Software Requirements Specification

for

Officer’s Mess Management System

Version 1.0 approved

Prepared by

Zannatul Tazree Ramisa (2054901004)

Fariha Nishat (2054901008)

Jafrin Akter Alo (2054901035)

Muhammad Mahbub Sarwar Shafi (2054901049)

Date: 20/08/2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1. Purpose

The main objective of this document is to give a detailed overview of our software system named "Officer's Mess Management System". This document describes the functional and non-functional requirement of the software describe by the client. This SRS will provide the foundation for the project. From this SRS the Officer's Mess Management System can be designed, implemented, and finally tested. This document is the contract between the user and the organization. It also works as a basis for validation and acceptance of the software system.

1.2. Document Conventions

- Entire document should be justified
- Convention for chapter title
  - Font face: Times New Roman
  - Font style: Bold
  - Font size: 16
- Convention for sub-title
  - Font face: Times New Roman
  - Font style: Bold
  - Font size: 14
- Convention for running text
  - Font face: Times New Roman
  - Font size: 12

1.3. Intended Audience and Reading Suggestions

This is a technical document this document is intended to be read by developers to overview the tasks that must be performed during the development of the software system, client for having a detailed overview on the software system, then by project managers for quality project planning to processed further, after that tester to ensure that the software created by developers is bug free or not.

1.4. Project Scope

The Officer's Mess Management System will be used to deal with the problems on managing mess activities and avoid the problems which occur when carried manually. It is used to improve clerical services for the employers(regular/irregular) who takes the services from mess department. The system will help the admin to maintenance of the food management, taking the order from the regular and irregular customer, mess menu,

maintenance of the bill and dues record. This will also help employees to check mess menu and mess fees and dues. Employees will get the notification of their activities every time.

1.5. References

- Software requirement specification:
  https://www.scribd.com/document/107024959/Mess-Management-System#user-util-view-profile
  Visited Date: 20/08/2022

2. Overall Description

2.1. Product Perspective

Officers' Mess Management by manual way is a monotonous process, since it involves a lot of work pressure and commitment of time. In the manual way, the managers must write cash receipts for every order, keep track of inventory, and manage the monthly billing system all by themselves repeatedly. This often results in different types of errors as well. Like- people with similar names can get their monthly bills exchanged or one person can get billed twice. The Officers' Mess Management System is an independent stand–alone system. With this automated system, we can easily manage the mess activities, manage inventory, billing system and receipts etc. Daily order receipts will be generated automatically when the customers will order, and monthly bills will also be generated at the end of the month and sent to the correct customers. So, the effort and time needed will decrease dramatically and chance of errors happening will be gone.

2.2. Product Features

In this software, the following activities will be available-

- Login system for both user(clients/customers) and admin(managers).
- Users can sign up or register as either regular or non-regular customers. The admins will approve of them.
- The menu can be updated by the admins for every mealtime(breakfast/lunch/tea-break).
- Users can request meals using the 'request meal' button during a certain time allotted by the officers' mess.
- Order receipts will be auto generated after the user has finished ordering. The bill will also be added in the database to the customer's id.
- A monthly bill will be generated by the system and sent to the user/customer.
- After the customer has paid their dues, the admin will press the 'done' button in the bill.

2.3. User Classes and Characteristics

Officers mess management system will support two types of users- user(clients/customers/officers) and admin(managers). Officers will have access to user module which will also be divided into two categories, which are- regular and non-regular customers. Regular customers are those officers who will have food regularly in the mess and non-regular customers will be those officers who will have to notify in the system within a certain time before coming for a meal. The mess managers will have access to the admin module.

2.4.Operating Environment

The officers mess management system is a software which will operate in all widely known browsers in the world, i.e., Google chrome, Mozilla Firefox etc. Operating environment of the system will be as stated below:

Operating system: Windows 7/8/10/11

Database: MySQL database

Platform: HTML/CSS/JavaScript/Bootstrap/PHP/Laravel

2.5.Design and Implementation Constraints

The officers' mess management system can have the following constraints:

- Development time: The project must be finished with the features as the client suggested within the allotted time.
- Operating system: The software will be compatible with the widely used Windows operating system.
- Web-based project: As it is a web-based project, so it needs a working browser to access the software and use it.
- Internet connectivity: The software needs internet connection for access and usage.
- MySQL server: The software uses MySQL server to store all the information including user information, menu items, menu updates, login id and password etc. So, MySQL is needed as SQL engine and database.
- Password protection: users must register and be approved by the admins in order to use the software. They will also need id and passwords to login.
- Error-free: The system should be able to handle the features accurately without any errors in code.

