



SOFTWARE REQUIREMENT SPECIFICATION

CLINICAL APPOINTMENT SYSTEM



SUBMITTED BY
NIKTI PAUL_ROLL: 34900119032
AYAN DHARA_ROLL: 34900119045
PRITHICK SAMUI_ROLL: 34900119029

Contents

Sl no.	Contents	Page
1	Problem Analysis	2
2	Project Planning	4
2.1	Resource Planning	4
2.2	Risk Management	4
2.3	Purpose	5
2.4	Solution	6
3	Identify scope objective and infrastructure	7
3.1	Scope	7
3.2	Objective	7
3.3	Overview	8
3.4	Infrastructure	8
3.4.1	Product Perspective	8
3.4.2	Product Diagram	9
4	Software Requirement analysis	12
4.1	Individual Phases	12
4.1.1	Design Phase	12
4.1.1.1	User Interface	12
4.1.1.2	System Interface	16
4.1.1.3	Database	17
4.1.2	Development Phase	17
4.1.3	Testing Phase	17
4.1.4	Deployment Phase	17
4.1.5	Maintenance Phase	18
4.2	Functional and Non-Functional Requirements	18

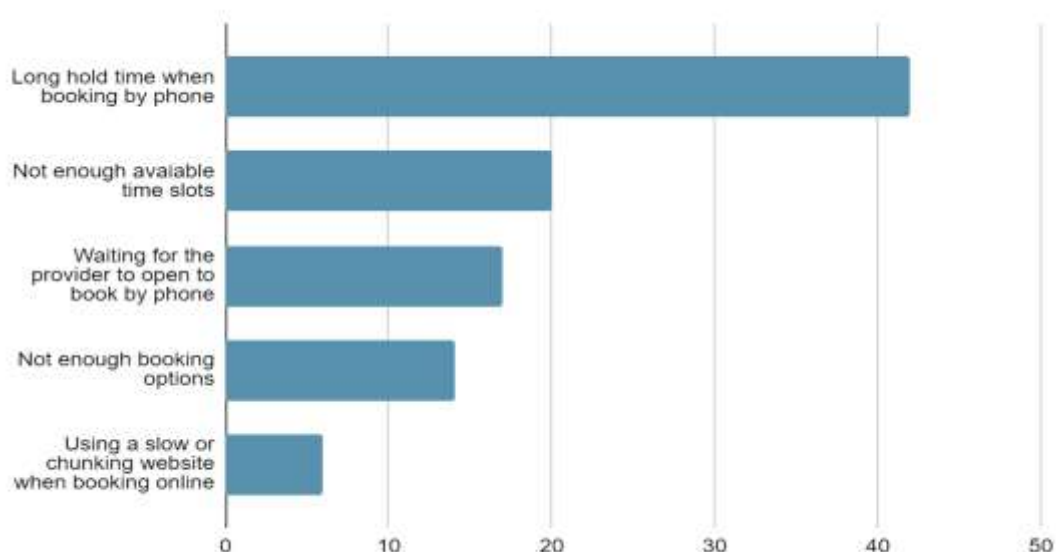
CLINICAL APPOINTMENT SYSTEM

1. Problem Analysis:

- **The offline appointment booking system is not efficient:** In the context of urgency, the efficiency of the Traditional offline appointment system is questionable. In this busy world, people find it difficult to go to the clinic and make appointments for their medical check-ups. Even such processes can be more complicated during rush hours. Sometimes a whole day gets occupied to do such a simple task.
- **Lacks when there is a schedule change:** In many cases, it has been seen that doctors are not free for appointments for a particular week or the clinic will remain closed for some particular days or any other reason, patients are unaware of such information or notices. They find it very difficult to get to know about this which sometimes affects their regular check-ups or any other medical emergencies. This may also interrupt their schedule.
- **Mass gathering and Violation of Social Distancing:** During this deadly pandemic it is very necessary to avoid mass gathering and maintain social distancing. But the traditional appointment system causes unnecessary gatherings. Places like doctors' clinics are the most unsafe area and those places are overly crowded by the patients.
- **No transparency in the queue:** In many cases, people complain about unfair appointment systems and list generation. In traditional systems there is always a "favouritism factor", the receptionist or the managing committee alters the list as per their will. Such malpractices are common because there is no transparency in the list. Patients are unaware of their order in the list.
- **Have to wait for your turn for a long time:** In many cases, it is seen that people wait for hours for their turn. The first person in the queue and a random person in the queue arrive at the clinic at the same time. They can not estimate their appointment time. This not only leads to an unnecessary crowd but is also a waste of the person's time.
- **Call and appointment system is also a failure:** Call and appointment registration system is also a failure in many cases. There are many clinics

around the city that book slots for the person only by accepting a registration fee, which is not possible through calls. Also transacting through UPI Ids doesn't confirm the person's identity or your booking. So there is always an authentication issue.

