

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

Hall Management Software

Version 1.0

Prepared by : 1. Krish (21CS10037)  
2. Tanishq (21CS30054)  
3. Sourodeep (21CS00064)

April 2, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Purpose . . . . .	4
1.2	Document Conventions . . . . .	4
1.3	Intended Audience and Reading Suggestions . . . . .	4
1.4	Product Scope . . . . .	4
1.5	References . . . . .	4
<b>2</b>	<b>Overall Description</b>	<b>5</b>
2.1	Product Perspective . . . . .	5
2.2	Product Functions . . . . .	5
2.3	User Classes and Characteristics . . . . .	5
2.4	Operating Environment . . . . .	6
2.5	Design and Implementation Constraints . . . . .	6
2.6	User Documentation . . . . .	7
2.7	Assumptions and Dependencies . . . . .	7
<b>3</b>	<b>External Interface Requirements</b>	<b>8</b>
3.1	User Interfaces . . . . .	8
3.2	Hardware Interfaces . . . . .	8
3.3	Software Interfaces . . . . .	8
3.4	Communications Interfaces . . . . .	9
<b>4</b>	<b>System Features</b>	<b>10</b>
4.1	Complaint resolution system . . . . .	10
4.1.1	Students . . . . .	10
4.1.2	Warden . . . . .	10
4.2	Hall Management . . . . .	11
4.2.1	Hall clerks . . . . .	11
4.2.2	Temporary Employees' Salary Management . . . . .	11
4.3	Fees and dues management . . . . .	12
4.4	HMC Employee Management . . . . .	13
4.4.1	Description . . . . .	13
4.4.2	Stimulus/Response Sequences . . . . .	13
4.5	HMC Expenditure Management . . . . .	13
4.5.1	Description . . . . .	13
4.5.2	Stimulus/Response Sequences . . . . .	14

4.6	Hall Occupancy Management . . . . .	14
4.6.1	Description . . . . .	14
4.6.2	Stimulus/Response Sequences . . . . .	14
4.7	Hall Expense Report . . . . .	15
4.7.1	Description . . . . .	15
4.7.2	Stimulus/Response Sequences . . . . .	15
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>16</b>
5.1	Performance Requirements . . . . .	16
5.1.1	Server Up-time . . . . .	16
5.1.2	Response Time . . . . .	16
5.1.3	System Dependability . . . . .	16
5.1.4	Prominent Documents . . . . .	16
5.2	Safety Requirements . . . . .	16
5.3	Security Requirements . . . . .	17
5.3.1	Secure Login . . . . .	17
5.3.2	Verifiable Access . . . . .	17
5.3.3	Student Information Must be Secure . . . . .	17
5.4	Software Quality Attributes . . . . .	17
5.4.1	Reliability . . . . .	17
5.4.2	Information Accessibility . . . . .	17
5.4.3	Maintainability . . . . .	17

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to describe the design of a web application that can handle the needs of the Hall Management Center of IIT Kharagpur by automating various book-keeping activities associated with the management of the different halls. This is the first version of the application. The application's scope covers an end-to-end system that is self-contained, possessing all the required interfaces and a corresponding backend for interacting with the system to achieve various management tasks.

## 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

This SRS is for developers and testers who will be directly working on the implementation of the application. It is suggested that the reader familiarize themselves with the standard operating procedures of the halls in IIT KGP since this would help rationalize the various design decisions described in this document.

## 1.4 Product Scope

This software system will be a Hall Management System for our institute. This system will be designed to streamline the process of everything related to the working of halls in our institute. It will serve as a portal for the students, wardens, hall clerks and the HMC Chairman to access information relevant to the respective parties. More specifically, this system is designed to facilitate communication between the different entities such as students and wardens, which would ensure smoother running of the halls and provide an organised way to handle expenses and allocations.

