Software Requirements Specification

for

Hostel Management System

Prepared by Group 6

Instructor: Dr. Saurabh Tiwari and Dr. Manish Khare

Course: IT314 Software Engineering

Teaching Assistant:

Date: 24/04/2023

Table Of Contents

1. Introduction

1.1. Problem statement: Hostel Management System

Hostel management system is a software application that helps in managing the day-to-day operations of a hostel. The system also allows for easy communication between hostel staff and students, making it a useful tool for both parties. The use of a hostel management system can greatly improve the efficiency and organization of a hostel, resulting in a better experience for both students and staff.

1.2. Purpose

The purpose of developing a hostel management system is to automate and streamline the various operations and processes involved in managing a hostel. With this system in place, the residents of the hostel can easily check the couriers that may arrive for them, search for an item in the lost and found list, directly post their complaints and also check the notices. Additionally, the system facilitates communication between staff and students, enabling them to share information, updates, and notifications quickly and easily.

The ultimate goal of implementing a hostel management system is to improve the efficiency and organization of the hostel, providing a better experience for both students and staff. The system can help eliminate manual errors, reduce paperwork, and save time, enabling staff to focus on providing better services to students.

In summary, the purpose of a hostel management system is to optimize the management of a hostel's day-to-day operations, leading to improved efficiency, better communication, and a better overall experience for all stakeholders.

1.3. Product Scope

- Complaint Management: The website should allow students to submit complaints, and administrators should be able to track and manage complaints effectively.
- Notices Management: The website should allow students to view latest notices and administrators to post notices.
- Couriers Management: The administrator should be able to post the newly arrived couriers in order to update the students regarding any couriers.
- Item Management: The user(admin or student) can post if any item is lost or found.

1.4. Intended Audience and Reading Suggestions

- The intended audience will be hostel managers, students and administrators who are responsible for the day-to-day operations of the hostel.
- This would include tasks such as notice board, complaints, courier, lost and found.

2. Overall Description

2.1. Product Overview

All the tasks in the hostel are done manually right now thus a system is required to organize and maintain hostel records. The following are the needs that we face daily:

- Announcement notices about the hostel to all residents.
- Managing the records of complaints and their status.
- To manage the courier service inside campus.
- Housekeeping management (sweeper, bathroom cleaners)
- A lost and found section to address the lost belongings.

2.2. User Classes and Characteristics

We have 2 user classes:

1. Student:
   Users who are seeking to see daily updates of the hostel and can view Notices and Couriers.
2. Admin:
   User who is responsible for updating daily affairs and can see complaints from students and post Notices.

Characteristics for each user class:

Student

- Level of access:
  Students would have access to view Notices from admin, couriers received every day and can post complaints to administrators and can view status of that complaint.

- Technical expertise:
  Students may have varying levels of technical expertise and familiarity with the website, so it should be designed to be user-friendly and easy to navigate.

- Role within the organization:
  Students are the primary users of the Hostel management system, and its main purpose is to get updates.

- Frequency of use:
  Students may use the Hostel Management System daily.

Admin

- Level of access:
  administrators will have access to view new complaints, update their status, and post Notices.

- Technical expertise:
  administrators may have varying levels of technical expertise and familiarity with the website, so it should be designed to be user-friendly and easy to navigate.

- Role within the organization:
  administrator is responsible to manage day to day operations of the Hostel.

- Frequency of use:
  administrator may use the website almost daily because he has to get updates from students and he needs to post updates to the website so students can view those things.

2.3. Operating Environment

- The website will be designed to run on a Node server engine that supports the required programming languages and frameworks.
- The website will require a NoSQL database management system such as MongoDB to store user data and job postings.
- The website's development and maintenance will require consistent efforts.

2.4. Assumptions and Dependencies

2.4.1. Assumptions

- The website will be used to connect students with the hostel supervisor and the staff.
- Students will be able to report complaints, submit lost/found items, check couriers and read notices uploaded by the admin.
- Admin members will be able to add/delete lost and found items, add couriers and its intended recipient, post notices and respond to the couriers using this system.
- The website will be compatible with major web browsers.
- The website will need to comply with relevant laws and regulations, such as those governing user data privacy and job advertising.

2.4.2. Dependencies:

- The website will require a database to store user information and the information generated by users like lost and found or complaints.
- The website will need to be hosted on a reliable and scalable web hosting platform to ensure high availability and performance.
- The website will need to be developed using appropriate web development tools and technologies, such as HTML, CSS, JavaScript, and a web framework like Node.js and React.

3. Specific Requirements

3.1. External Interface Requirements

- User Interface: The system should have a user-friendly interface that is easy to navigate and understand. It should have a clean and modern design with clear instructions and feedback for users.
- Browser Compatibility: The system should be compatible with different web browsers such as Google Chrome, Mozilla Firefox, and Safari, to ensure accessibility for all users.

