
Software Requirements Specification

for

Hostel Management System

Version 1.0

Prepared by 20CS052,20CS055

CSPIT-CSE

02-02-22

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Objective	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Project Scope	1
1.5 References	2
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Features	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	3
3. System Features	3
3.1 System Feature 1	3
4. External Interface Requirements	4
4.1 User Interfaces	4
4.2 Hardware Interfaces	5
4.3 Software Interfaces	5
4.4 Communications Interfaces	5
5. Other Nonfunctional Requirements	5
5.1 Performance Requirements	5
5.2 Safety Requirements	6
5.3 Security Requirements	6
5.4 Software Quality Attributes	6
6. Other Requirements	6
Appendix A: Glossary	6
Appendix B: Analysis Models	7
Appendix C: Issues List	7

Revision History

Name	Date	Reason For Changes	Version

• 1 Introduction

1.1 Introduction

We are designing a website for a Hostel that will provide a computerized process that is stress free, reliable and quick through the use of PHP computer programming language and MySQL database. Here, Student/User can log in and view room details and register him/her self easily. The user can change their password anytime. Admin can add/manage rooms, courses, students and many more.

1.2 Document Conventions

We have chosen fonts and highlighted certain information which will be reader friendly.

1.3 Intended Audience and Reading Suggestions

Hostel staff or Hostel management who have requirements of managing many hostel activities like add/manage rooms,rooms related information,course,students related information will use this system.

1.4 Project Scope

This project "Hostel Management System" will help hostel admin in managing records of the hostel in an efficient manner.This project manage records of the students,hostel room and other things related to the hostel.

1.5 References

- <https://www.w3schools.com/php/>
- <https://www.w3schools.com/js/>
- <https://getbootstrap.com/>

Reference has also been taken from web designing and development tutorials.

• 2 Overall Description

2.1 Product Perspective

we have user interfaces.User interfaces will be divided like

➤ **Administration Interfaces**

➤ **User Interfaces**

➤ **Interfaces for Administrator :**

- In this view, At the start there will be a login screen where admin has to enter its login email and password to authenticate himself or herself.
- After the successful login, Admin can manage rooms, students, and courses.

➤ **Interfaces for User :**

- In this view, At the start there will be a login screen where the user has to enter its login email and password to authenticate himself or herself. and if the user is not registered on the system , then the user can register himself or herself using the user registration option.
- After the successful login, User can update its registration details using the “My profile” option except the email id and password ,though user can change its password using “change password” option, user can also book hostel by providing personal information , address and room related information , after successfully providing the all information user can see its all information using “room details” option.

2.2 Product features

“Hostel Management System” enables to perform the following features.

- ❖ Maintaining room information.
- ❖ Maintaining the resident information.
- ❖ Searching, sorting and retrieval of data.
 - The various other functions covered by the requirement specifications which follows are provided to meet the requirements of database administrator students and administrator with role based updating and viewing rights.

The following general features will be included:

- ❖ The student details are filled by as:

- Student Name
- Student father's Name
- Student year
- Room No
- ❖ Following information given by the administrator:
 - Student details
 - Room details
- ❖ Updating by the database administrator:
 - Student details
 - Room details

2.3 Product featuresUser Classes and Characteristics

User Characteristics:

- **Educational level:**
 - At least user of the system should be comfortable with English language.
- **Technical Expertise:**
 - Users should be comfortable using general purpose applications on the system.

User Classes with Functionality:

- **Administrator:**
 - The administrator can manage room related information , students related information and course related information.
- **Student:**
 - Every student who have room in the hostel have a database and a student account to access his data.Student can check his/her data.Also he/she can update the information.

2.4 Operating Environment

Operating environment for the Hostel management system is as listed below.

- *Distributed Database*
- *Client/Server System*
- **Operating system:** Windows.
- **database:** MySql database
- **platform:** PHP/html/css/JavaScript

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).>

3 rd Sem	4 th Sem	5 th Sem	6 th Sem	7 th Sem
Not required	Not Required	Not Required	Mandatory	Mandatory

2.6 User Documentation

- The developed system should run under any platform i.e. Windows.
- All mandatory fields should be filled by an individual.
- There can be security risks involved.
- Details provided by the individual during his sign up should be stored in database.
- Student details can update or change by administrator/database manage or by student.

2.7 Assumptions and Dependencies

- The details related to the student, rooms, mess.
- Administrator is created in the database already.
- Roles and tasks are predefined.

● 3 System Features

3.1.1 Login/Access user account

3.1.1.1 Description and Priority

- The user can access the account by using unique Login ID and Password.
- As the priority of this feature is high.

3.1.1.2 Stimulus or Response Sequence

Use Case: Login/Access user account

Actor: User, Database

Scenario:

- The user enters a login name.
- The user enters a password.
- The local database confirms the password through the local database.

3.1.1.3 Functional Requirements

1. User needs to provide correct Login ID.
2. User needs to provide correct password.

3.1.2 User Registration

3.1.2.1 Description and Priority

- The user can create the account by entering the user details including email id and password.
- As the priority of this feature is high.

