VIT Hostel Management system for validation of students.
6. **Communications Interfaces**
7. The system will use HTTP protocol for transfer of information between the client and the web server.

# Functional Requirements

1. **Use Case Diagrams**



.

1. **Use Case Scenario Description**

|  |  |  |
| --- | --- | --- |
| Use Case ID | 1 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Caterer Registration | |
| Summary | The System will update the database to hold the following details for a caterer   * Caterer name * Mess\_code * Mess Manager name * Types of food provided in the mess(Special or (North/South) Indian (Veg/Non-Veg) or Chinese or Food Mall * Certification * Location | |
| Preconditions | The mess manager should have a written permission from the hostel office to enroll in the system.  The hostel management officer entering the caterer details should be registered with the system | |
| Success End Condition | Caterer Registration Successful Confirmation Message is displayed to the user | |
| Failed End Condition | Please fill in the required fields Message to the User when any of the given fields is blanks or all zeroes | |
| Primary, Secondary Actors | Mess Manager, Hostel management, Food Mall manager | |
| Trigger | This use case is initiated based on the request from the hostel management officer registering the caterer | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the Caterer Registration link |
|  | 2 | The user enters the name, mess code, manager name, certification, location and in the form. |
|  | 3 | He selects the types of food provided by the mess in a checkbox input type. |
|  | 4 | The End User Clicks on the Submit Button |
| EXTENSIONS | Step | Branching Action |
|  | 1 | The system updates the data into the database.  The system updates the database with the timestamp. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 2 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Student Registration | |
| Summary | The System will update the database to hold the following details for a student   * Name * Reg. no. * Mess name * Block * Room no. * Type of mess | |
| Preconditions | The hostel management officer entering the student details should be registered with the system | |
| Success End Condition | Student registration successful message is displayed to the user. | |
| Failed End Condition | Please fill in the required fields Message to the User when any of the given fields is blanks or all zeroes | |
| Primary, Secondary Actors | Hostel management officer, student | |
| Trigger | This use case is initiated based on the request from the student to register for the mess. | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the student registration link |
|  | 2 | The End user enters the name, reg no., block, room no. into the system. |
|  | 3 | The end user selects the name of the mess from the dropdown list. |
|  | 4 | The end user selects the type of mess selected from the dropdown list. |
| EXTENSIONS | Step | Branching Action |
|  | 1 | The system updates the data into the database.  The system updates the database with the timestamp. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 3 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Mess Menu data entry | |
| Summary | The System will update the database to hold the mess menu for the whole month based upon the following fields   * Breakfast * Lunch * Snacks * Dinner | |
| Preconditions | The Hostel management officer who enters the Mess menu details should be registered with the system | |
| Success End Condition | Mess menu updated for the current month message will be displayed to the user. | |
| Failed End Condition | Please fill in the required fields Message to the User when any of the given fields is blanks or all zeroes | |
| Primary, Secondary Actors | Hostel Management officer | |
| Trigger | This use case is initiated based on the request from the hostel management officer entering the menu | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the Update Mess Menu link |
|  | 2 | He enters the menu in the breakfast, lunch, snacks and dinner textbox for each day of the month. |
|  | 3 | The end user clicks the update button to update the data in the table. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 4 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Search student details | |
| Summary | The hostel management officer and mess manager can view or search for a particular student by entering the student’s Reg. no or name. | |
| Preconditions | * The hostel management officer should be registered with the system * The student must be registered with the system | |
| Success End Condition | The search results are displayed to the user if any. | |
| Failed End Condition | The student is not enrolled into the system. | |
| Primary, Secondary Actors | Hostel Management officer, Mess Manager | |
| Trigger | This use case is initiated based on the request from the hostel management officer or mess manager to view student details. | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the student search link. |
|  | 2 | The End user Selects the parameter to search the student upon, for instance student first name or last name or his reg. no. |
|  | 2 | The end user enters the selected information and presses search button |
|  | 3 | The details of the registered student is displayed to the user or no students found message is displayed. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 5 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Modify student details | |
| Summary | The System will update the database to hold the following modified details for a student   * Name * Reg. no. * Mess name * Block * Room no. * Type of mess | |
