Software Requirements Specification

for

NITC Health Centre Application

Version 1.0

Prepared by

Sheenam Waris
MXXXXXCA

Project Manager: Dr. XXXXXXX

Course: CS3099D Project

Date: March , 2023

This template is based on the one available from GMU site by Dr. Rob Pettit. Modifications specific to NITC are made and will be used for academic purpose only.

Mentor's Sign:

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	SHEENAM WARIS	System Overview diagram use cases diagram Functional Requirements System User interface Non – Functional Requirements Writing document Safety and Security Requirements	17/03/23
1.1	SHEENAM WARIS	Technology Stack Database design tables Class Diagram	27/03/23

1 Introduction

This project aims to develop an automated Health Management System for health centre of NIT Calicut. The entire process of health centre done manually by providing each student a health card and these cards stored in drawers batch wise which is handled manually can be automated through this application with an easy to understand and simple user interface. Current process is paper based and automating the process will improve the user experience.

1.1 Document Purpose

The purpose of this document is to help people to understand the detailed description of the functional requirements proposed by the system to maintain the objectives. A short discussion accompanies each requirement, to add the background and framework necessary to explain the functionality. It also describes non-functional requirements and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

1.2 Product Scope

This software system will be an online health care portal for NITC wishing to manage their patient records and their appointment needs online. More specifically to design and develop a simple and intuitive system which shall cater the health centre needs of NITC. The system shall provide features to the patient to book appointment and request for medical certificate. Currently this application is for NITC community and they need NITC mail id to register.

1.3 Intended Audience and Document Overview

Primarily this project is being developed for the students, faculties and staff member of NITC, that is basically those who have unique ID given by NITC. With this Patient can check their visits and request for appointment or medical certificate online.

This is a working document and, as such, is subject to change. In its initial form, it is incomplete by definition, and will require continuing refinement. Requirements may be modified and additional requirements may be added as development progresses and the system description becomes more refined. This information will serve as a framework for the current definition and future evolution of the NITC Health Centre Portal.

1.4 Definitions, Acronyms and Abbreviations

DB:	Database
GUI:	Graphical User Interface
IEEE:	Institute of Electrical and Electronics Engineering
ID:	Identity
MC:	Medical Certificate
NITC:	National Institute of Technology, Calicut
SRS:	Software Requirement Specification

UML:	Unified Modelling Language
UI:	User Interface
VS-code:	Visual Studio Code

1.5 Document Conventions

this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text are single spaced and maintain the 1" margins found in this template.

1.6 References and Acknowledgments

- R.S Pressman, Software Engineering: A Practitioner's Approach, Mc-Graw-Hill, Edition-7 (2010).
- https://app.diagrams.net/
- https://uizard.io/
- https://mocdoc.in/blog/the-functional-and-nonfunctional-requirement-for-hms