- **Failure to get an appointment due to full booking of slots in traditional offline booking method:** In many cases, it has been seen that a person goes to the clinic for an appointment but is unable to make it since all the slots are occupied. Ultimately their time is wasted due to being unaware of that information.
- **Emergency cases in fully computerized and automated systems:** For a computerized system without manual entry and a machine learning algorithm, emergency cases during regular check-ups may affect the whole list.
- **The problem for those people who are unaware of modern technology:** Also a fully automated system may cause a problem for those who are not friendly with online booking and android phones.
- **Offline prescription problems:** Keeping the offline prescription can be problematic. It might get lost, or can get damaged. Due to different problems, there can be multiple prescriptions, keeping track of all those can be a hassle.
- **Appointment booking problems via different methods:**



2. Project Planning:

· 2.1 Resource planning:

1. The Android-based application will be used to create the application for an online appointment booking system and queue management.
2. Adobe XD will be used to create the Prototype for the User Interface of the application.
3. Java will be used as the frontend stack.
4. Node js with an express framework will be used to do the backend of the application.
5. Machine learning algorithms with python will be used to monitor and manage the queue system.
6. Git and Github will be used to collaborate with teammates and version control of the application.

· 2.2 Risk Management:

1. There are a lot of online appointment booking apps that are available on the market. Initially, people may not accept this app because there are strict protocols that need to be maintained for seamless and smooth service. In such a case, the application must be smooth, user-friendly, and unique to overcome this risk.
2. Due to server down issues, the application might face difficulty in efficient queue management. To overcome such issues a snapshot is taken every time a new patient is added to the queue. The local machine will be used to store the snap. The current snapshot gets replaced with the previous one.

· 2.3 Purpose of the Project

1. **Proper management of appointments:** Appointments will be managed properly by implementing data transparency.
2. **Helps users to save time:** In many cases, a patient visits for a check-up and medical treatment but has to wait for hours for the appointment. This system will reduce the waiting time by 70 to 50% and help the user to save time.
3. **Maintaining queue:** Queue will be maintained based on live reports, emergency cases, and other priority issues. A fair and generalized queue will be generated using different sorting algorithms.
4. **Replacement of traditional appointment system:** Replacing traditional appointment system by digital appointment system using the android application and web services.
5. **Maintain COVID Protocol:** Helps to prevent mass gathering and to maintain social distancing.
6. **To maintain authenticity:** The generated queue or the list will be authentic and can not be altered by the receptionist. The access will be available only to the admins.
7. **Proper attendance system:** Helps to maintain a proper attendance system by digitalizing the entry and exit of the attendant.
8. **Providing Slotting System:** To provide particular slots to users based on machine learning algorithms.
9. **Generating Analytics:** To generate digitized surveys and reports based on the total number of attendants, their purpose of visit, and the maximum number of visits on a particular time and date.
10. **Keep tracking patient records:** This software will help the user as well as the admins to track patients' profiles and store the digital format of prescriptions provided by the consulting doctor

2.4. Solution of Problems mentioned in Problem Analysis:

2.4.1. Problem: Offline booking system is not efficient.

Solution: The best way to increase efficiency in the appointment system is by digitizing the whole system. An online appointment system will help the users to book slots from anywhere, anytime, and will provide flexibility.

2.4.2. Problem: Lacks when there is a schedule change.

Solution: Since the system is digital, users will get an instant notification and all the necessary information regarding the clinic and its new schedules.

2.4.3. Problem: Mass gathering and violation of Social Distancing.

Solution: The digital system will help to book appointments online and its algorithm will also provide an appropriate time for the appointment. That is, people will not have to wait for 3 - 4 hours for their name, they will get an appointment time. This will help to reduce gatherings and will help to maintain COVID protocols.

2.4.4. Problem: No transparency in the queue.

Solution: Since everything is digital, there will be transparency in the list generation. All the patients can see their name and order in their android application. The list will be so secured that not even the receptionist or the managing committee can alter it. The application will have three user interfaces one for users, one for receptionists, and one for the admins. Receptionists will have only pushing access, they cannot alter the list.

2.4.5. Problem: Emergency cases in fully computerized and automation:

Solution: Naturally, Emergency cases may arise in any clinic. In such a case to avoid conflict in appointments we will use a priority queue and machine learning algorithm which will help to reschedule the whole queue instantly.

2.4.6. Problem: Problem for those people who are unaware of technology.

Solution: This is a different problem that cannot be solved directly but the best possible solution is by adding a feature in the receptionist end that if somebody wants, they can go to the clinic and book an appointment manually. Receptionists can add the person's name from his/her device and make an entry on the behalf of the patient.

3. Identify project scope, objective, and infrastructure:

· 3.1 Scope:

- a. **Helpful for Businessman and Office Person:** This project will help the person who has a very tight and busy schedule. They can easily book appointments online without visiting the clinic.
- b. **Clinics will widely accept the software:** In medical shops, they use software to keep records of the purchases, medicines and other stuff which help them to manage things properly likewise this software will help them to manage appointments properly throughout the whole week. Digitalizing the system will be beneficial for them.
- c. **User-friendly Interface;** Almost 62% population of India has a smartphone and among them, maximum people are comfortable using it. The user interface is so friendly that maximum people will love to use it.
- d. **Helpful for people living in another town;** Many people book appointments from another town over a phone call or have to take help from another person living in that town. So, this system will be beneficial for them.

· 3.2 Objective:

This software will be an online appointment booking and data managing system for any medical clinic or any service-providing shop wishing to replace traditional offline appointment systems. More specifically to design and develop a simple and fair-queue management system using different algorithms which will not only save attendants time but also be a transparent, authentic, and effective way to provide services. There will also be a manual method of slot booking for those who are still unaware of modern technology to cover all possible users. This will also generate analytics that will help the service provider to improve or update their service time and list management.

