

Project Design Doc:

Team-24: Madhusudan

Rhythm

Khushi

Pranav

Introduction:

A software systems project design document is a document that outlines the technical and functional specifications of a software project. Its purpose is to provide a clear and concise description of the system design, architecture, and requirements to all stakeholders involved in the project. It serves as a roadmap for the development team and ensures that everyone understands the project requirements and expectations. It is an essential tool for ensuring the successful completion of a software project.

System Overview:

The doctor appointments app is a software system that allows patients to book and manage doctor appointments online. The app includes video call and chat features, providing patients with the option to have virtual consultations with doctors. The app also displays doctor and patient details, including their contact information and appointment history, and enables patients to check the hospital's open hours.

In a real-life scenario, a patient could use the app to book an appointment with their doctor, view their medical history and previous appointments, and join a video call for a virtual consultation. Alternatively, the patient could use the chat feature to communicate with the doctor directly, asking questions or requesting a prescription refill. The app's ability to display the hospital's open hours can also save the patient time and hassle by avoiding arriving at the hospital when it's closed.

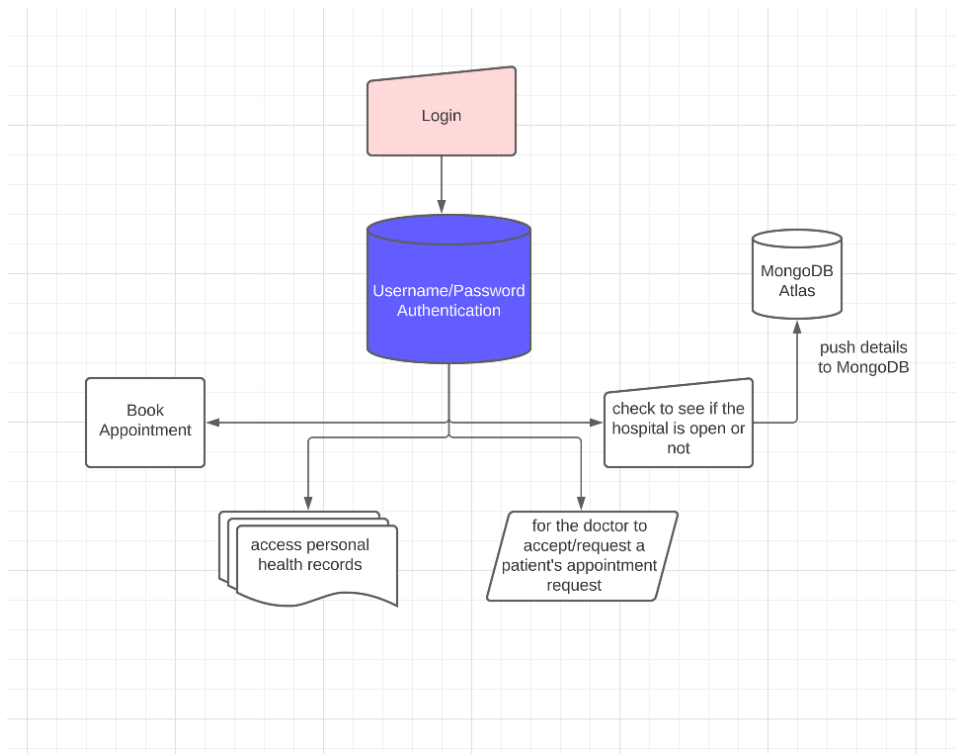
The app can solve several problems for patients, including the inconvenience of waiting in long lines to book an appointment, difficulty finding a doctor that fits their schedule, and the need to travel to the hospital for non-emergency consultations. By providing virtual consultations, the app can also improve access to healthcare for patients who live in remote areas or have limited mobility.

For doctors, the app can simplify appointment management, providing an easy-to-use interface for booking and rescheduling appointments, communicating with patients, and viewing patient medical records. Overall, the doctor appointments app is a useful tool that can improve patient access to healthcare and streamline appointment management for doctors.