2 Overall Description

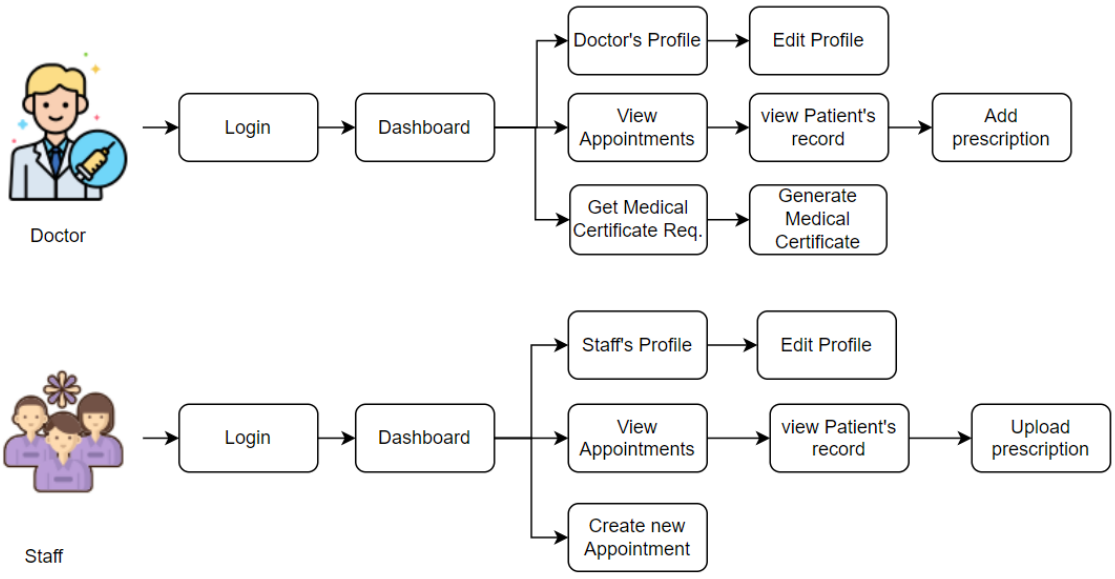
2.1 Product Overview

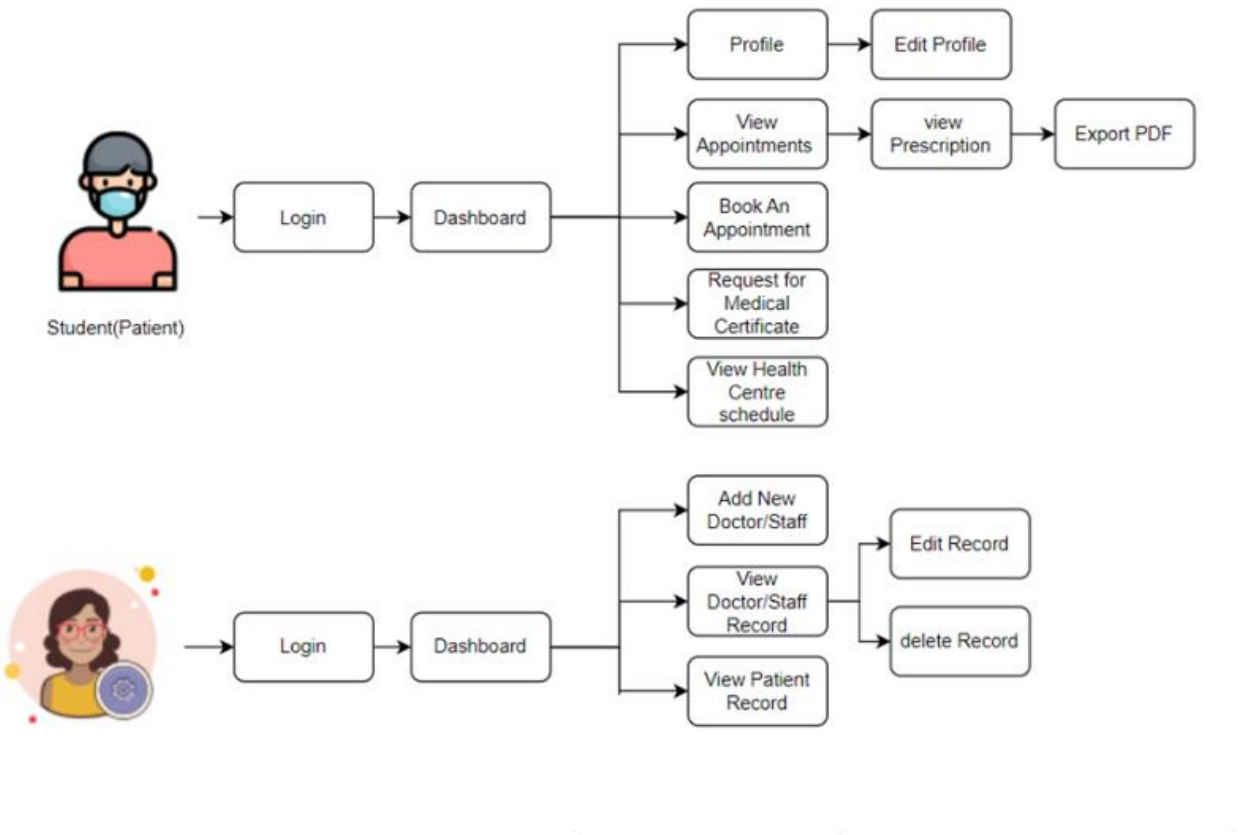
NITC Health centre application is meant to serve as a common platform where management of everyday staff and patients' tasks can be carried out conveniently. Our goal is to develop a replacement to the manual health card used at health centre making it more users friendly and to provide hassle-free facility among the users.

This Application has two modules, one module has three users of the application staff, Doctor and patients(students) and other has Admin module. First Admin will register Doctors, staff and students to the application and send their login to credentials to its users. A student login to the application then it requests for appointment by selecting what type of problem he/she is facing, then it will be approved by staff. Student can check his appointment history and also request for medical certificate. Doctor is able to check its list of appointments and generate medical certificate. The admin is able to check the list of all appointments of all the doctors available. The doctor can also give prescriptions online or staff will upload an image of prescription online.

System Overview

Users (Doctor/Staff/Student)





2.2 Product Functionality

This application will contain two modules – the admin module and the user module.

ADMIN -

- Login to the account
- Create, update and delete account of doctor
- Create, update and delete account of staff
- View all previous Appointments

USERS -

1. **Doctor Functionality**
   - Login to the account
   - Medical history of patient
   - Generate Medical Certificate
   - View Today's Appointments

2. **Staff Functionality**
   - Login to the account
   - Medical history of patient
   - Upload Prescription's image
   - Approve Appointment Request

- Create New Appointment
- View Today's Appointments

3. **Student Functionality**
   - Login to the account
   - Edit Profile
   - New Appointment request
   - Previous Appointment history
   - Request Medical Certificate

2.3 Design and Implementation Constraints

- This application will be a mobile application using Nodejs (JavaScript) as the backend language.
- To maintain the hierarchy of projects we will be using GIT version control to properly and concurrently develop the application and use all the helpful features of GIT like staging area, commit it, push etc. This will help in continuous and smooth development for the project.
- A fast and reliable database with good storage is needed which will store the user details. This project will be using SQL for the database. It handles large volumes of data at high speed with a scale-out architecture.
- This application also needs a good server for hosting as the application is used by many students and faculties at the same time.
- One of the major points of this mobile application is that it is user friendly and easily accessible.

2.4 Assumptions and Dependencies

Most of the portals have a lot of redundant features which are rarely used in institutes. Our system focuses on the features which are most important to the users of an academic institute along with introduction of some new features.
We have made the following assumptions:

- The intended user and doctor of the application must be a part of NITC, that is they should have a unique ID given by NITC.
- Admin will deal with the reports that will be given by the users if someone is not behaving appropriately with them.
- It requires proper internet connection to request an appointment.
- The admin has to update the schedule of the doctor when he/she will be on leave and also update the schedule when there is any change required.

