
Software Requirements Specification

for

MEAL-On

Prepared by: 1)Ankita Vaid 2) Akanksha Mahajan 3)Musadiq Showkat

College of Engineering, Pune.

13-02-2020

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Stakeholders.....	1
1.4 Definitions, Acronyms and Abbreviation	1
1.5 References.....	2
2. Overall Description.....	2
2.1 Product Perspective.....	2
2.2 Architecture	2
2.2 Technology Stack	3
2.3 Dependencies and Assumptions	3
2.4 User Classes and Characteristics	3
2.5 Design and Implementation constraints	5
3. Specification Requirements.....	5
3.1 External Interface Requirements.....	5
3.2 Functional Requirement.....	6
4. Non-Functional Requirements.....	13
4.1 Performance	13
4.2 Usability Requirements.....	13
4.3 Security Requirements	13
4.4 Reliability Requirements	13
4.5 Availability Requirements	13
5. Rules	14

1. Introduction

1.1 Purpose

- Hostel Mess has to deal with on-paper manual work.
- Mess Bills are calculated in the big registers.
- Kitchen inventory is managed by the working staff.
- With all these considerations, digitalizing the mess related laboring task is the sole purpose of this project.

1.2 Scope

- The software system will consist of two separate components.
 - Mess admin side component
 - Customer side component
- The Mess admin side component will be able to manage the customers, their bills, distribute food menu, predict the number of plates to be made for lunch/dinner and manage the inventory stock inside Mess.
- Customer side will be able to see the menu, give feedback/suggestions and mark if they would attend the Mess.

1.3 Stakeholders

- Mess manager
- Mess members
- Students (mess members)

1.4 Definitions, Acronyms and Abbreviation

- SRS: System Requirement Specification
- JSP: Java Server Page
- SQL: Structured Query Language

1.5 References

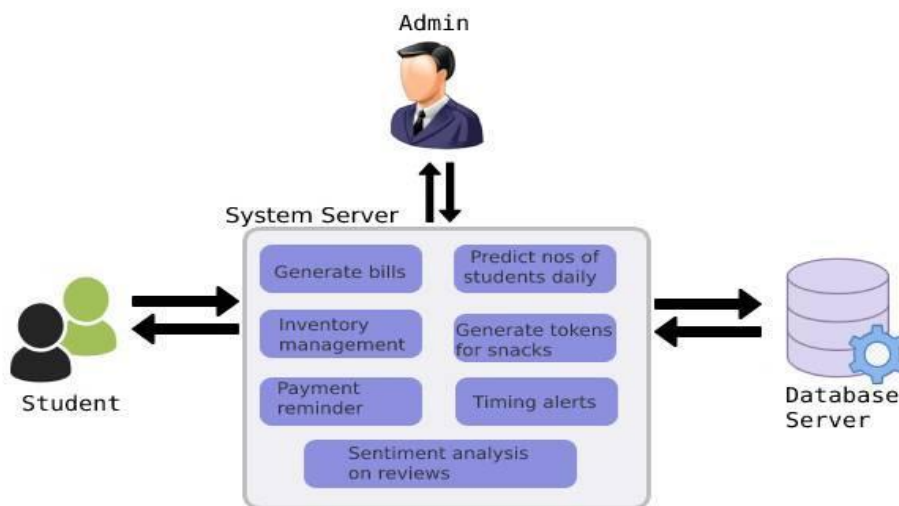
- tgmc-2008 sample synopsis:
https://www.ibm.com/developerworks/community/files/form/anonymous/api/library/3cf803c7-f973-4051-99a8-2949fd4ceab1/document/e6bd87f7-a158-4997-b5b6-2a8e2f3e11ed/media/SRS_Sample.doc
- <https://www.ijert.org/research/mess-management-system-implementation-IJERTCONV3IS24003.pdf>
- https://www.tutorialspoint.com/adaptive_software_development/sdlc_iterative_incremental_model.htm

2. Overall Description

2.1 Product Perspective

Digital Mess will completely be an independent system. Our product will not be part of any other system. It has two user interfaces i.e. Student and Mess Admin. The students authenticated by mess admin will use web portal / Android application on which they can login with unique ID and password and avail all the facilities. The Mess Admin has a different interface on his side with login ID and password. A relational database is used to store all the data like students list, employees list, previous data about number of students (for ML prediction) attending mess on corresponding dates, feedback, inventory data, etc.

