SOFTWARE REQUIREMENTS SPECIFICATION

for

Hall Management Software

Version 1.0

Prepared by :
1. Ojas Dubey (22CS30039)
2. Shreya Bose (22CS30050)
3. Nived Roshan Shah (22CS10049)

April 2, 2024

Contents

1 Introduction

1.1 Purpose

This document aims to outline the design of a web application tailored to meet the requirements of the Hall Management Center at IIT Kharagpur. The application streamlines various administrative tasks associated with managing the different halls. It represents the initial iteration of the application, encompassing a comprehensive system with self-contained functionalities, including necessary interfaces and a corresponding backend to facilitate interactions for accomplishing diverse management functions

1.2 Document Conventions

Bold text is used for software frameworks and tools. *Italicized* text is used for specific end-users of the application.

1.3 Intended Audience and Reading Suggestions

The Software Requirements Specification (SRS) targets developers and testers directly involved in implementing the application. It is recommended that readers acquaint themselves with the standard operating procedures of the halls at IIT Kharagpur, as this understanding will facilitate the comprehension of the design decisions outlined in this document.

1.4 Product Scope

The proposed software system will function as a comprehensive Hall Management System tailored for our institute. It aims to enhance the efficiency of all hall-related processes within our institution. Serving as a centralized portal, it will enable students, wardens, hall clerks, and the HMC Chairman to access pertinent information pertinent to their roles. Specifically, the system is engineered to foster communication among various stakeholders, such as students and wardens, thereby promoting seamless operations within the halls and establishing a structured approach to managing expenses and allocations.

1.5 References

- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- SRS for ACM SIGSOFT Newsletter Content Management & Generation System-II

2 Overview

2.1 Product Perspective

This application is essential for consolidating various bookkeeping operations susceptible to errors and irregularities. The manual handling of these tasks separately by different halls at IIT Kharagpur can result in redundancies and inconsistencies. Moreover, students lack a straightforward mechanism to manage their interactions with their respective halls. For instance, filing complaints often necessitates students to personally approach multiple individuals for resolution, sometimes without any centralized complaint records, thus lacking accountability. Numerous such scenarios offer opportunities for digitization and the associated benefits. This document outlines our vision for a web application designed to address these challenges in a user-friendly manner.

2.2 Product Functions

1. User Registration
2. Complaint resolution
3. Notifying students
4. Dues settlement
5. Grant management
6. Budgeting
7. Personnel management

2.3 Client Classes and Characteristics

The following client classes have been categorized according to the subset of product features they utilize. Each client class is provided with a distinct interface within the application, accessing functions that are pertinent to their respective roles.

1. *Students*: Pay various fees and dues, register and view complaints, View notices.
2. *Mess managers*: Handle and log the day-to-day expenditures of the mess, Generate menu on a monthly basis.

Figure 2.1: User classes

3. *Hall managers*: Manages Registration of students into the Hall and allocates rooms in an automated fashion, Manage the employment and salaries of temporary hall employees, Manage Hall expenses, Creates notices, generates action taken reports(ATR).

4. *Wardens*: View ATRs, Manage payments to hall and mess managers, handle financial grants from the HMC, Generate dues for the student, allocate finances to hall and mess

5. *The HMC*: View various expenditures, Allocation of grants to different halls

2.4 Operating Environment

The application is designed to utilize the platform-agnostic **Python 3.0+** interpreter along with associated frameworks. The application server responsible for handling incoming requests can run on any platform equipped with a TCP/IP stack supporting **Django 4.0+** and **Python 3.0+**. As a result, the web application can be deployed on Windows Server or Linux-based distributions that support these tools.
For optimal functionality, the server should be connected to the local area network provided by the Computer and Informatics Centre at IIT KGP and appropriately configured to handle incoming requests. Any relevant firewall settings must be adjusted to allow high traffic to the server without dropping any requests.

2.5 Design and Implementation Constraints

Developers are limited to using Python for providing the backend services for the web application for the following reasons:

1. Professors have enforced a rule necessitating the usage of either **C++**, **Java** or **Python**. This eliminates the possibility of writing the codebase with just **HTML**, **CSS** and **JavaScript** for the front-end combined with **JavaScript** on the backend (**Node.js**, etc.)