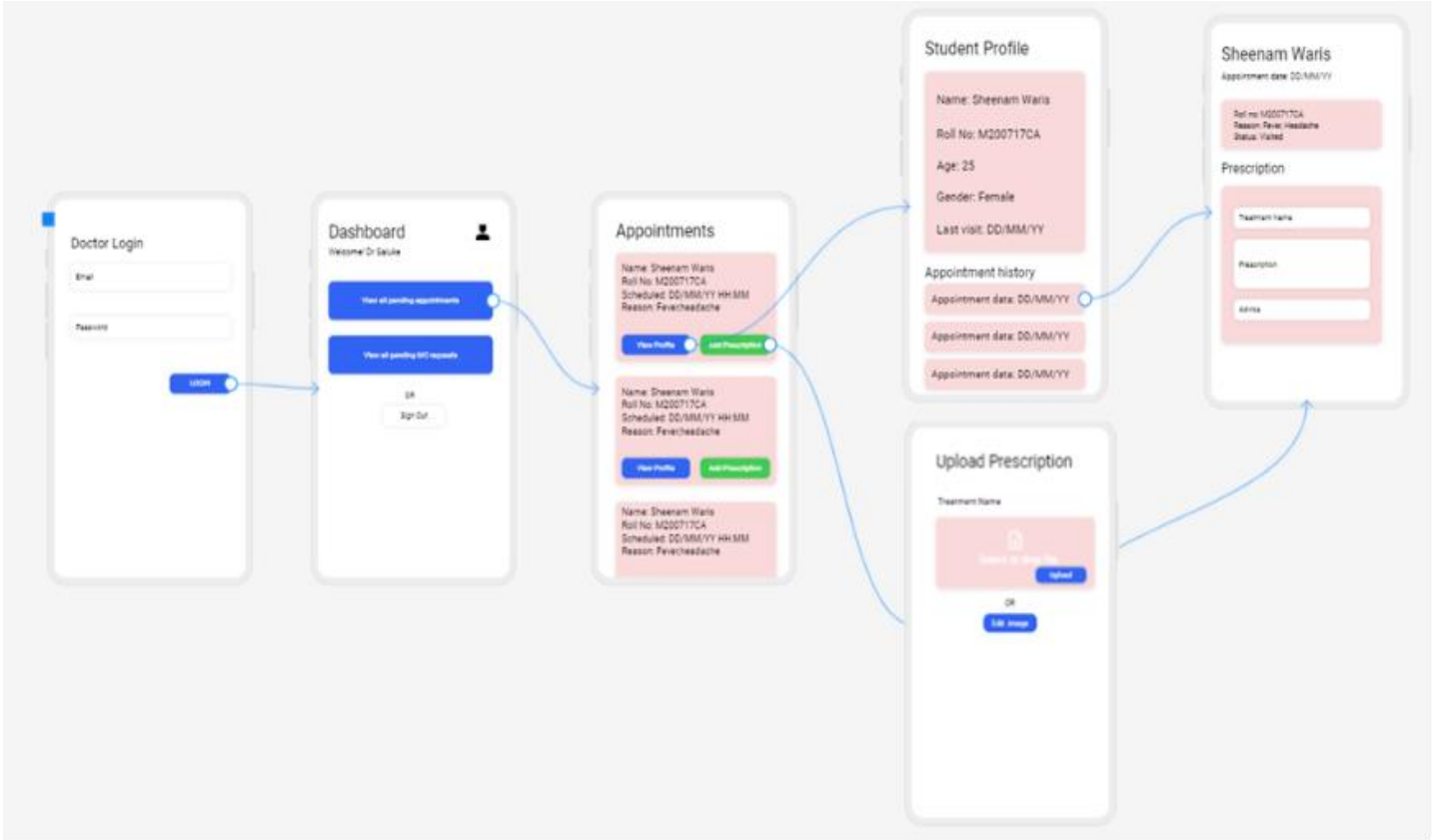
3 Specific Requirements

3.1 External Interface Requirements

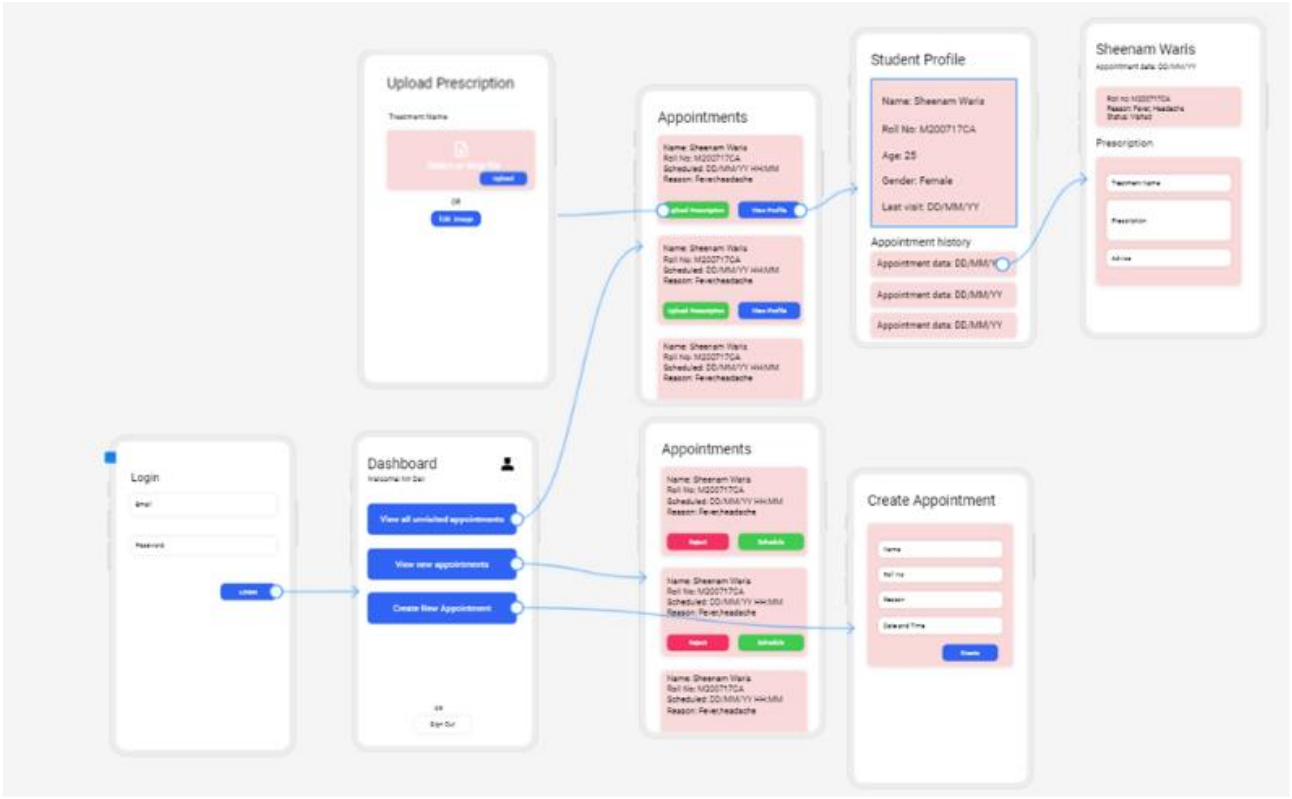
3.1.1 User Interfaces

The NITC Health centre platform is a GUI dependent system that has already set up all user interfaces. The Project Module layout of user interfaces for (Admin and Users) should be consistent with the standard activity modules. Each user interface's content will depend on the specific user type and requirements specified for that user.
The platform aims to provide an intuitive and user-friendly experience, making all user interfaces easy to use. To ensure users can navigate the platform with ease, help documentation will be available to support them. This documentation will include step-by-step instructions, diagrams, and other relevant materials to help users understand how to use the various features and functionalities of the system.
Overall, the user interfaces in the NITC Health centre Application platform will be designed with the user's needs and experience in mind, providing a seamless and efficient user experience.

