Software Requirements Specification

for

HostelDesk

Version 1.0

Prepared by

Group No - 3

Sanil Mishra	B200057CS	sanil_b200057cs@nitc.ac.in
Daruri Kaustubh	B200825CS	daruri_b200825cs@nitc.ac.in
Gowri B Kumar	B200699CS	gowri_b200699cs@nitc.ac.in
Hari Narayanan J	B200741CS	harinarayan_b200741cs@nitc.ac.in
Kaustubh Bahl	B200833CS	kaustubh_b200833cs@nitc.ac.in
Juby Johnson	B200051CS	juby_b200051cs@nitc.ac.in
Elsa Mary Jibiachen	B200049CS	elsa_b200049cs@nitc.ac.in
Manideep Reddy	B200823CS	belaganti_b200823cs@nitc.ac.in
Mohamed Afthab E K	B200719CS	mohamed_b200719cs@nitc.ac.in
Solanki Himalay Harijvan	B200855CS	solanki_b200855cs@nitc.ac.in
Gigil James	B200771CS	gigil_b200771cs@nitc.ac.in

Contents

1 Introduction

1.1 Document Purpose

The Software Requirements Specification (SRS) provides a complete description of the requirements needed for the HostelDesk Project. It provides an understanding of the framework of the HostelDesk. This SRS is the basic foundation for the project. The project team uses it for the future stages of the project. Also, the end users can know whether this project will build a system that fulfills their requirements.

1.2 Product Scope

The HostelDesk software will automate the current working of the hostels and mess in NITC. It allows the students to enroll and reserve their rooms. It also allows them to give a preference order for their mess. The hostel managers can approve rooms for the eligible students. The students can inform the hostel authorities about any complaints regarding the hostel. The system's end users are hostel managers, mess managers, and students. The administrators can have access to all three system functionalities without any restrictions. The students will have limited access to the system. To keep restrictions for different end-user levels, HostelDesk will create distinct login functions.

1.3 Objective

The main objective of the HostelDesk is to simplify the day-to-day processes of the hostels and mess. It reduces data redundancy and human error to some extent. The software can provide a solution for a large amount of file handling in the hostels and mess. Safety, ease of usage, and, most importantly, the efficiency of information retrieval are some benefits that the project team puts forward with this system. The system will be user-appropriate, easy to use, and have a high overall end-user subjective satisfaction.

1.4 Document Convention

This document was created using Microsoft Word and uses the font Arial. A font size of 18pt has been used for the headings, 17pt for the sub-headings, and 16pt for the content.

2 Overall Description

2.1 Project Overview

HostelDesk is a web application that mainly aims to automate the hostel room allocation and the process of choosing a mess. Currently, our students are filling up forms and submitting them in the respective hostel offices, which involves much paperwork and is less efficient.

2.2 Product Functionality

The Web Application has the following:

1. Administrators
2. Hostel Managers
3. Mess Managers
4. Students

Students can apply for room allocation and are allotted rooms according to vacancy. The Administrator appoints Hostel Managers. Students can also choose their mess through the portal. Mess Managers can, in return, see the students registered under their mess. The Administrator also appoints Mess Managers.

3 Use Case Model

3.1 Use Case #1: Appoint Hostel Manager

Purpose - This use case demonstrates the appointment of a hostel manager.

Preconditions - The appointed hostel manager should not be in charge of any other hostel.

End User - Administrator

3.2 Use Case #2: Remove Hostel Manager

Purpose - This use case deals with removing a previously appointed hostel manager

Preconditions - The hostel manager that is being removed should be previously appointed.

End User - Administrator

3.3 Use Case #3: Appoint Mess Manager

Purpose - This use case demonstrates the appointment of a mess manager.

Preconditions - The appointed mess manager should not be in charge of any other mess.

End User - Administrator

3.4 Use Case #4: Remove Mess Manager

Purpose - This use case deals with removing a previously appointed mess manager.

Preconditions - The mess manager that is being removed should be previously appointed.

End User - Administrator

3.5 Use Case #5: View Students

Purpose - This use case shows that the end-user can view details of the students by entering their roll number.

Preconditions - No specific condition.

End User - Administrator

3.6 Use Case #6: Expel Student

Purpose - This use case shows that the end-user can permanently remove the student.

Preconditions - The student must be initially part of some hostel.

End User – Administrator

3.7 Use Case #7: Vacate Student

Purpose - This use case shows that the end-user can vacate the students in the database.

Preconditions - No specific condition.

End User - Hostel Manager

3.8 Use Case #8: View Student Details

Purpose - This use case shows that the end-user can view details of the allotted students.

Preconditions - The student must reside in the same hostel as the manager manages.

End User - Hostel Manager

3.9 Use Case #9: Submission of Hostel Application

Purpose - This use case deals with the provision that allows the user to submit hostel allocation forms.

Preconditions - No specific condition.

End User – Student

3.10 Use Case #10: Choosing Mess

Purpose - This use case deals with allowing users to choose their mess.

Preconditions - No specific condition.

End User - Student

3.11 Use Case #11: View Profile

Purpose - This use case deals with the provision that allows user to view their profile

Preconditions –No specific condition

End User - Student

3.12 Use Case #12: View Student Enrolled

Purpose - This use case shows that the end-user can view whether or not a student is allotted to their mess by roll number.

Preconditions - The student must be allotted to the same mess as the manager manages.

End User - Mess Manager

3.13 Use Case #13: View List of Students

Purpose - This use case shows that the end-user can view the list of students allotted to the mess in a CSV file.

Preconditions - No specific condition

End User - Mess Manager

4 System Requirement Specification

4.1 Functional System Requirements

This section gives the functional requirements that apply to the HostelDesk. These are the sub-modules in this phase.

4.1.1 Administrator Module

The Administrator can:

1. Appoint a Hostel Manager
2. Remove a Hostel Manager
3. Appoint a Mess Manager
4. Remove a Mess Manager
5. View the details of the students
6. Expel students permanently from the hostel

4.1.2 Hostel Manager Module

The Hostel Manager can:

1. Vacate students enrolled in their hostel
2. View details of students enrolled in their hostel

4.1.3 Mess Manager Module

The Mess Manager can:

1. View relevant details of students enrolled in their mess

4.1.4 Student Module

The students can:

1. Submit the application form for hostel allocation
2. Choose their mess

4.2 Non-Functional System Requirements

4.2.1 Performance Requirements

Some performance requirements needed are listed below:

1. The database should be capable of storing more than 15000 records.

2. The software should support multiple users at a time.

4.2.2 Safety Requirements

There are chances of crashes in a database at any time due to malware attacks or system failure. So, it is necessary to back up the database.

4.2.3 Security Requirements

Some factors identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below.

1. Later version of the software will incorporate encryption
2. Keep specific log or history data sets.
3. Check data integrity for critical variables.
4. Restrict communication between some areas of the program

4.2.4 Software Quality Attributes

1. Less human error
2. Manual effort can be reduced
3. High Security
4. Data consistency
5. Easy to handle a large volume of data
6. Simpler updation and deletion of data
7. More organized form of maintaining records
8. Data redundancy can be avoided to some extent

5 Hardware Requirements

1. CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU
2. Cores: Single (Dual/Quad Core is recommended)
3. RAM: 4 GB (6 GB recommended)
4. Graphic Accelerators: Nvidia or ATI with support of OpenGL 1.5 or higher
5. Display Resolution: 1280x1024 is recommended, 1024×768 is minimum
6. Hard Disk: 64GB or more