

**Software Requirements Specification  
for  
Mess Refund System**

**Version 1.0**

**Harshavardhan G  
Manas Kinkar  
D Praneetha**

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

## **Table of Contents**

<b>Table of Contents</b>	<b>2</b>
<b>Revision History</b>	<b>3</b>
<b>1.0. Introduction</b>	<b>4</b>
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
<b>2. Overall Description</b>	<b>5</b>
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
<b>3. External Interface Requirements</b>	<b>6</b>
3.1 User Interfaces	6
3.2 Hardware Interfaces	7
3.3 Software Interfaces	7
3.4 Communication Interfaces	7
<b>4. System Features</b>	<b>7</b>
4.1 System Feature 1	8
4.1.1 Description and Priority	8
4.1.2 Stimulus/Response Sequences	8
4.1.3 Functional Requirements	8
4.2 System Feature 2 (and so on)	9
<b>5. Other Nonfunctional Requirements</b>	<b>9</b>
5.1 Performance Requirements	9
5.2 Safety Requirements	9
5.3 Security Requirements	9
5.4 Software Quality Attributes	10
5.5 Business Rules	10
<b>6. Other Requirements</b>	<b>10</b>
<b>Appendix A: Glossary</b>	<b>11</b>

<b>Appendix B: Analysis Models</b>	<b>11</b>
<b>Appendix C: To Be Determined List</b>	<b>11</b>

## **List of Figures**

<b>Table of Contents</b>	<b>2</b>
--------------------------	----------

## **Revision History**

<b>Name</b>	<b>Date</b>	<b>Reason For Changes</b>	<b>Version</b>
MRS 1.0	17-01-19	Initial Draft	1.0

## 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***

*TO BE UPDATED....*

### ***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 tract 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***

*TO BE UPDATED....*

## **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***

To Be Decided....

## ***2.6 User Documentation***

None as of now...

## ***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.

### ***3.2 Hardware Interfaces***

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

### ***3.3 Software Interfaces***

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

### ***3.4 Communication Interfaces***

*<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***

*<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***

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

#### **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.>*