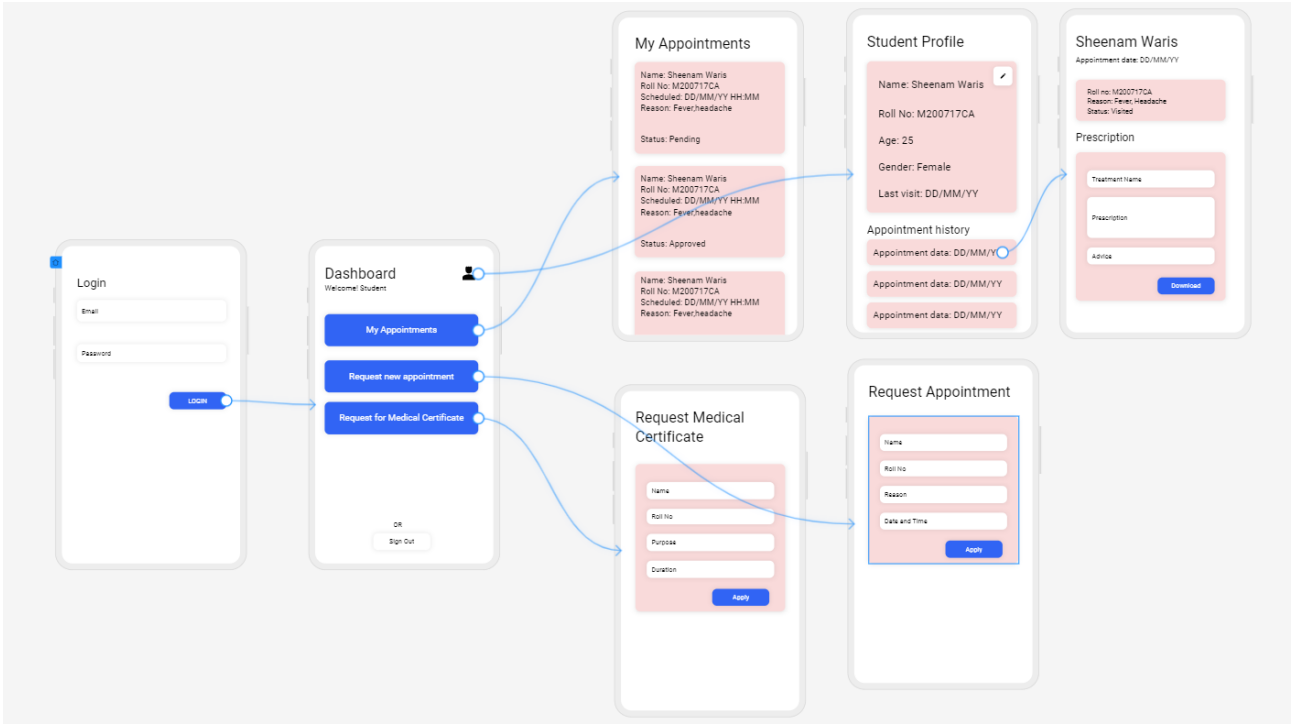
Doctor's UI



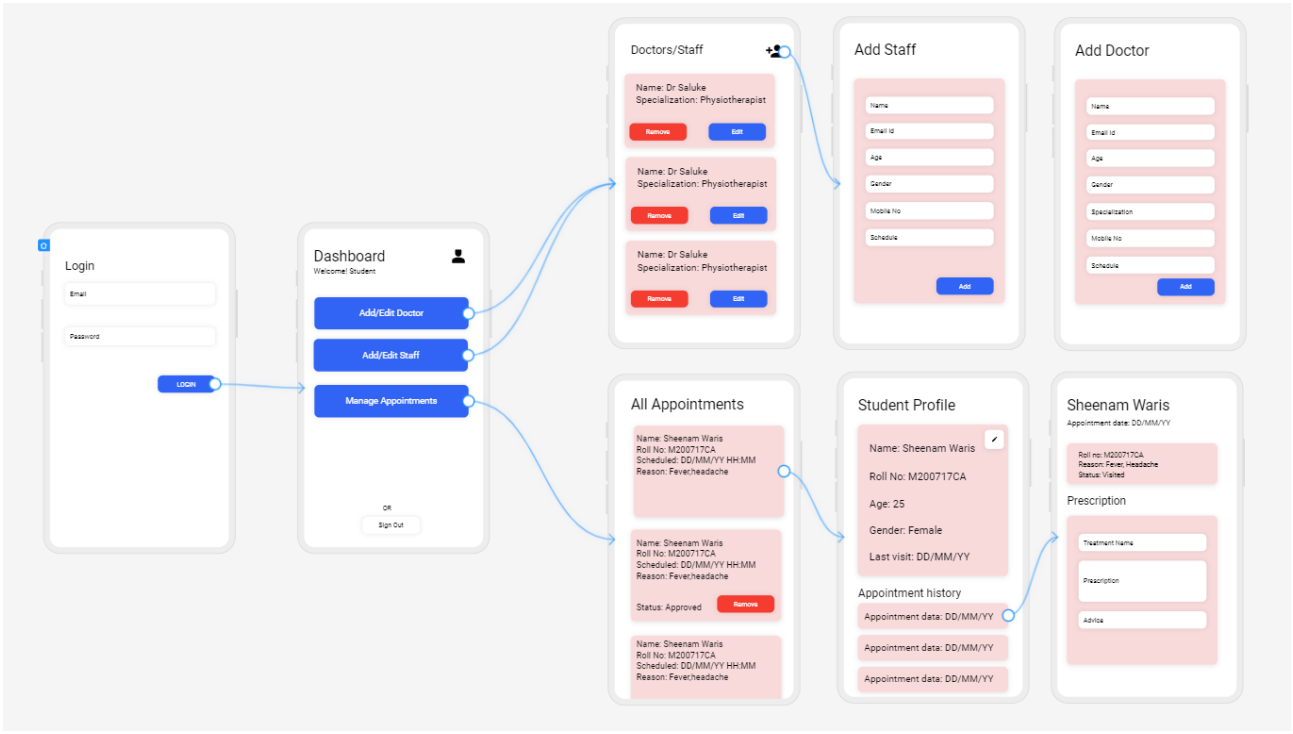
Staff's UI



Student's UI



Admin's UI



3.1.2 Hardware Interfaces

This hardware Constraints refers to the server-side and user-side required hardware specifications so that the module can be run without any problems.

Server System Specification:

Description: Any server system can host the product provided they have the software required (Refer to Software Constraints).

Constraints: The server system must have the minimum hardware which the required software specifies

User System Specification:

Description: Any Android device can run the product provided they have the software required.

Constraints: The user's Android device must have the minimum hardware which the required software specifies.

3.1.3 Software Interfaces

Server System Software:

Description: The server system must have the minimum software to host the final Android app.

Constraints: Prisma Client is one of the most powerful external modules of the node.js.it used to translate the code and its representation from SQL to the Node.js server.

User System Software:

Description: The user's computer system must have the minimum software to run this product.

Constraints: The user's android device must have minimum Android 5.0 or above version to access this application.

3.2 Functional Requirements

ADMIN-

F1: Allows the admin to securely access their account with proper authentication
F2: Enables the admin to manage the doctor's accounts by creating, updating, or deleting them as per the requirement.
F3: Allows the admin to manage the staff's accounts by creating, updating, or deleting them as per the requirement.
F4: Provides the admin with access to view all the previous appointments made by patients, doctors, and staff for record-keeping purposes.

DOCTOR-

F5: Allows the doctor to securely access their account with proper authentication.
F6: Gives the doctor access to the medical history of the patient for reference during diagnosis and treatment.
F7: Allows the doctor to generate a medical certificate for the patient to provide proof of illness and the need for medical leave.
F8: Provides the doctor with access to view the appointments scheduled for the current day for efficient time management.

