

# Android Project: Healthcare Appointment Management System (HAMS):

SEG2105A - Introduction to Software Engineering

Fall 2023

School of Electrical Engineering and Computer Science  
University of Ottawa

Professor: Dr. Hussein Al Osman

Group 39

Steven Yang (#300250780)

Daniel Guo (#300291390)

Balpreet Singh (#300322603)

Leon Matthews

Tarek Rehouli

Dipak Chinnasamy Selvam

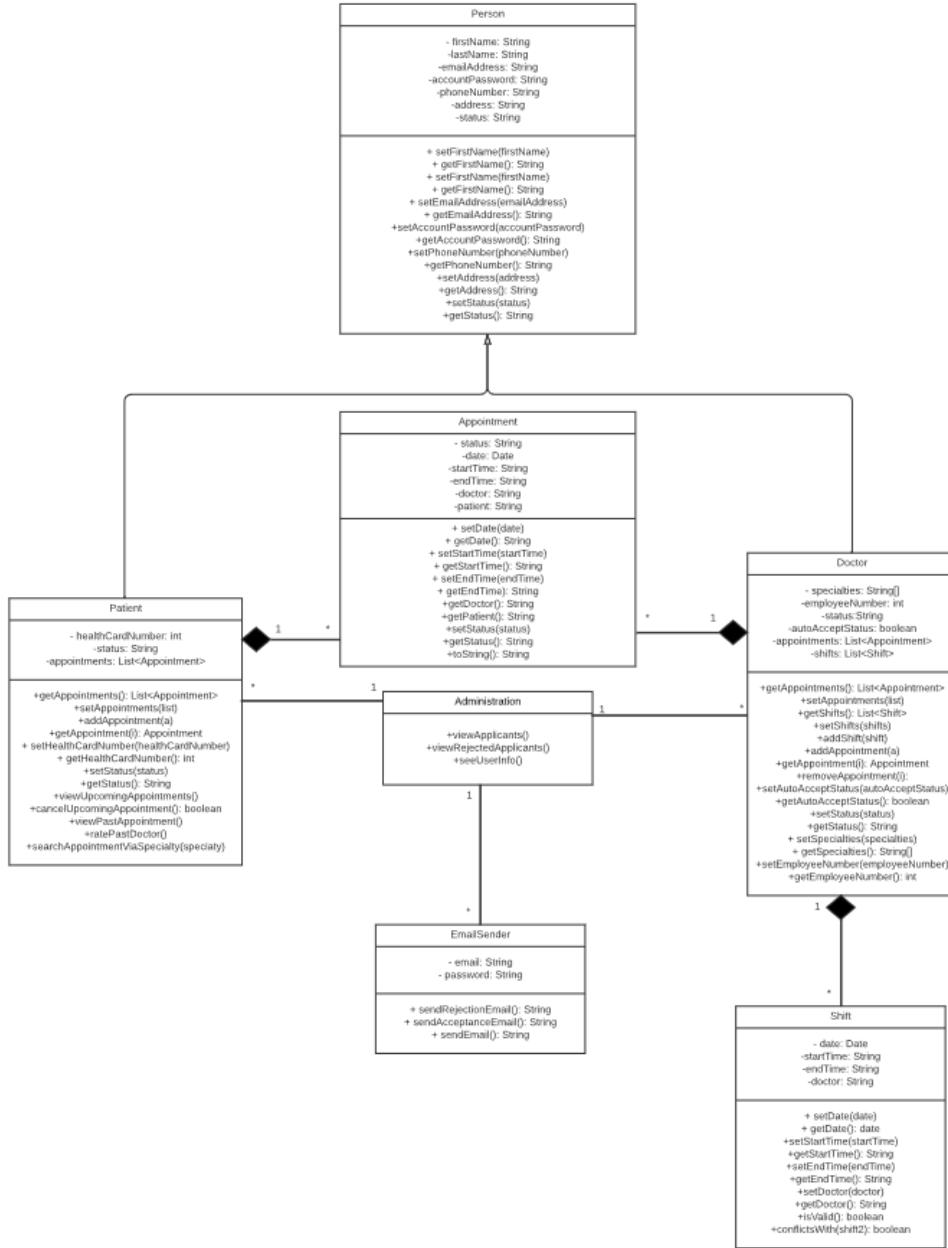
Submission date : 2023-12-04

## **Introduction:**

This project is a collaborative group effort. In our group, we have 6 members contributing to this project. In this project, we are required to develop a healthcare Appointment Management System (HAMS) for a telehealth clinic. This is a mobile application developed to help patients, doctors and administration streamline the process of healthcare appointment scheduling and management. This application is easy to use and can significantly help doctors, patients and administrators schedule and manage healthcare appointments. Android studios, firebase and Git are all softwares we used to help us build this application. The front end of the application is mainly built using android studios while the back end is developed using Java. The project consists of four deliverables and a final report (included in deliverable 4).

Overall, this report will include the final UML diagram, a table specifying individual contributions, screenshots of the application and lessons learned.

## **Final UML Diagram:**



## Individual Contributions:

	Balpreet	Daniel	Steven	Leon	Terek	Dipak
Deliverable 1	-log in, log off -integration -firebase -patient class	-Home screen -Person class -uml diagram	-Patient and doctor registration forms	-log in		

Deliverable 2	-create admin page -patient login	-update classes, add status attribute -uml diagram	-doctor patient login changes			
Deliverable 3	-doctor view patient info -doctor approval -doctor approve rejected patient -doctor view list of past appointment -view list of shifts	-auto accept setting -update doctor and patient class -uml diagram	-update doctor and patient class -doctor add shift -doctor delete existing shift			
Deliverable 4	-final report -patient view past, upcoming, canceled appointments -cancellation possibility. -The Doctor cannot delete a shift if it is associated with one or more Patient appointments	-uml diagram -doctor rating -final report	- Search for appointments by specialty -A booked time slot is not listed when Patients look for appointments			
Extra						

## Screenshots of the Application:

### Lessons Learned:

Overall, throughout this project, every member of this project has learned many valuable lessons.