2.2 Architecture



2.3 Technology Stack

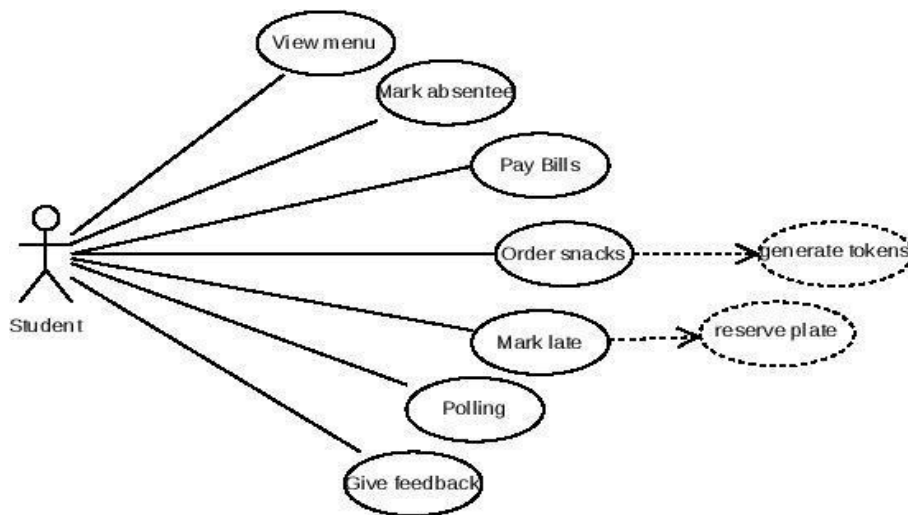
- Frontend: React
- Backend: Java SpringBoot
- Database: MySQL
- Android

2.4 Dependencies and Assumption

1. Previous year dataset about daily presence of students in mess for prediction.
2. Roles and works are predefined
3. System will work only according to the rules of COEP Mess.

2.5 User Classes and characteristics

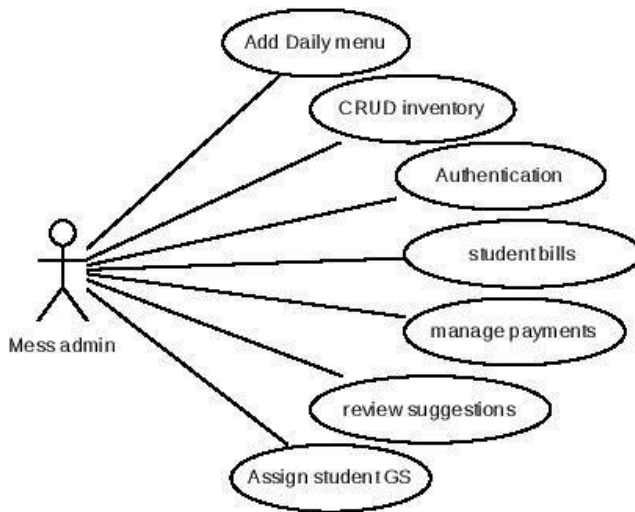
1. Student:



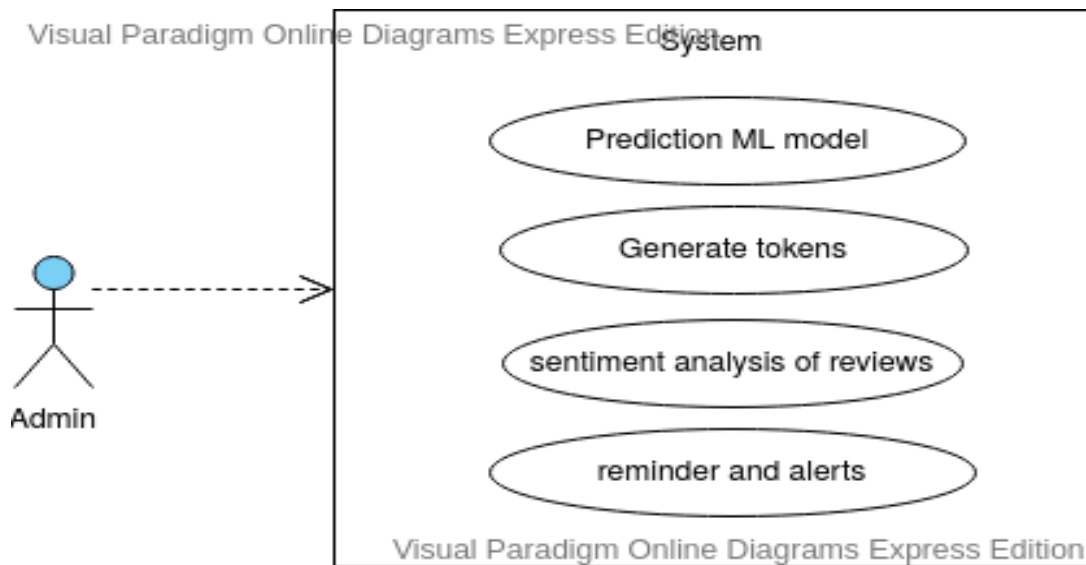
1. Students can view the menu before every meal and mark themselves as present/absent within the time bound.