2. **C++** and **Java** do not possess easy to use, batteries included frameworks like **Python**. The development time available for this project is less than three weeks. This necessitates the usage of pre-built mechanisms and libraries that Python possesses.

   With these two constraints in mind, we have decided to use **Python** along with the **Django** framework for the web application.

2.6 User Documentation

The web application will feature a self-documenting, intuitive interface that eliminates the need for specific manuals for users. All relevant information regarding the usage of the web application will be strategically placed within the different webpages and displayed as needed, ensuring ease of use and accessibility for all users.

2.7 Assumptions and Dependencies

1. The existence of a local network within the KGP campus. If this is absent, additional configuration would be required in terms of DNS, hosting, ISP handling, etc.

2. End users should be able to connect to this network, otherwise access to the application will be limited to those with direct access to the servers.

3. The network should not have significant downtimes that can affect the utility of the application and its ability to handle incoming requests.

3 External Interface Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

1. The hardware used for this needs to be server computer with a processor fast enough to handle a large number of requests simultaneously. Preferably a multi-core system with distributed processes across the core will be suitable.

2. Solid state drives instead of hard disks to boost the speed of information retrieval. Enables faster request handling.

3. RAID systems to manage data redundancy. Ensures no user data is lost because of storage device failures by maintaining redundant copies as backup.

3.3 Software Interfaces

1. A Web browser needed on the end-users' systems to access the web application

2. Graphical user interfaces combined with the browsers needed. Command-line interface will not be able to meaningfully handle the user interfaces provided in the web application.

3. In order to set up the system, some web server like Heroku may be needed for deployment.Currently the system runs on the local device of the user.

