

**Software Requirements Specification
for
Mess Refund System**

Version 3.0

**Harshavardhan G
Manas Sanket Kinkar
D Praneetha**

**Submitted in partial fulfillment
Of the requirements of
IT350 Software Engineering
31 January 2019**

Table of Contents

Table of Contents	2
Revision History	3
1. Introduction	4
1.1. Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience and Reading Suggestions	4
1.4 Scope of Project	4
1.5 References	5
2. Overall Description	5
2.1 Product Perspective	5
2.2 Product Functions	5
2.3 User Classes and Characteristics	5
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	6
2.6 User Documentation	6
2.7 Assumptions and Dependencies	6
3. External Interface Requirements	7
3.1 User Interfaces	7
3.2 Hardware Interfaces	7
3.3 Software Interfaces	7
3.4 Communication Interfaces	8
4. System Features	8
4.1 QR Code Scan	8
4.1.1 Description and Priority	8
4.1.2 Stimulus/Response Sequences	8
4.1.3 Functional Requirements	9
Major functions of the Login and Registration System	9
4.2 System Feature 2 (and so on)	9
5. Other Nonfunctional Requirements	9
5.1 Performance Requirements	9
5.2 Safety Requirements	9
5.3 Security Requirements	10
5.4 Software Quality Attributes	10
Availability	10
Security	10
Usability	10

5.5 Business Rules	11
6. Other Requirements	11
Appendix A: Analysis Models	12

List of Figures

Table of Contents	2
--------------------------	----------

Revision History

Name	Date	Reason For Changes	Version
MRS 1.0	17-01-19	Initial Draft	1.0
MRS 2.0	30-01-19	Adding unfinished content	2.0
MRS 3.0	20-02-19	Finishing the rest of the document	3.0

1. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Mess Refund System. It will explain the purpose and features of the system, the interface of the system, what the system will do, the constraints under which it must operate and how the system will react to external entity. This document is intended for both the Professor of the course and the developers of the system and will be proposed to the authorities for the approval of the project.

1.2 Document Conventions

Title/subtopics in bold. The number in the square bracket represents links.[1]

Point something refers to sub parts under a specific module.

1.3 Intended Audience and Reading Suggestions

The Document is intended for the User of the System, Developers, The Professor of the Course, Concerned Authorities and other members.

1.4 Scope of Project

This software system will be a Mess Refund System for the students of the Institution designed as a part of Software Engineering Course. This system will be designed to provide transparency in the system and will enable the students to avail their respective refunds.

More specifically, this system is designed to allow the user to have a clear idea about the refund that will be acquired by him and also will make sure that the student has a say over what he gets and also the authorities can keep a track of how the mess is performing as this

system also acts as an attendance management system, hence the authorities will be able to periodical statistics related to the mess.

1.5 References

[1] Django QR Code Documentation Release 1.0.0:

<https://media.readthedocs.org/pdf/django-qr-code/latest/django-qr-code.pdf>

[2] Django QR Code:

<https://django-qr-code.readthedocs.io/en/latest/pages/README.html>

2. Overall Description

2.1 Product Perspective

The product has an independent existence and is not a sub part of any other module. The product though can be made as a subsidiary of a larger module comprising of automation of the institute working and student affairs.

2.2 Product Functions

- Student registers with the software.
- Student scans the Quick Response code before having his/her meal that the mess.
- Student receives a success code and this acts as an attendance for that meal.
- Hence the complete statistics of the student is maintained and at the end of the semester the refund is calculated and provided to the student.

2.3 User Classes and Characteristics

The users included in this project are students, mess managers, and the administrator handling the application operations.

- *Students:*
They are one of the most important users of this application. The users are required to create an account, and scan the QR code provided by the respective mess.

- *Mess Managers:*
They need to create an account to link their messes for the generation of the respective code from the mess's QR code.
- *Administrator:*
Manages the server when needed. Verifies the accounts of both students and mess managers.

2.4 Operating Environment

This will be a web application and hence it is operating system/platform independent. Therefore the application can be used in Windows, Linux, macOS or any other. Only requirement is that the system must have a working browser that is updated to the latest version and a working internet connection.

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software)>

2.6 User Documentation

The Web Application will be integrated with the User Guide Page which will describe all the functionalities and working of of the features for a particular *Role*.

The User Guide page will be designed according to the format given [here](#)

2.7 Assumptions and Dependencies

The mess refund is being calculated per semester and is being distributed equally among the students. Also, it is assumed that most of the students of the institution possess a smartphone or have access to a smartphone at a given instant.