3.3 Overview:

Avoiding mass gathering and maintaining social distancing has become very crucial during this **Covid-19** outbreak. **Appointment System** software will not only avoid mass gathering but also help to reduce human effort and save time by sorting the queue fairly and transparently. In the initial phase, it will take care of all the complications related to sorting queues, human effort, saving time, digitalizing appointments, and proper management of the list. The best part of this project is that it is scalable. We can add more features like having a digital record of all the prescriptions of the patients given by the doctor. To sum up, in its initial phase it is capable of solving smaller problems but in the future, it can be developed as per the requirement to perform much more complicated tasks.

3.4 Infrastructure:

4.4.1. Product Perspective: (CAS) Clinical Appointment System is meant to serve as a common platform where queue management is carried out conveniently. It replaces the traditional system by making it more efficient and effective.

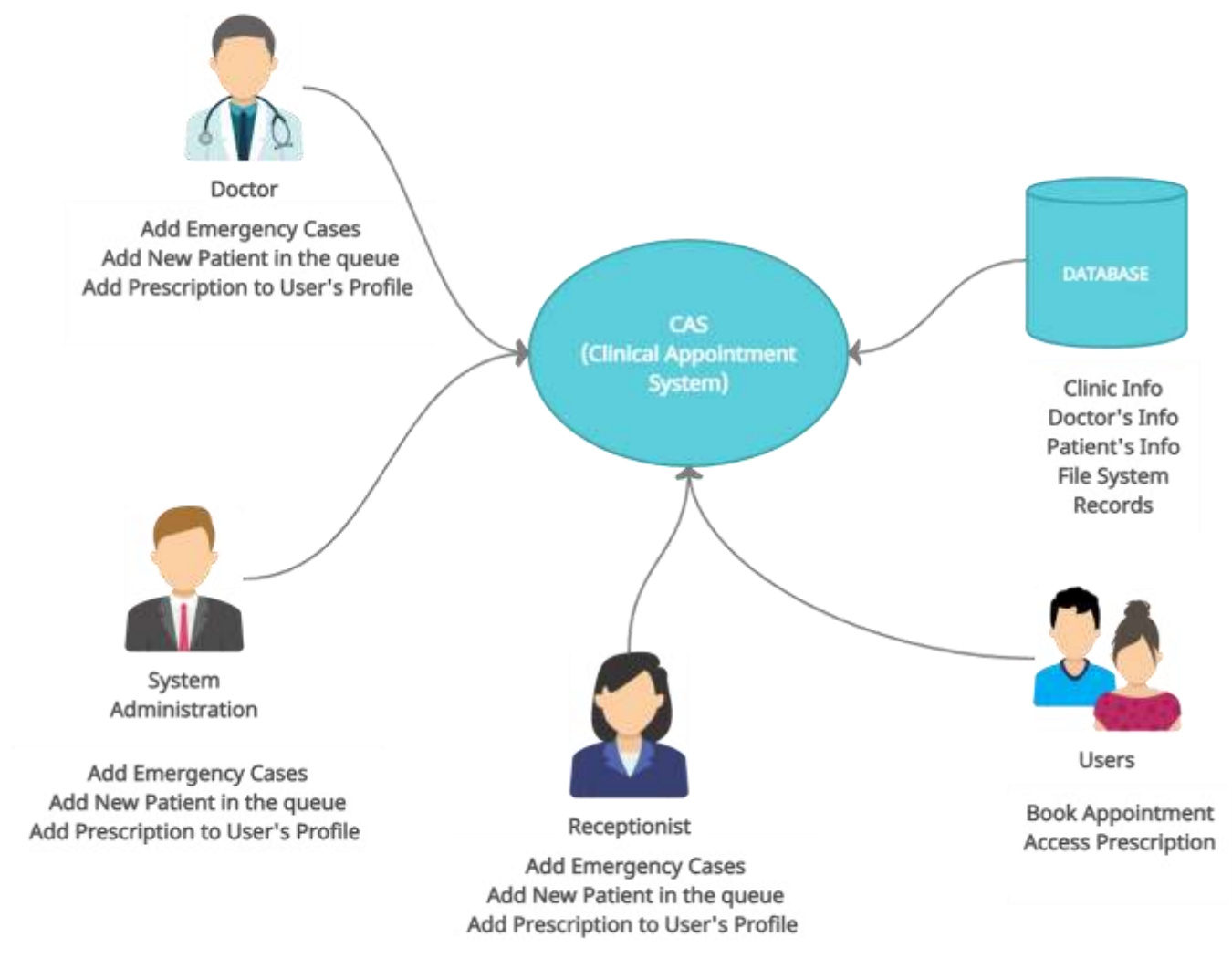
- a. System Interface:** Node-js will be used as a web server and all the API requests will be carried out by the express-js framework written on Node js. Users will be authenticated using their credentials and all previous data will be stored in the server.
- b. User Interface:** By keeping in mind that there is a huge diversity of users, the UI will be very simple and friendly so that anybody can book an appointment easily as well as the management of data by the clinics will be a child's play. Users can easily navigate through their profiles, doctor's profile, their stored prescription and their order in the list. The prototype of the UI will be first designed using Adobe XD and after a survey the application will be built using JAVA and XML.
- c. Hardware Interface:**
 - I. Server side:** The web application will be hosted on a web server which is listening on the HTTPS web standard port, port 443.