2.6. User Documentation

- Login/ register:

  There will be an option for register as a customer in the homepage of the website. The officers can register as either regular or non-regular customer. Regular customers are those officers who will have food regularly in the mess and non-regular customers will be those officers who will have to notify in the system within a certain time before coming for a meal. After registering, an admin will approve of the user. After that, the officer can login through the homepage and use the website's services.

- Request meal:

  There will a request meal button in the homepage. The users can use the button to request meals throughout the 24 hours. There are certain time criteria for certain meal requests, so if the user tries to request meal in the wrong or exceeded time limit, it will show 'service not available at this moment'. The user can request breakfast, lunch, or tea-break meals throughout the day. Breakfast and lunch requests are mandatory for non-regular users. But tea-break request is not mandatory.

- Generate daily/monthly receipt:

  The daily or monthly receipt can be generated using this feature.

- Admin module:

  Admins can login as admin in the website and approve meals, update menu, item prices and check 'done' on monthly bill receipts when the officers pay their dues for the month.

2.7.Assumptions and Dependencies

The assumptions and dependencies of officers' mess management system include:

- MySQL server is needed to store all the information in database.
- A certain life cycle model should be followed throughout the project.
- The developers need to have experience with the developing languages.
- Working browsers and internet connection is needed to access and use the software.
- Operating system will need to be compatible with the software for access.
- The project will be developed from re-used or modified codes from other implemented projects.

3. System Features

3.1. Access Management (Client)

3.1.1. Sign-Up

A new client(officer) can avail the application's benefits by signing up. The client has to fill up the form with necessary details. And hit the "Sign-up" button. Email verification would be done by the application. Admin side would verify the client's authenticity and client would be added to the database.

3.1.2. Login

Clients can login in their respective accounts using their provided email address and password. If password forgotten, new password can be set using phone number.

3.1.3. Response Sequences

- Client Sign-Up
  Client during sign up enters email and necessary information→Email verified → New account opening request sent to admin.
- Client-side Login:
  Client enters email and password→verified account→access account.
- Client enters email and password→not verified account→cannot access account.
- Password forgotten→OTP sent to verified email→set new password

3.1.4. Functional Requirements

REQ-1: Client can sign up
REQ-2: Verify email
REQ-3: Login
REQ-4: Recover password
REQ-5: Remember Login

3.2. Access Management (Admin)

3.2.1. Login

Admin side can easily login entering the set email address and password.

3.2.2. Response Sequences

Admin enter password and email→ Correct password and email →Logged in admin side.

3.2.3. Functional Requirements

REQ-1: Login

3.3.Non-regular Customer Meal Request

3.3.1. Meal Request

Non-regular customer can request a meal (breakfast or lunch). Breakfast request can be made prior a day and Lunch request can be made before 9A.M. the same day. client can request meal beforehand by selecting dates.

3.3.2. Response Sequences

Client side request a meal→ admin side would confirm→confirmation email would be sent, and client can see the update in his meal history along with the billing.

3.3.3. Functional Requirements

REQ-1: User can browse meal
REQ-2: User can confirm a meal
REQ-3: Get the billing

3.4. Regular Customer Meal Management

3.4.1. Subscription Cancellation

Regular customer can cancel their meal subscription. The user would be then considered as non-regular client. Can request extra meal.

3.4.2. Response Sequences

- Client side turns of regular meal→ Vendor side would confirm→confirmation email would be sent.
- Client-side profile would be updated.

3.4.3. Functional Requirements

REQ-1: Cancel subscription

3.5. Extra Meal Request and Tea break

3.5.1. Extra meal request

Client can request extra meal to be prepared for guests he would be bringing. Or bring guests who would be served tea break items.

3.5.2. Response Sequences

- Client-side choses number of guests and time and date → admin side would confirm →confirmation email would be sent with billing.
- Client brings guests→Tea break items served→Bill added to client account and emailed.