3. External Interface Requirements

3.1 User Interfaces

Proper implementation of various human computer interaction techniques like ergonomics, etc.

Also the application must be user friendly and should not contain a lot of details and should basically be a neat and tidy application. For achieving this, we use bootstrap and CSS.

3.2 Hardware Interfaces

Smartphone(scanning the qr code and display the student details), tablet containing QR code/Paper.

3.3 Software Interfaces

The project is done on Ubuntu, using django (version 2.x) framework, the databases used in mysqlite, scanning the QR code is done using a django package called 'django-qr-code', handling the QR code image is done using 'pillow'.

The current versions we're working with are:

- Django 2.1.7 - The basic framework of the project.
- mySqlite 3 - Handles all the data required for the functioning of the project.
- django-qr-code 1.0.0 - One of the basic elements needed for fulfilling the purpose of the project. This is an application that provides tools for displaying QR codes. This application depends on the qrcode python library which requires the Pillow library in order to support the PNG image format.
- Pillow 5.4.1 - As mentioned above. Manages PNG images. Helps in displaying the QR code as an image.

- Ubuntu 18.04 - The operating system being used for the project.

3.4 Communication Interfaces

The product follows HTTP as a Communication Standard between server and receiver, Message passing after QR scan .

The system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

4. System Features

The mess refund system works on outputs generated by the respective QR codes. The major service provided by the application is the exact calculation of refund a student needs to get back with respect to mess usage and the amount of students making use of mess facility provided by the college. Preferences of mess can also be inferred from this data.

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 QR Code Scan

Scanning the QR code and generating a automated link.

4.1.1 Description and Priority

High priority

The QR code should be verified and tested for any errors. The link generated using the QR code should be handled by the server and a quick and secure response should be given back to the client.

4.1.2 Stimulus/Response Sequences

The list of sequences is as follows:

- Every mess is registered in the app and the system response is the generation of the corresponding QR code.

- Now, the user has to scan the QR code, which would generate a response which would uniquely identify the said user.

4.1.3 Functional Requirements

Major functions of the Login and Registration System

- Authenticate and Login user to the webapp.
- Enable a registered user to change his password if he forgets his password.
- Enable a registered user to update his profile which includes his location, skills and profile picture.
- Enable a registered user to view his profile.
- Enable user to change his/her Mess.

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, In case of opening databases, sorting questions and evaluation there are no delays and the operation is performed in less than 2 seconds for opening ,sorting, computing, posting > 95% of the files.

5.2 Safety Requirements

Information transmission should be securely transmitted to the server without any changes in information. Proper Authentication must be done. The information shared should be in correlation with the permission, i.e. specific user should receive a specific request.

As the system provides the right tools for discussion, problem-solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

5.3 Security Requirements

The data used in this project is that of the students registered in the college. The database includes images, as well as monetary information of the students. Thus, this data needs to be protected. The server on which the project resides will have its own security to prevent unauthorized write/delete/read access. This is handled by the administrator.

The user authorization process is also handled by the Admin.

The PCs on which the user classes reside will have its own security mechanisms.

5.4 Software Quality Attributes

Availability

If the internet service gets disrupted while sending information to the server, the information can be send again for verification.

Security

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The tablet id registration is way to spam check for increasing the security. Hence, security is provided from unwanted use of recognition software.

