Software Requirements Specification

for

Hostel Management System

Version 1.0 approved

Prepared by Badri Ravali(15IT207)

National Institute of Technology, Surathkal

8th Jan,2018

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
SRS	**18.02.2018**	**Updated 4.5 and 4.11 of Functional Requirements**	**v1.1**

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Hostel Management System v1.0” . It will illustrate the purpose and complete development of system, what the system will do and the system constraints under which it must operate and interfaces. This document is intended for the users of the software and also potential developers.

1.2 Document Conventions

This document was created based on the IEEE template for System Requirement Specification Documents. This document uses Times New Roman font throughout the entire document. Major header lines use bold, size 18 font. Smaller subsections use bold, size 13 font. Any part of this document that is not a header of some kind uses size 12. The one inch margins are maintained throughout this document as well.

1.3 Intended Audience and Reading Suggestions

This document is intended for typical users such as students who join a hostel , hostel administrations who manage a hostel and programmers who are interested in working on the project by further developing it or fix existing bugs.

The remainder of this document provides the general product description and a technical outline for the requirements of this system. Section two will give a high level description of the project functionality and implementation details. Section three will give a more detailed description of the specific requirements for different components of the application, which include various interfaces and functional requirements.

1.4 Product Scope

The “Hostel Management System” is a web-based application which helps people managing a hostel. The application should be free to access and use. Hostel administration can provide their information using the web portal. They can update the fees of different rooms in their hostels and change the fees of the mess. The administration can manage the user information through the web portal. The users of this website, students can update their profile and choose rooms and pay fees for it.
The objective and goal of this web application is to help hostel management administration with their work. Their work which usually involves man force to maintain records and registers is made easy by computerizing the whole system. Also students need not visit offices to get their work done and can get their work done online through this web based portal.

1.5 References

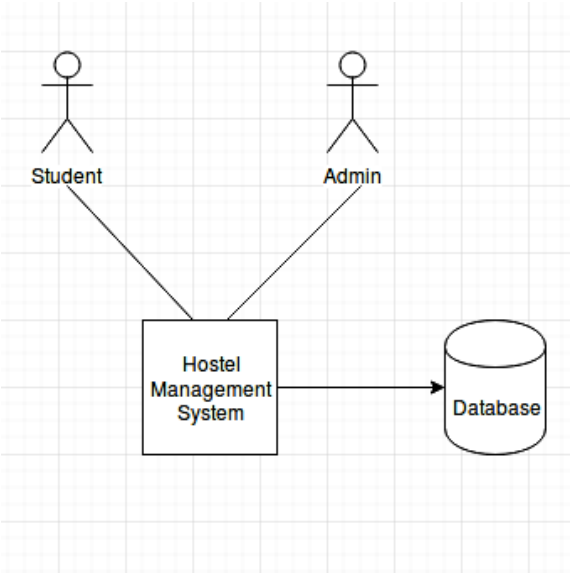
[1]IEEE Template for System Requirement Specification Documents,
https://web.cs.dal.ca

Other references regarding interface style guides and technologies used will be specified in later versions.

2. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with the users and introduce the basic functionality of it. At last, the constraints and assumptions for the system will be presented.

2.1 Product Perspective



This is a new, self-contained project. Initially, there is no framework or system in place to build upon; therefore, this software will be built from the ground up.
This system will consist of a backend and a frontend. In the frontend, the user will interact directly with a webpage using the interface to retrieve any and all data requested. In the backend, our website will allow us to aggregate the data and produce results which will be of value to the users.The whole project will be designed as two parts, the front and back end, that will be used by end users and Hostel Management respectively. The front end will be a web application where the user may interact with to see a visual representation of his profile and related information.

The web application will be built on PHP , HTML, CSS and interface with the mySQL server to store and pull relevant data. The backend will be the mySQL server that will store and push the data that is being used by the application. The backend will be primarily used by hostel managements who have registered with the website to have access to the data. The idea originated so as to replace the existing manual record based system of hostel management.

2.2 Product Functions

- The user must authenticate himself or herself with the system.
- The user will be able to update his or her profile.
- The user will able to see the fees of different hostel rooms and different mess options available.
- The user can pay the fees for the hostel room he chooses and the mess option he chooses.
- User can also choose some additional facilities offered by hostel that are not mandatory.
- Users can also signup for various hostel events.
- Admin must also authenticate himself or herself with the system.