II. Client Side: Only an Android phone will be needed to install the CAS (Clinical Appointment System) Software.

III. Communication Interface: The HTTP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

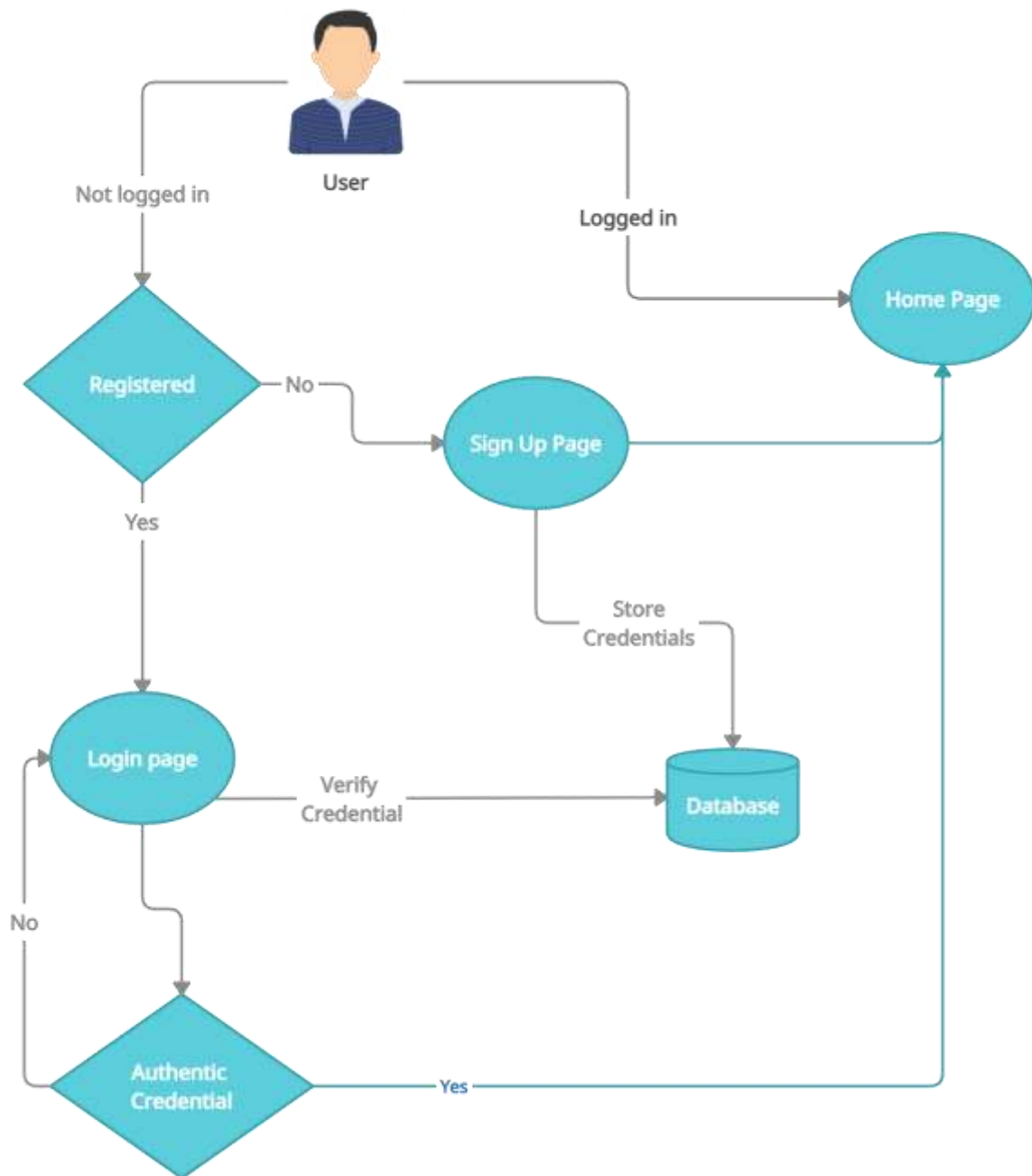
3.4.2. Product Function:

3.4.2.1. Context Diagram:

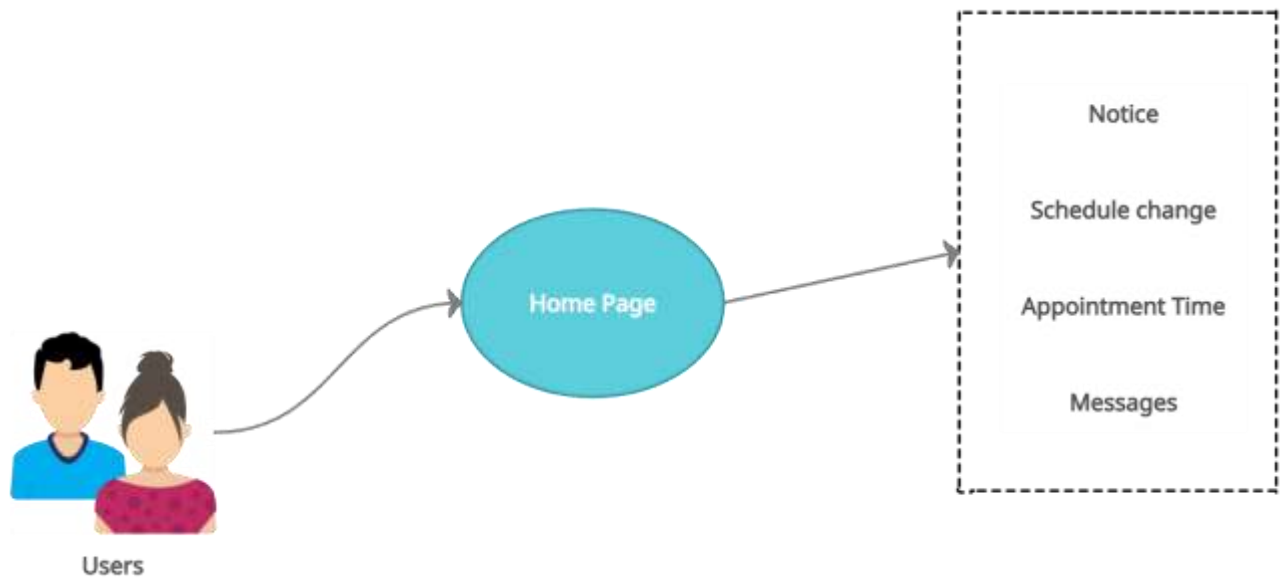


3.4.2.2. User Case Diagram:

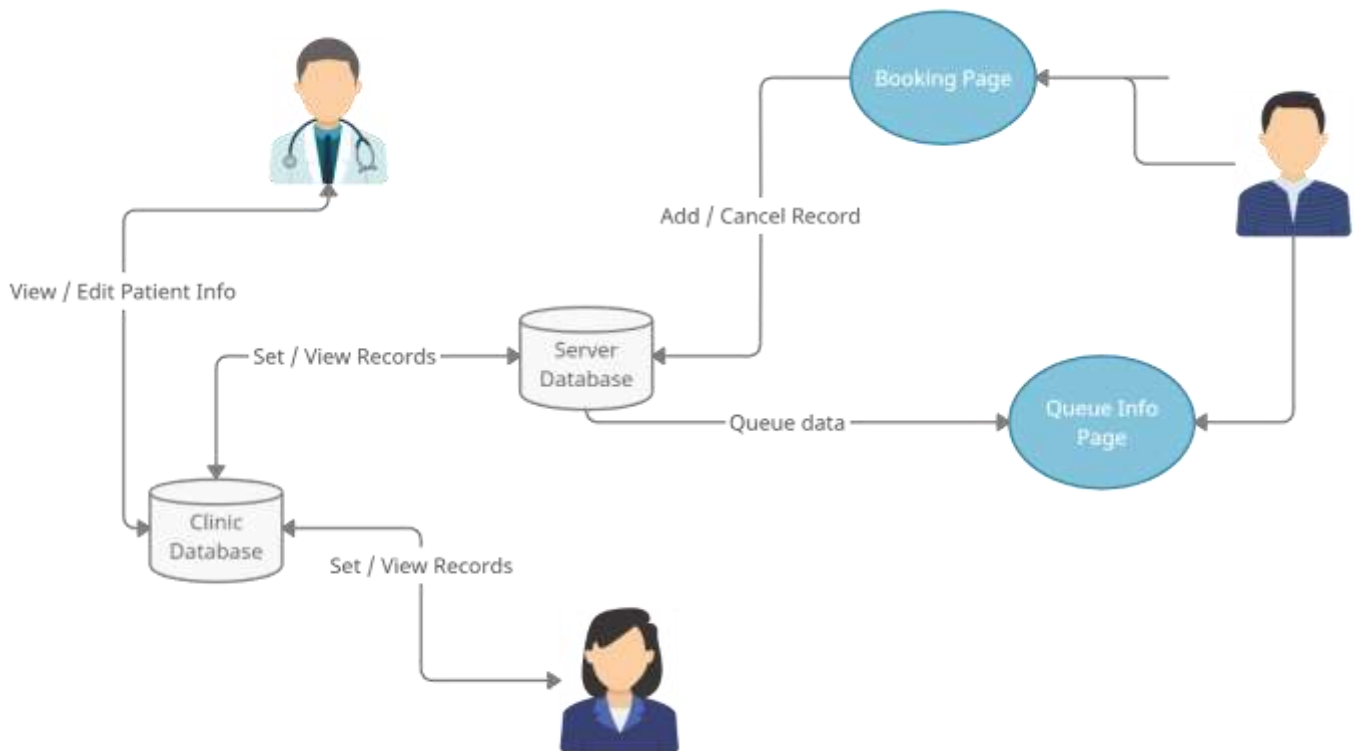
3.4.2.2.1. Login Page:



3.4.2.2.1. Push Notification:



3.4.2.2.1. Records:




4. Software requirement Analysis:

4.1 Individual Phases:

4.1.1 Design Phase:

4.1.1.1 User Interface:



The illustration shows a doctor in a white coat with a stethoscope around his neck, holding a small plant. He is standing next to a patient in a green dress who is also holding a small plant. They are both smiling and looking at each other.

