

---

# Software Requirements Specification

for

## Health++

Version 1.1

Prepared by

**Group: 5**

**Group Name: StarBugs**

Name	Roll no.	E-Mail
Anuj	210166	<a href="mailto:anuj21@iitk.ac.in">anuj21@iitk.ac.in</a>
Pulak Gautam	210791	<a href="mailto:pulakg21@iitk.ac.in">pulakg21@iitk.ac.in</a>
Dev Gupta	210318	<a href="mailto:devgupta21@iitk.ac.in">devgupta21@iitk.ac.in</a>
Abhimanyu Sethia	190023	<a href="mailto:sethia@iitk.ac.in">sethia@iitk.ac.in</a>
Deepanshu	210310	<a href="mailto:deepanshu21@iitk.ac.in">deepanshu21@iitk.ac.in</a>
A. Atulya Sundaram	210001	<a href="mailto:aatulya21@iitk.ac.in">aatulya21@iitk.ac.in</a>
Kushagra Bhatnagar	210550	<a href="mailto:kushagrab21@iitk.ac.in">kushagrab21@iitk.ac.in</a>
Pinkesh Mahawar	210719	<a href="mailto:pinkeshm21@iitk.ac.in">pinkeshm21@iitk.ac.in</a>
Pradeep Chahal	210733	<a href="mailto:pradeepc21@iitk.ac.in">pradeepc21@iitk.ac.in</a>
Vaishnavi	211140	<a href="mailto:vaishnavi21@iitk.ac.in">vaishnavi21@iitk.ac.in</a>

**Course:** CS253

**Mentor TA:** Anuj Shrivastava ([anujsh@cse.iitk.ac.in](mailto:anujsh@cse.iitk.ac.in))

**Date:** 10 February 2023

<b>CONTENTS .....</b>	<b>II</b>
<b>REVISIONS .....</b>	<b>II</b>
<b>1 INTRODUCTION .....</b>	<b>1</b>
1.1 PRODUCT SCOPE .....	1
1.2 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	1
1.4 DOCUMENT CONVENTIONS .....	1
1.5 REFERENCES AND ACKNOWLEDGMENTS.....	2
<b>2 OVERALL DESCRIPTION .....</b>	<b>2</b>
2.1 PRODUCT OVERVIEW .....	2
2.2 PRODUCT FUNCTIONALITY .....	3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS .....	3
2.4 ASSUMPTIONS AND DEPENDENCIES .....	3
<b>3 SPECIFIC REQUIREMENTS .....</b>	<b>4</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS .....	4
3.2 FUNCTIONAL REQUIREMENTS .....	4
3.3 USE CASE MODEL.....	5
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>6</b>
4.1 PERFORMANCE REQUIREMENTS.....	6
4.2 SAFETY AND SECURITY REQUIREMENTS.....	6
4.3 SOFTWARE QUALITY ATTRIBUTES .....	6
<b>5 OTHER REQUIREMENTS .....</b>	<b>7</b>
<b>APPENDIX A – DATA DICTIONARY .....</b>	<b>8</b>
<b>APPENDIX B - GROUP LOG .....</b>	<b>9</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.1	All Team Members	Made all the diagrams and completed all the sections that are there in the document	10/02/23

# 1 Introduction

## 1.1 Product Scope

Health++ is a web application to make various hospital management processes efficient and effective. The product is designed in particular for the Health Centre in the campus, but can be generally extended with some modifications to any hospital.

The current system of operations at the Health Centre is painstaking and inefficient for all stakeholders, including campus residents, doctors and the staff at the HC. Our software aims to automate the various processes including: requesting for appointments, requesting for urgent blood requirement, calling for/tracking ambulance, applying for medical leaves, generating prescriptions and maintaining digital health record.

This automated system has a plethora of benefits:

1. For the students, HC related processes would become faster and much more accessible – the software will eliminate the need for physical presence for booking appointment or processing medical leave applications.
2. The software will completely eliminate the need for a physical health booklet, as authentication as well as maintenance of health records and prescriptions will be digitized by the application.
3. The application will allow the patient to directly book an ambulance and relay it immediately to an active ambulance driver. This will reduce the delay in rendering emergency services and efficiently utilize ambulance infrastructure.
4. Since all appointments and medical records are stored on a digitized/ computer-readable database, the HC will be able to identify any spikes thereby, identifying community transmission of flues and other communicable diseases well in time.
5. The process of approving or rejecting medical leaves for the administration/SUGC would be streamlined. Currently, they have to deal with a large number of emails and manually check each medical proof. Instead, our automated leave management system would directly attach authenticated medical records and provide SUGC with an interface to approve or reject leaves.

## 1.2 Intended Audience and Document Overview

This document is intended for:-

1. Developers:
  - a. to inculcate the desired system requirements in the implementation
  - b. to understand the scale, security and resilience requirements of the system
  - c. and accordingly, make technical choices for databases and architectures to be used
2. Project Managers
  - a. to understand the functional requirements of the system even without having the domain experience
  - b. to understand the working constraints of the software
  - c. to comprehend how various components of the software are interconnected
  - d. to determine if the project under development aligns with the client's desired product
  - e. to help control and assess the flow of the project development
3. Users
  - a. to comprehend the product's interface
  - b. to understand the software's functionalities and constraints
  - c. to understand if the system provides the expected requirements

The document must ideally be read in order, as it first describes the overview of the application, then dives deeper into the functionality and use cases and ultimately describes the non-functional requirements of the application.

For the users, section 2 (Overall Description) and Section 3 (Specific Requirements) are most relevant. They may entirely skip Section 4 (Non-functional Requirement) as that is primarily relevant for the developers.

### 1.3 Definitions, Acronyms and Abbreviations

- CSS: Cascading Style Sheets
- CC: Computer Centre
- HC: Health Centre
- IITK: Indian Institute of Technology Kanpur
- SUGC: Senate Undergraduate Committee (used to refer to the SUGC Convenor in common parlance)

### 1.4 Document Conventions

This document follows the IEEE formatting requirements. Graphical representation follows UML formatting.