#### 1. Importance of communication

First and foremost, the entire group underestimated the importance of communication during the group project. At first, we scheduled a meeting once every two weeks. However, we realized that it was not sufficient to meet once every two weeks and communicate via text messages in between the meetings. Many miscommunications and misinterpretations occurred which ultimately hindered our app development process. This could have easily been avoided if meetings were scheduled more frequently. For example, many parts of the assignment used the java class Patient. The Patient class has an attribute called “status” which was supposed to be stored as a type String. However, we did not communicate effectively about this specific attribute as it was stored as a type boolean. As a result, whenever we needed to use this attribute for the application, we tried using this as a type String which caused issues. The issue could have been avoided if we communicated more often and if we were more specific about the implementation during the meeting.

## 2. Starting and asking for help early

Another problem we encountered was group members not asking for help early enough. For example, they would either be busy with midterms and assignments and they would save the deliverables until the last minute. This would result in other group members having to help them and finish the deliverables under the time constraint. Overall, it was difficult to balance all of the work we had; however, it is important to manage our time effectively and start as early as possible.

## 3. Taking initiative

Another lesson that we all learned was the importance of taking initiative. Whenever a group member needed help for a certain part, someone eventually had to help. After a few weeks into the project, although it was not obvious initially; we realized that we are all in the same boat and the work had to be done as a group. Unlike previous group projects, this one is very large and we cannot get away with simply everyone “doing their part”. Overall, taking initiative is a very valuable lesson learned and we believe that it will help us later on as well.

## 4. Constantly implementing testing

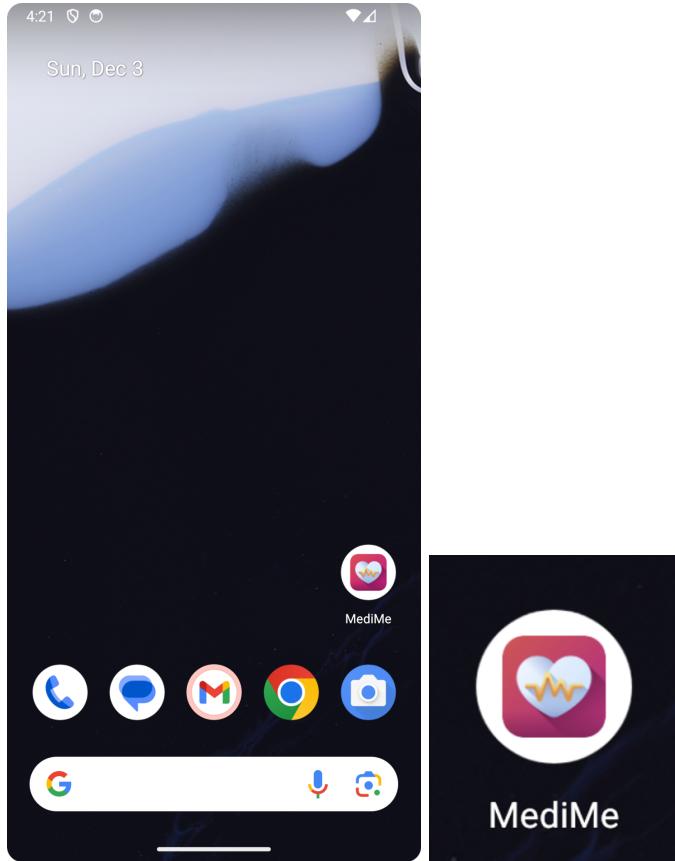
One important technical skill we learned during this group project was to constantly test the software during implementation. Even after every small additional implementation, it was important to run test cases. We ran the test cases manually for most of this assignment. At first, we only conducted test cases after the entire group had done their parts. However, the problem with that was it was hard to back track and find the bug. After the first deliverable, we decided to test the program after each additional feature implementation. This was important since writing programs as a group was different from writing programs alone. In a group, many of us are unfamiliar with ways other people write code. This results in many bugs when we merge everyone’s parts. Thus, it was important to reduce the number of bugs to frequently test the project.