Appointment System

Registered Email

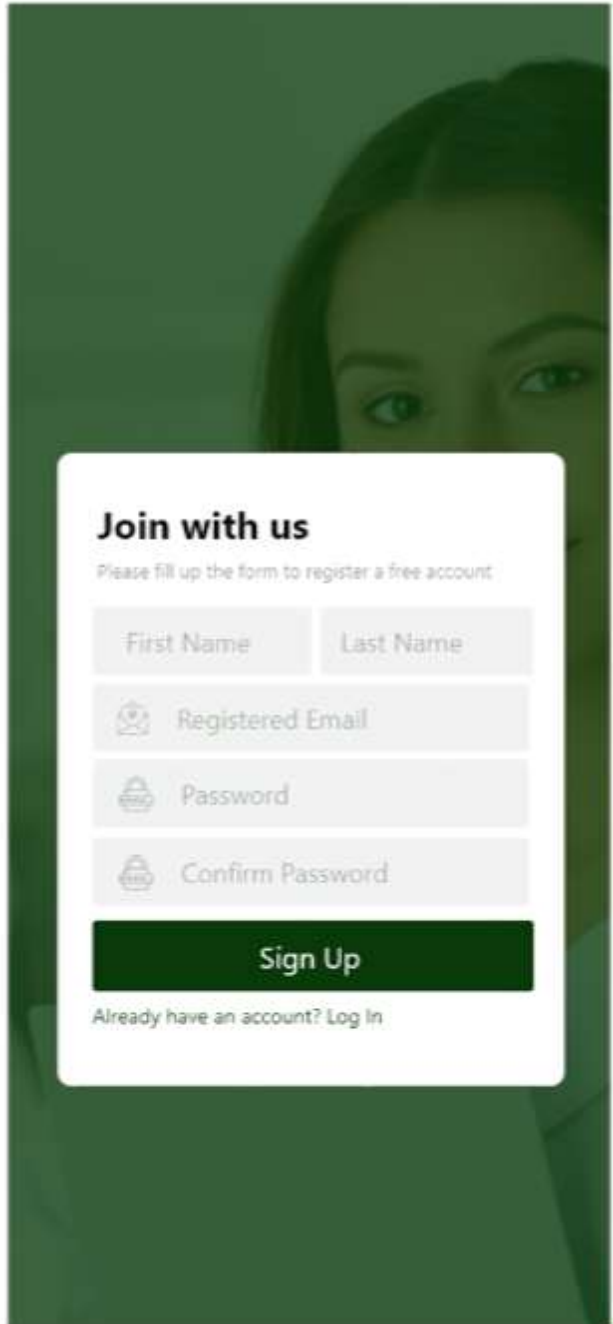
Password

Log In

New User? Sign up

Made in India

Log In



The background is a dark green color with a faint image of a woman's face. The sign-up form is a white box with a dark green header and footer.

Join with us

Please fill up the form to register a free account

First Name Last Name

Registered Email

Password

Confirm Password

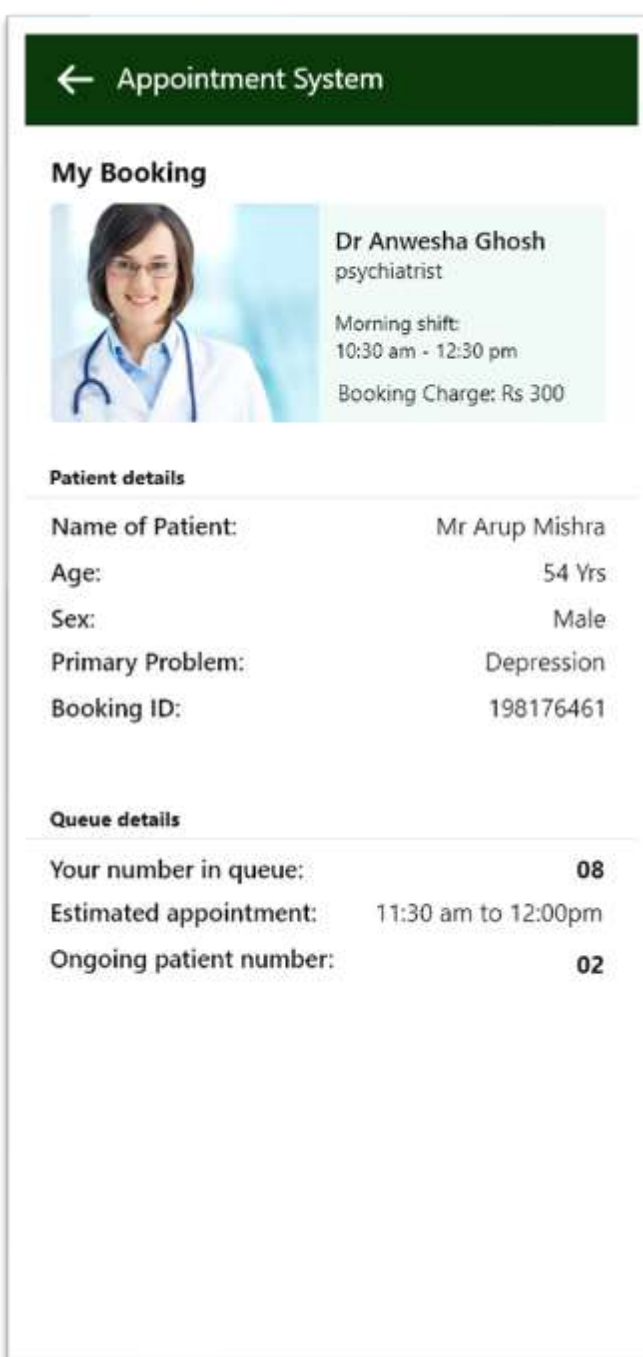
Sign Up

Already have an account? Log In

Sign Up



Home Page



Booking Status

← Appointment System

Sunday

Dr Anwesha Ghosh	10:30 am - 12:30 pm
Dr Rishi Banerjee	01:30 pm - 04:30 pm
Dr Anwesha Ghosh	10:30 am - 12:30 pm