## 1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

## 2 Overall Description

### 2.1 Product Perspective

This application is needed to centralize the various book-keeping operations that are prone to errors and malicious failures. Manual management of these activities separately by the different halls in IIT KGP can lead to redundancies, lack of uniformity, etc. Students do not possess a simple mechanism to manage their interactions with their halls. For example, filing complaints would require the students to personally approach various individuals for resolution, sometimes with no accountability since there are no centralized complaint records available. Similarly, there are various such use cases that offer the potential for digitisation and the benefits associated with it. This document describes our vision for a web application that will handle all these situations in an accessible manner.

### 2.2 Product Functions

1. Complaint resolution
2. Fee management
3. Grant management
4. Expenditure control
5. Staff management

### 2.3 User Classes and Characteristics

The following user classes have been differentiated based on the subset of product features used. Each user class possesses a different interface into the application and are exposed to functions that are relevant to them.

1. *Mess managers*: Need to handle the dues, fees, and complaints relevant to the messes handled by them.
2. *Students*: Pay various fees and dues, register and view complaints.
3. *Hall clerks*: Manage the employment of temporary workers

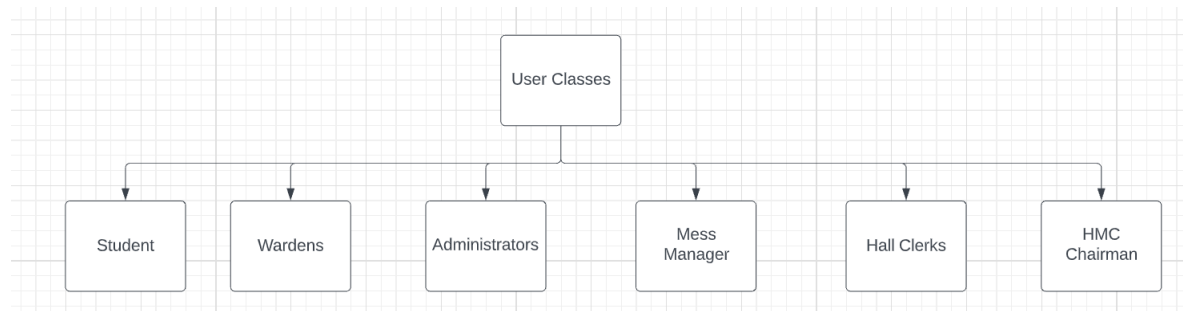


Figure 2.1: User classes

4. *Wardens*: Handle complaints, manage payments to mess managers, handle financial grants from the HMC
5. *HMC chairman*: View various expenditures, allocation of grants to different halls
6. *Administrators*: Manage student records and hall allocation

## 2.4 Operating Environment

The application is designed to use the platform agnostic **Python** interpreter along with frameworks associated with it. The application server that handles incoming requests can be run on any platform that possesses a TCP/IP stack with support for **Django** and **Python**. The web application can therefore be run on **Windows Server** or **Linux**-based distributions which support the above tools. The server should be connected to the local area network provided by the Computer and Informatics Centre in IIT KGP and should be appropriately configured to handle incoming requests. Any relevant firewall settings must be tweaked 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 possess a self-documenting, easy to use interface that does not require a specific set of manuals for any of the users. All the relevant information regarding the usage of the web application will be placed at the appropriate locations inside the different webpages and displayed when needed.