3.5.3. Functional Requirements

REQ-1: Client can confirm extra meal
REQ-2: Billing information can be seen

3.6. Month End Billing (Client)

3.6.1. Month end Bill receipt

Client would be sent an automated receipt of officer's mess bill. Due bills (if any) would be added to the current bill. If payment done, confirmation bill paid email would be served.

3.6.2. Response Sequences

Client-side receipt sent→ Bill paid→Confirmation email

3.6.3. Functional Requirements

REQ-1: Client can see his due bill.

3.7. Billing (Admin)

3.7.1. Month End Summary

Admin can see all the clients who didn't clear their bill. Admin can select clients who have more than one month of due bill. Can generate bills for specific dates. Admin can check whether client paid or didn't pay their bills.

3.7.2. Response Sequences

- Admin views bill→sees total dues and total paid bill
- Admin chooses a client and date (from and to)→generates bill

3.7.3. Functional Requirements

REQ-1: Admin can view total due bill
REQ-2: Admin can view client status of bill (paid or unpaid)

3.8. Meals and Pricing (Admin)

3.8.1. Update Meal and Pricing

Admin can update their meal menu and price. According to the price, billing would be done.

3.8.2. Response Sequences

- Admin updates meal→database updated→client can see new meal menu
- Admin updates price of meal→database updates→billing done according to the new price.

3.8.3. Functional Requirements

REQ-1: Admin can update meal and price of each meal.
REQ-2: Bill is auto-updated

3.9. Request Management (Admin)

3.9.1. Meal Request

Admin can see total number of requests for breakfast and lunch per day.

3.9.2. New Client Request

Admin would be notified of a new client and after admin verifies the new client, the client account would be enabled.

3.9.3. Response Sequences

- Admin views meal request→confirms request→ views total number of request that day.
- Admin views new client request →verifies client and confirms→Client account activated

3.9.4. Functional Requirements

REQ-1: Admin can view total number of meals he needs to serve.
REQ-2: Admin can accept new clients.

4. External Interface Requirements

4.1. User Interfaces

- Sign-Up Interface:

Client is asked to enter ID number, name, email address and a password. If invalid email is entered, error message is shown. If ID number is already registered, error message is shown.

- Logging Interface:

Users are asked to enter email address and password. If the input values do not match the database result error message is shown.

- Requesting meal interface (non-regular client):

Client can see meal menu and request accordingly. And can request meal in advance. Moreover, client can request extra meals. Client can also apply for being "regular-client".

- Requesting meal interface (regular client):

Client can request meal in advance and client can request extra meals. Here client cancel subscription of his regular meal.

- Billing interface (Client):

History of meals requested, and date is shown along with billing information.

- Billing interface (Admin):

Due bills with client names are shown. Admin can select client and date then see his bill specifically.

4.2. Hardware Interfaces

The system shall run on a Microsoft Windows based system

4.3. Software Interfaces

The system shall interface with an Oracle or Access database

4.4. Communications Interfaces

This system will use some communication protocols such as HTTP and SMTP. HTTP protocol will be used to browse the website and use the database by the user, admin, and service providers. The SMTP protocol may be used to communicate with the users to via email where the transaction history may be provided to the non-regular users.