- Admin will be able to manage the students or users of the hostel.
- Admin can add various hostel events.
- Admin can update room fees and mess fees.
- Admin will be able to send personal notifications to the users of the system.

2.3 User Classes and Characteristics

There are two types of users that interact with the system : Students(users) and Admin.
The initial users of the software front end will be conscious individuals who are familiar with navigating a web browser and managing a desktop, laptop, or mobile device.
The user may take a moment to familiarize him/her self with the layout and interface of our web service, but should easily understand what he/she is looking at.
The users of the backend, or the mySQL database, will be Hostel management team. Therefore, it is expected that these individuals have a solid foundation with mySQL databases and are capable of navigating themselves. The administration is managing the overall system so there is no incorrect information within it.

2.4 Operating Environment

- Hardware Requirements:

  Processor : Intel Core Duo 2.0 GHz or more.
  RAM : 1GB or more.
  Hard Disk Space : 10GB or more.

- Software Requirements:

  Operating System : Windows 7 or more, Ubuntu
  Front-end : HTML, CSS
  Back-end : PHP, MySQL Database.

2.5 Design and Implementation Constraints

Design:
- Language : HTML5, CSS, PHP
- Database : MySQL
- Server : Apache server

Constraints:
- Must run on Google Chrome or Firefox 18
- User must authenticate himself or herself at the login
- Processing requirement : as little processing as possible.
- The Internet connection is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.
- The web portal will be constrained by the capacity of the database. Since the database is shared between both user and admin it may be forced to queue incoming requests and therefore increase the time it takes to fetch data.

2.6 User Documentation

Online help:

- Step by step instructions on setting up accounts.
- Viewing will be intuitive and self-explanatory.

Everything will be as simple as possible. Ideally, it will require no extra explanation. Every step for account setup and viewing options will be clearly labeled and described as the client progresses through the web page.

2.7 Assumptions and Dependencies

- We assume the user has or has access to a computer or laptop.
- We assume the user has an internet connection in order to navigate to the website and login
- It is assumed that the user is familiar with an internet browser and also familiar with handling the keyboard and mouse.
- We depend on the correct operations between database and the application.
- Users must have their correct usernames and passwords to enter their online accounts and perform intended actions.

3. External Interface Requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1 User Interfaces

- Login Page
- Registration Page
- Student Profile page
- View Hostel fee
- View Mess fee
- Pay Room fee
- Pay Mess fee
- See notifications
- Sign up for events
- Update profile
- Admin profile page
- Update college information
- Add events
- Send notifications
- Update Mess fee
- Update hostel fee
- Manage student information

The logical characteristics and design of each interface between the software product and the user and sample screen images will be defined in the later versions as the design is not finalized yet.

3.2 Hardware Interfaces

- Laptop/Desktop:
- The user may also access our software through a web browser on their desktop or laptop.
- Under the assumption that the user is using a traditional computing environment, the user may interact via mouse clicks and keyboard inputs.

3.3 Software Interfaces

- SQL Server (MySQL community edition 5.x.x) Interface
  The SQL tables will contain all the information collected. It will be accessible by our software via the PHP. Data transferred include fee payments, user and account information, etc.
  The communication between the database and the web portal consists of operation concerning both reading and modifying the data,
- Windows
- Linux
  Both Windows and Linux OS will interact with our web interface (via web browser) to display the data to the user via a web browser.

3.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems and the web portal.

4. System Features

This section demonstrates the most prominent features of the hostel management system software and explains how they can be used and the results they will give back to the user.
The system design is divided in to two portions. The Administrator section and the User(student's) section.

4.1 Registration

4.1.1 Description

The purpose of this feature is to facilitate the registration of a new user by providing a form and requesting for necessary details.
The feature is of high priority as registration is necessary to avail the services of this system. Both user and admin can register through this page.

4.1.2 Stimulus/Response Sequences

The user and the admin fill in the required details asked by the registration page. There is a verification to check if the user in not robot. This is done by giving a simple math question to solve and all the fields are verified for integrity. Once the user hits on the Register button. All the details will be inserted into the database.

4.1.3 Functional Requirements

REQ-1: Once the user hits the register button, the system should be able to make sure that all the required fields are filled
REQ 2: All the details should be inserted into the database without any errors and delays.