Typographical conventions-

- *Italic* text refers to comments
- **Bold** text refers to abbreviations and acronyms

Naming Conventions:

- Use case models are named as U#<index>
- Functional Requirements are named as F#<index>

### 1.5 References and Acknowledgments

- Software Engineering, 6th Edition by Ian Sommerville
- [IEEE - Manuscript Templates](#) on IEEE's official webpage
- [Unified Modeling Language](#)

## 2 Overall Description

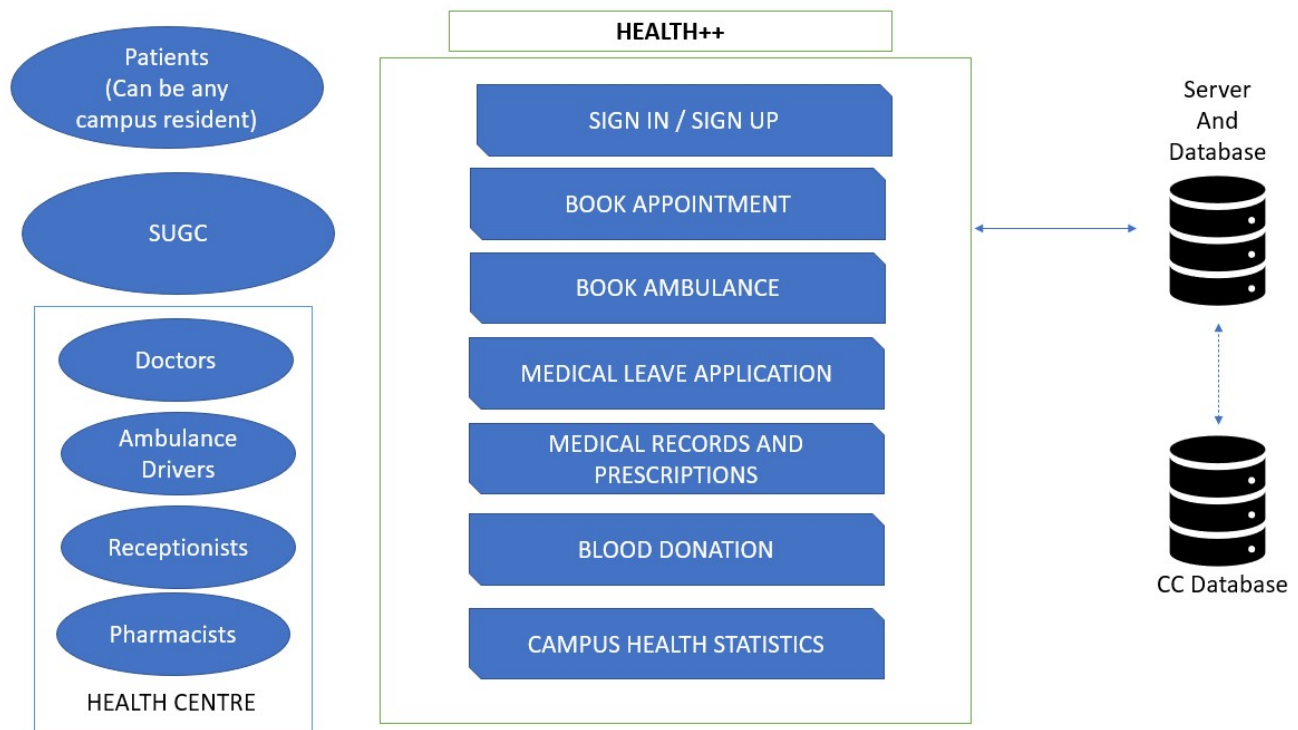
### 2.1 Product Overview

Health++ aims to improve the current HC management system. The current system for the most part is physical or uses primitive online tools.

- It requires every patient to have a health booklet. Doctors write prescriptions in the Health Booklet, which is then shown to the pharmacy store for obtaining the medicines. This will be completely substituted by our application.

- For ambulance booking, a patient needs to call the HC reception – which then calls the Ambulance driver and gives them instructions. This process will again be substituted by our application, which will allow patients to directly book ambulance on the app and the allotted ambulance driver (allotted by the app itself) to get information about their next pick-up. Moreover, the patient may still book an ambulance via calling the reception – in which case, the receptionist can use our app to book the ambulance on their behalf.
- Currently, leave applications system works on emails. The student is supposed to send an email to the SUGC, with an attached scan of the physical Health booklet record. The SUGC is then supposed to evaluate the emails and forward their approval/rejections to the DoAA staff. This process will be integrated on our application, where the students can directly apply for leaves – by attaching their digital health record – and the SUGC can (from their sign-in) approve or reject the request.

The application is self-contained, does not necessarily need any integrations. However, an optional integration with the CC database (for using CC logins for authentication) could be helpful.



## 2.2 Product Functionality

The product will have the following core functionalities:

1. User Login (Sign Up/Sign In)
2. Online Appointment Request and Allotment System – patients can view and book available slots; doctors can update their availability; staff and doctors can view the appointments schedule
3. Online Medical Records/Prescription – the doctors can add or view past medical entries and prescriptions; pharmacy staff can view prescription
4. Ambulance Booking – patients or staff can book a new ambulance; ambulance driver will be auto-allotted and given instructions for next pick up

5. Medical Leave Application – students can generate new leave application, directly attach medical records with the application and view status of their application; SUGC can accept or reject leave application
6. Blood Donation Facilities – any user can raise a request for blood for a given blood-group; notifications will be broadcasted across
7. Campus Health Statistics – System admin can see macro statistics on the number of appointments sought, number of prescriptions given etc. and thereby, gauge any spikes quickly

The actors in our software scheme would be (each of these will have their own logins):

1. Student / campus resident / patient
2. HC Receptionist/Manager
3. HC Staff (Nurses/Pharmacy staff)
4. Doctor
5. Ambulance Driver
6. SUGC Convenor/Institute Administration

## 2.3 Design and Implementation Constraints

- Database & Memory: The system is supposed to cater to all campus residents. Hence, the potential number of users could be as much as 15,000. The database must be able to store and efficiently query the medical records of such a large number of users.
- Compatibility: Our web-app must work on the recent versions of all major web browsers, across both Windows and Linux systems.
- The initial setup as well as training of staff would need to be done by our engineers. Once all debugging is done, a System Admin may be appointed at the HC, responsible for updating the roles, adding or deleting accounts (in cases when staff joins or leaves,
- Timing: Certain features such as ambulance booking and blood donation alerts need to work without any significant lag across users, as these are supposed to be used in emergencies. For example, the ambulance driver must receive the update for a new pickup as soon as the ambulance is requested on the app by the patient.
- Security Constraints: Since the app would have a lot of personal data of the users, we must avoid using third-party APIs.
- Language Constraints: For now, the software will use only English language.