3.2. Functional Requirements

A. The Administrator can:

- Post notices regarding the hostel.
- Modify the student records.
- They can see registered complaints and set status.
- They can manage facilities and status of people/ workers associated with it.

B. The User can:

- It makes the registration forms(Complaints/Permissions) accessible to the various users.
- They can file a complaint as well as see their previous complaint status.
- They can see notifications from the Notice Board.
- They can view and search Couriers.
- Claim a lost item or post an entry for a found lost item.

3.3. Non Functional Requirements

- **Security constraint:**
  - The user's login information should be secure enough so that anyone except for the developers will know the user's username/email and password.
  - When the user types their password to login, it should be hidden.
  - The system should be secure enough so that a user's personal information is not disclosed to other users.

- **Performance Requirements**
  - Any link on the website must be functional and should take the user to the intended webpage.
  - Updating and fetching information should be efficient.

- **File type constraint:**
  - It should be a web application.

    The development of web application is more feasible otherwise in android application will affect the economical constraints. Considering the application must be compatible in IOS too.

- **Accessing:**
  - The users can access the web application from anywhere. For instance a user should be able to view notices from anywhere regardless of their location.

- **Ease of use:**
  - The application aids users to easily communicate with the staff.
  - An easy to operate interface should be implemented to facilitate satisfied user experience.

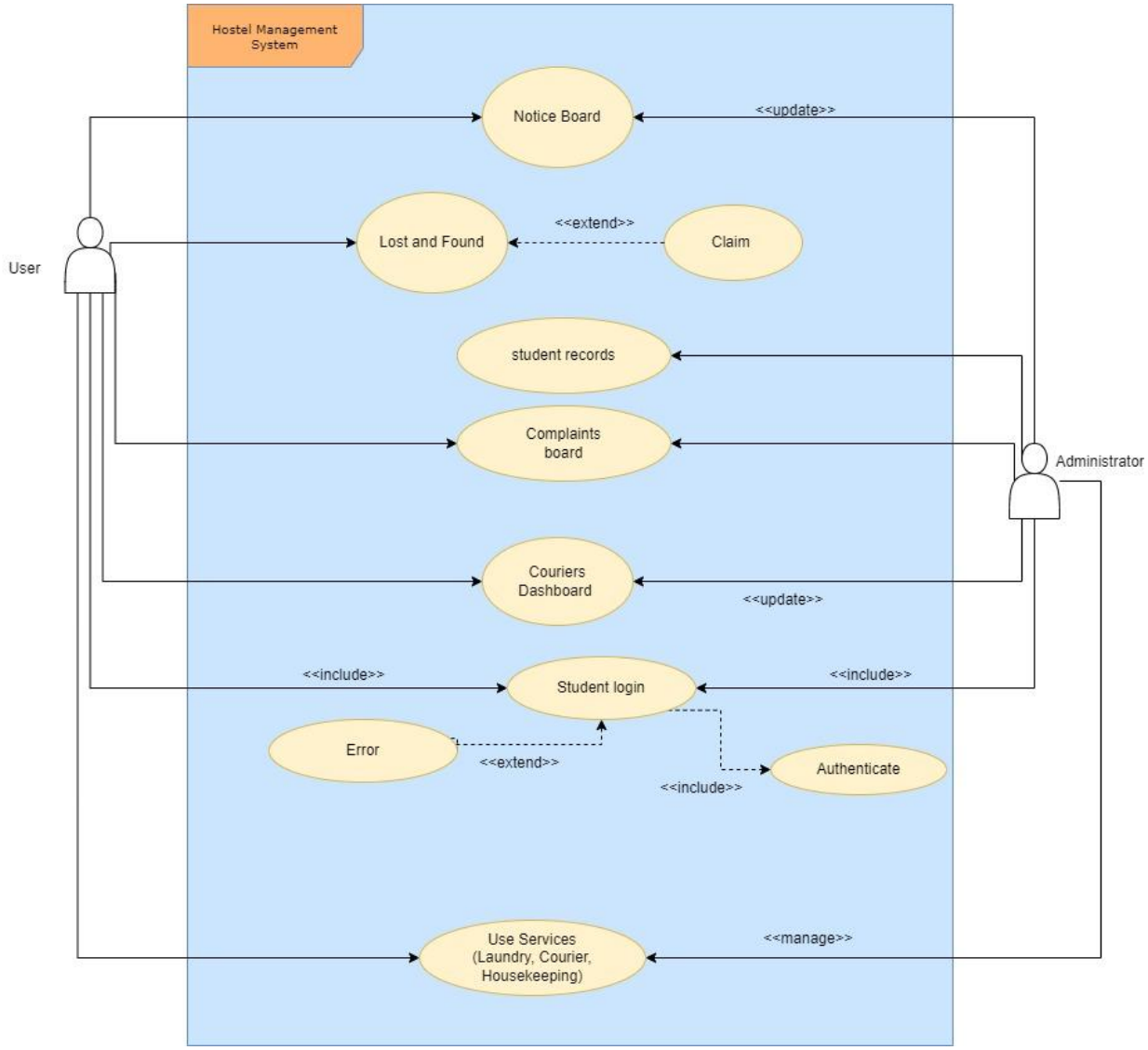
- **Maintenance:**
  - Post deployment the smooth functioning of the application is desired.
  - Every functionality of the system should be up and running as expected.