4. The client will need an active email address for verification.

3.4 Communications Interfaces

1. The system uses a standard TCP/IP stack available on most operating systems.

2. The system uses the **HTTP** protocol to deliver web pages and handle user login

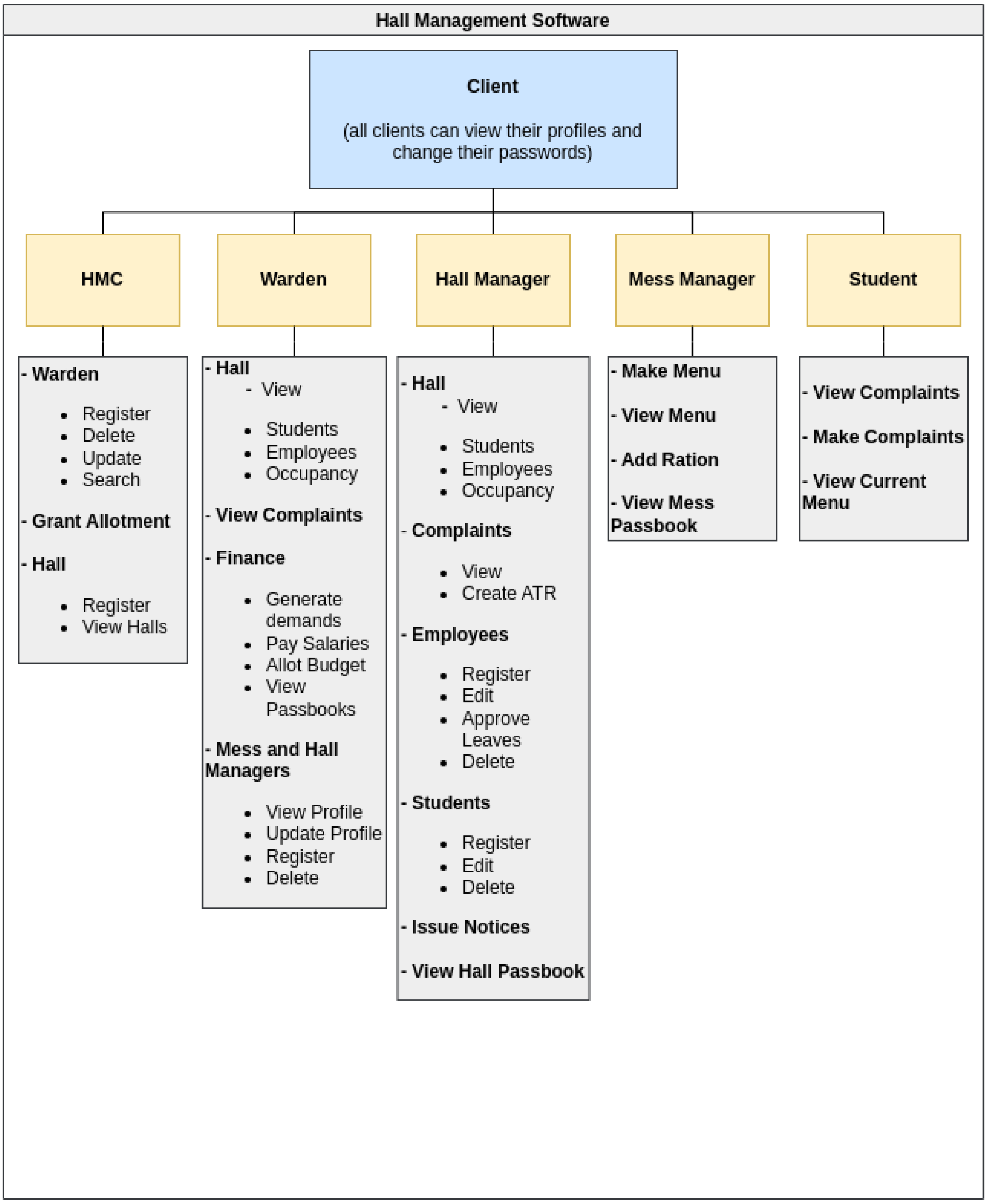

Figure 3.1: Classes and their functionalities

4 System Features

4.1 User Registration/Deletion

4.1.1 HMC

Description

The HMC is created in the django admin page. This is to be created by the System maintainer.

Stimulus/Response Sequences

- Stimulus: The college wishes to register a Hall management committee into the system
- Response: Maintainer can create an HMC from the django admin interface.

Functional Requirements

- **REQ1**: Django admin authorisation
- **REQ2**: Digital form for creation of HMC with superuser privileges

4.1.2 Hall

Description

Halls along with there residence specifications can be created by the HMC. Only the HMC and the Warden has access to this page.

Stimulus/Response Sequences

- Stimulus: A new Hall needs to be registered in the system.
- Response: System adds the hall and creates rooms in the database along with their specified rent and occupancy.
- Stimulus: The status of the hall needs to be checked regarding occupancy.
- Response: System queries the database and displays the status of all rooms in the hall.

Functional Requirements

- **REQ4**: Digital forms for the creation of halls.
- **REQ5**: Automated filling of the database with room numbers based on the specifications provided regarding the blocks and floors.
- **REQ6**: Page to show the data of each hall based on the current and the maximum occupancy.

Clarity in occupancy of the hall makes it easier for the HMC to approximate allowable funds for the ration etc for the hall.

4.1.3 Warden

Description

Wardens can be allocated to halls by the HMC. The details can be added by a page only accessible to the HMC.

Stimulus/Response Sequences

- Stimulus: HMC wishes to allocate a warden to the hall via the registration page.
- Response: The form needs to be filled and the email needs to be verified by the warden in order to activate their account.

- Stimulus: HMC wishes to update the information for the allocated hall warden.
- Response: The database is queried and the data is pre-furnished in the form in order to update the data of the warden.

- Stimulus: HMC wishes de-allocate a hall to a warden.
- Response: Confirmation page for the deletion for the warden. The system deletes the warden from the hall data.

Functional Requirements

- **REQ7**: Digital forms for the registration and allocation of the wardens.
- **REQ8**: Email verification needs the existence of a unique email per each warden.
- **REQ9**: User authentication page before deletion of warden data.

