
Software Requirements Specification

for

MyCompanion

Version 1.0

Prepared by

Group Name: To_Be_Decided

Ayush Tyagi
Chirag Setia
Chirag Madan
Tanish Charaya

101603070
101603079
101610021
101611056

atyagi_be16@thapar.edu
csetia_be16@thapar.edu
cchirag_be16@thapar.edu
tcharaya_be16@thapar.edu

Instructor: Dr. Vinay Arora

Course: Software Engineering

Lab Section: COE6

Teaching Assistant: *Deepali Bhagat*

Date: 05-09-2018

Contents

CONTENTS.....	II
REVISIONS.....	II
1 INTRODUCTION.....	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATION	1
1.5 DOCUMENT CONVENTIONS.....	1
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	1
2 OVERALL DESCRIPTION.....	2
2.1 PRODUCT OVERVIEW	2
2.2 PRODUCT FUNCTIONALITY.....	2
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	2
2.4 ASSUMPTIONS AND DEPENDENCIES	2
3 SPECIFIC REQUIREMENTS.....	3
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	3
3.2 FUNCTIONAL REQUIREMENTS	4
3.3 USE CASE MODEL.....	6
4 OTHER NON-FUNCTIONAL REQUIREMENTS.....	7
4.1 PERFORMANCE REQUIREMENTS.....	7
4.2 DESIGN CONSTRAINTS.....	7
4.3 STANDARD COMPLIANCE.....	7
4.4 RELIABILITY.....	7
4.5 AVAILABILITY.....	7
5 TEST CASES.....	9
5.1 TEST CASES.....	9

1 Introduction

The purpose of this application is to automate the hostel mess functionality and provide both the user and the admin a smart platform to interact with each other.

1.1 Document Purpose

The purpose of this document is to present a detailed description of the Mess Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and what kind of system interactions take place.

1.2 Product Scope

The Mess Management System helps the user to access all the functionalities of the mess without having to visit the mess physically. Our product aims at providing an automated system for hostel mess billing, leave entry, inform the mess authorities when he/she won't be eating in the Mess, complaints and reviews. It will replace the current system in which leave entry has to be filled by the student in the leave diary, complaints are to be reported directly to the management office.

By using our product hostel students will be able to access all the above mentioned features with just few clicks on his/her device. The product makes the work of both the students as well as mess management team easy and efficient.

The system will utilize a relational database for handling all the data such as menu items, prices, student IDs, consumption statistics, student feedback, etc.

1.3 Intended Audience and Document Overview

This document is intended for the students residing in Hostel-J of Thapar Institute as well as the mess management team of the hostel itself. Later the scope can be increased to other hostels of the university.

1.4 Document Conventions

Arial font size 11 was used throughout this document except for headings. Margin of 1" was used.

1.5 References and Acknowledgments

IEEE. *IEEE Std. 830-1998 IEEE Recommended Practice for Software requirements Specifications*. IEEE Computer Society, 1998.

2 Overall Description

2.1 Product Overview

In various institutes, universities and offices people depend on mess/canteens for their meals as it is more economical and easily reachable. But to serve so many people a mess organization should be able to receive and work on feedbacks from their customers. Moreover to prevent any losses it should have efficiency and accuracy in sections related to finance and database related to stock and database of each of its customer.

So in this project we provide automation of mess management system. This will enable the organizations to work efficiently.

2.2 Product Functionality

- Student / management login
- User can check mess menu and also receives timely notifications for the mess food
- User can make a leave entry
- User can check his/her mess bill and make the payment on the app itself
- Mess management can upload student bills and receive complaints, reviews filled by students

2.3 Design and Implementation Constraints

- Database server will limit the number of audience
- Users should have an android smart phone

2.4 Assumptions and Dependencies

- Users have an android smartphone with internet connection
- Each user has its own username and password for login
- It is assumed that mess management team will keep the database record up-to-date

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Login Screen**
 - The login screen will provide authenticate students with their user Id and password. It will also contain login for admin (mess management in this case).
- **Main activity (the home screen)**
 - All the utilities of the app will be available on this screen. Users will have option to give feedback, reviews, check their mess bill, register a leave entry.
- **Register leave entry**
 - On this screen user will have to provide leave period including start date and end date. The app will validate the entered leave period and after location verification through GPS will pass on the leave to the database.
- **Give feedback/complaint**
 - Users will specify category of feedback/complaint and can describe their views.
- **Check mess bill**
 - Users will get detailed info about their mess bill, previously registered leaves.
- **Check mess menu**
 - Users can check the weekly mess menu, mess timings.
- **Notifications/ messages**
 - Information sent by the mess management for the students will be accessible in this section.