Monday

Dr Anwesha Ghosh	10:30 am - 12:30 pm
Dr Rishi Banerjee	01:30 pm - 04:30 pm
Dr Anwesha Ghosh	10:30 am - 12:30 pm

Tuesday

Dr Anwesha Ghosh	10:30 am - 12:30 pm
Dr Rishi Banerjee	01:30 pm - 04:30 pm
Dr Anwesha Ghosh	10:30 am - 12:30 pm

Wednesday

Dr Anwesha Ghosh	10:30 am - 12:30 pm
Dr Rishi Banerjee	01:30 pm - 04:30 pm
Dr Anwesha Ghosh	10:30 am - 12:30 pm

Thursday

Dr Anwesha Ghosh	10:30 am - 12:30 pm
Dr Rishi Banerjee	01:30 pm - 04:30 pm
Dr Anwesha Ghosh	10:30 am - 12:30 pm

Friday

Dr Anwesha Ghosh	10:30 am - 12:30 pm
Dr Rishi Banerjee	01:30 pm - 04:30 pm
Dr Anwesha Ghosh	10:30 am - 12:30 pm

Saturday

Dr Anwesha Ghosh	10:30 am - 12:30 pm
Dr Rishi Banerjee	01:30 pm - 04:30 pm
Dr Anwesha Ghosh	10:30 am - 12:30 pm