## 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

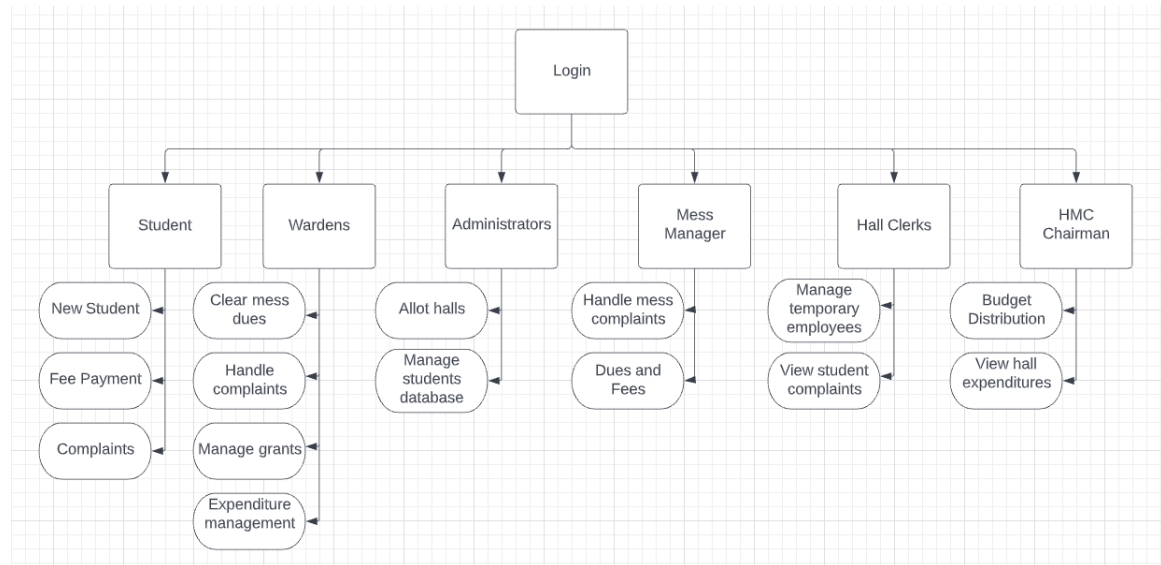


Figure 3.1: User interfaces

### 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. Web browsers needed on the end-users' systems to access the web application



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

### **3.4 Communications Interfaces**

1. Will be using a standard TCP/IP stack available on most operating systems.
2. Will use the **HTTPS** protocol to deliver webpages and handle user login

## 4 System Features

### 4.1 Complaint resolution system

#### 4.1.1 Students

##### Description

- Mess related: A student can file complaints directly with the mess manager regarding food-related issues.
- Hall related: Issues such as broken plumbing, absence of working electrical equipment, malfunctioning washing machines, etc., can be filed.
- Fees related: Any issues related to the payment of dues and fees can also be handled.

##### Stimulus/Response Sequences

- Stimulus: Student wishes to file a new complaint.
- Response: Form is displayed where the student can enter the details of their complaint.
- Stimulus: Student wants to review the action take report submitted by the warden or other authorities.
- Response: System retrieves the report if submitted by the authorities via their portal and delivers it to the student.

##### Functional Requirements

- **REQ1:** Digital forms for complaint registration
- **REQ2:** Tables for displaying complaint history
- **REQ3:** Webpage for displaying action take report

#### 4.1.2 Warden

##### Description

The warden must be able to view all complaints registered by students and file action taken reports against them when necessary.

### **Stimulus/Response Sequences**

- Stimulus: Warden requests complaint history.
- Response: System uses its internal complaint database and returns the list of complaints.
- Stimulus: Warden wishes to file an action taken report.
- Response: Form is displayed where the warden can enter the details and upload any relevant documents.

### **Functional Requirements**

- **REQ4:** Digital forms for the filing of action taken reports
- **REQ5:** Tables for displaying complaint history

Students will be able to view these complaints and the action taken reports filed by the warden. This enables transparency in the resolution process and helps in timely handling of issues.

## **4.2 Hall Management**

### **4.2.1 Hall clerks**

#### **Description**

The hall clerks must be able to add employees and register any petty expenses.

### **Stimulus/Response Sequences**

- Stimulus: Hall clerk wants to add new employee.
- Response: System brings up form for clerk to add the employee, asking for required details.
- Stimulus: Hall clerk wants to add new petty expense of the hall.
- Response: System brings up form for adding the new expense.

### **Functional Requirements**

- **REQ6:** Access of clerk to database of respective hall's employees.

### **4.2.2 Temporary Employees' Salary Management**

#### **Description**

Hall clerks must be able to mark the days when temporary employees don't show up for work, and deduct pay accordingly. Priority is 6/10.