## 2.4 Assumptions and Dependencies

- Internet Connectivity: We assume that all users of the software have access to stable internet. This is a fair assumption as IITK gives all its residents free-wifi across the campus.
- Server Availability: We assume that reliable servers will be available round-the-clock to process all requests efficiently and handle the expected traffic without compromising on performance.
- Framework dependencies: The software development will involve utilizing pre-existing frameworks and libraries, such as Bootstrap-CSS for styling and ReactJS/AngularJS for front-end development, as per the design needs.
- Hardware Availability: We assume that the HC staff, doctors and ambulance drivers will have one device each (computer/laptop/mobile phone) to interact with the application.

## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

The user's interaction will occur majorly through the following interfaces:

1. Sign-Up Page:

This is the page where a new user will register. It will consist of a form asking for the following data fields at registration:

- (a) Full Name
- (b) Role (Student, Faculty, etc.)
- (c) Email ID / CC ID
- (d) Roll Number/ID
- (e) Blood Group
- (f) Date of Birth
- (g) Password
- (h) Confirm Password

After signing up, the user will be redirected to the login page. If an integration with the CC database is enabled, then we can simply use the CC ID and password for authentication.

2. Login Page

The login page will ask the user to enter their Email ID and password. On successful validation of the credentials, the user will be redirected to the homepage, where they will have access to the various features, depending on their role.

3. Appointments Module

- a. Doctor: can change the availability for a particular time and date; will be able to see the booked appointments for any day
- b. Staff: can view the booked appointments for any day
- c. Patients: can see the available slots for any doctor on any day; book among the available slots; view booked appointments
- d. Receptionist: can book new appointments for a given ID with a preferred doctor; can view booked appointments for any day

4. Blood Donation Facilities:

- a. Any user: can post a request for blood with prompts (blood group, required volume, comments/remarks if necessary)
- b. Doctor/Receptionist: blood donation requests posted by the doctor and receptionists can be optionally sent to all users as an urgent notification
- c. Any user: can respond to the request by consenting to donate blood

5. Ambulance Booking:

- a. Receptionist/ Patient: can book an ambulance with the prompts for location and contact number
- b. Ambulance Driver: will be able to view current pick-up patient allotted along with their location and contact number; are also supposed to update on the app when they have picked up and dropped off the patient
- c. Receptionist: will be able to view the number of patients the ambulance is picking up (so that they can prepare accordingly for emergencies)

6. Health booklet/Prescription Module:

- a. Doctor: Can add a new entry; can view past entries
- b. Patient, Staff, Receptionist: Can view past entries

7. Leave Application Module:

- a. Patient: Can generate a leave application (attaching relevant medical record entry), can view status of the leave application

- b. SUGC: can approve or reject pending leave applications

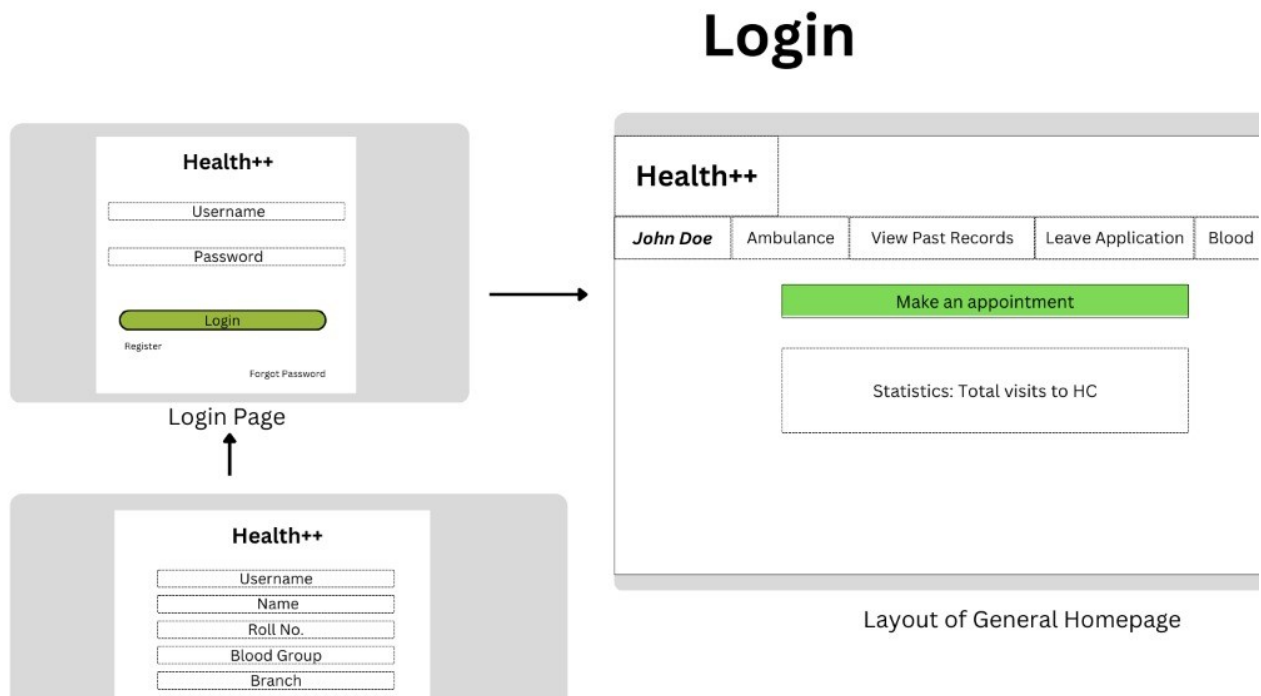
### 3.1.2 Hardware Interfaces

List of Hardware Devices we must interact with are:

1. Health Centre Server: For storing the user database and medical records.
2. Receptionist's Computer
3. Doctor's Computer or Phone
4. Patient's Computer or Phone
5. Ambulance Driver's Phone
6. Pharmacist's Computer