4.2 Login

4.2.1 Description and Priority

The purpose of this feature is to facilitate the user with the access of all services provided by the system by authenticating him or her.
The feature is of high priority as authentication is necessary to avoid unregistered and unauthenticated users from modifying the data.

4.2.2 Stimulus/Response Sequences

The user or student is asked for his username, ID and password to authenticate.
The admin is prompted to provide username and password.
Once the user hits the login button, the details given by the user are verified against the details stored in the database and if there is no error the use is granted access. Otherwise, access is denied.

4.2.3 Functional Requirements

REQ-1: Once the user hits the login button, the system should be able to make sure that all the required fields are filled and verify the details against the details stored in the database.
REQ 2: The user should be taken to his profile page once the access is granted.

4.3 Update Profile

4.3.1 Description and Priority

The purpose of this feature is to facilitate the user to update his or her profile by giving primary details of his/her background.
The feature is of low priority as it has nothing to do with the working of other functionalities.

4.3.2 Stimulus/Response Sequences

The user is prompted to fill a form and provide basic information. There is a button named Update. Once the user finishes filling the document he or she can hit the update button to insert the data into database.

4.3.4 Functional Requirements

REQ-1: Once the user hits the update button, the system should be able to make sure that all the required fields are filled and update them into the database.
REQ 2: Database access should be as quick as possible.

4.4 View Mess Fee and Hostel Fee

4.4.1 Description and Priority

The purpose of this feature is to facilitate the user to view the available mess and hostel options. User can choose any one of the options from the available options and see the mess fee or hostel fee for the chosen option.
The feature is of medium priority.

4.4.2 Stimulus/Response Sequences

The user is prompted with a drop down list and the user can choose option of his interest. Accordingly the system will show the fees corresponding to the option chosen.

4.4.3 Functional Requirements

REQ-1: Once the user selects an option, fees should be viewed correctly corresponding to the option chosen.
REQ 2: Database access should be as quick as possible.

4.5 Pay Mess Fee and Hostel Fee

4.5.1 Description and Priority

The purpose of this feature is to facilitate the user to pay the room or mess fee. He can also choose some additional facilities provided by the hostel.
The feature is of medium priority.

4.5.2 Stimulus/Response Sequences

The user is prompted with the total amount to be paid after the user chooses additional services in addition to hostel and mess.

4.5.3 Functional Requirements

REQ-1: Once the user selects an option, fees should be viewed correctly corresponding to the option chosen.
REQ 2: Database access should be as quick as possible.
REQ 3 : If the maximum no. of seats get filled, student should not be allowed to choose that particular mess or hostel.

4.6 View Notifications

4.6.1 Description and Priority

The purpose of this feature is to facilitate the user to view any notifications sent to him by the admin. The feature is of medium priority.

4.6.2 Stimulus/Response Sequences

The user is shown a screen with the notifications sent to him.

4.6.3 Functional Requirements

REQ-1: Database access should be as quick as possible.

4.7 Sign up for Events

4.7.1 Description and Priority
The purpose of this feature is to facilitate the user to sign up for the events held by the hostel office. The feature is of low priority.

4.7.2 Stimulus/Response Sequences

A screen with all the events is shown to the user and the user can choose to register for any one of the events.

4.7.3 Functional Requirements

REQ-1: Once the user signs up for an event it is updated in the database.
REQ 2: Database access should be as quick as possible.

4.8 Update Hostel Information

4.8.1 Description and Priority
The purpose of this feature is to facilitate the admin to change the description of his or hostel. The priority of this feature is low as it has nothing to do with the other functionalities of the system.

4.8.2 Stimulus/Response Sequences

A form will be provided to the admin to fill or change the hostel information. Once the update button is hit the information is updated.

4.8.3 Functional Requirements

REQ-1: Once the user hits the update button the information is updated in the database.
REQ 2: Database access should be as quick as possible.

4.9 Manage Student Information

4.9.1 Description and Priority
The purpose of this feature is to facilitate the admin to manage the student information. He can view basic information about the student. Admin has full authority to delete the student. The priority of this feature is medium.

4.9.2 Stimulus/Response Sequences

There are no sequences for this feature.

4.9.3 Functional Requirements

REQ 1: Actions performed by the admin have to reflect in the database.
REQ 2: Database access should be as quick as possible.

4.10 Update Mess and Hostel Fee

4.10.1 Description and Priority
The purpose of this feature is to facilitate the admin to change hostel and mess fee for different hostels and mess options available. The priority of this feature is high.