##### 5. Documenting/logging every step and detail

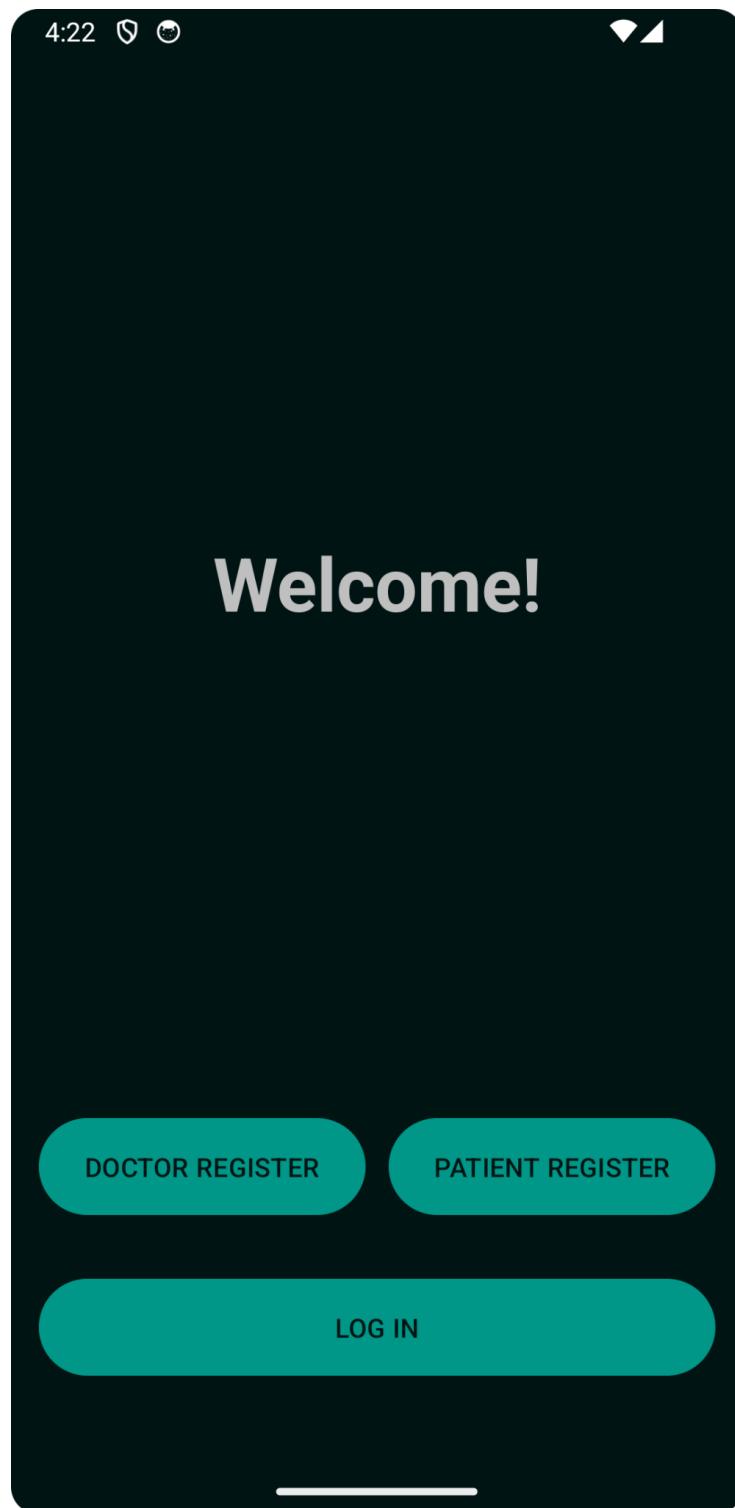
In this project, since it is a group of six, it was very easy to get lost on who does what. As a result, we created a log that tracks everybody's tasks. In addition, we added due dates to keep track of a timeline of what everyone has to do and what they have done. In addition, using a log was beneficial since we can check off every completed task. This can allow other group members to start the next part whenever a prerequisite task is completed. We also added an attendance list so if a group member could not attend a meeting due to any circumstances, someone else can provide a summary of what was discussed during that particular meeting.

Screenshots:

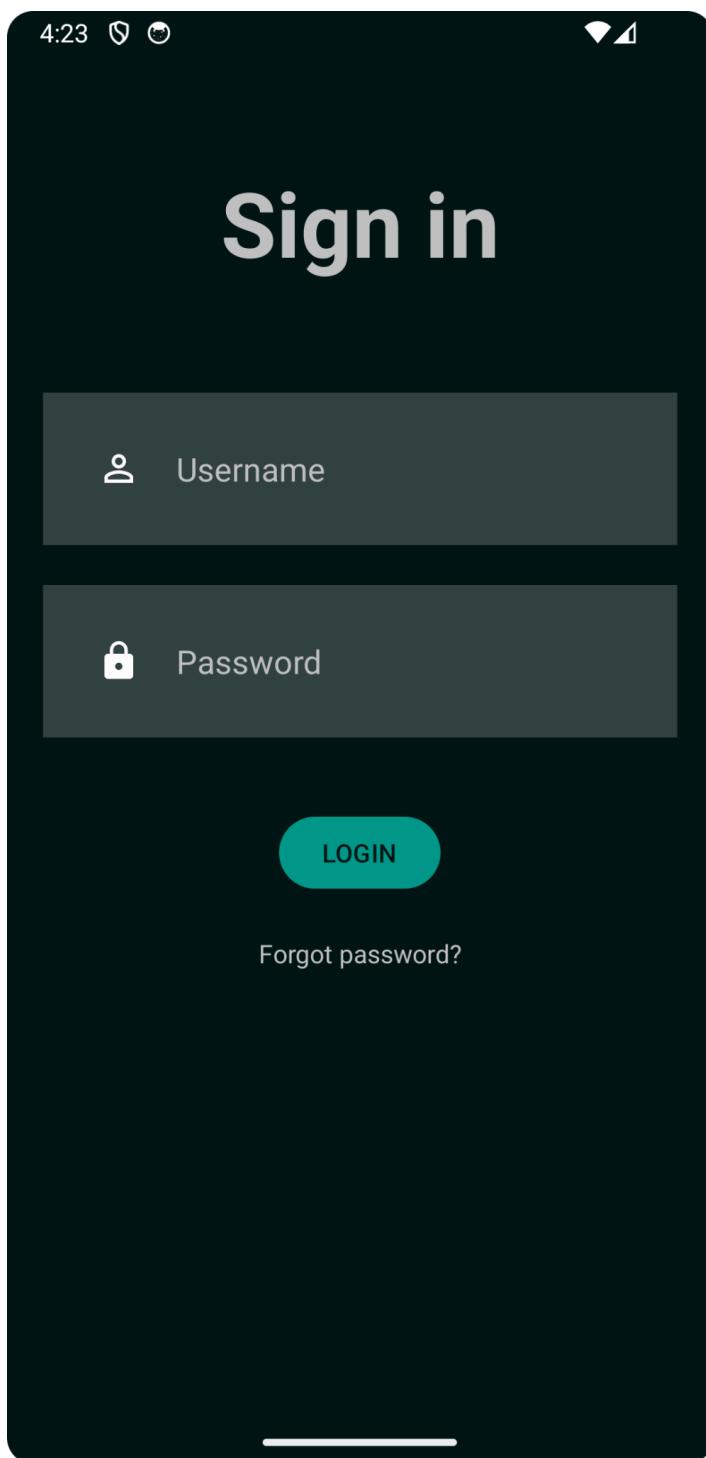
**App icon in home screen:**



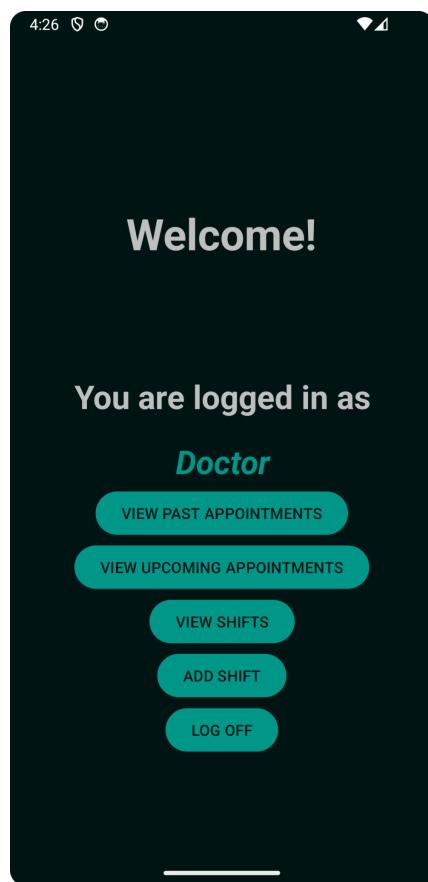
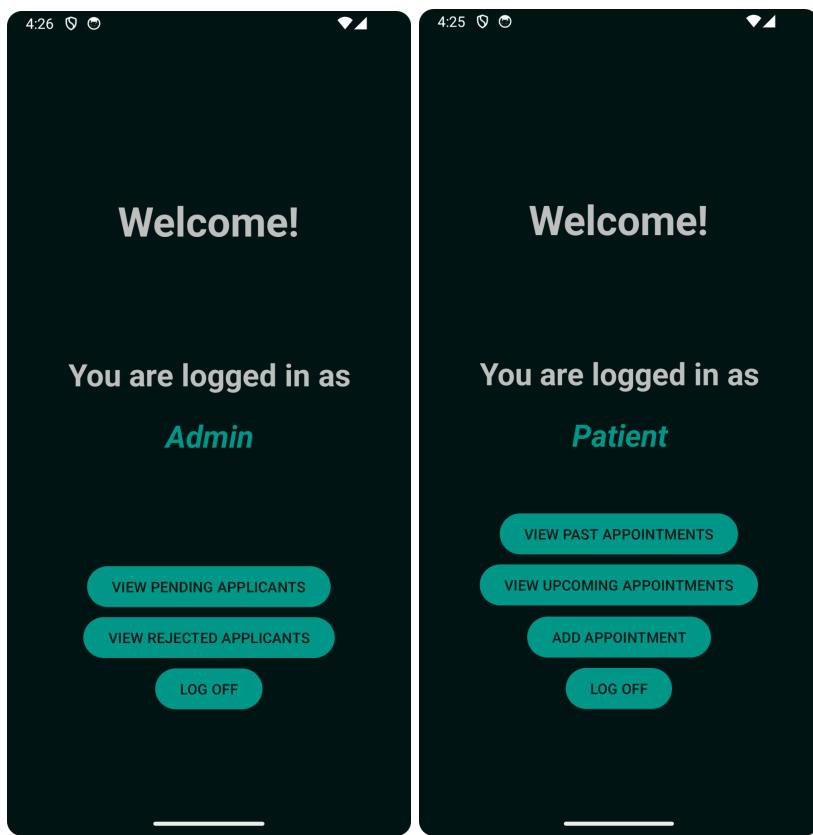
Home page:



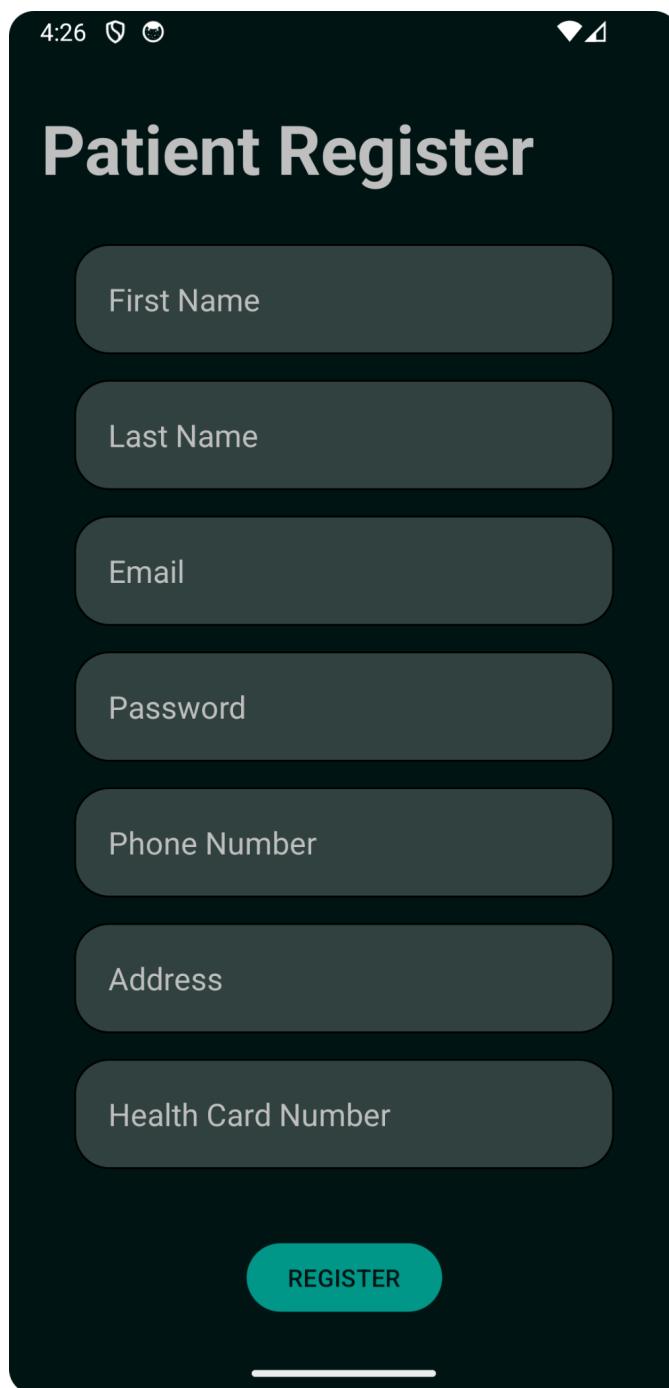
Login page:



Successful login page (as an Admin, Doctor and Patient):



## Patient Register page:



## Doctor Register page (with sample input):

The image displays two side-by-side screenshots of a mobile application titled "Doctor Register". Both screenshots show a dark-themed interface with rounded rectangular input fields.

**Left Screenshot (Initial State):**

- First Name: Placeholder text "First Name"
- Last Name: Placeholder text "Last Name"
- Email: Placeholder text "Email"
- Password: Placeholder text "Password"
- Phone Number: Placeholder text "Phone Number"
- Address: Placeholder text "Address"
- Employee Number: Placeholder text "Employee Number"
- Specialties: Placeholder text "Specialties"

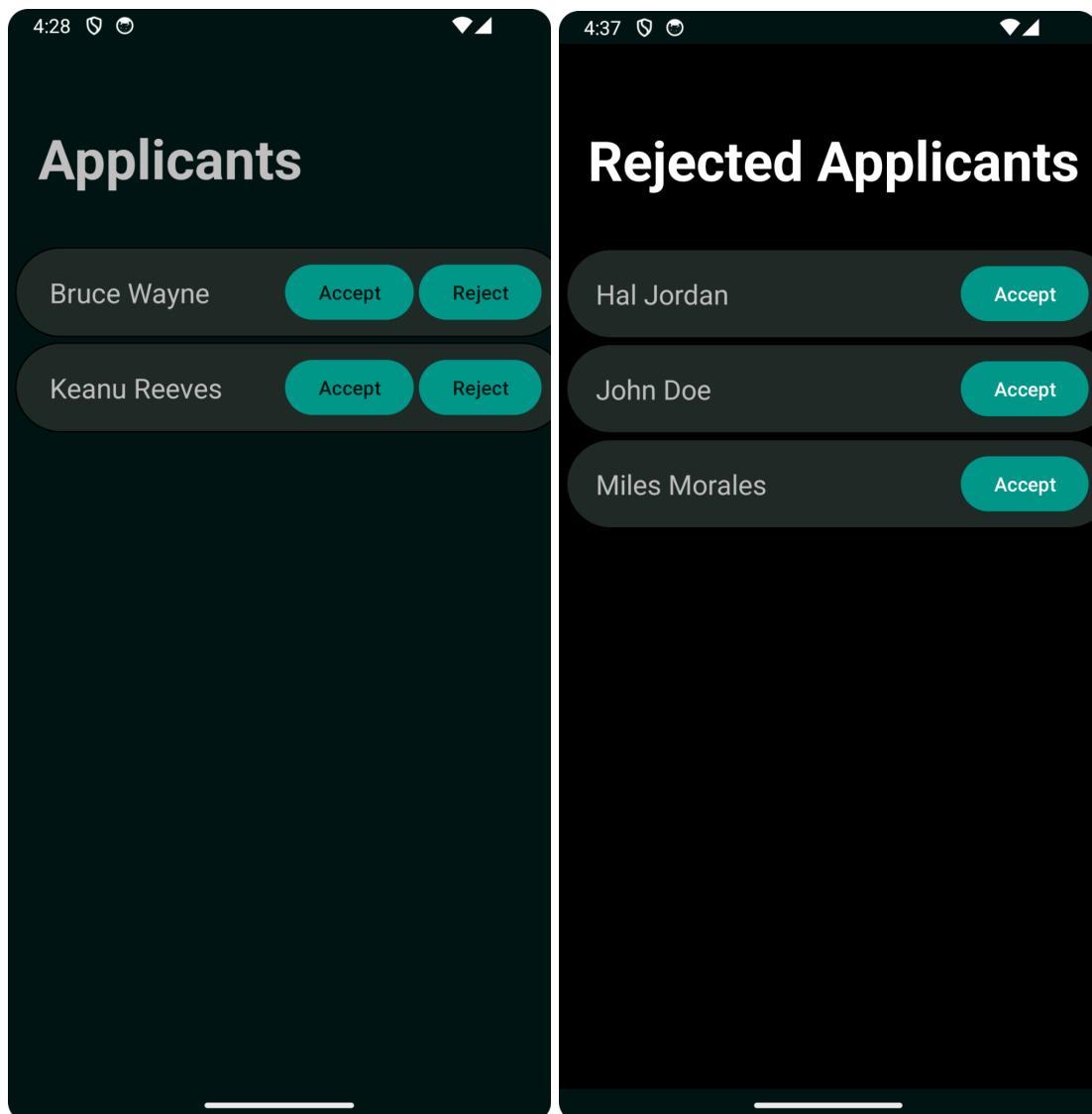
At the bottom, there is a "AutoAccept" toggle switch (disabled, grey) and a teal-colored "REGISTER" button.