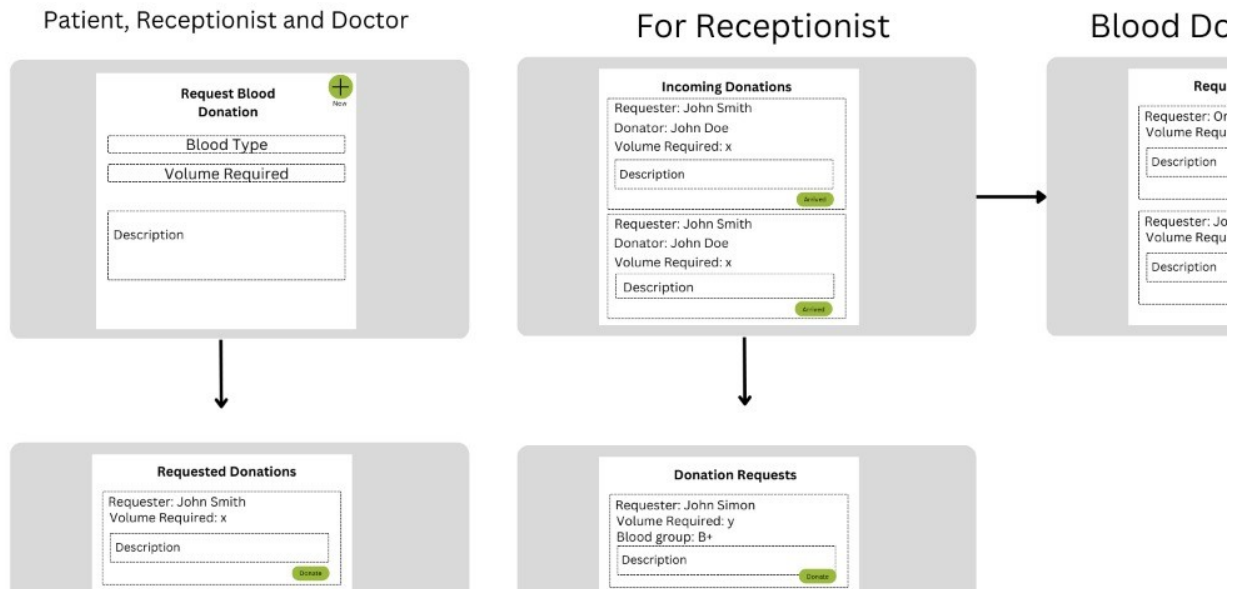
Since we are developing a web-app, just the presence of a browser on the hardware with internet connectivity will work well.