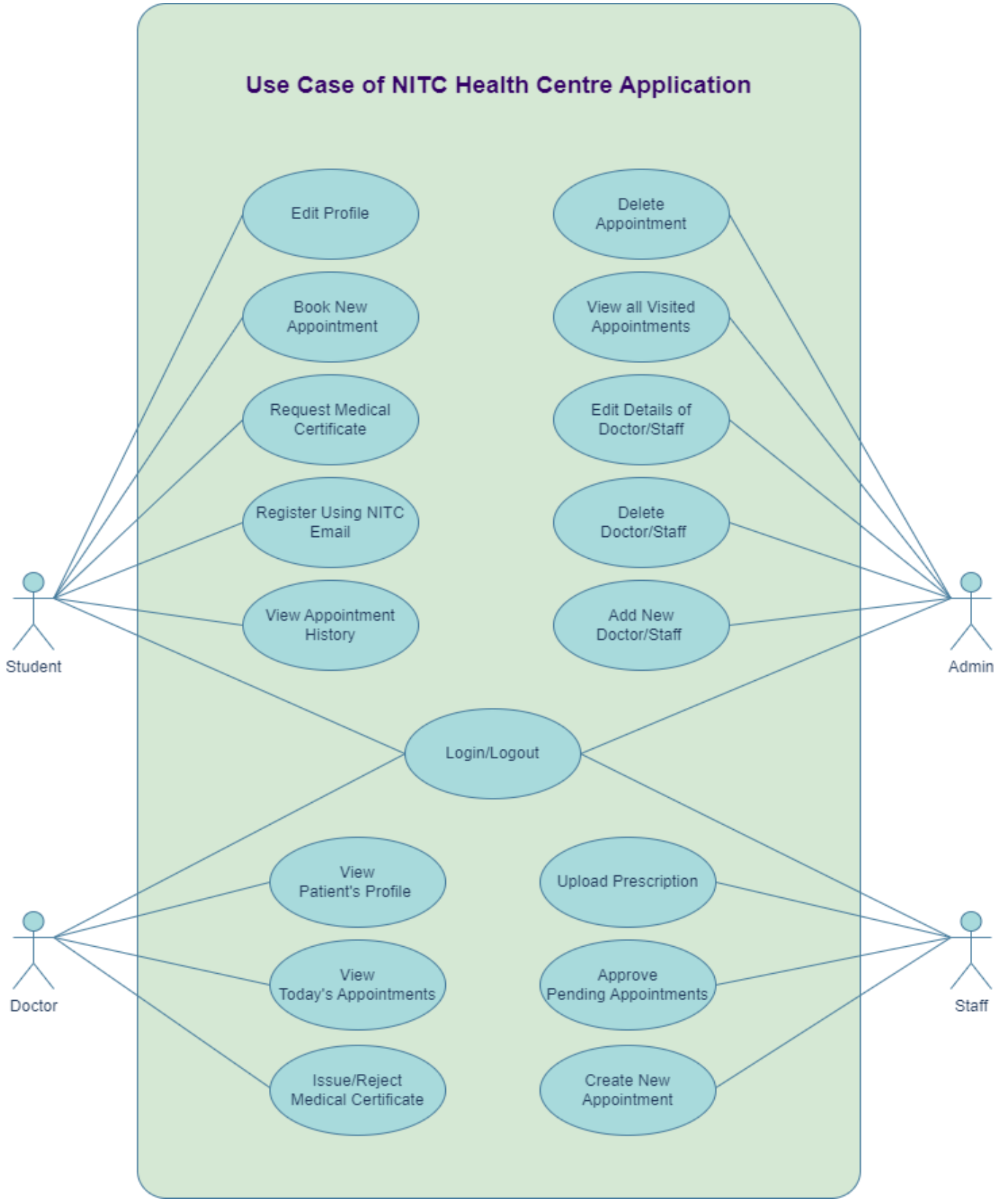
STAFF-

F9: Allows the staff to securely access their account with proper authentication.
F10: Gives the staff access to the medical history of the patient for reference and support for the doctor.
F11: Enables the staff to upload or edit the image of the prescription provided by the doctor for record-keeping and medication preparation purposes.
F12: Allows the staff to approve the appointment request made by the patient based on availability and scheduling requirements.
F13: Enables the staff to create a new appointment for the patient as per their request and availability.
F14: Provides the staff with access to view the appointments scheduled for the current day for efficient time management and scheduling.

STUDENT-

F15: Allows the student to securely access their account with proper authentication.
F16: Enables the students to upload the personal details as per the requirement.
F17: Enables the student to edit their profile details, such as contact information and medical history, for accurate records.
F18: Allows the student to request a new appointment with the doctor or staff for diagnosis and treatment.
F19: Provides the student with access to view their previous appointment history for reference and follow-up.
F20: Enables the student to request a medical certificate from the doctor for illness and medical leave purposes.
F21: Enables the student to authenticate using NITC email and register.

3.3 Use Case Model



3.3.1 Use Case #1 Login (Admin, Doctor, Staff, Student) U1

Author – Sheenam Waris

Purpose - The purpose of this use case is to get access to the software. The user will provide its credentials to get access.

Requirements Traceability – F1, F5, F9, F15

Priority - High

Preconditions - The user has to enter the credentials to get logged to the account.

Post conditions - This will give the user access to the software.

Actors – Human (Admin, Doctor, Staff, Student).

Extends – Verification of user's entered credentials

Flow of Events

- **Basic Flow** - The user will enter his/her email id, password, and role (Admin, doctor, Student) to the system. If the credentials are correct the user will be logged in to his account.
- **Alternate Flow:** If the credentials are invalid then an error message is displayed.
- **Exceptions**: Server is not responding.

Includes: Registered user (already registered by the admin).

3.3.2 Use Case #2 ADD Doctor/Staff U2

Author- Sheenam Waris

Purpose- The system will allow admin to register a doctor/Staff that is an employee of NITC health centre.

Required Traceability - **F2, F3**

Priority - High

Precondition- Admin should be logged in to the application, to register any new Doctor/Staff that is employed in NITC health centre.

Postcondition- Admin will see a successfully registered message, after doctor/Staff data is saved in the database.

Actor- Admin (Human)

Extends- None.

Flow of Events:

- **Basic flow-** Admin needs to enter Name, email Id, password, branch, college ID, phone number, Medical Specialist in/Staff type and click on submit button. If the doctor/Staff data is not present in the system, then it gets successfully saved and gives a message " Doctor/Staff added successfully".