3.1.2 Hardware Requirements

A webserver will be required so that the students and the mess admin can connect to it to exchange information. The server will have a database to store all the data entries.

3.1.3 Software Requirements

- Programming language: Android SDK and SQLite
- Operating system: Android
- Eclipse IDE

3.2 Functional Requirements

3.2.1 User/ management login

The system will provide three different type of login options

- For students
- For admin
- Guest login

For student/ admin login preassigned user id, password needs to be entered.

The accessibility of functionality will depend on type of user logged in.

The student may click on the forgot password link on the login page to have his password changed after having it emailed to him.

3.2.2 Dashboard

On the dashboard will show the present day mess menu for breakfast, lunch, dinner.
From the dashboard users can navigate to the other functionality of the app.

3.2.3 View Mess Menu

The student gets to view the menu for the week. He can also choose to view a particular menu like Monday's lunch menu or the Night mess menu. The query from the student is sent to the database where the data is read and is sent back to the student to be displayed on the screen.

3.2.4 Complaints

Users can register complaints with the app. The complaint will be sent to admin portal with the user id of the user logged in. They can also check status of their complaints.

The status will be updated by the admin.

This functionality will **not** be available in guest login.

3.2.5 Register mess leave

This functionality lets user register leave entry on the app itself after specifying start date and end date of the leave.

For security and to eliminate false users will not be able to fill leave entry when they are in Patiala. The system will validate this by using GPS services from the device.

This functionality will **not** be available in guest login.

3.2.6 Check mess bill

Logged in users can check their present mess bill in detail, previously filled leaves.

3.2.7 Logout

The student logs out of the system so that nobody else can modify his mess option or book orders from his account.

3.2.6 Management portal (For admin login only)

3.2.6.1 Update menu

The administrator can update the menu for any meal for any day of the week or the mess menu by selecting new food items to add to the menu or by removing unpopular food items. The changes after the updation are saved in the database so that the students can see the updated menu.

3.2.6.2 View Feedback/ Reviews/ Complaints

The mess admin can view the feedback submitted by the students. The database will be queried for feedback and complaints and the admin can check all these forms and take the required action on them. He can also update the feedback status as read or the complaint status as per its current state.

3.2.6.3 Check leave entries

The administrator can check the leave entries filled by the students. The query will be sent to the database and records fetched will be shown to the admin.

3.2.6.4 Update bill

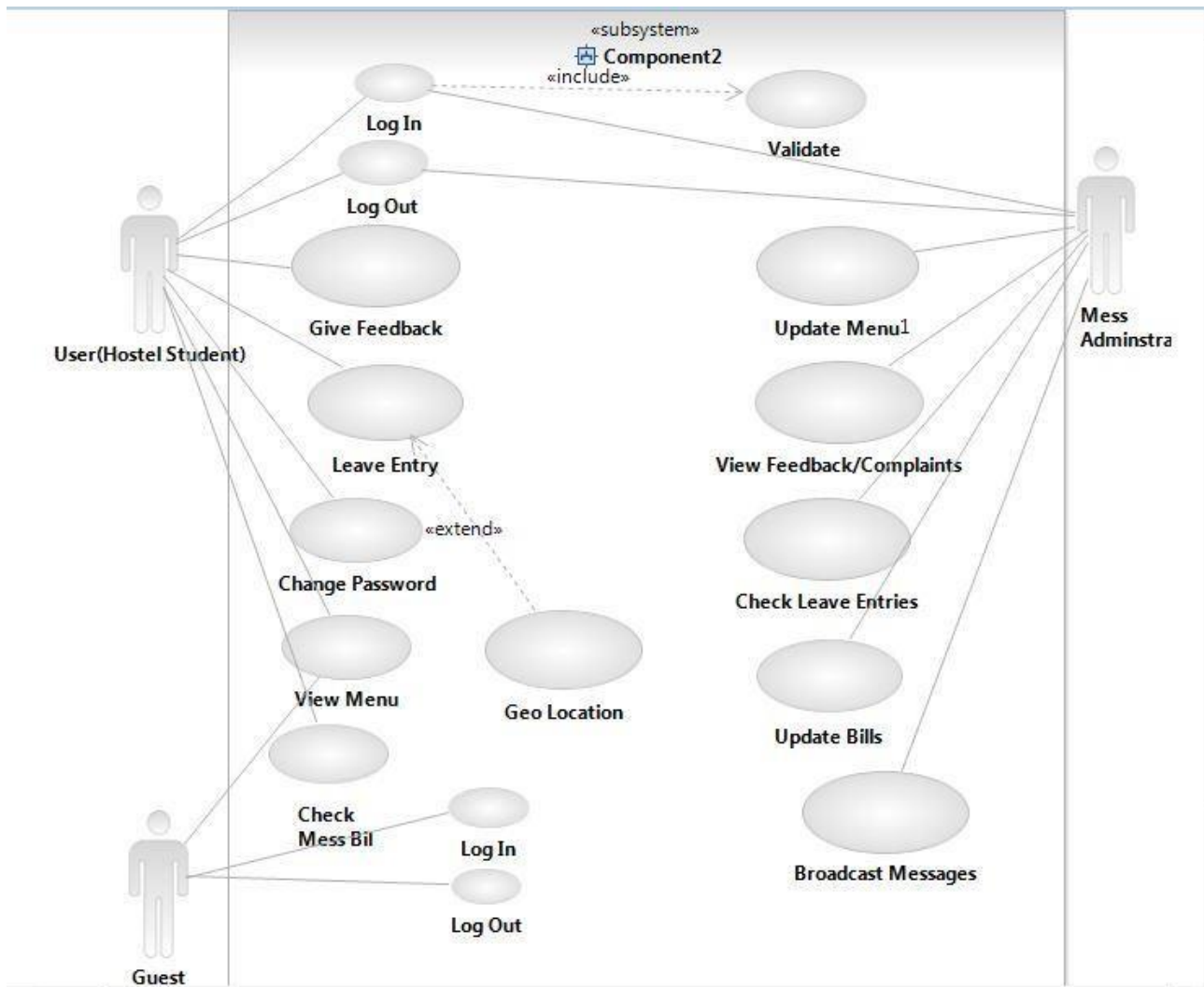
The administrator can upload bills to the database so that students can view them.