Usability

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

5.5 Business Rules

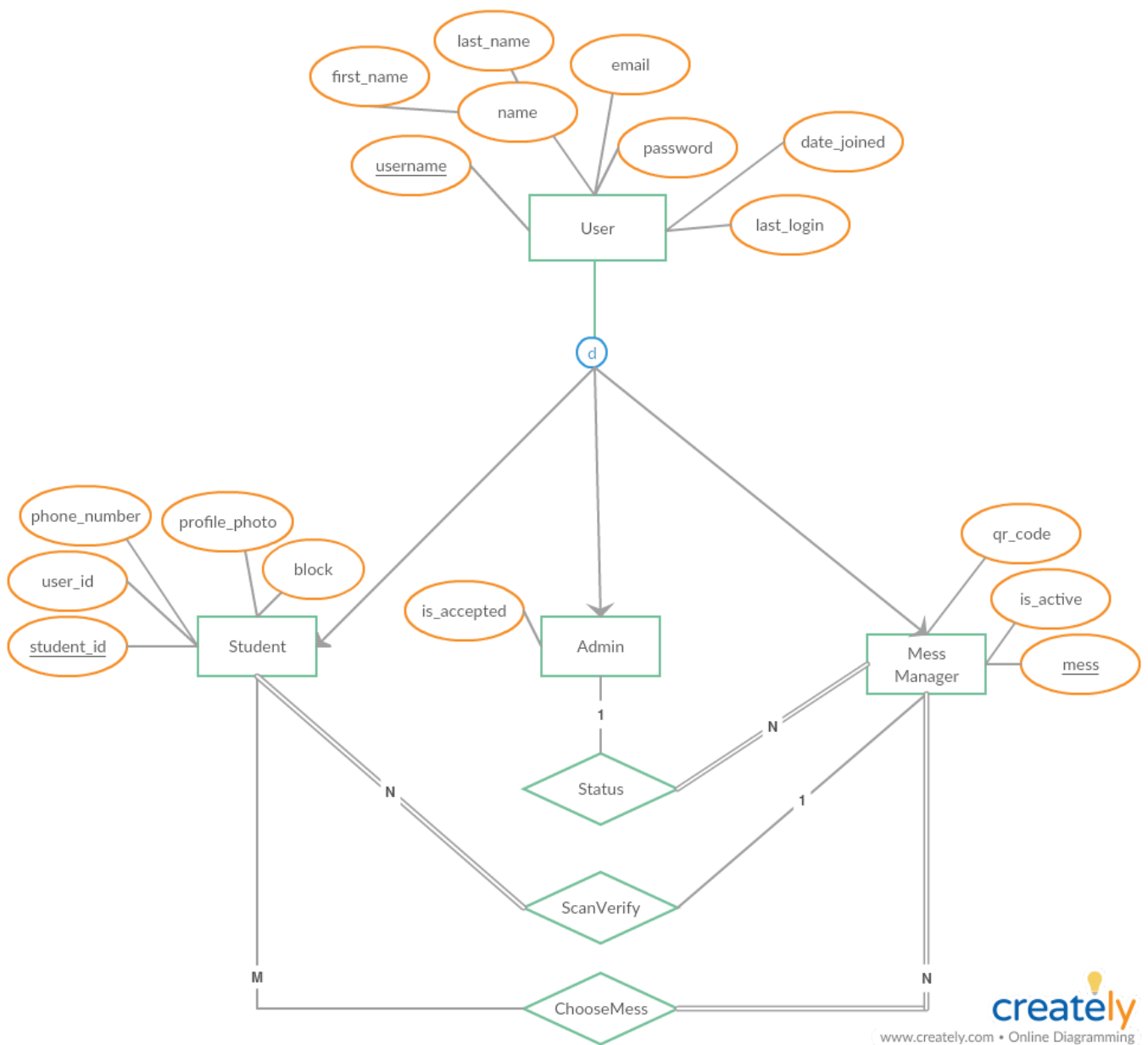
<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6. Other Requirements