### 3.1.3 Software Interfaces

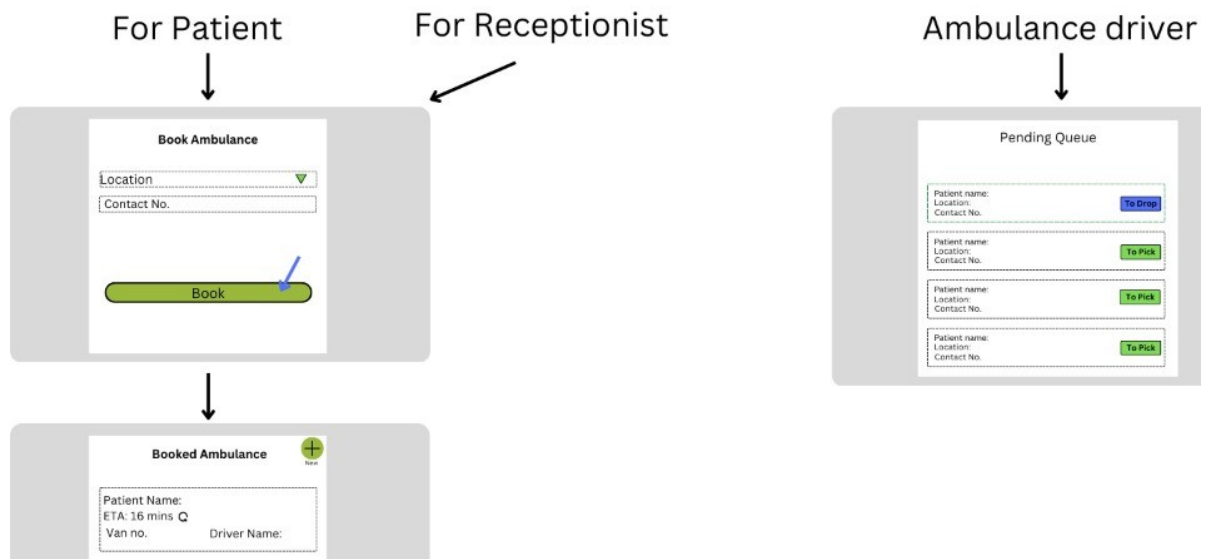




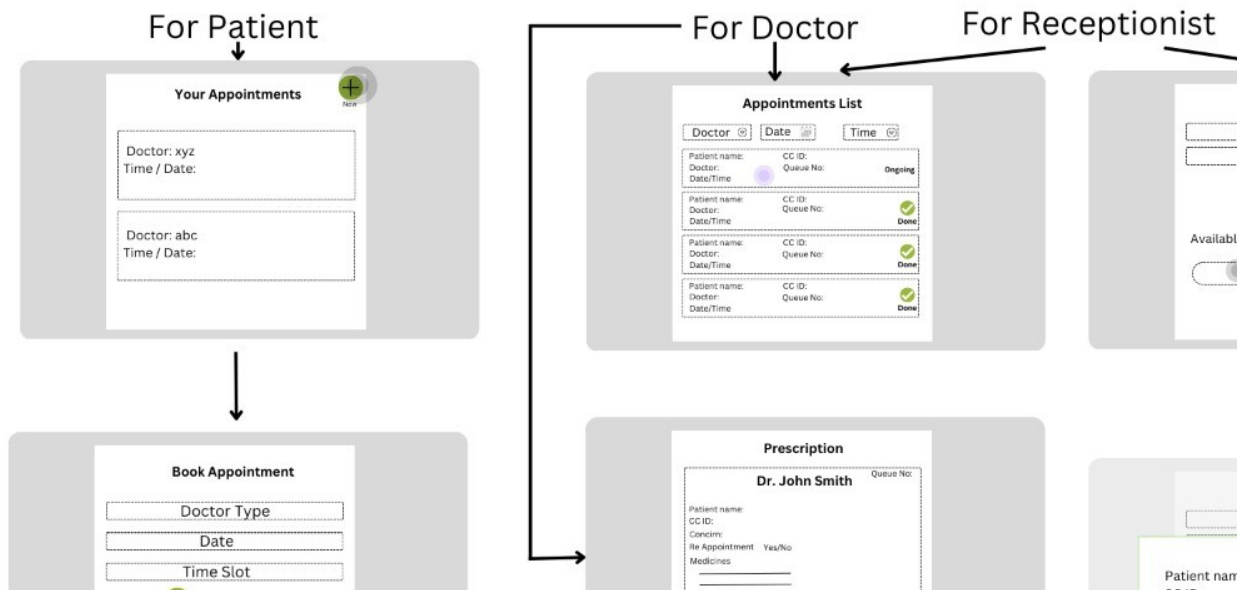
## Blood Donation



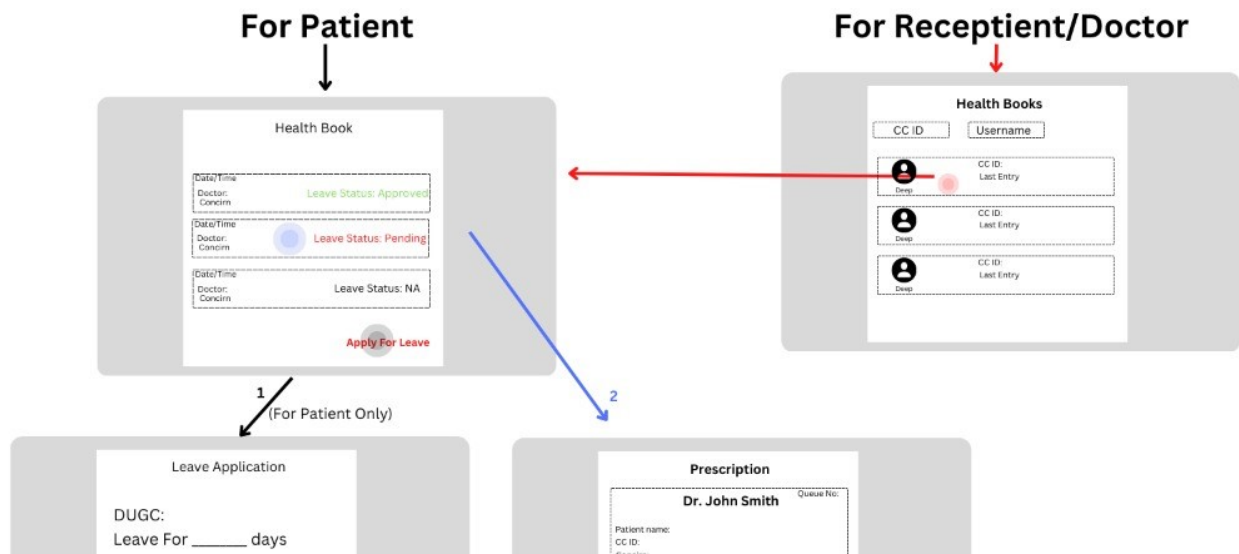
## Ambulance Booking



# Booking Appointment



# Health Booklet/Leave Application



## 3.2 Functional Requirements

### 3.2.1 F1: User Sign In/Sign Up

- The signup form will consist of personal details which are name, date of birth, blood group, phone number along with a password and e-mail ID to be used for logging in.
- All users of this software can login and access the facilities available according to their role.

### 3.2.2 F2: Appointment Booking

- The software will act as an interface for all the patients to pre-book appointments for doctors.
- Doctors can edit their availability status in case they are on leave or unavailable on certain hours.
- Patients can choose the doctor they want to consult. Thereby, they will be shown the available time slots of all doctors and can accordingly choose a slot and book it. The bookings will be made on a first-come-first-serve basis. Patients can then view their scheduled appointments.
- Receptionists can assist patients in booking appointments and checking doctors' schedules, benefiting those who have difficulty using the platform or lack digital literacy.
- Receptionist, staff and doctors can view their appointment schedule to be prepared for the day or plan other logistics.

### 3.2.3 F3: Digital Medical Records/Prescription

- Doctor can create new or view old medical records and/or prescriptions of the respective patient
- Any medical record entry will consist of:
  - (a) Patient's vitals like Temperature, BP, oxygen levels if checked.
  - (b) Medicines/tests prescribed
  - (c) Ailments diagnosed with
  - (d) Any comments/remarks of the doctor

### 3.2.4 F4: Ambulance Booking

- Patients can book ambulance by giving their address and contact number.
- The application will automatically designate the next available driver to the patients
- Ambulance drivers will be provided with details of their next pickup. They are also supposed to update when they have picked up and dropped off a patient.
- The receptionist can also book an ambulance on behalf of the patient, in case the patient wants to book ambulance by calling up.
- The receptionist can view all running pick-ups and thereby prepare in advance for incoming emergency patients.

### 3.2.5 F5: Medical Leave Application

- Students can directly apply for medical-leave applications on this software with a doctor's medical record entry and track the application's status.
- The relevant authority (such as **SUGC**) will be able to view the list of all pending leave applications and approve or deny them based on the doctors' comments.
- If the leave is approved, the patient will receive an application reference ID which the student can e-mail to the concerned professors for make-up lab/quiz requests.

### 3.2.6 F6: Blood Donation Facility

- There will be a blood donation forum where users will be able to post requests for the blood group they require.
- When a request is made, the software will search the database for users with matching blood types and send them an urgent notification.
- Organisations that conduct blood donation camps can also use this software by posting the camp's date and venue information, so that willing donors can easily check out all ongoing camps.