**Right Screenshot (Sample Input Entered):**

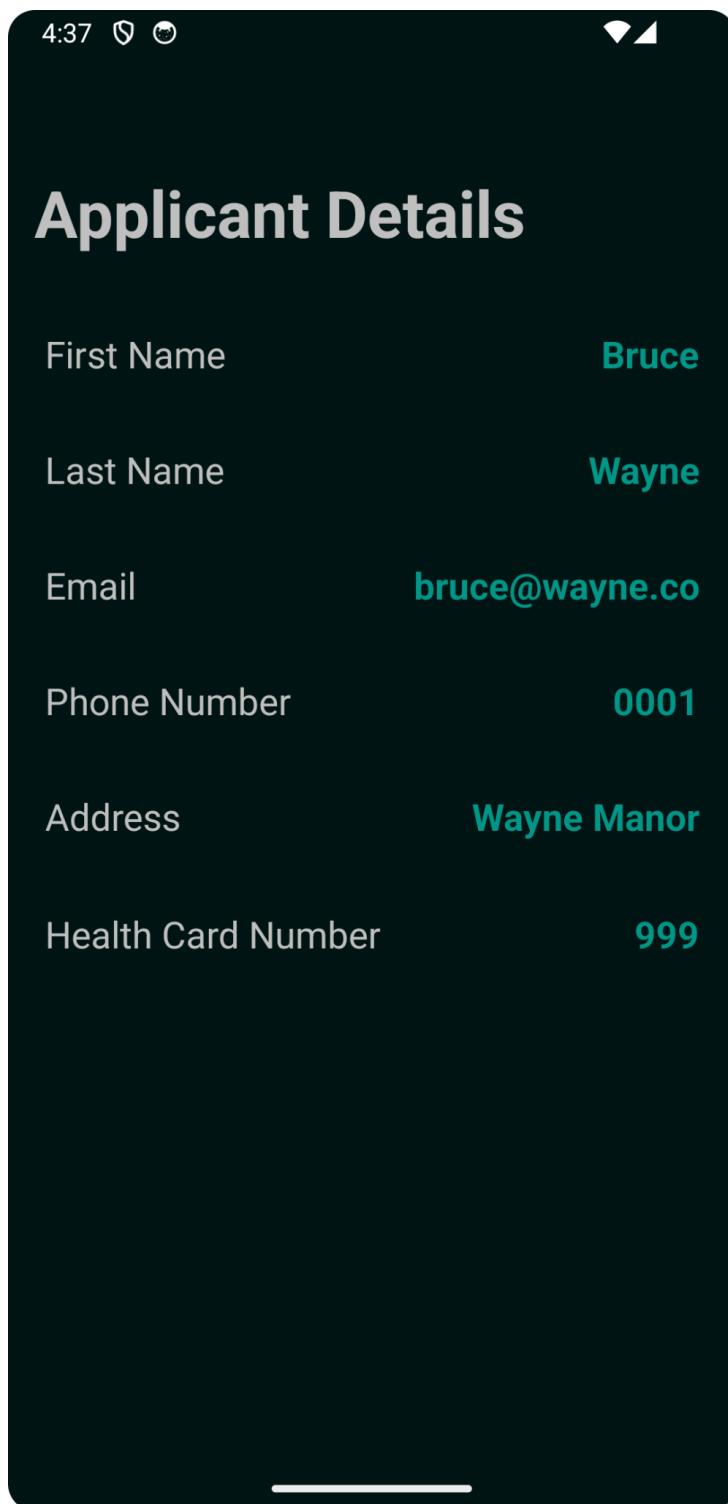
- First Name: "Hal"
- Last Name: "Jordan"
- Email: "hal@myspace.com"
- Password: Placeholder text "•"
- Phone Number: "456"
- Address: "Gotham City"
- Employee Number: "987"
- Specialties: "Arms,Legs|"

At the bottom, there is a "AutoAccept" toggle switch (enabled, teal) and a teal-colored "REGISTER" button.

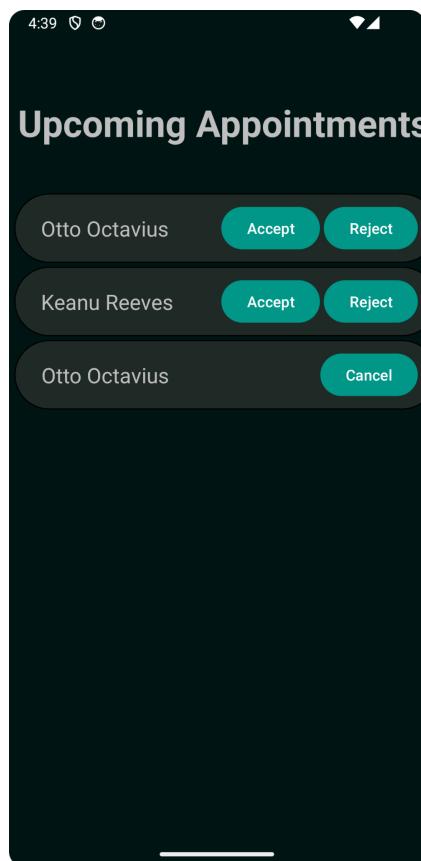
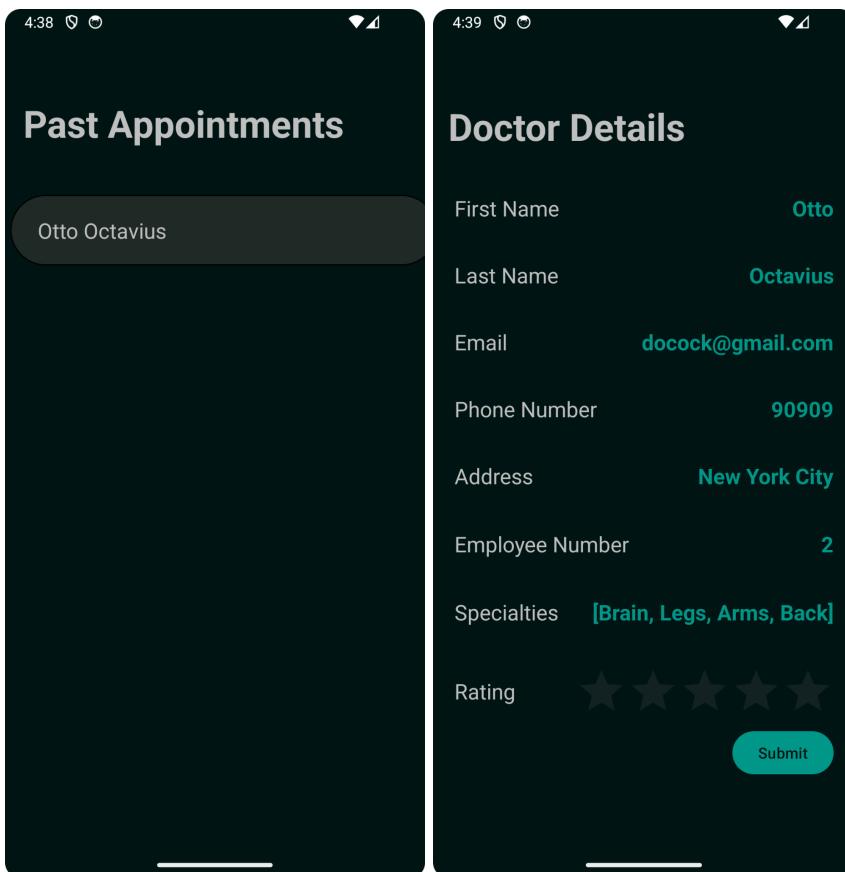
**View pending and rejected applicants (as an Admin):**