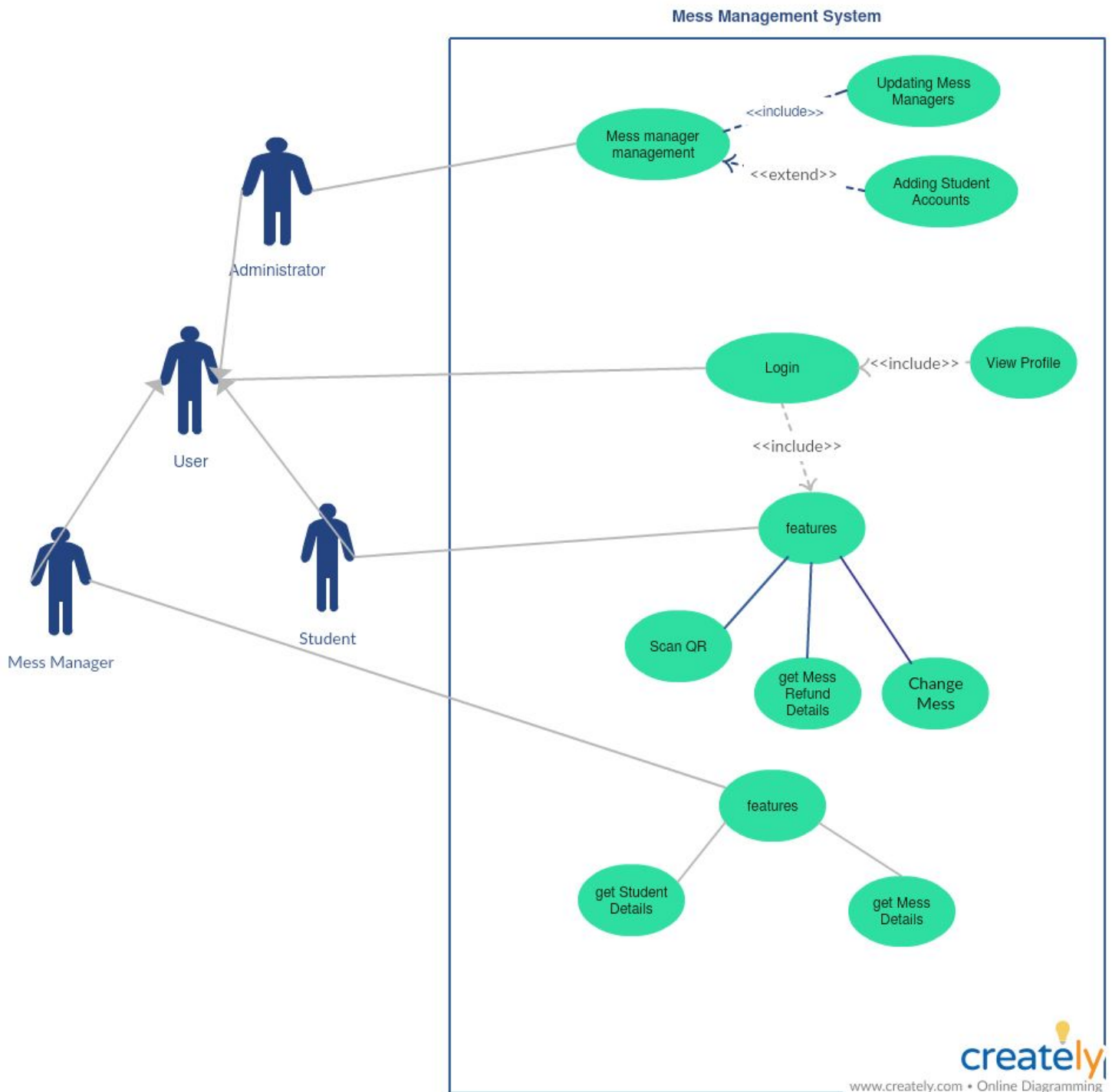
<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Analysis Models

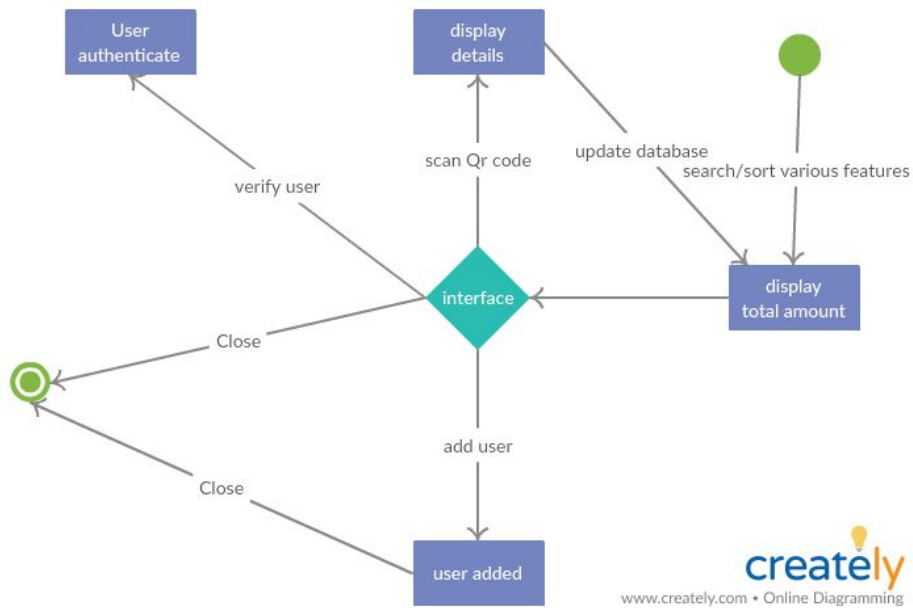
1. ER Diagram



2. Use-Case Diagram



3. Sequence Diagram



4. Navigation Diagram