5. Behavioral Requirements

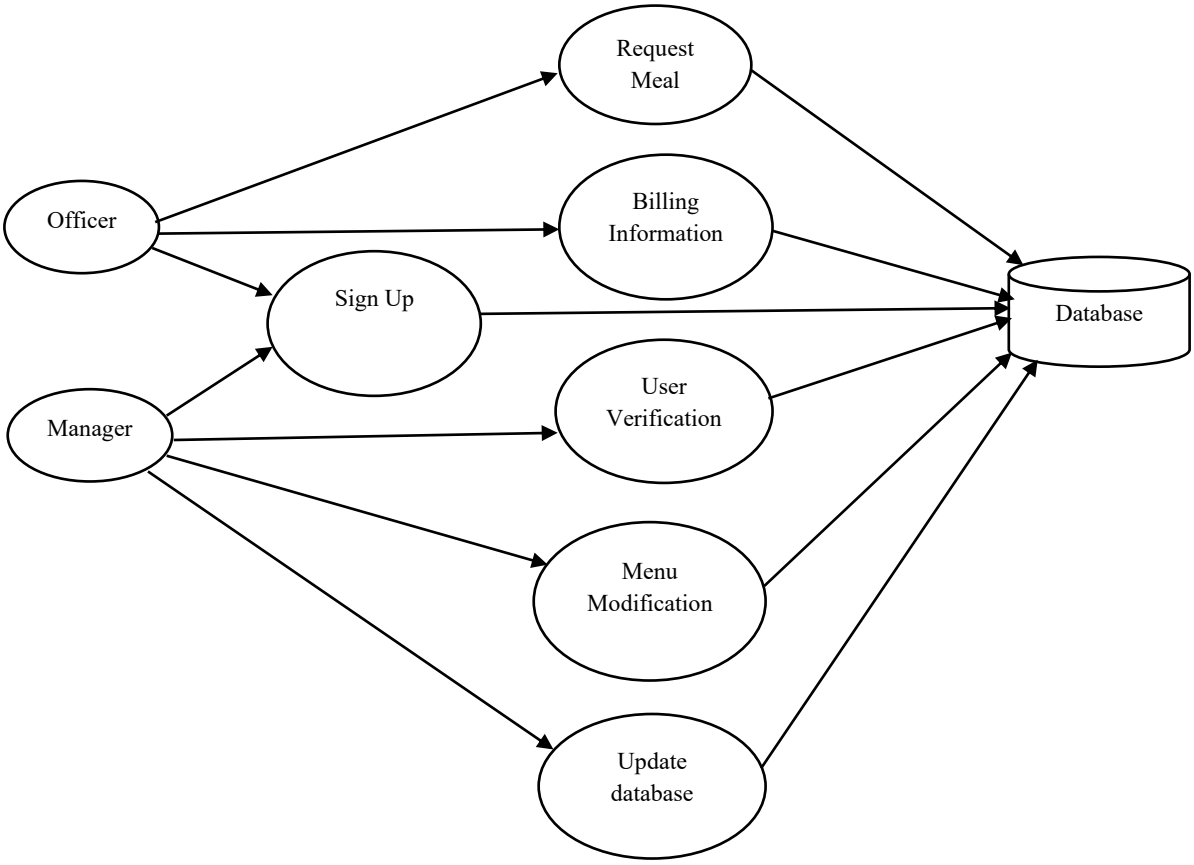

Figure: Officers Mess Management System

6. Other Nonfunctional Requirements

6.1. Performance Requirements

This defines the acceptable response time for system functionality.

- The loading time for user interface should not take so much time.
- The login information should be verified within few seconds.
- The system should change few functionalities over time.
- The system should be able to handle large amount of data and users.

6.2. Safety Requirements

The data can be backup from the system in timely manner for the security of data. Administrators can backup the log files or can be schedule backups monthly. Non-regular users can get notifications each time they use the service through email.

6.3. Security Requirements

This system will be web application based so

- System will use secured database.
- System will have different types of users and every user has access constraints.
- Normal users can read information, but they can't edit or modify anything except their personal and some other information.
- Proper user authentication should be provided.
- There should be separate accounts for admin and users such that users can only order or mark him/herself as non-regular or regular customer but can't access database.
- Admin has the access to the database and can accept the orders from the users.

6.4. Software Quality Attributes

- **Maintainability:** This system is developed in modular way so it can be maintained easiest way.
- **Availability:** The system will be available 24/7 but services will be provided based on mess timing.
- **Reliability:** The system will be 100% reliable and data will be stored safely.
- **Accessibility:** This system is web application so admin, students and service providers will be able to access it when needed.
- **Flexibility:** The system can be understandable to the users can be used faster and easily.
- **Correctness:** This system will maintain the database in correct way thus users' transactions won't conflict.

Appendix A: Glossary

The following are the list of conventions an acronym used in this document and the project as well:

- Administrator: A login id representing a user administration privileges to the software.
- User: A general login id assigned to most users.
- Client: Intended users for the software.
- SQL: Structured Query Language; used to retrieve information from a database.
- MySQL: A SQL server; used to store data in an organized format.