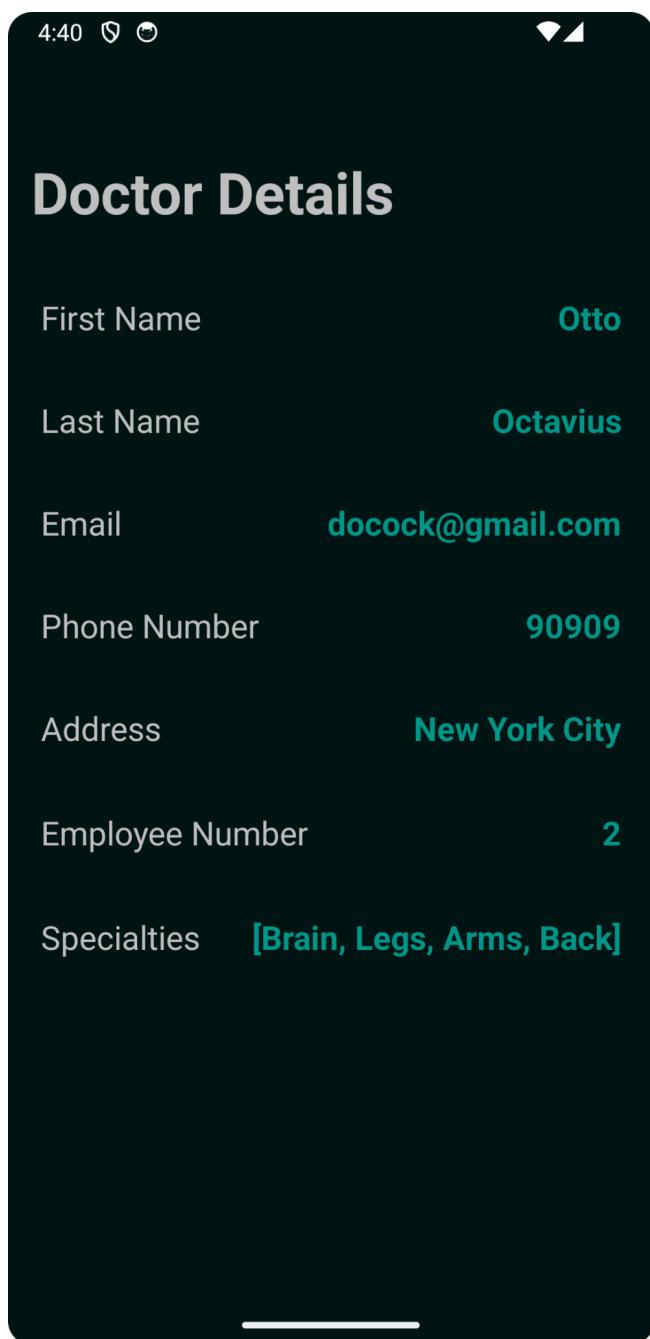
View Applicant Details page (as an Admin):