2. Students are notified about the monthly bills and deadline. In fact, there is a payment gateway to pay bills.
3. Daily feedback about each food item can be given by the student.
4. Suggest improvements in the functioning of the mess system and food quality.
5. Student polls are considered to be chosen by the secretary of mess.
6. For snacks, students order the food online and receive the token. Double spending problems will be avoided completely.
7. Token numbers will be string of combination of date, time and id number.

2. Admin:



1. Admin shall add daily menu which is notified to the students daily and
2. Admin shall manage the inventory and keep track of the amount of raw materials available in storage.
3. Admin shall authenticate students and employees.
4. Admin shall manage student bills and salary of employees.
5. Admin can review suggestions given by students directly.
6. Admin shall have final authority to choose student secretary.
7. Admin shall maintainer of system where system provide following functionalities:
 1. Predict the number of students that will be present for the meal by analysing past data and menu of meal.
 2. Remind students about the payments.
 3. Generate tokens for snacks.
 4. Sentiment analysis on reviews given by students



2.6 Design and Implementation constraint

- The developed system should run under any platform i.e. Windows, Linux, Mac OS.
- Details provided during signup are stored.
- There can be security risk involved
- Student details can further be changed only by admin
- Time constraint

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Mess Admin(s)** : The interface for mess admins will be in the form of a web-app. It will be designed to be more functional with simplicity and simple understandable minimal styling so that it is convenient for mess administrator to operate with minimal discomfort.
- **Students** : The interface for students (users) will be an android app.

3.1.2 Hardware Interfaces

A webserver will be used to connect mess admins to students for exchange of information. The server will be hosted on AWS. The server will have a database to store the data.

3.1.3 Software Interfaces

The webserver will be hosted using Apache2. It will also have a MySQL database to store data will be used to connect, store and retrieve data from database.

3.1.4 Communication Interfaces

The communication protocol used will be HTTP(s). It will be used to transfer information back and forth from client to the server. HTTP GET and POST will be used.

3.2 Functional Requirements

3.2.1 Students

3.2.1.1 Student Login

Use Case Name	Login (first time)
Trigger	The Student opens application.
Precondition	The Student has entered his login details on the login page.
Basic Path	<ol style="list-style-type: none">1. The student navigates to the login page.2. The student enters the username and password.3. The student clicks the login button.4. Data is sent to the server.5. The server will compare the login data with the password stored in the database.6. If login credentials are verified, the student is logged in.7. If not, the student is prompted to enter the login details again.

Post Condition	The Student is logged in and is taken to his account page.
Exception Path	The Student may terminate the login at any time.

3.2.1.2 Change Student Password

Use Case Name	Change Password
Trigger	The student clicks on the change password button on his account page.
Precondition	The student is logged in.
Basic Path	1. The student enters the previous password and the new desired password. 2. The data is sent to the server if it is not empty. 3. The server validates the user's password and then updates his password in the database.
Alternative Paths	The student may click on the forgot password link on the login page to have change his password after verification.
Post Condition	The user's password is updated in the database.
Exception Path	The attempt may be abandoned at any time.

3.2.1.3 View Menu

Use Case Name	View Menu
Trigger	The student clicks the view menu button in home screen.
Precondition	The student is logged in.

Basic Path	1. The student can choose which day's menu he wants to view. 2. The choice is sent to the server where the database is queried for that particular menu.
Alternative Paths	The student can also choose to view the breakfast menu along with the cost of the food items.
Post Condition	The Menu is displayed on the screen.
Exception Path	The student may abandon the operation at any time.
Other	Menu consists of menu items and their details.

3.2.1.4 Order Breakfast

Use Case Name	View Menu
Trigger	The student clicks the order breakfast button in home screen.
Precondition	1. The student is logged in. 2. Time is from 7:30 AM to 9:00 AM(Breakfast) or 3:30 PM to 5:30 PM (supper – not on Sundays)
Basic Path	1. The student can choose the items he wishes to order. 2. The choice is sent to the server where the database is queried for those items.
Alternative Paths	The student can also check if an item is available or not.
Post Condition	The student gets a token number for his order (if items available).
Exception Path	The operation can't be cancelled midway.
Other	Orders once confirmed can't be revoked.

