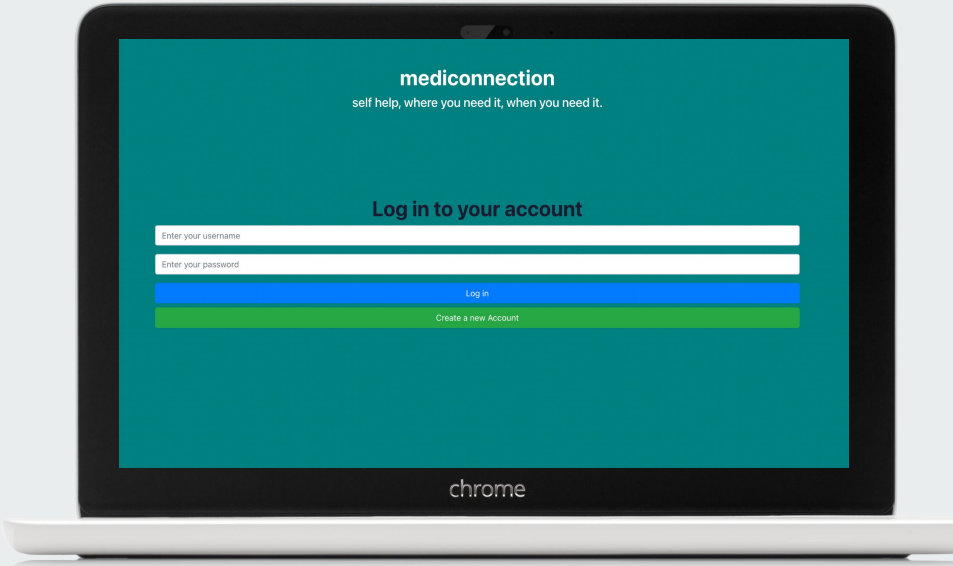
MediConnection

Problem statement and
solution proposal



Outline

Problem Statement

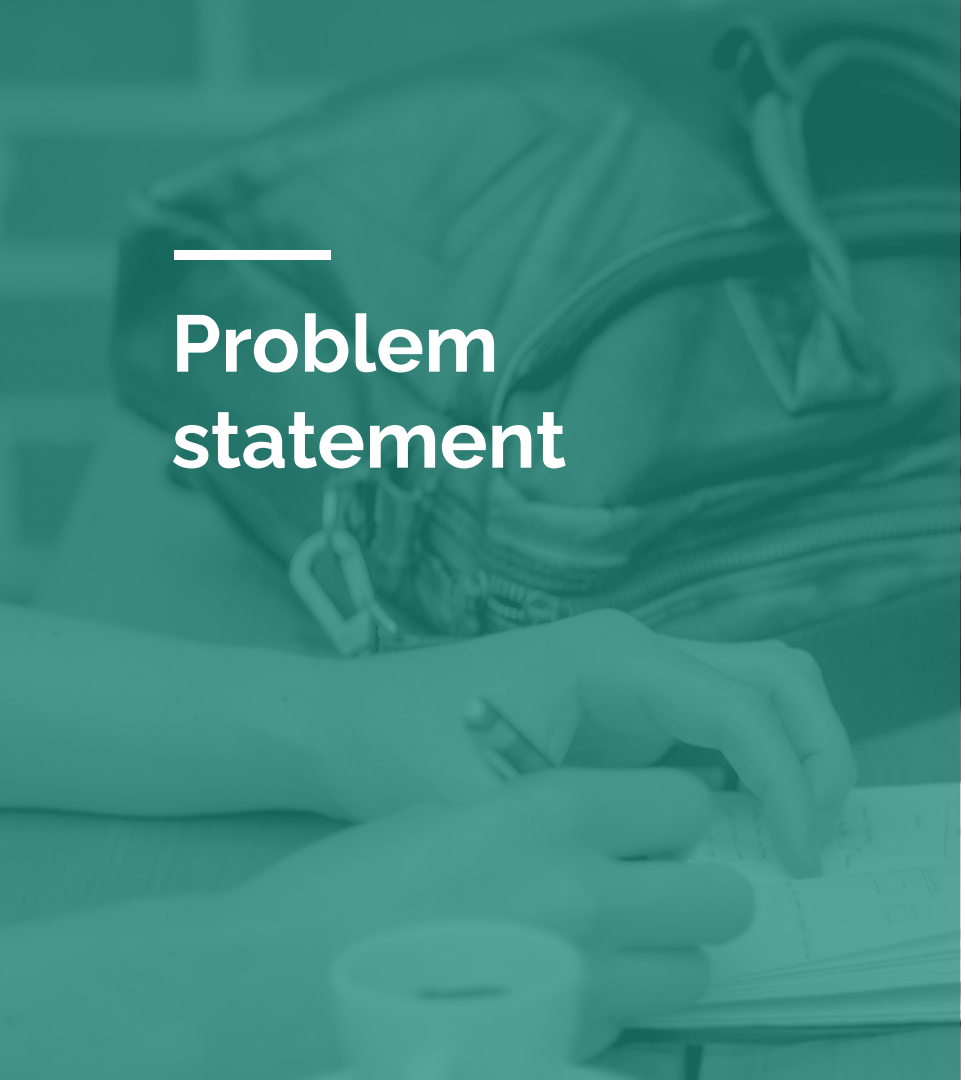
Motivation

Solution

Implementation

Demo

The Problem



Problem statement

There is a need for a solution that allows patients and healthcare providers to connect remotely, enabling easier access to medical advice, improved communication, recommendations, medication tracking, and prescriptions without in-person visits.