| Preconditions | * The hostel management officer should be registered with the system * The student must be registered with the system | |
| Success End Condition | Student Details Modification Successful Confirmation Message is displayed to the user | |
| Failed End Condition | * Please fill in the required fields Message to the User when any of the modified fields is blanks or all zeroes * Invalid Reg. no. | |
| Primary, Secondary Actors | Hostel management officer | |
| Trigger | This use case is initiated based on the request from the Police Officer modifying the crime details | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the “modify student details” link on the details page obtained after student search. |
|  | 2 | The user enters the modified details into the system. |
|  | 3 | The user clicks the update button |
|  | 4 | The end user presses ok on the confirmation dialog box. |
| EXTENSIONS | Step | Branching Action |
|  | 1 | The modified details get updated into the database. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 6 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Mess change | |
| Summary | The student can apply for mess change by entering the following details into the system:   * New mess name * Type of mess taken | |
| Preconditions | The student should be registered with the system. | |
| Success End Condition | Applied for mess change message displayed to the user. | |
| Failed End Condition | The following message is displayed to the user, “you can apply on first 25 days of the month only.” | |
| Primary, Secondary Actors | Student | |
| Trigger | This use case is initiated based on the request from the student to change the mess. | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the ‘Change the mess’ link on his home page |
|  | 2 | He enters the following details in the following form:   * New mess name * Type of mess |
|  | 3 | The user clicks the apply button. |
|  | 4 | A success message is displayed to the user |
| EXTENSIONS | Step | Branching Action |
|  | 1 | The database is updated with the reg. id of the student, his current mess id and new mess id. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 7 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Mess Leave | |
| Summary | The student can apply for a leave from mess for greater than or equal to 7 days by selecting the start date and end date from the calendar. | |
| Preconditions | The student must be registered with the system.  The leave days should be greater than or equal to 7 days. | |
| Success End Condition | Mess leave application accepted | |
| Failed End Condition | The no. of days is less than 7. | |
| Primary, Secondary Actors | Student | |
| Trigger | This use case is initiated based on the request from the student to take a leave. | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the mess leave link. |
|  | 2 | * He selects the start date from the calendar drop-down. * He selects the to date from the calendar drop down. * He clicks apply button. |
|  | 3 | A confirmation dialog is displayed to the user displaying him the total no. of days he needs the leave for. |
| EXTENSIONS | Step | Branching Action |
|  | 1a | The database is updated with the user request. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 8 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Absent notification | |
| Summary | The student can notify the mess manager if he would not be coming for a particular time during the mess slots. | |
| Preconditions | The student must be registered with the system.  He can register him absent for the nex to next slot onwards.eg till 2’o clock(lunch slot), he could mark him absent for the dinner slot. | |
| Success End Condition | Absent request granted | |
| Failed End Condition | You cannot register to be absent today. | |
| Primary, Secondary Actors | Student, Mess Manager | |
| Trigger | This use case is initiated based on the request from the student to remain absent for a particular time slot. | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the ‘notify absence’ link. |
|  | 2 | * He selects the time slot he would be absent for. |
|  | 3 | A confirmation dialog is displayed to the user accepting his request. |
| EXTENSIONS | Step | Branching Action |
|  | 1a | The database is updated with the user request. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 9 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Food quality review system | |
| Summary | Every student can rate each item each day which he has already taken based upon 3 categories:   * Good * Bad * Awesome   He can also issue a written complain regarding unavailability of the food item or its bad quality. | |
| Preconditions | * The student registering the complaint or reviewing the food item should be registered with the system. * He can only review food items in his own mess. | |
| Success End Condition | Your response has been acknowledged message will be displayed to the user | |
| Failed End Condition | * You have not selected any option * You have not written anything in the complain box. | |
| Primary, Secondary Actors | Student | |
| Trigger | This use case is initiated based on the request from the Student who wants to complain or review the provided food. | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user is displayed with previously unchecked radio buttons having following options:   * Good * Bad * Awesome   Or he could click the report button to lodge his complain in the provided text box. |
|  | 2 | He selects any one of the provided options and clicks on the submit button |
|  | 2 | A pop up message is displayed to him saying that his response has been recorded. |
| EXTENSIONS | Step | Branching Action |
|  | 1a | The database is updated with the Reg. no. of the student and his ratings for a particular food item. |