3.2.1.4 Mark Absentee

Use Case Name	Mark Absentee
Trigger	The student clicks on Mark Absentee button in his/her account page.
Precondition	1. The student is logged in. 2. Time is from before 10:00 AM(for lunch) or 4:00 PM (dinner – not on Sundays)
Basic Path	1. The student can inform mess admin that he will not come for lunch / dinner. 2. The request will be sent to the server and the database is updated accordingly.
Alternative Paths	The student can also check if an item is available or not.
Post Condition	The student gets a token number for his order (if items available).
Exception Path	No Exception
Other	Student can revoke absentee only if mess admin accepts it.

3.2.1.4 Pay Bill

Use Case Name	Pay Bill
Trigger	The student clicks on the pay bill button on his account page.
Precondition	The student is logged in.
Basic Path	The student chooses the option he wishes to pay with.
Alternative Paths	The student will be redirected to his chosen gateway.

Post Condition	The student's bill amount is updated in the database.
Other	In case of any issue, the mess admin needs to verify the payment.

3.2.1.5 Daily Review

Use Case Name	Daily Review
Trigger	The student clicks on the Give review.
Precondition	1. The student is logged in. 2. The student has had food.
Basic Path	1. The student gives reviews and ratings. 2. The reviews are stored in database.
Post Condition	Mess admin gets an overview of the review.

3.2.1.6 Reserve Plate

Use Case Name	Reserve Plate
Trigger	The student clicks on the reserve plate.
Precondition	1. The student is logged in. 2. It should be done before 30 minutes from closure of mess.
Basic Path	1. The student requests to reserve plate. 2. The mess admin receives an notification about it.
Post Condition	A plate is kept aside for student.
Other	The request can be rejected by the admin, depending on availability of food.

3.2.2 Mess Admin

3.2.2.1 CRUD Inventory

Use Case Name	CRUD Inventory
Trigger	The admin clicks on CRUD inventory.
Precondition	The admin is logged in.
Basic Path	1. The admin updates inventory stock. 2. The stock is updated in the database.
Alternative Path	The admin can also add new items in the database. Also the mess admin needs to update prices as they change.
Post Condition	Mess admin can see updated database.

3.2.2.2 Verify Students

Use Case Name	Verify Students
Trigger	The admin clicks on verification portal.
Precondition	The admin is logged in.
Basic Path	1. The admin verifies the students in that particular mess. 2. The database is updated accordingly.
Post Condition	The verified students are added to that mess.
Other	Only verified students can give reviews and ratings.

3.2.2.3 Give Mess Menu

Use Case Name	Give Mess Menu
Trigger	The admin clicks on give menu.

Precondition	The admin is logged in.
Basic Path	1. The admin gives the mess menu. 2. The menu is updated in the database accordingly.
Post Condition	The students can see the menu.
Other	There may be some no non-veg days.

3.2.2.4 Check Daily Ratings and Reviews

Use Case Name	Check Reviews and Ratings
Trigger	The admin clicks on feedback button.
Precondition	The admin is logged in.
Basic Path	The mess admin clicks to view review and ratings.
Alternative Path	The admin may choose to view ratings of particular day/ food only.
Post Condition	The admin is given summary of the review and daily ratings.

3.2.2.5 Add Employees

Use Case Name	Add Employees
Trigger	The admin clicks on manage employees.
Precondition	The admin is logged in.
Basic Path	The mess admin adds employees, their details.
Alternative Path	The admin may edit details of employees or delete existing employees
Post Condition	The admin is given summary of the review and daily ratings.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

- The information is refreshed when some update in the data has been made.
- The system shall respond as fast as possible from the time of the request submittal.
- The system shall be allowed to take more time when doing a large processing, e.g. redirection to payment portals

4.2 Usability Requirements

- The system shall allow the admins and students to access the system from the browser.
- Students shall be provided with android application for handy access.

4.3 Security Requirements

- Other students will not be able to mark absentee/pay bills for other student unless account access is given.
- Tokens once generated will not be able to replicate it without another payment.
- Money cannot be refunded after generating tokens.

4.4 Reliability Requirements

- The system will be 100% reliable.
- Predicting the amount of food to be made on each day might not accurate for initial phase due learning phase.

4.5 Availability Requirements

- The system will be available for 24 hours – 7 weeks – 365 days.
- For updating the system later, it will be down for maintenance purpose for about 1-2 hours at mid-night.

5. Rules

- Mess admins must keep in mind that the prediction of number of meals to make each day might differ.
- Admins should authenticate students to board on the portal.
- The Student can login only after he is verified by mess admin.
- The student can give feedback only if he has had food.
- A student can mark himself absent before four hours from dinner/ lunch.
- Tokens for breakfast/ supper once generated can't be cancelled.