The Motivation

Motivation

To enhance the efficiency and convenience of healthcare services by enabling remote access to medical care and improved patient care through a mobile application

Solution Proposal

Information Gathering

1. Interviews:

   one on one interviews with students and healthcare professionals

2. Questionnaires:

   list of questions designed in google forms

Questionnaire

Section 1: General Information
1.1 Please provide your name, title, and organization:
1.2 What is the size of your organization (number of employees)?
1.3 What is the primary function of your organization within the healthcare industry?
Section 2: Communication
2.1 What communication channels do you currently use to interact with patients (e.g., phone, email, text messaging, mobile app)?
2.2 Are there any specific communication features you would like to see in the software (e.g., secure messaging, video conferencing)?
2.3 What types of communication do you want to facilitate between healthcare providers and patients (e.g., appointment reminders, test results, treatment updates)?
2.4 What types of communication do you want to facilitate between healthcare providers (e.g., consultation, referrals, case management)?
Section 3: Appointment Scheduling
3.1 What are the key features you require for appointment scheduling (e.g., online booking, automated reminders, appointment rescheduling)?
3.2 Do you require integration with any existing scheduling or calendar systems?
3.3 What are your preferred methods for patients to request, change, or cancel appointments?
3.4 Do you have any specific requirements for managing appointment waitlists or prioritizing urgent cases?
Section 4: Medication Tracking
4.1 What specific features do you require for medication tracking (e.g., prescription management, refill reminders, dosage tracking)?
4.2 Do you require integration with any existing electronic prescription systems or pharmacy management systems?
4.3 What types of medication-related alerts or notifications do you want to send to patients and/or healthcare providers?
4.4 How do you want patients to report medication adherence or any side effects experienced?

To achieve this, the following system architecture/models can be implemented:

1. Client-Server Architecture
2. Real-time Communication
3. Security and Privacy

Why it’s better than existing solutions?

1. Convenience
2. Quick Appointments
3. Real-time Communication

First, why the MERN stack?

Deciding choice of development environment

-Framework Choices

- The MERN stack is a popular combination of technologies used for building web applications.
- Why our choice?
- It allowed us to use Full-Stack JavaScript from frontend to backend.
- This made us more efficient by reducing our learning curves since we all had some experience using JavaScript.
- The MERN stack has a large and active community of developers who contribute to open-source projects and provide support through tutorials, and other resources.

CONSTRUCTION

-Viewing through the lens of major use cases

Tackling Communication
The Chat System

-Major Use Case

-Priority: High

-Primary Stakeholder: Patient and Doctor

-Secondary Stakeholder: Patient and Doctor

Real-time Physician/Patient Chat

- Chat System works like regular messaging apps like whatsApp, albeit, trimmed to a patient and physician with an appointment between them.
- Relevant database Schemas used are the Chat and Message databases.
- A chat object stores the idea of what a chat is, and every object has a chat_Id.
- A chat has a bunch of messages that are linked to it through its id.
- Once a route to a chat is triggered, we search the message database for all messages linked to that chat, and display them in order of their timestamps and who sent what.
- History saved for future logins.

Handling Searches
The Search functionality

-**Major Use Case**

-**Priority:** High

-**Primary Stakeholder:** Patient and Doctor

-**Secondary Stakeholder:** Patient and Doctor

Patient and Physician Search

- Searching was the trickiest and one of the most difficult parts of the project.
- We aimed to make the search responsive (we wanted entries to be queried as soon as they changed).
- Also we wanted to be able to get search suggestion so far as the entry is an attribute of what is being searched.
- So what we did here was to accept the entry as a placeholder for the string attributes of say a Physician.
- So in running through the database, we do not only check the entry against usernames, but also other attributes like specialization, etc.
- This made the search quite slower and we hope to make some changes in the future.

Limiting in-person Scheduling
The Appointment booking system.

-**Major Use Case**

-**Priority:** High

-**Primary Stakeholder:** Patient

-**Secondary Stakeholder:** Doctor

Appointment Booking Flow

- Patients request for appointments in the system.
- A patient first searches for a physician in the hospital system and selects a service offered by the physician.
- The patient is then prompted to enter the details for the appointment they wish to book: date, time, etc.
- After this, the system uses the entered details to create an appointment object which is later sent to the Physician for approval/rejection.
- After an appointment request has been sent, the can view the list of requests that have been sent and accept or reject them.
- Acceptance of request opens up chat and medication tabs for the parties.
- Main hurdle was data storage/fetch.

Don’t Forget Your medications
Access Medication

-**Major Use Case**

-**Priority:** High

-**Primary Stakeholder:** Patient

-**Secondary Stakeholder:** Doctor

Access Medication Flow

- Patient clicks on"Appointments" on their dashboard
- Systems displays all appointments associated to the user
- Patient clicks on a particular appointment
- Patient clicks on Medication tab
- System checks if there are medications prescribed for
- Depending on the the previous situation, the system either shows a empty catalogue of medications or shows list of medications and their dosage

Don’t Forget Your medications
Medication Tracking

-The failed Use Case

Medication Alert System

- One main goal of ours was to integrate a calendar API once a medication had been added by a doctor for a patient in an appointment.
- The dosage and medication names were supposed to be taken and used in tandem with the daily dosage frequency to generate a calendar file (ics) which could be downloaded by a patient to keep themselves alerted as to when to take their medications and how much quantity they should take.
- Unfortunately, we did not manage to implement the .ics file generation after being able to add medications.

TESTING

Questions?