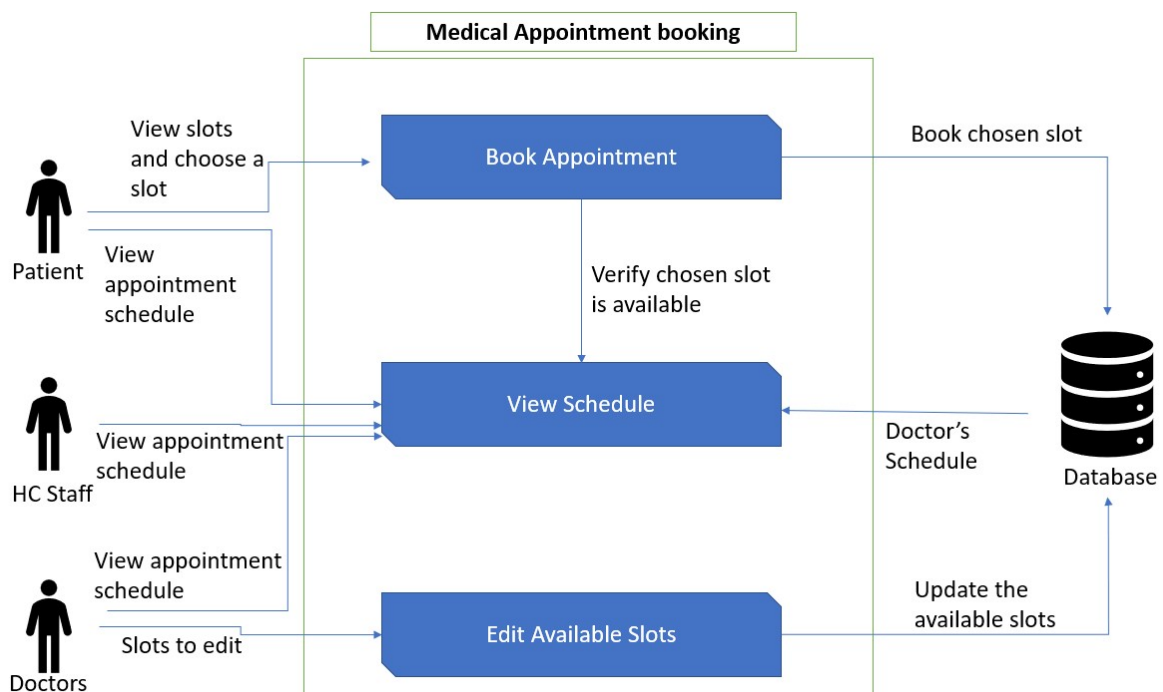
### 3.2.7 F7: Campus Health Statistics

- The software will provide information about the current spread of illnesses among the campus community using data visualizations.
- Some of the possible stats that can be shared are:
  - (a) Plots of number of patients vs time.
  - (b) Number of cases of major diseases like COVID, dengue, malaria etc.
  - (c) Illnesses can be classified into broader classes like viral infections, physical injuries, chronic diseases etc. and the number of cases for each class of diseases vs time can be plotted to bring out more relevant trends.
- Medical researchers can use the software for their research and analysis, however, personal data of any individual patient will not be made publicly available

## 3.3 Use Case Model

TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.

### 3.3.1 U#1: Appointment Booking



**Author** – Abhimanyu, Anuj

**Purpose** – Demonstrate how a user will book an appointment

**Requirements Traceability** – F2

**Priority** - High

**Preconditions** – user is logged in

**Post conditions** – Redirected to the homepage

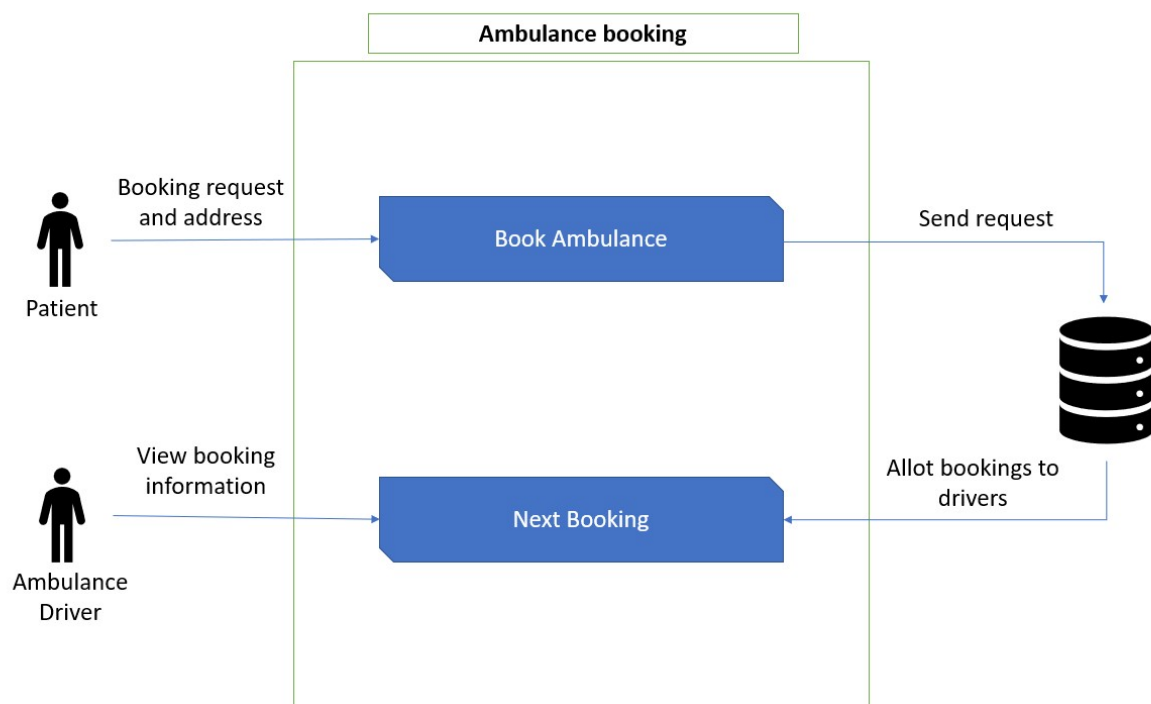
**Actors** – Patient, Doctor, HC Staff, Appointments Database

