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

List of Figures

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

1.0. 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 a 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 .

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

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.

4.1 System Feature 1

Scanning the QR code and generating a automated link.
<Don't really say "System Feature 1." State the feature name in just a few words.>

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.

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

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.

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

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

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

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: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>