3.2.6.5 Broadcast messages

Admin can also broadcast any important message to also the users. The message will be sent as notifications to the users.

3.3 Use CaseModel

3.3.1 Use Case



4 Other Non-functional Requirements

Non - Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

4.1 Performance Requirements

Performance requirements define acceptable response times for system functionality. The load time for user interface screens should be minimum. The log in information should be verified quickly. Queries and response should also be minimum.

4.2 Design Constraints

The Mess Management System shall be a stand-alone system running in a android environment. The system shall be developed using Java or MySQL database.

4.3 Standards Compliance

There shall be consistency in variable names within the system. The graphical user interface shall have a consistent look and feel.

4.4 Reliability

The data transferred through the app should be correct and reliable. The app will store the data onto the database server.

4.5 Availability

The system shall be available during normal hostel operating hours. It will be easily accessible for the hostel students.

4.6 Maintainability

The Mess Management System is being developed in Java. Java is an object oriented programming language and shall be easy to maintain.

4.7 Portability

The Mess Management System shall run in any Android device.

Test Cases

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TU01	Check Student Login with valid Data	<ol style="list-style-type: none"> Go to Login Screen Enter Roll Number Enter Password Click Log in 	Roll Number = 101610021 Password = 1234	User should Login into application	As Expected	Pass
TU02	Check Student Login with invalid Data	<ol style="list-style-type: none"> Go to Login Screen Enter Roll Number Enter Password Click Log in 	Roll Number = 101603079 Password = 4567	User should not Login into application	As Expected	Pass
TU03	Check Student Login with empty password	<ol style="list-style-type: none"> Go to Login Screen Enter Roll Number Enter Password Click Log in 	Roll Number = 101610021 Password =	Application should ask for password	As Expected	Pass
TU04	Check Admin Login with valid Data	<ol style="list-style-type: none"> Go to Admin Login Screen Enter Admin Id Enter Password Click Log in 	Admin id = 1234 Password = 1234	Admin should Login into application	As Expected	Pass
TU05	Check Admin Login with invalid Data	<ol style="list-style-type: none"> Go to Admin Login Screen Enter Admin Id Enter Password Click Log in 	Admin id = 1234 Password = 12345	Admin should not Login into application	As Expected	Pass
TU06	Check if feedback	<ol style="list-style-type: none"> Enter empty feedback 	NA	Feedback should not	As expected	Pass

	content is empty or not On enter feedback screen	2. Click on submit button		be sent and error should be raised.		
TU07	Check if leave entry data is stored in database or not	1. Enter leave entry data on leave entry page 2. Submit	NA	Leave Entry details should be uploaded to leave entry table in database	As Expected	Pass