Email verification provides 2-step authentication system for security. Security during the deletion of the registered hall warden ensures only authenticated users can de-allocate wardens from the hall.

4.1.4 Hall/Mess manager

Description

Wardens can appoint an register managers for their Hall. The details can be added by a page only accessible to the Warden.

Stimulus/Response Sequences

- Stimulus: The warden wishes to allocate a manager to the hall via the registration page.
- Response: The form needs to be filled and the email needs to be verified by the manager in order to activate their account.
- Stimulus: The warden wishes to update the information for the allocated manager.
- Response: The database is queried and the data is displayed pre-filled in the form in order to update the data of the manager.
- Stimulus: Warden wishes to de-register a manager.
- Response: Confirmation page for the deletion for the hall manager. The warden must enter their password before progressing for deletion.

Functional Requirements

- **REQ10**: Digital forms for the registration and allocation of the manager.
- **REQ11**: Email verification needs the existence of a unique email per each manager.
- **REQ12**: User authentication page before deletion of manager data.

Email verification provides 2-step authentication system for security. Security during the deletion of the registered manager ensures only authenticated users can de-allocate managers from the hall.

4.1.5 Students

Description

Hall managers have access to the page to register the students in their halls. Like other registration pages, Email verification is mandated here as well for the students. Additionally, automated room allocation based on the preferences of the students is also possible.

Stimulus/Response Sequences

- Stimulus: Hall management wishes to register students in the hall.
- Response: Students are added to the database on the form input via Email verification.
- Stimulus: While registration, Student wishes to declare their room preferences.
- Response: Up to 3 room preference can be added in the form. The system will allocate one of the rooms of preference if available, otherwise a room is randomly allocated.
- Stimulus: The manager wishes deletion of student data from the hall database.
- Response: Up to 3 room preference can be added in the form. The system will allocate one of the rooms of preference if available, otherwise a room is randomly allocated.

Functional Requirements

- **REQ13**: Digital forms for the registration and room allocation for a student.
- **REQ14**: Email verification needs the existence of a unique email per each student.
- **REQ15**: Choices by a student can be filled in a priority ordering.

Deletion page enables the hall to clear the data of the student in the event of a hall re-allocation or graduation.

4.1.6 Hall employees

Description

Hall managers have access to the page to employ workers for the hall. As the hall employee is not a direct user of the system, no email verification is needed here. Roles and salaries can be fixed during registering.

Stimulus/Response Sequences

- Stimulus: Hall management wishes to employ a hall worker.
- Response: Page with personal details, job, salary bracket is furnished.

Functional Requirements

- **REQ16**: Digital forms for the employee details to be furnished by the manager.

Salary brackets are currently restricted to a few limited options to be selected by the manager.

4.2 Budget allocations

4.2.1 HMC

Description

Grants may be allocated to the halls based at demand via a restricted page. The funds are added to the Warden's budget to manage from that point onwards.

Stimulus/Response Sequences

- Stimulus: Allocation of grants to the halls by the HMC.
- Response: The System renders the form for the allotment after validating user. Allotted budget is added to the warden passbook.

Functional Requirements

- **REQ1**: Digital form for budget allocation.
- **REQ2**: Allocated budget is displayed in the passbook of the warden.

4.2.2 Warden

Description

The warden, upon receiving budget from the HMC can decide how much allocation of funds is needed to the mess and the hall facilities respectively.

Stimulus/Response Sequences

- Stimulus: Warden wants to distribute the funds to the mess and the hall.
- Response: System verifies the password of the user and proceeds to allocate the funds to the hall and the mess passbooks respectively.
- Stimulus: Warden wishes to view the past allocations to the hall and mess.
- Response: The warden passbook page displays the past allotments with their timestamp

Functional Requirements

- **REQ3**: Digital forms for the filing of action taken reports.
- **REQ4**: Passbook to display the past allocations both to and by the warden.

4.3 Hall employee handling

4.3.1 Salaries

Description

The warden is authorised to grant the salaries to the employees out of the hall budget. This is logged into the hall passbook.