4.10.2 Stimulus/Response Sequences

The admin can choose the specific hostel or mess and update the fees accordingly. The database should give back the correct hostel or mess options as chosen by the administrator.

4.10.3 Functional Requirements

REQ 1: Actions performed by the admin have to reflect in the database.
REQ 2: Database access should be as quick as possible.

4.11 Send notifications

4.11.1 Description and Priority
The purpose of this feature is to facilitate the admin to send any sort of messages or notifications to the users or students. The priority of this feature is medium.

4.11.2 Stimulus/Response Sequences

The admin can send the notifications to the students. The admin is prompted with a dropdown list which has the IDs of all the students. The admin is supposed to choose one of the students from the list and can type in his message for the student or choose from preset messages.

4.11.3 Functional Requirements

REQ 1:Messages are stored in the database and are viewed to the user as and when requested by the user or student.
REQ 2: Database access should be as quick as possible.

4.12 Add Events

4.12.1 Description and Priority
The purpose of this feature is to facilitate the admin to add events organized by the hostel. The priority of this feature is medium.

4.12.2 Stimulus/Response Sequences

The admin is prompted with a form where in he can fill all the details regarding the events. Once the Add Event button is hit the details of the event will be added in the database.

4.12.3 Functional Requirements

REQ 1:Details added by the admin have to reflect in the database.
REQ 2: Database access should be as quick as possible.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Clicking any link on the navigation bar should take less than 2 seconds.
- Updating and saving any personal information should take less than 2 seconds.
- Database access should be performed in less than 3 seconds.

- Login verification should be done in less than 2 seconds.
- Loading of images should not consume time and resources.

5.2 Safety Requirements

- The user's login information should be secure enough so that anyone except for the developers will know that user's username/email and password.
- When the user types his or her password to login, it should be hidden.
- The system should be secure enough so that the user's personal information will not be disclosed to unauthorized users.
- The system should be secure enough to prevent the login of data of unauthorized users.
- The system should log out the user from the session if there is no activity for a long period of time.

5.3 Security Requirements

There are no security requirements for this website as of now. If required, will be mentioning them in the later versions.

5.4 Software Quality Attributes

5.4.1 Availability

- The system must deliver services to the client when requested.
- The application should be available 99.9% of the time.

5.4.2 Dependability

- The application must be reliable enough so that when different users perform the same task, the expected outcome does not differ.
- The application must be reliable enough so that when users perform normal functions, the system does not fail.

5.4.3 Usability

- The application must be easy enough to learn so that users know how to use the product entirely on their first try.
- The application must be more efficient than similar products.

5.5.4 Flexibility

- The application must be flexible enough to support Google Chrome and Firefox 18 browsers
- The application must be able to support Windows and Linux operating systems
- The application must be flexible enough to support users with different requirements.

5.5.5 Maintainability

- The hostel administration should maintain the correct data and information.

5.5.6 Business Rules

User can pay only his/her hostel fee.
User can only view the notifications sent to him and not the messages of other users.
Administration has full permission to manage user information.
There are no stringent corporate policies enforced as of now.

6. Other Requirements

Database access should be as quick as possible. This project doesn't have any legal requirements and internationalization requirements.

Appendix A: Glossary

Term	Definition
Author	Person submitting an article to be reviewed. In case of multiple authors, this term refers to the principal author, with whom all communication is made.
Database	Collection of all the information monitored by this system.
HTML	Hyper Text Markup Language. A language used for the front end of the website.
PHP	Hypertext Processor.
IEEE	Institute of Electrical and Electronic Engineers.
CSS	Cascading Style Sheets.
Apache	Web Server Software
MySQL	Open-Source Relational Database Management System
Admin	System administrator who is given specific permission for managing and controlling the system.
HMS	Hostel Management System
User	Reviewer or Students
RAM	Random Access Memory
Administration	Owner of the Hostel or the Hostel Management System
TBD	To be determined list

Appendix B: Analysis Models

All the data flow diagrams, class diagrams, state-transition diagrams and entity-relationship diagrams will be given in the later versions as the project is not completely bulit.

Appendix C: To Be Determined List

TBD Item	**Status**	**Date Closed**
References regarding interface style guides [1.5]	Not closed	-
Logistic characteristics and design of user interfaces [3.1]	Not closed	-
Security Requirements [5.3]	Not closed	-
Analysis Models [Appendix B]	Not closed	-