### **Stimulus/Response Sequences**

- Stimulus: Hall clerk wishes to mark a leave of absence when a certain temporary employee doesn't show up.
- Response: A webpage with a form is displayed that lets the clerk enter the relevant details.
- Stimulus: Hall clerk requests the total amount payable to temporary employees.
- Response: A webpage with a form is displayed that lets the clerk enter the relevant details of the employees and returns the due amounts in an orderly fashion.

### **Functional Requirements**

- **REQ7:** Ability to track leaves of absence on a digital calendar.
  - **REQ8:** Calculation of amount payable based on attendance
1. Employee management of temporary employees such as gardeners, etc. and their payments.

## **4.3 Fees and dues management**

### **Description**

Mess managers need to collect money from the students for food. Halls charge rents from their boarders. Apart from this, halls involve various other fees and costs. All these fees need to be recorded on a per-student basis and collected without irregularities.

### **Stimulus/Response Sequences**

- Stimulus: Mess manager wishes to enter a due amount against a certain student
- Response: A webpage with a form is displayed that lets the manager enter the relevant details.
- Stimulus: Student requests the total amount owed with different filters (such as mess, rent, common room fees, etc).
- Response: A webpage with the relevant information is displayed.
- Stimulus: Warden wishes to set the various fees associated with their hall such as rent.
- Response: A form is presented where the relevant information is collected and the internal database is updated to reflect the change.

### **Functional Requirements**

- **REQ9:** Ability to track the dues owed by different students.
- **REQ10:** Portal for mess managers to record the amount owed to them by each student.
- **REQ11:** Ability to set rent, etc, by wardens.

## **4.4 HMC Employee Management**

### **4.4.1 Description**

The employees working in the HMC will be registered with the web application. The payroll associated with the HMC will be handled. Salaries will be computed on a monthly basis and cheques will be issued to the employees.

### **4.4.2 Stimulus/Response Sequences**

- Stimulus: Administrator wishes to register a new staff.
- Response: A form that allows the admin to enter the details is displayed. The data is inserted into the database, and computations for their salary is started.
- Stimulus: Administrator requests the total amount payable to a certain employee
- Response: A form is displayed after which the relevant information is displayed based on the input.

### **Functional Requirements**

- **REQ12:** Ability to track, add and delete registered employees.
- **REQ13:** Ability to view payments owed to different employees

## **4.5 HMC Expenditure Management**

### **4.5.1 Description**

- HMC Chairman receives grant that is to be distributed to the various halls. This needs to be digitised.
- HMC incurs petty expenses such as repair works carried out, news paper and magazine subscriptions, etc. It should be possible to handle these expenses through the application.

### 4.5.2 Stimulus/Response Sequences

- Stimulus: HMC Administrator wishes to enter a new petty expense.
- Response: A form that allows the admin to enter the details is displayed. The data is inserted into the database.
- Stimulus: Warden wishes to enter expenditures against grants.
- Response: A form that allows the warden to enter the details is displayed. The data is inserted into the database and is shown to the HMC chairman when requested.
- Stimulus: Administrator requests the petty expense report for a certain duration.
- Response: A form is displayed after which the relevant information is displayed based on the input after accessing the internal database.

### Functional Requirements

- **REQ13:** Ability to record petty expenses.
- **REQ14:** Ability to view expense reports on a timely basis.
- **REQ15:** Wardens should be able to enter expenditures against grants.

## 4.6 Hall Occupancy Management

### 4.6.1 Description

Wardens need to view which rooms are occupied and who is occupying them. This information needs to be quickly retrieved without having to go through paper records. The application helps them do that digitally.

### 4.6.2 Stimulus/Response Sequences

- Stimulus: Warden requests an occupancy report.
- Response: A form that allows the admin to apply various filters is displayed. Based on the form input, the database is filtered and the relevant information is presented, such as number of rooms occupied.
- Stimulus: Warden wishes to see who is occupying a certain room
- Response: A form is displayed after which the relevant information is displayed based on the input after accessing the internal database.

### Functional Requirements

- **REQ16:** Ability to see who is occupying a certain room
- **REQ17:** Ability to view an occupancy report.

## 4.7 Hall Expense Report

### 4.7.1 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.

### 4.7.2 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.