Stimulus/Response Sequences

- Stimulus: Salaries are to be paid to the hall employees.
- Response: System verifies password of current logged in user. The page is available only to the hall warden.

Functional Requirements

- **REQ1**: The salaries of each type of employee is queried and aggregated and added to the hall passbook along with the timestamp.

4.3.2 Leaves

Description

The hall manager can register the duration of permitted leave and whether or not the leave will be paid based upon documentary uploads.

Stimulus/Response Sequences

- Stimulus: Hall employee leave requests need to be processed by the hall manager.
- Response: System loads the form for the hall manager and they can select the duration of the leave. Provision to upload any documents if needed also exists.

Functional Requirements

- **REQ2**: Ability to enter leaves of absence on a digital calendar.
- **REQ3**: Calculation of amount payable based on attendance.
- **REQ4**: If the leave is to be paid, The manager must upload a document in support of it.

4.4 Fees and dues management

Description

Students are to pay for the hall and the mess charges. Both the requests must be processed separately for easy tracking. These payments must be recorded and collected without irregularities.

Stimulus/Response Sequences

- Stimulus: Students dues are to be generated and displayed to them.
- Response: The warden will generate dues to the student for both hall and mess charges separately. They must validate themselves first before generating this demand.

- Stimulus: Student requests the total amount owed with different filters (such as mess, rent, common room fees, etc).
- Response: A web page with the relevant information is displayed.

- Stimulus: Student wishes to clear the dues generated for him.
- Response: Payment portal is rendered where the student can pay their dues.

Functional Requirements

- **REQ1**: Ability to track the dues owed by different students.
- **REQ2**: The request generation is password protected to avoid errors.
- **REQ3**: Stripe payment is used here. To customize the payment options, contact the developers.
- **REQ4**: Student passbook keeps a record of all the payments made by the student. This can be downloaded by the student in PDF format.

4.5 Complaints management

4.5.1 Student

Description

Students should be able to generate complaints and view the corresponding action taken reports (ATR).

Stimulus/Response Sequences

- Stimulus: Student intents to register a complaint
- Response: System loads the form for the student. The complaint details can be filled in a text format in the form. Provision to upload any images of the complaint if needed also exists.

Functional Requirements

- **REQ1**: The complaint details can be filled in a text format in the form.
- **REQ2**: Image format can be uploaded and viewed by the manager.
- **REQ3**: The ATR can be downloaded by the student if the issue has been resolved.

4.5.2 Hall manager

Description

The manager should be able to view the complaint and assign an employee to deal with it if needed. They should also be able generate an ATR if needed.

Stimulus/Response Sequences

- Stimulus: Warden wishes to assign workers to deal with an issue raised by a student.
- Response: The warden can enter the assigned worker and the details of the action taken in a text format. The status of the complaint then can be marked as completed or not accordingly by the manager.

Functional Requirements

- **REQ4**: The ATR details can be filled in a text format alomg with the status in the form.
- **REQ5**: PDF can be generated by the manager, which cam be downloaded by the student.

4.6 Hall notices

4.6.1 Hall Manager

Description

The hall manager must have the option of adding notices in order to notify each student of the hall.

Stimulus/Response Sequences

- Stimulus: Hall manager intends to add notices.
- Response: A form that allows the hall manager to upload images and text based notices which will show up on the pages of each of the student which reside in that hall.

Functional Requirements

- **REQ1**: Ability to add text to the notice.
- **REQ2**: Ability to upload images to the notice board.
- **REQ3**: The hall manager and the warden should be able to preview the uploaded notice.

4.6.2 Student

Description

All the students should be able to view the notices uploaded by the hall management.

Stimulus/Response Sequences

- Stimulus: Students intends to view the hall notice board.
- Response: The student can navigate to the notice board page post login.

Functional Requirements

- **REQ4**: Access the notice board with only view only permissions.

4.7 Mess Management

4.7.1 Menu

Description

Mess managers can perform various functions like generating the monthly mess menu.