- **Reliability**:
  - The system should be available and reliable 24/7, with minimal downtime for maintenance or upgrades.
  - The system should be able to recover from any failures or errors without losing data or causing disruption to hostel operations.

- **Compatibility:**
  - The application must be flexible enough to support Google Chrome and Firefox 18 browsers.
  - The application must be able to support Windows and Mac OS operating systems.
  - The application must be flexible enough to support users with different power companies.

4. Software Model

4.1. Use Case Model



4.2. User Stories:

1. Notice Board

Name: Notice board

- Identifier: UC1
- Actors: Administrator, Student
- Goal: Post important announcements on website visible to all students
- Pre-conditions: The students should be logged into their account before accessing the notices webpage.
- Description:
    1. Administrator selects to make the announcement.
    2. The HMS asks the administrator to enter the title and description of the announcement.
    3. Administrator selects post option.
- Post-conditions:
    - All students can see the announcement.
- Exceptions:

3.a. Administrator tries to post without filling in all the information
3.a.1. HMS continues to step 2.

2. Lost Found

Name: Lost Found

- Identifier: UC2
- Actors: Student
- Goal: Students should be able to Post lost items.
- Pre-conditions: None
- Description:
    1. Students select to add lost items to the list.
    2. The HMS asks the Students to enter the title and description of the item lost.
    3. Student selects the post option.

- Post-conditions:
    - All students can see the found items list.

- Exceptions:

3.a. Student tries to post without filling in all the information

3.a.1. HMS continues to step 2.

3. Complaint Box

Name: Complaints board

- Identifier: UC5
- Actors: Administrator, Student
- Goal: Students can enter their complains and see its status
- Pre-conditions: None
- Description:
    1. Student selects to lodge a complaint.
    2. The HMS asks the student to enter the title and description and concerned department of the complaint.
    3. Student selects the post option.
    4. Administration can change the status of the complaints

    (Posted -> In progress -> Completed)
- Post-conditions:
    - All students and administration can see the complaint and its status.

- Exceptions:
  3.a. Student tries to post without filling in all the information
  3.a.1. HMS continues to step 2.

4. Courier Dashboard

Name: Courier Dashboard

- Identifier: UC6
- Actors: Administrator, Student
- Goal: Post couriers received on website visible to all students
- Pre-conditions: None
- Description:
  1. Administrator chooses to update the courier information.
  2. The HMS asks the administrator to enter the courier details and specify the respected owner.
  3. Administrator selects post/update option.
- Post-conditions:
  ○ All students can see the courier details.
- Exceptions:
  ○ 3.a. Administrator tries to post without filling in all the information
  3.a.1. HMS continues to step 2.

5. Login

Name: Login

- Identifier: UC8
- Actors: User, Administrator
- Goal: Users and Administrator can access all the appropriate functionalities of the system after successful login.
- Pre-conditions: User must have institute email id.
- Description:
  1. The HMS asks the administrator/user to enter an email address and password.
  2. User enters its email address and password to login and view the dashboard when verified.
- Post-conditions:

- ○ All students can access the HMS functionalities.
- ○ administrators can access its functionalities.

- Exceptions:

3.a. Administrator/Users enters email address and password to signup

3.a.1. HMS continues to step 2.

6. Authentication

Name: Authentication

- Identifier: UC9
- Actors: Administrator, Student
- Goal: When a user tries to login, check whether the entered details match with the database.
- References to requirements: <number in requirement>
- Pre-conditions: None
- Description:

1. The HMS checks the entered password with their database.

- Post-conditions:
  - ○ User or Administrator redirected to the home page.
- Exceptions:

1.a. User enters invalid password, redirect to login page.

7. Error

Name: Error

- Identifier: UC10
- Inherits from: UC8
- Actors: Administrator, Student
- Goal: Show error message if the login fails.

Pre-conditions: None

- Description:

1. Display login failure message

- Post-conditions:

- All students can see the courier details.

- Exceptions:

  3.a. Administrator tries to post without filling in all the information

  3.a.1. HMS continues to step 2.