3.1.2.2 Stimulus or Response Sequence

Use Case: User registration

Actor: User, Database

Scenario:

- The user enters all user details.
- The local database store all user information.

3.1.2.3 Functional Requirements

1. User needs to provide all registration details.
2. User needs to provide unique registration number.

3.1.3 Add/Manage course

3.1.3.1 Description and Priority

- Admin can add courses or manage the already added courses.
- As the priority of this feature is high.

3.1.3.2 Stimulus or Response Sequence

Use Case: Add/Manage courses

Actor: Admin, Database

Scenario: Add course

- The Admin enters the course short name and course full name.
- The local database store the information.

Scenario: Manage course

- The Admin edit/delete course.
- The local database update/delete information.

3.1.3.3 Functional Requirements

1. Admin needs to provide all information while adding course.

3.1.4 Add/Manage Room

3.1.4.1 Description and Priority

- Admin can add room or manage the already added room.
- As the priority of this feature is high.

3.1.4.2 Stimulus or Response Sequence

Use Case: Add/Manage room

Actor: Admin, Database

Scenario: Add room

- The Admin enters the room number and fees per student.
- The local database store the information.

Scenario: Manage room

- The Admin edit/delete room.
- The local database update/delete information.

3.1.4.3 Functional Requirements

1. Admin needs to provide all information while adding room.

3.1.5 Manage Students

3.1.5.1 Description and Priority

- Admin can manage the student's information.
- As the priority of this feature is high.

3.1.5.2 Stimulus or Response Sequence

Use Case: Manage students

Actor: Admin, Database

Scenario:

- The Admin edit/delete student's information.

→ The local database update/delete the information.

3.1.6 Update Registration-details (My Profile)

3.1.6.1 Description and Priority

- The user can update his registration information except the email address.
- As the priority of this feature is high.

3.1.6.2 Stimulus or Response Sequence

Use Case: My Profile

Actor: User, Database

Scenario:

- The user enters the information which he need to change/update.
- The local database update that user information.

3.1.6.3 Functional Requirements

1. User needs to provide informormation.

3.1.7 Change Password

3.1.7.1 Description and Priority

- The user can update his password.
- As the priority of this feature is high.

3.1.7.2 Stimulus or Response Sequence

Use Case: Change Password

Actor: User, Database

Scenario:

- The user enters the new password.
- The local database update the password.

3.1.7.3 Functional Requirements

1. User needs to provide old password.

3.1.8 Book Hostel

3.1.8.1 Description and Priority

- The user can book hostel by entering his information(room related info,personal info,address).

→ As the priority of this feature is high.

3.1.8.2 Stimulus or Response Sequence

Use Case: Book hostel

Actor: User, Database

Scenario:

- The user enters the room related information, personal information and address.
- The local database store all user information.

3.1.8.3 Functional Requirements

2. User needs to provide all mandatory information.

● 4 External Interface Requirements

4.1 User Interfaces

Front-end Software: Visual studio code.

Back-end Software: phpMyAdmin.

4.2 Hardware Interfaces

Windows

Browser which supports CSS,HTML,javascript

4.3 Software Interfaces

Following are the software used for our Hostel Management System.

Software used	Description
Operating system	We have chosen the Windows operating system for its best support and user-friendliness.

Database	To save the room,students related information we have chosen MySql database.
Visual studio code	To implement the project we have chosen PHP,HTML,CSS,javascript language for its more interactive support.

4.4 Communications Interfaces

This project supports all types of web browsers.

• 5 Other Nonfunctional Requirements

5.1 Performance Requirements

As the project is web based so there are no specifications for performance but in further updates of the project the user will be able to select the products they wish to buy and will also be able to generate a rough expenditure(Quotation) for the products.

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.>

3 rd Sem	4 th Sem	5 th Sem	6 th Sem	7 th Sem
Not Required	Not Required	Required, if requireme nt needs	Required, if requireme nt needs	Required, if requireme nt needs

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.>

3 rd Sem	4 th Sem	5 th Sem	6 th Sem	7 th Sem
Not Required	Not Required	Required, if requirement needs	Required, if requirement needs	Required, if requirement needs

5.4 Software Quality Attributes

- **Responsive website:** each pages will be responsive.
- **Usability:** website interface is very easy to use from user side.
- **maintainability:** maintainability for this website is very easy from server side.
- **Availability:** all student wise information will be available to admin.

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.>

3 rd Sem	4 th Sem	5 th Sem	6 th Sem	7 th Sem
If applicable	If applicable	If applicable	If applicable	If applicable

Appendix B: Analysis Models

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

3 rd Sem	4 th Sem	5 th Sem	6 th Sem	7 th Sem
If applicable	If applicable	If applicable	If applicable	If applicable

Appendix C: Issues List

< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>

3 rd Sem	4 th Sem	5 th Sem	6 th Sem	7 th Sem
If applicable	If applicable	If applicable	If applicable	If applicable

References

<https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>