Stimulus/Response Sequences

- Stimulus: Mess manager wishes to generate the mess menu for the month.
- Response: The system generates a form for a week as menus are weekly based in halls. Here the mess manger can set the fields for 4 meals a day, 7 days a week.
- Stimulus: The student wishes to view the mess menu generated.

- Response: The system fetches the menu database and filters and displays the menu based on thew hall of the student.

Functional Requirements

- **REQ1**: Both the student and mess manager can view the mess menu in a neat tabulated manner.
- **REQ2**: Only the mess manager is granted access to the menu editing page.

4.7.2 Rations and Mess passbook

Description

The mess manager can log the rations used on a daily basis

Stimulus/Response Sequences

- Stimulus: Mess manager intends to add the ration used per day.
- Response: The system generates the form to .

- Stimulus: Mess manager wishes to view the expenditure of the mess.
- Response: The mess passbook page enables the mess manager to view and download the rations consumed in PDF format.

Functional Requirements

- **REQ3**: The aggregate price of the rations are calculated and added to the mess passbook automatically.
- **REQ4**: The mess passbook is neatly tabulated and the expenses are labelled according to their nature.

4.8 Hall Expense Report

4.8.1 Description

Wardens need to view the cash flow and various expenses owed. This information needs to be quickly retrieved without having to go through paper records. Mess fees, rent, grants, etc, can be centralized in a single location for the warden to review.

4.8.2 Stimulus/Response Sequences

- Stimulus: Warden requests various expenses.
- Response: A form that allows the warden to apply various filters is displayed. Based on the form input, the expenses database is filtered and the relevant information is presented, such as the amount of owed by students, amount owed to mess managers, etc.
- Stimulus: Warden requests a full accounting report
- Response: The system produces a table with all the expenses and payments accounted for. Cheques will automatically be generated. Warden will be able to print this report and the generated cheques.

Functional Requirements

- **REQ17**: Ability to generate accounting reports
- **REQ18**: Ability to generate cheques.
- **REQ19**: Ability to submit the generated reports for audit.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Server Up-time

The system needs to keep running for long periods of time for accessibility so the hardware needs to be capable of that. Moreover the maintenance frequency needs to be low to have a stable experience for the end users.

5.1.2 Response Time

The system needs to have a good response time as it will be repetitively used by various stakeholders in the Institute and the response time needs to keep up with the incoming requests. At the same time less used tasks like printing bills at the end of the month can be a little slow since they will be accessed much less frequently.

5.1.3 System Dependability

It determines the failure rate of the system, that is, the cases when the system receives an incorrect input or there is a random failure. The end user needs to be notified in such a scenario.

5.1.4 Prominent Documents

Since the system is also being used to maintain cash receipts and expenditure through the halls and generate official documents for the students' stay in their halls, the output needs to be clear, concise and consistent in a .pdf or another similar printable format.

5.2 Safety Requirements

There needs to be a regular audit to the cash flow through the hall since there is no check for the information entered by the mess manager. Moreover, audit also needs to be done for the budget distribution as there might be cases for fraud in the long run if left unchecked since the system is not verifying the data in any manner.

5.3 Security Requirements

5.3.1 Secure Login

The system should be protected against forced and malicious login attempts to access the software. The web interface needs to be continuously patched to fix bugs as and when they crop up.

5.3.2 Verifiable Access

The system is designed to serve multiple parties and their levels of access and their permissions are decided by their login IDs. Thus, the login and access to functionality should be robust and extremely secure since the system also deals with the finances of the institute.

5.3.3 Student Information Must be Secure

All the personal student information stored by the system should be safe and secure.

5.4 Software Quality Attributes

5.4.1 Reliability

The system must be dependable, which means it must constantly perform as expected with very low downtime and rate of failure. In absence of this the HCM will not be able to function smoothly.

5.4.2 Information Accessibility

To save the login credentials and the myriad types of data that the software uses, the application should be able to save, access and alter files on the system so that the information is retained even on successive runs.

5.4.3 Maintainability

The program should be easy to modify and add features to. The code should be clearly commented and the features well documented to allow for easy maintenance. The frameworks used should be well supported by their developers well into the future.