Design Overview:

Architectural Design:

We're building a mobile application using React-Native, NodeJS and MongoDB. React Native is an open-source JavaScript framework, designed for **building apps on multiple platforms like iOS, Android, and also web applications**, utilizing the very same code base. It is based on React, and it brings all its glory to mobile app development. Node.js is a popular programming environment that can be used for **building high-scale applications that need to support multiple concurrent requests**. Single-threaded non-blocking I/O makes it an excellent choice for both real-time and data streaming applications, too. MongoDB is a document database used **to build highly available and scalable internet applications**. With its flexible schema approach, it's popular with development teams using agile methodologies.



System Interfaces:

User Interface:



Parijatham

The Gynaecology Clinics

Login



Welcome back.

Email

Password

[Forgot your password?](#)

Login



Hello Doctor!

Welcome

Hospital: Not Found

[Logout](#)

Your Appointments For Today

All Appointments

Dr. Doctor

Patient
26th April
6:30

Pending

[Logout](#)

Your Current Patients

Patient

Chat

Patient Request

Back

hi patient
7:27 PM

Apr 20, 2023

wow it is working
7:04 PM

hello
10:06 PM

Apr 24, 2023

hi
5:16 PM

here you can see the real time chat working
5:17 PM

so this is what we did for chat till now
5:17 PM

Now I would like to show how the appointments booking is done with our app
5:17 PM

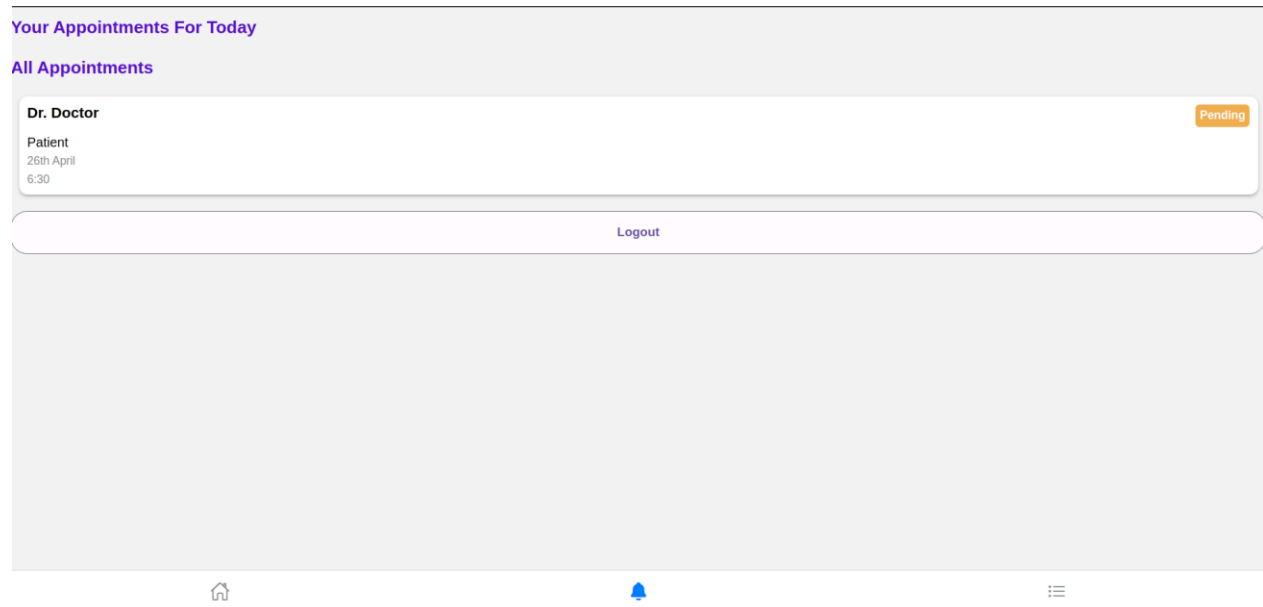
you could see the appointments on the page
5:19 PM

the real time chat
5:26 PM

hospital status is shown here
5:27 PM

Thank you

Type your message here...