**Exceptions** - None

**Includes** (other use case IDs)

**Notes/Issues** -

### 3.3.2 U#2: Ambulance Booking



**Author** – Abhimanyu, Anuj

**Purpose** – Demonstrate how a user will book an ambulance

**Requirements Traceability** – F3

**Priority** - High

**Preconditions** – User is logged in

**Post conditions** – Redirected to the ambulance driver allotment details

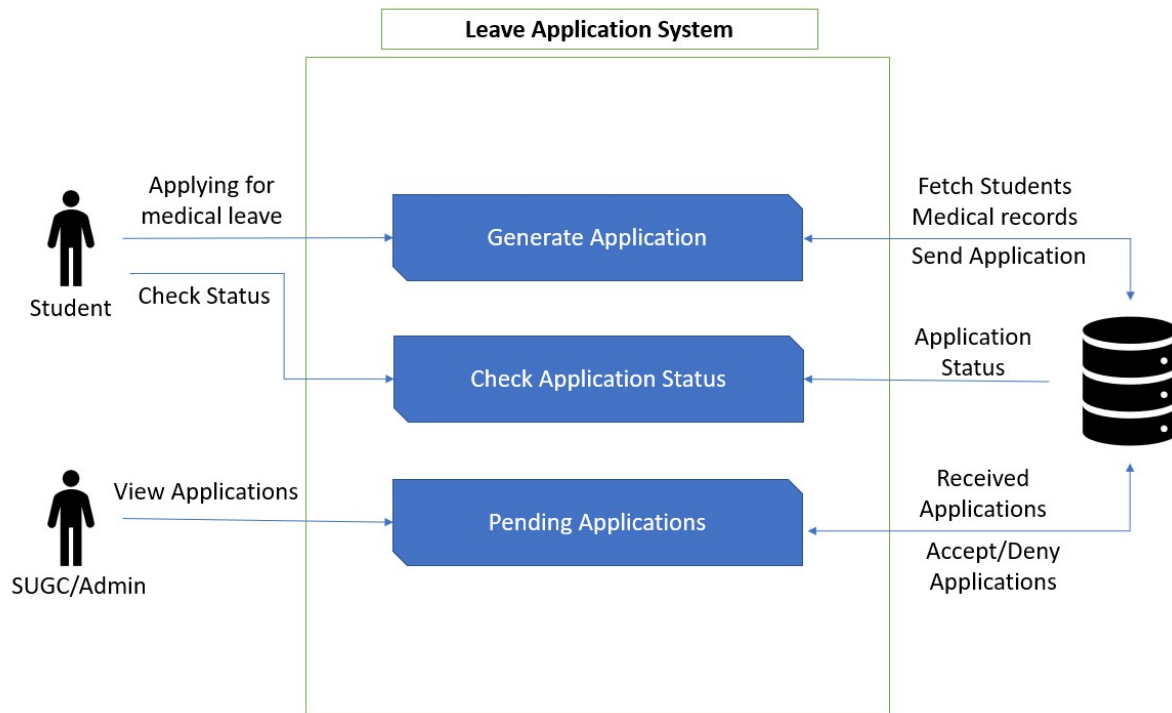
**Actors** – Patient, Ambulance Driver

**Exceptions** – No ambulance driver available

**Includes** (other use case IDs)

**Notes/Issues** -

### **3.3.3 U#3: Leave Application System**



**Author** – Abhimanyu, Anuj

**Purpose** – Demonstrate how a user will generate a leave application

**Requirements Traceability** – F4

**Priority** - Low

**Preconditions** – User is logged in

**Post conditions** – Redirected to leave application status page

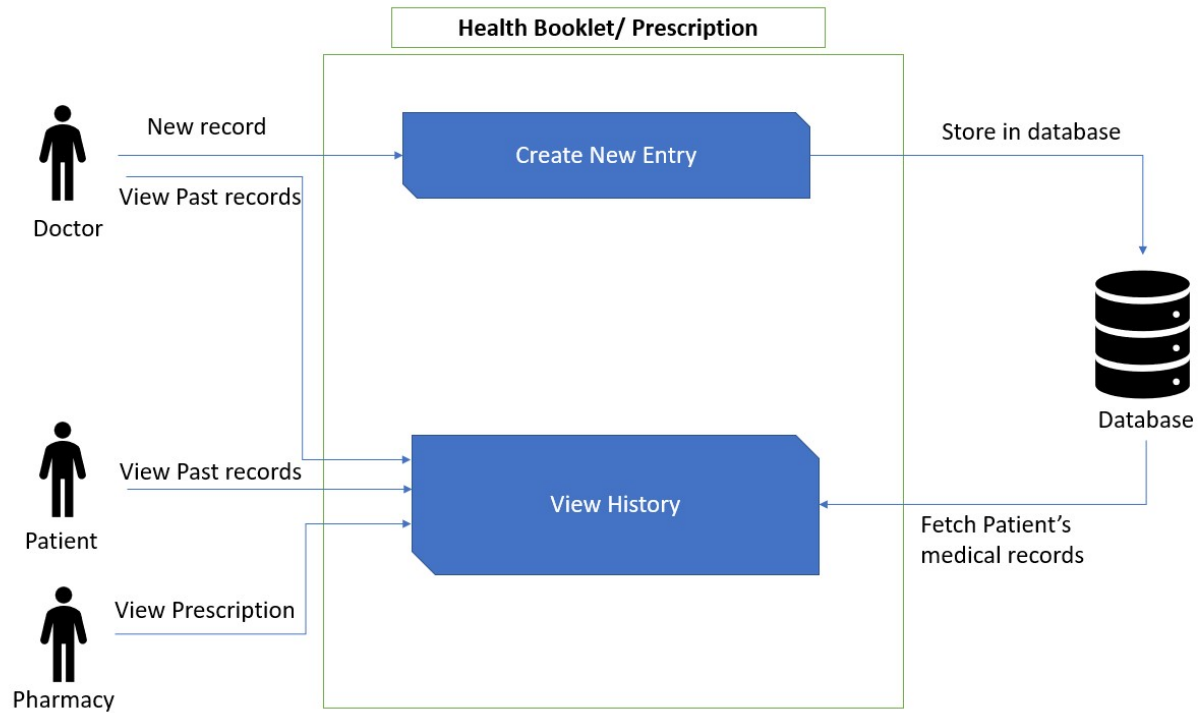
**Actors** – Student, SUGC

**Exceptions** – No medical record found

**Includes** (other use case IDs)

**Notes/Issues** -

### 3.3.4 U#4: Medical Record/Prescription



**Author** – Abhimanyu, Anuj

**Purpose** – Demonstrate how the application will digitize health booklets

**Requirements Traceability** – F5

**Priority** - Moderate

**Preconditions** – User is logged in

**Post conditions** – Redirected to homepage

**Actors** – Student, Doctor, Pharmacy Staff

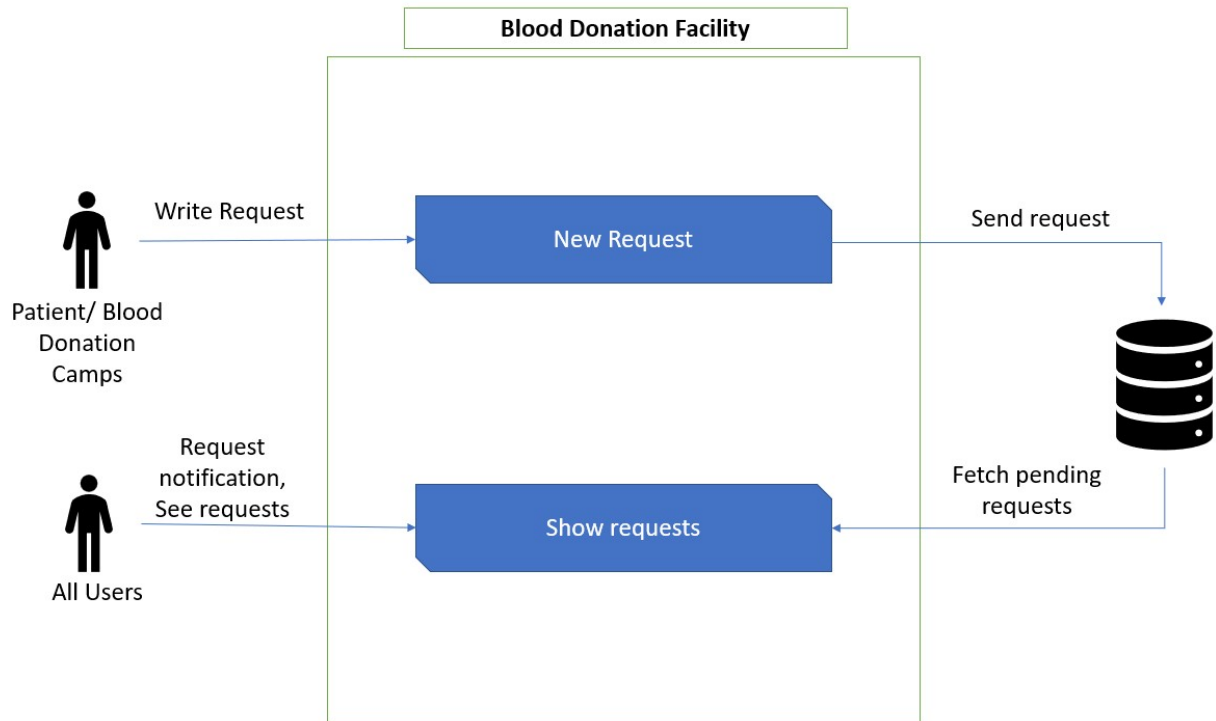
**Exceptions** – None

**Includes** (other use case IDs)

**Notes/Issues** -

### 3.3.5 U#5: Blood Donation Facility





**Author** – Abhimanyu, Anuj

**Purpose** – Demonstrate how a user can request for blood

**Requirements Traceability** – F6

**Priority** - High

**Preconditions** – User is logged in

**Post conditions** – Redirected to homepage

**Actors** – All users

**Exceptions** – none

**Includes** (other use case IDs)

**Notes/Issues** -

## **4 Other Non-functional Requirements**

### **4.1 Performance Requirements**

- Capacity: The software is supposed to cater to the entire campus community. Hence, it must be able to handle a load of about 15k users.
- Timing is crucial, especially for emergency services like ambulance booking and blood donation module. A blood request must be communicated to all users of that blood group without any notable delay.
- Similarly, any request for ambulance must be relayed to the available ambulance driver without any notable delay

### **4.2 Safety and Security Requirements**

- Availability:  
The programme should be accessible round-the-clock to be prepared for crises.  
The software should be able to display which doctors are available right now.
- Maintainability:  
The software needs to be able to back up the data and information of users and staff also.  
The software should record all errors.
- Usability:  
Users should be able to learn how to operate the interface quickly.
- Reliability:  
The software must function continuously for a long time without fail.  
The software needs to obtain all relevant information in case of an error.

### **4.3 Software Quality Attributes**

- Availability:  
The programme should be accessible round-the-clock to be prepared for crises.  
The software should be able to display which doctors are available right now.
- Maintainability:  
The software needs to be able to back up the data and information of users and staff also.  
The software should record all errors.
- Usability:  
Users should be able to learn how to operate the interface quickly.