- **Alternate Flow-** If the doctor/Staff data already existed into the database then it will give an error message "Doctor/Staff Exists".

- **Exceptions**- Database server is not responding.

Includes- None.

3.3.3 Use Case #3 Register Student U3

Author: Sheenam Waris

Purpose: The purpose of this use case is to allow student to upload details. So that students can access the application.

Required Traceability: **F21**

Priority: High

Precondition: student should be login to their NITC email for authentication, to register personal data.

Postcondition: NITC student's will be added to system.

Actor: Student (Human)

Extends: None.

Flow of Events:

- **Basic flow:** Student needs to upload personal info data through and current student's details will be added to database.

- **Alternate Flow**: If the user data does not exist into the database, then it will give an error message "Invalid User".

- **Exceptions**: Database server is not responding.

Includes: None.

3.3.4 Use Case #4 Remove Doctor/Staff U4

Author- Sheenam Waris

Purpose- The system will allow the admin to remove any doctor/staff from the application. This is the case when a doctor/Staff leaves NITC.

Required Traceability- F2, F3

Priority- Low

Precondition- The doctors/ staff's data should be present in the application and admin should be logged in.

Post Condition- The doctor/ staff's data is successfully deleted from the application.

Actors- Admin (Human).

Extends- None.

Flow of Events

- **Basic Flow:** The admin will give the doctor/ staff's Email ID and role(optional) and click on the "Retrieve data" button to fetch details of the doctor/staff. Once admin verifies the doctor/staff selected is correct. then it will click on the "Remove" button to delete data.

- **Alternate Flow:** If no such email id is present in the application database, then it will show an error message "No such user exists, check email id!".

- **Exceptions**: None.

Includes- None.

3.3.5 Use Case #5 All Appointment History (Admin) U5

Author- Sheenam Waris

Purpose- The purpose of this functionality is to show all appointment history to Admin

Required Traceability- F4

Priority- Medium

Precondition- The Admin should be logged in to the application

Postcondition- The Admin will see all completed appointment history and pending appointment.

Actor- Admin (Human)

Extends- None.

Flow of Events:

- **Basic flow:** The Admin will login to the application and then move to medical history to check all the details of appointments which has been completed previously and upcoming appointments.
- **Alternate Flow**: If no appointments is done yet, the page will be blank.
- **Exceptions**: Server not responding.

Includes- None.

3.3.6 Use Case #6 Appointment History U6

Author- Sheenam Waris

Purpose- The purpose of this functionality is to show medical history to staff, doctor and patient.

Required Traceability- F7, F12, F20

Priority- Medium

Precondition- The staff, student and doctor should be logged in to the application

Postcondition-The student will see his own medical history and doctor, staff will see medical history of the patients.

Actor- Doctor, Staff, Student (Human)

Extends- None.