← Appointment System

Sort by: Doctor ▾

Dr Rishi Banerjee

11/03/2004

PEDIATRICS UNLIMITED
1000 University Drive
Wellington, NM 88230

March 10, 2009
ne: Kevin Zadnick

July 28, 09
NKDA
16 pounds

Ferrous Sulfate 4 mL PO TID

Dispense one bottle

Refills: 4
Dr. Mc

Dr Rishi Banerjee

11/03/2004

PEDIATRICS UNLIMITED
1000 University Drive
Wellington, NM 88230

March 10, 2009
ne: Kevin Zadnick

July 28, 09
NKDA
16 pounds

Ferrous Sulfate 4 mL PO TID

Dispense one bottle

Refills: 4
Dr. Mc

Uploaded Prescription

Doctors Routine

Booking appointment for



Dr Anwesha Ghosh
psychiatrist

Morning shift:
10:30 am - 12:30 pm

Booking Charge: Rs 300

Name of Patient

Age

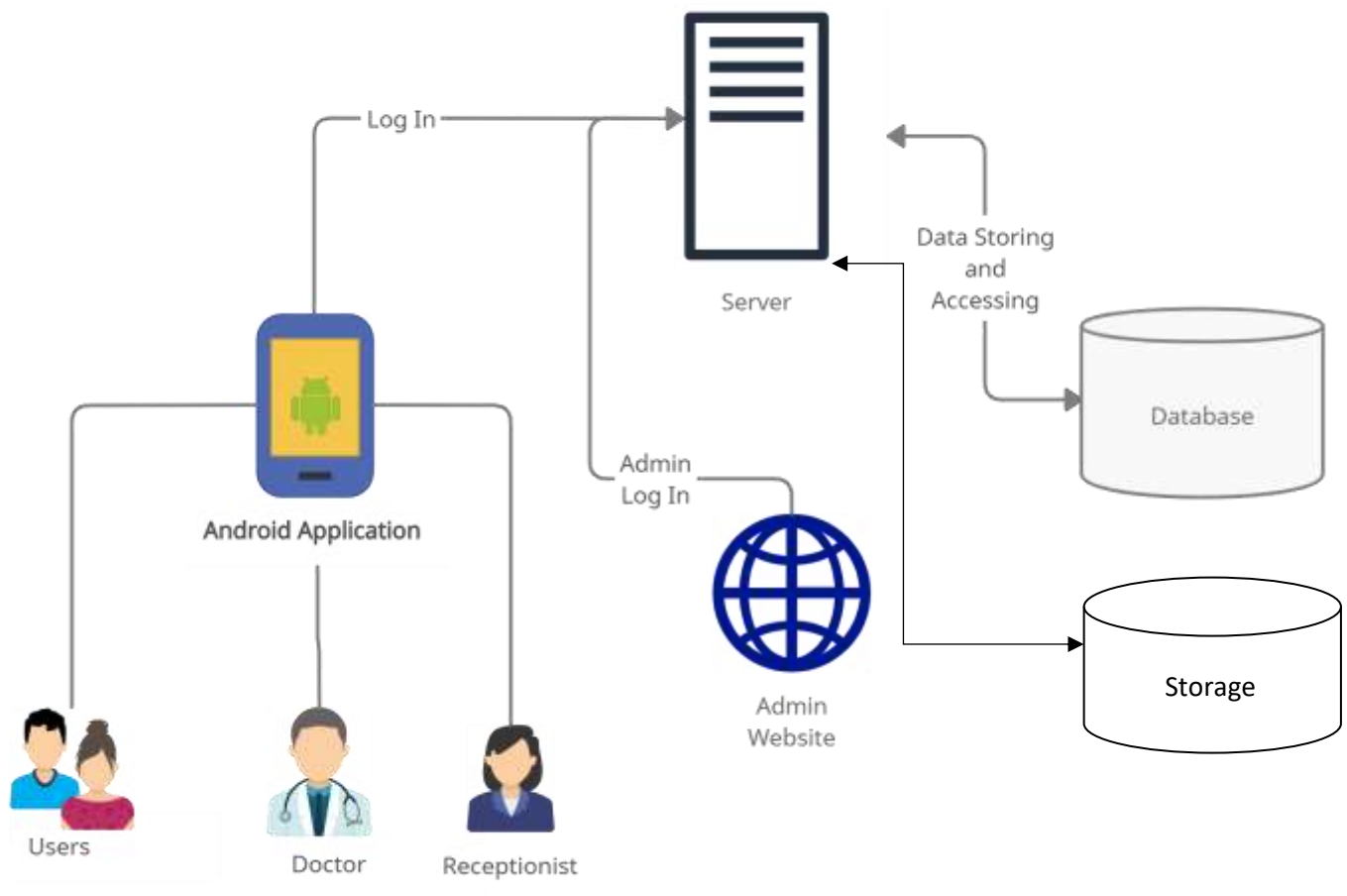
Sex

Primary Symptom

Book Now

Booking Appointment

4.1.1.1 System Interface:



- **Android Device:** Anyone with an android device running on android 5 and above can install and run this application. The application will be of at most 20 MB and compatible with devices more or equal to 500 MB of RAM. Maximum of extra 500 MB storage can be required.
- **Server:** Since the application is very light, **1 GB** RAM and one CPU is enough to run the whole application data for a metropolitan area.
- **Storage:** This application is going to store patients' records, previous prescriptions and doctor details, so it can need approximately 2 TB of storage and database initially.

4.1.1.2 Database:

The application is going to store all patient details, manage queue, Login details, doctor details, announcements, etc. So, to maintain the concurrency of data and to improve the performance of the whole server, we are going to use Postgres database. It can need approx. of 5 GB of database space.

4.1.2 Development Stage:

- The application will need approximately 1.5 months to build, where UI design will need 5 days, 10 days are required to build front end designs using XML and rest part and the backend will need 30 days.
- Total of 3 engineers are enough to built the application in a team. There roles will be based on frontend, backend and Algorithm required in the application for sorting and management of queue.

4.1.3 Testing Stage:

- To initially test the application, the three engineers are enough to do the task.
- After initial testing, an alfa version of the application will be given to 10-20 users.
- Based on the feedbacks and the user experiences, bugs will be fixed and more changes will be applied (if needed).

4.1.4 Deployment Stage:

- Once the testing is done, and the product is ready for deployment, it is released for customers to use.
- Now the application is ready to be published on Google Play store and backend will be hosted on AWS with S3 bucket for storing all the data.
- The size of the project determines the complexity of the deployment. The users are then provided with the training or documentation that will help them to operate the software.

- Again, a small round of testing is performed on production to ensure environmental issues or any impact of the new release.

4.1.5 Maintenance Stage:

The actual problem starts when the customer actually starts using the developed system and those needs to be solved from time to time. Maintenance is the last phase where the developed product is taken care of. According to the changing user end environment or technology, the software is updated timely.

4.2 Functional and Non-Functional Requirements

4.2.1 User Case Scenario

4.2.1.1 Login Frame

Purpose	User logs in to system using existing profile.
User	A user with an existing profile.
Input Data	Profile username and password.
Output Data	Corresponding page data.
Pre-Condition	User is not logged in to a profile, input profile exists in data base, user password matches profile
Post-Condition	User's device will save the logs of the user
Basic Flow	Redirect to the home page (if already logged)
Alternative Flows	Invalid password, invalid username, or mismatched username and password redirect to error message.

4.2.1.2 Booking Function

Purpose	To book an appointment
User	A user with an existing profile.
Input Data	Patient name, age, sex, primary problem
Output Data	Status in queue and estimated appointment time.
Pre-Condition	User needs to be logged in.
Post-Condition	User will able to book an appointment.
Basic Flow	Redirect to the Payment gateway and then to the home page.

4.2.1.3 Receptionist function

Purpose	To manage the patients in a queue based on priority for emergency cases.
User	Receptionist of the clinic.
Input Data	Receptionist are not allowed to modify the queue
Output Data	Generate Queue
Pre-Condition	User need to have a receptionist permission
Post-Condition	NA
Basic Flow	Redirect to Queue List itself

4.2.1.3 Prescription and Patient details.

Purpose	To save the prescriptions and details of the patients.
User	Doctor and his/her assistant.
Input Data	Prescription and patient details.
Output Data	Prescription details (if any)
Pre-Condition	User need to have doctor permission.
Post-Condition	NA
Basic Flow	NA