- Reliability:  
The software must function continuously for a long time without fail.  
The software needs to obtain all relevant information in case of an error.

## Appendix A – Data Dictionary

Class	Field	Datatype	Description	Use
<b>LOGIN_DATA</b>	Name	String	Name of the user	Sign in/ Sign up
	Roll Number	Integer	For students	Sign in/ Sign up
	CC ID	String	For students and faculties	Sign in/ Sign up
	Staff_ID	String	For HC Staff	Sign in/ Sign up
	Phone Number	String	Contact details of patient	Sign in/ Sign up
	Address	String	Residential Address of user	Sign in/ Sign up
	Blood Group	String	Blood Group	Sign in/ Sign up
	Role	String	The role assigned to the user	Sign in/ Sign up
<b>APPOINTMENT_DATA</b>	Doctor Type	String	The speciality or field of study of doctor	Book Appointment
	Date	Date	Date of Appointment Booking	Book Appointment
	Time Slot	Time Interval	Time slot to book appointment	Book Appointment
	Availability	Boolean	availability of a doctor in a time slot	Book Appointment
<b>PRESCRIPTION_DATA</b>	Doctor Name	String	Name of doctor who gave the	Prescription

			prescription	
	Ailments	String	description of diseases being diagnosed	Prescription
	Re_Appointment	Boolean	whether a reappointment is prescribed	Prescription
	Medicines	StringList	list of prescribed medicines	Prescription
	Vitals_Info	String	details of patients temperature, blood pressure etc.	Prescription
<b>AMBULANCE_DATA</b>	Location	String	Location of patient	Ambulance booking
	ETA	Time	Expected time of arrival of ambulance	Ambulance booking
	Driver_Name	String	Name of Ambulance Driver	Ambulance booking
	Driver_Contact	String	Phone number of Ambulance driver	Ambulance booking
	Ambulance_No	Integer	Vehicle No. of the Ambulance	Ambulance booking
<b>LEAVE_DATA</b>	Date	Date	date for which leave is to be applied	Medical Leave Application
	Status	Boolean	Approval status of leave	Medical Leave Application
	Description	String	Doctor's comments for the leave	Medical Leave Application
<b>BLOOD_DONA</b>	Requester	String	Name of	Blood

<b>TION</b>			requester	Donation Facility
	Volume_RequireDe scription	Integer	Volume of blood required	Blood Donation Facility
	Blood Type	String	Requested Blood group	Blood Donation Facility
	Requester_comme nt	String	comments by requester for making the request	Blood Donation Facility
<b>STATS_DATA</b>	Cases_Count	Integer	number of consultations given on a day	Health Statistics
	Date	Date	Date for which statistics is being reported	Health Statistics
	Illnesses	StringList	List of all diseases observed in the campus	Health Statistics

## Appendix B - Group Log

- 14th January- First group meeting to decide on broadly what our product will do and what basic features it will have. Divided work.
- 21st January- Second Group meeting with mentor TA to address queries and seek suggestions.
- 22nd January- Third Group meeting to discuss suggestions by TA, decide on exact functionalities of the software
- 23rd January- Fourth Group Meeting to redivide work among the teammates:
  1. Section 1.3,1.4,1.5 - To be done at the end
  2. Section 2-> Pradeep, Pulak, and Kushagra
  3. Section 3->Deepanshu, Atulya, Abhimanyu, Anuj,
  4. Section 4-> Vaishnavi, Pinkesh, Dev
- 25th January- First draft was completed. Necessary changes were discussed. Some flowcharts and diagrams had to be redone.
- 26th January – finalized leftover sections (mainly Section 1)
- 27th January- Editing was done to remove errors and inconsistencies. This was followed by proofreading.