Flow of Events:

- **Basic flow-** The doctor and staff will login to the application and then move to medical history to check all the details of his appointments he has previously taken and his upcoming appointments. Student is also able to check his appointment date, time and download prescription.
- **Alternate Flow-** If no appointments is done yet, the page will be blank for that person.
- **Exceptions**- Server not responding.

Includes- Create Scheduled Appointment.

3.3.7 Use Case #7 Verify & Generate MC U7

Author- Sheenam Waris

Purpose- The system will allow the doctor to print medical certificate for the student who requested for medical certificate and is not able to attend classes/ examination because of suffering from severe problem and need to be admitted to hospital. After successful recovery the student will need doctor and admin permission to re-join campus.

Required Traceability- F9

Priority- High

Precondition- The doctor's data should be present in the application, student should have a hospital medical certificate to proof of his recovery from illness.

Post Condition- After Successful verification of student submitted document, the students can re-join campus.

Actors- Doctor (Human).

Extends- None.

Flow of Events

- **Basic Flow:** The student will request for medical certificate online. Doctor will verify and generate medical certificate. Also, patient should meet in person once to doctor for his verification. After successfully verified, the doctor will print a medical certificate for the student to get admitted to hospital. To re-join campus again, the student will have to verify to doctor using this same use case itself.

- **Alternate Flow:** If the doctor finds that any of the provided documents by the student is faulty then his request is cancelled and he again needs to apply for it with correct documents.

- **Exceptions**: None.

Includes- None.

3.3.8 Use Case #8 Upload/Edit Prescription U8

Author- Sheenam Waris

Purpose- The system will allow the staff to upload or edit a prescription's image that will be visible in the patient's profile.

Required Traceability- F13

Priority- Medium

Precondition- The staff should be logged in to the application and open patient profile.

Post Condition- Message will pop up " Successfully Uploaded" and then redirected to the patient appointment page.

Actors- Staff (Human).

Extends- None.

Flow of Events

- **Basic Flow:** The staff will log in to the application then click on the view patient's appointment then upload prescription's image, then click on the upload button.
- **Alternate Flow:** Staff can edit the prescription if they found out any mistakes while uploading.
- **Exceptions**: Doctor already added prescription online.

Includes- None.

3.3.9 Use Case #9 Approve Appointment Request U9

Author- Sheenam Waris

Purpose- The purpose of this functionality is to allow staff to approve appointment requested by student.

Required Traceability- F14

Priority- High

Precondition- The staff should be logged in to application and he/she should have scheduled requested appointments on that date and time.

Postcondition- The student will get notified when their request has been approved.

Actor- Staff (Human)

Extends- None.

Flow of Events:

- **Basic flow:** The staff logged in to the application and moved to the "new appointment" page. Here, then give approval. After successful approval the patient will get notified.

- **Alternate Flow**: If the approval is not possible for some reason, then it would reject the appointment of that date.

- **Exceptions**: Server not responding.

Includes- None.

3.3.10 Use Case #10 New Appointment Request U10

Author- Sheenam Waris

Purpose- The purpose of this functionality is to allow users (Student, Staff) of this application to book an appointment.

Required Traceability- F15, F19

Priority- High

Precondition- The user should be registered to the application and logged in to it.

Postcondition- Appointment is booked successfully.

Actor- User (student, staff) (Human)

Extends- User Verification.

Flow of Events:

- **Basic flow:** The user first login to the application and then move to the book appointment page then according to his illness problem will select the option and then describe his problem in the textbox after that click on the "Request appointment" button. Users will get notified when booking is confirmed.

- **Alternate Flow**: If the student was not able to book then staff can book new appointment whenever student visits

- **Exceptions**: Server could not be reached.

Includes- None.

3.3.11 Use Case #11 View Appointment U11

Author- Sheenam Waris

Purpose- The purpose of this functionality is to show doctor/staff current day's appointment of patient.

Required Traceability- F10, F16

Priority- High

Precondition- The doctor/staff should be logged in to application and he/she should have scheduled requested appointments on that date and time.

Postcondition- The doctor/staff will plan accordingly based on appointments.

Actor- Staff, Doctor (Human)

Extends- View patient Profile, create prescription

Flow of Events:

- **Basic flow:** The Doctor/staff logged in to the application and moved to the "new appointment" page. Here, views all current scheduled appointment.
- **Alternate Flow**: None.
- **Exceptions**: Server not responding.

Includes- None.

3.3.12 Use Case #12 MC Request(student) U12

Author- Sheenam Waris

Purpose- The purpose of this functionality is when a user has a severe illness then it will allow him to take leave from campus to get admitted to a hospital.

Required Traceability- F21

Priority- Medium

Precondition- The user should be logged in to the application. User should have permission given by the NITC doctor describing his illness and have Original College ID.

Post condition- User will get Medical leave certificate from Doctor in person/online.

Actor- Student (Human)

Extends- None.

Flow of Events:

- **Basic flow-** The user will login to the application then move to the " Medical leave request page". Then it will submit the above mentioned pre condition documents as a pdf to the admin and click on the "Submit" button. Then the user should meet in person to admin in person for verification, after successfully verifying he/she will get a "Printed Medical Certificate".

- **Alternate Flow-** If in the verification some faults occur in his documents then he again needs to submit the request with correct documents or come up with special permission from Dean.

- **Exceptions**- None.

Includes- Print Medical Certificate.

3.3.13 Use Case #13 Update Profile U13

Author- Sheenam Waris

Purpose- The purpose of this functionality is when a user enabled to update his/her details and the changes should reflect in database.

Required Traceability- F18

Priority- Low

Precondition- The user should be logged in to the application. The student must be a registered patient.

Post condition- The record of patient is updated in database.

Actor- Student (Human)

Extends- None.

Flow of Events:

- **Basic flow-** The user will login to the system. He/she view his record then selects update details. Now user may change the necessary fields. Popup of update details.

- **Alternate Flow-** None

- **Exceptions**- None.

Includes- None.

3.3.14 Use Case #14 Health Centre Schedule U14

Author- Sheenam Waris

Purpose- The purpose of this functionality is when a student wants to book an appointment then they can check availability of their preferred doctor.