APIs:

We are implementing REST API for this project

The APIs we used in this project are:

- 1) <https://localhost:3000/api/patients>
 - a. We use these apis to to post and receive data related to patients.
- 2) <https://localhost:3000/api/doctors>
 - a. We use these apis to to post and receive data related to doctors.
- 3) <https://localhost:3000/api/chats>
 - a. We use these apis to to post and receive data related to chats.
- 4) <https://localhost:3000/api/appointments>
 - a. We use these apis to to post and receive data related to appointments.
- 5) <https://localhost:3000/api/login>

We use 5th API to handle login.

Model:

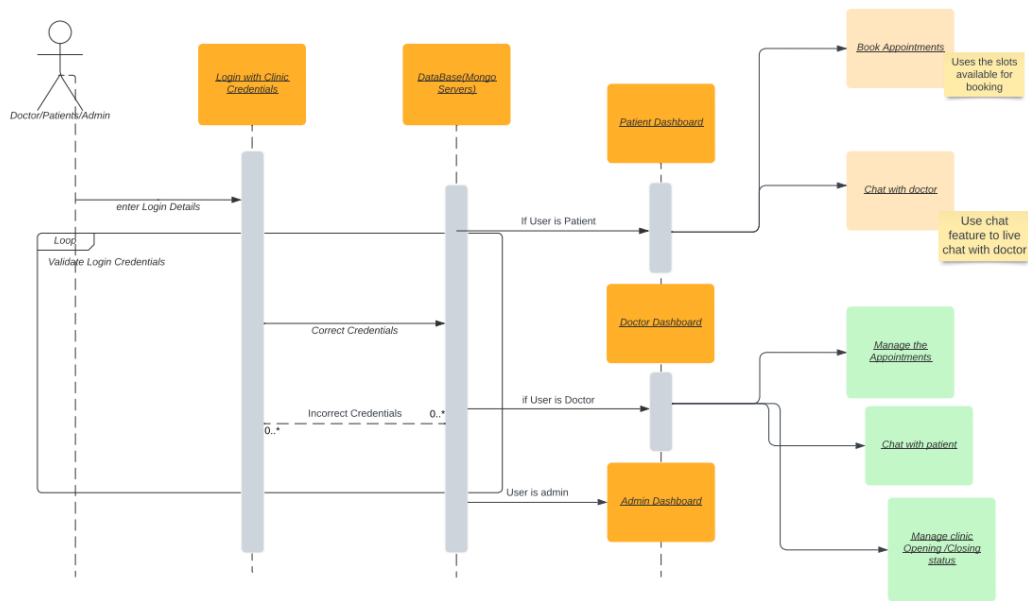
Doctor	<p>Doctor State</p> <ul style="list-style-type: none"> The class is responsible for storing doctor's details (email ID and the password) <p>Class behaviour</p> <ul style="list-style-type: none"> The class, in real world, has the following methods: <ul style="list-style-type: none"> login(): authenticates a doctor's login
Patient	<p>Patient State</p> <p>The class is responsible for storing patient's details (email ID and the password)</p> <p>Class behaviour</p> <ul style="list-style-type: none"> The class, in real world, has the following methods: <ul style="list-style-type: none"> login(): authenticates a patient's login
User	<p>Class behaviour</p> <ul style="list-style-type: none"> The class, in real world, has the following methods: <ul style="list-style-type: none"> register(): user registers as patient or doctor

Sequence Diagrams:

Use Case 1:Patient Booking an appointment

Use Case 2:Chat with Doctor

Use Case 3:Patient Viewing Hospital Opening/ Closing times



Design Rationale:

- 1) We need to improve our User Interface. We will be making it cleaner and make the navigation bar look much better so that our app feels like it has quality.
- 2) We still need to figure out how to implement the video call feature.
- 3) We still need to figure out a way to deploy which is considered as the most important part of our project.