|  |  |  |
| --- | --- | --- |
| Use Case ID | 10 | |
| [Use Case name](http://requirmentengineeringsrs.blogspot.in/) | Generation of mess bill | |
| Summary | The System will update the database at the end of the month displaying the no. of days each student has eaten in the mess. | |
| Preconditions | * The student should be registered with the system | |
| Success End Condition | A report will be generated displaying the the no. of days the student has eaten in the current month and the balance amount would be displayed. | |
| Failed End Condition | * The student is not registerd with the system * The student does not belongs to the particular mess. | |
| Primary, Secondary Actors | Mess Manager, Student, Hostel management officer | |
| Trigger | This use case is initiated based on the request from the Mess manager to generate the bill details for each students at the end of each month. | |
| DESCRIPTION | Step | Action |
|  | 1 | The End user selects the generate bill link. |
|  | 2 | The system displays the name of each student and no. of days he attended the mess in the particular mess |
|  | 4 | The hostel management officer can click submit to save the details in the database. |
| EXTENSIONS | Step | Branching Action |
|  | 1 | The modified details get updated into the database. |

# Non Functional Requirements

This section describes in detail all the non-functional requirements

## Usability

* 1. The system shall allow the users to access the system from the Internet using HTML. The system uses a web browser as an interface.
  2. The end users will be able to able to adapt to the system with a minimum training of 4 hours
  3. Key board shortcuts will be available for all functions of the system.

## Security

1. Login requirements -
   1. Hostel Management officers will be provided access to the system after they are provided the main username and password provided by the developer.
   2. Students can login into the system using the reg no. and password provided to them by email.
   3. Mess managers can login into the system by entering the mess id and password provided to them by the hostel management office.
2. Password requirements
3. Passwords must have a minimum length of 6 characters.
4. Passwords must meet at least 3 out of the 4 requirements for quality:
   * at least 1 lower case letter
   * at least 1 upper case letter
   * at least 1 number.
5. **Inactivity timeouts**
6. System should timeout when there is no activity for fifteen minutes.

## Performance

1. Response time
2. The response time will be less than 3 seconds for 95% requests made to the system depending upon the net speed of the user.

## Capacity

1. Throughput
2. The application shall be able to successfully handle 500 requests per hour.
3. Storage
4. Hard disk space –

450 GB – Content

50 GB – Transaction Logs

## Recovery

1. Recovery time scales
2. The system will be recovered within four hours from the down time
3. Backup Frequencies
4. The system would be backed up once per month

## Availability

1. Hours of operation
2. The system will be available on all days 24\*7

## Reliability

1. Mean Time between Failures
2. The mean time between failures for the system will be 15days.

## Maintainability

1. Mean Time to Recovery
2. The Mean Time To Recovery (MTTR) shall not exceed one person day.

## Portability

1. The system will run on windows 95/98/2000/NT/XP/Vista/Windows7/Windows 8/Mac/Linux or basically any device supporting a web browser and connected to the internet.

## Privacy

1. Any information can be changed by the hostel management officer.
2. The student cannot make any change to his information details.
3. The Mess manager cannot many any changes to his information as stored in the database.
4. The mess manager cannot make any changes to the student information.
5. The mess manager can update the mess bill details.
6. The student cannot see the details of other students.