Required Traceability- F22

Priority- Low

Precondition- The user should be logged in to the application. The student must be a registered patient.

Post condition- The schedule will be visible to student

Actor- Student (Human)

Extends- None.

Flow of Events:

- **Basic flow-** The user will login to the system. He/she view health centre schedule. Now user may book the appointment.
- **Alternate Flow-** None
- **Exceptions**- None.

Includes- None.

4 Other Non-functional Requirements

This section describes the non-functional requirements of the System, such as performance requirements, database requirements, usability, security, reliability, maintainability and scalability. This section also describes the hardware constraints and software constraints of the system. This is to ensure the usability and effectiveness of the entire system.

4.1 Performance Requirements

- User email and password verification at the time of login.
- Delay must be minimized to enhance user experience with an interactive interface.
- The UI will be user friendly and most of its part will be predefined; the user only needs to select the relevant option.
- User only needs basic internet connection since the system does not be too heavy, but only at the time of video and audio call it will need a reliable internet connection.
- User can see his history of booking at any time since its all data is stored at cloud storage which proves faster access.

4.1.1 Interface Loading Time:

The loading time of each page/interface from the Product should be consistent with the other pages already in product. Each page/interface should load in less than 30 seconds. Constraints of the static figure are based on experimentation with the existing system. This requirement can be dependent on the user's android device or the server itself.

4.1.2 Multiple Access

The NITC Health Centre Application shall allow multiple users to access it.

4.2 Safety and Security Requirements

- To secure a password, the first password is encrypted and then stored at the cloud database.
- Private user data will not be available to anyone; only admin have the access to use it.
- Email provided by the user at the time of registration should be his/her NITC id only.
- The product delivery safety is maintained and will be at high priority.

4.2.1 Server System Specification

The server system must have the minimum hardware which the required software system specifies. refer to software specification constraints

4.2.2 User System Specification

Any android device which has an android version greater then equal to 5.0 lollipop will be able to use our application.

4.3 Software Quality Attributes

Availability: This application is available to those who have NITC mail id or related to NITC.

Robustness: The application provides complete security to user credentials and is safe at our cloud storage, only admin have the access to the data.

Maintainability: The application will be developed with the help of VS Code and will be maintained through it.

Extendibility: The application is designed in such a way the "optimal" functionality requirements will be implemented at later dates. In later dates extra functionalities can also be added to the application.

Portability: The product is designed in such a way that it can be added to any Online health application system. It is also necessary that all the frontend part should be implemented with flutter and the backend is developed with NodeJS and mongo db.

5 Other Requirements

If possible, this project can be made open source and stored on GitHub and made public for NITC community, so that anyone can improve this app and add more amazing features to this app. This will be useful in the evolution of the app by the support of NITC student developers.

Appendix A - Activity Log

Date	*Discussion with Mentor*
23/01/2023	*Distinct Project Ideas proposal*
25/01/2023	*Project Idea (NITC Health Centre) Approved*
27/01/2023	*Literature Survey of Health Centre*
01/02/2023	*Basic description of the project,* *Functional requirements* *Who are users of this application*
06/02/2023	*Completed Abstract*
27/02/2023	*Problem Definition*
06/03/23	*Discuss on the UML Diagram* *How to write the use cases*
08/03/2023	*Completed Initial version of problem definition document*
09/03/2023	*Started working on SRS*
13/03/2023	*Non-functional requirements* *Other requirements*
15/03/2023	*External User Interface*
17/03/2023	*Completed Initial version of SRS*

6 Technology Stack

Frontend Development: Android (Java)
Backend Development: NodeJS
Database: SQL Server

7 Database Design Tables

Patient's Database Design

S.no.	Attribute Name	Data Type	Constraint	Description
1	Patient_ID	int	Primary Key	Contains unique ID
2	Name	Varchar	-	Contains Name
3	Gender	Varchar	-	Contains gender
4	DOB	Varchar	-	Contains Date of birth
5	Email	Varchar	-	Contains NITC email
6	Roll No	Varchar	-	Contains Student's roll number
7	Address	Varchar	-	Contains student's home/local address
8	Hostel details	Varchar	-	Contains student's hostel name and room no.

Admin/Staff/Doctor's Database Design

S.no.	Attribute Name	Data Type	Constraint	Description
1	User_ID	Varchar (50)	Primary Key	Contains unique ID
2	Password	Varchar	-	Contains Hashed password
3	Name	Varchar (50)	-	Contains Name
4	Gender	Varchar (50)	-	Contains gender
5	DOB	Varchar (50)	-	Contains Date of birth
6	Email	Varchar (50)	-	Contains NITC email
7	Mobile No.	Integer	-	Contains contact number
8	Role	Varchar (20)	-	Contains Admin/Staff/Doctor As category of user

Appointment's Data Design

S.no.	Attribute Name	Data Type	Constraint	Description
1	Appt_ID	Varchar	Primary Key	Contains unique ID
2	Patient_ID	Varchar	Foreign Key	Contains Student's roll Number
3	Reason	Varchar	-	Contains Medical reason for visit
4	Schedule time	Date and Time	-	Contains date and time
5	Status	Varchar	-	Contains approved, pending, reject as status
6	Reporting time	Time	-	Contains time of patient's visit

7	Treatment Name	Varchar	-	Contains treatment name
8	Image	Binary String	-	Contains image of prescription

Medical Certificate (MC) Requests Data Design

S.no.	Attribute Name	Data Type	Constraint	Description
1	MC_ID	Varchar	Primary Key	Contains unique ID
2	Patient_ID	Varchar	Foreign Key	Contains Student's roll Number
3	Appt_ID	Varchar	-	Contains appointment ID
4	Purpose	Varchar	-	Contains Medical reason for visit
5	Duration	Date	-	Contains date
6	Req_Date	Date & time	-	Contains date of request
7	Status	Varchar	-	Contains approved, pending, reject as status
8	Approval_Date	Date & time	-	Contains date of approval
9	Image	Binary String	-	Contains image of MC to download

8 Class Diagram