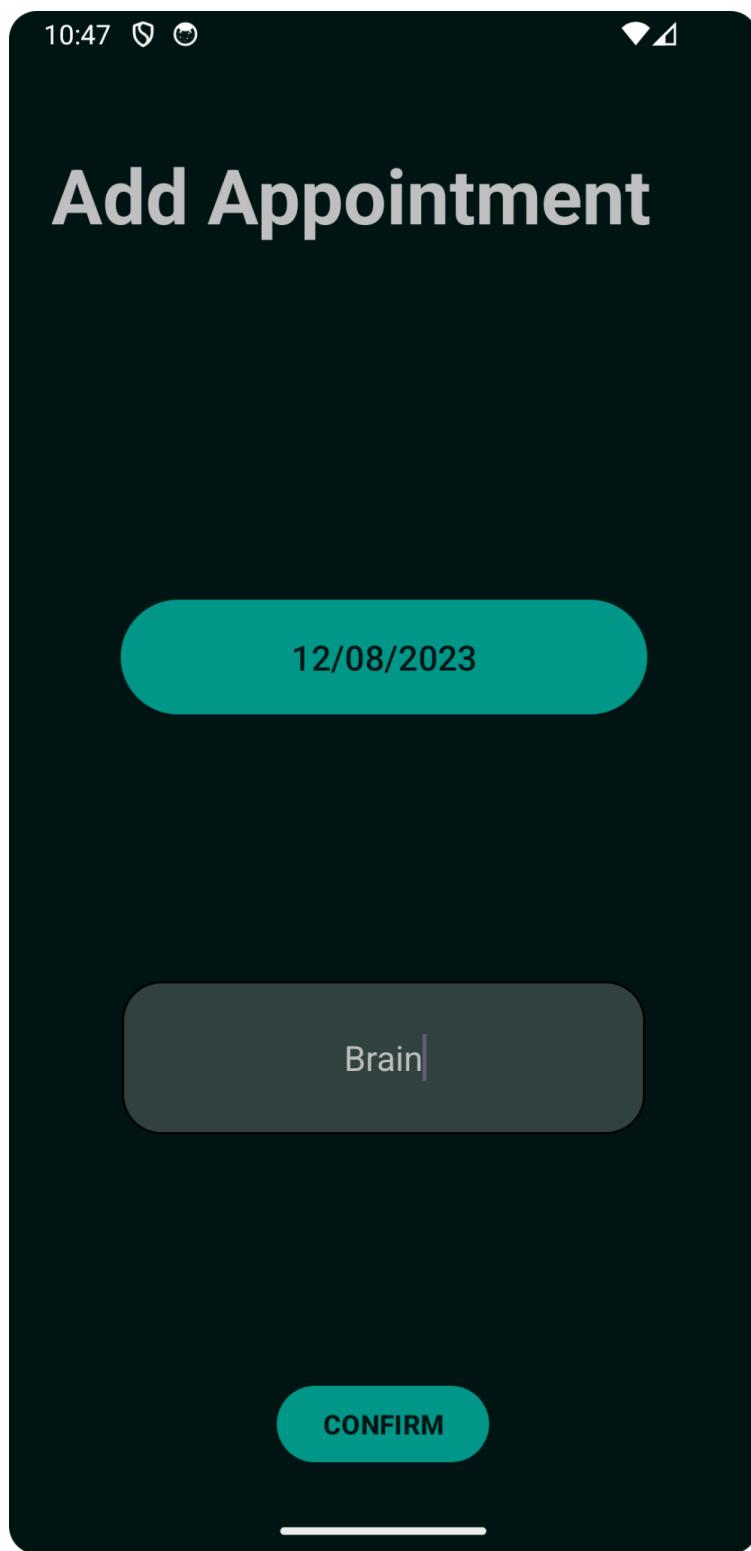
**View Past and Upcoming Appointments page (as a Patient, with ability to rate doctor):**



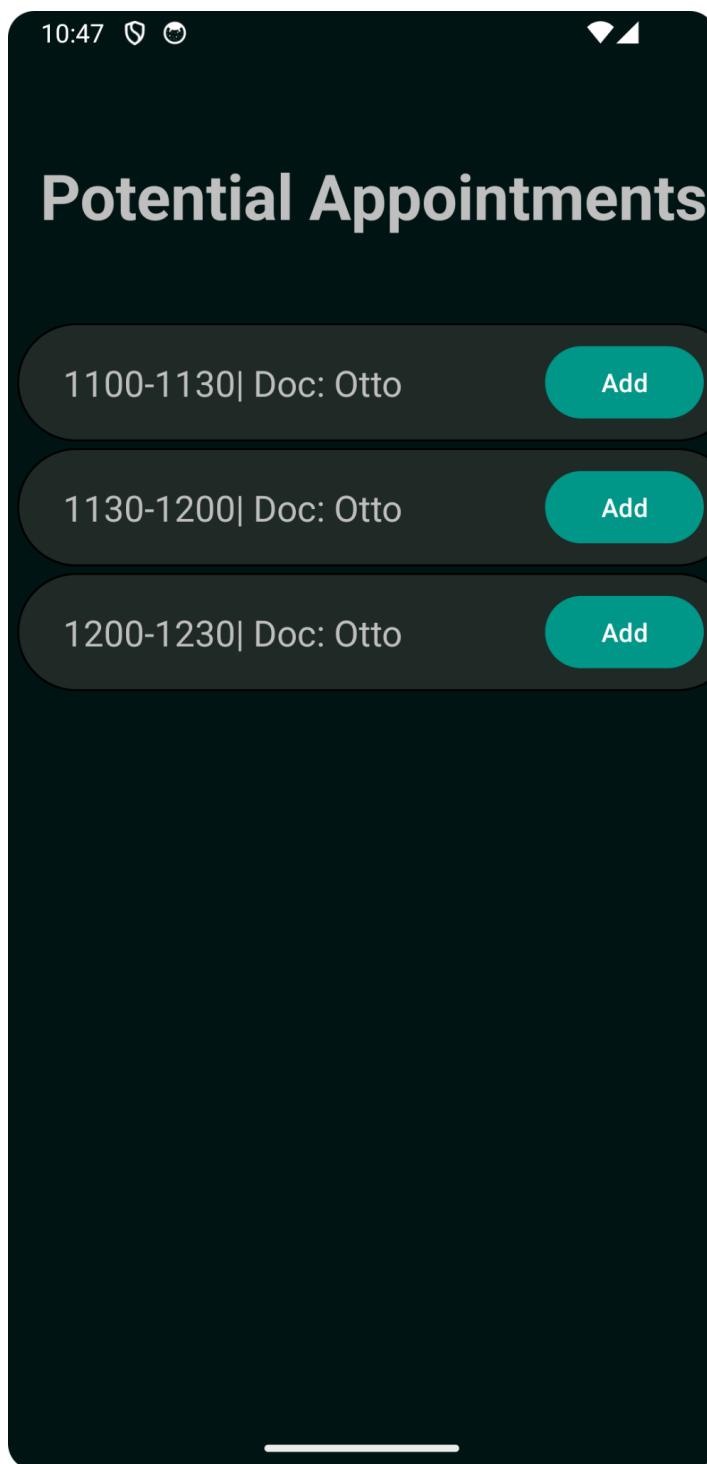
View Doctor Info page (as a Patient):



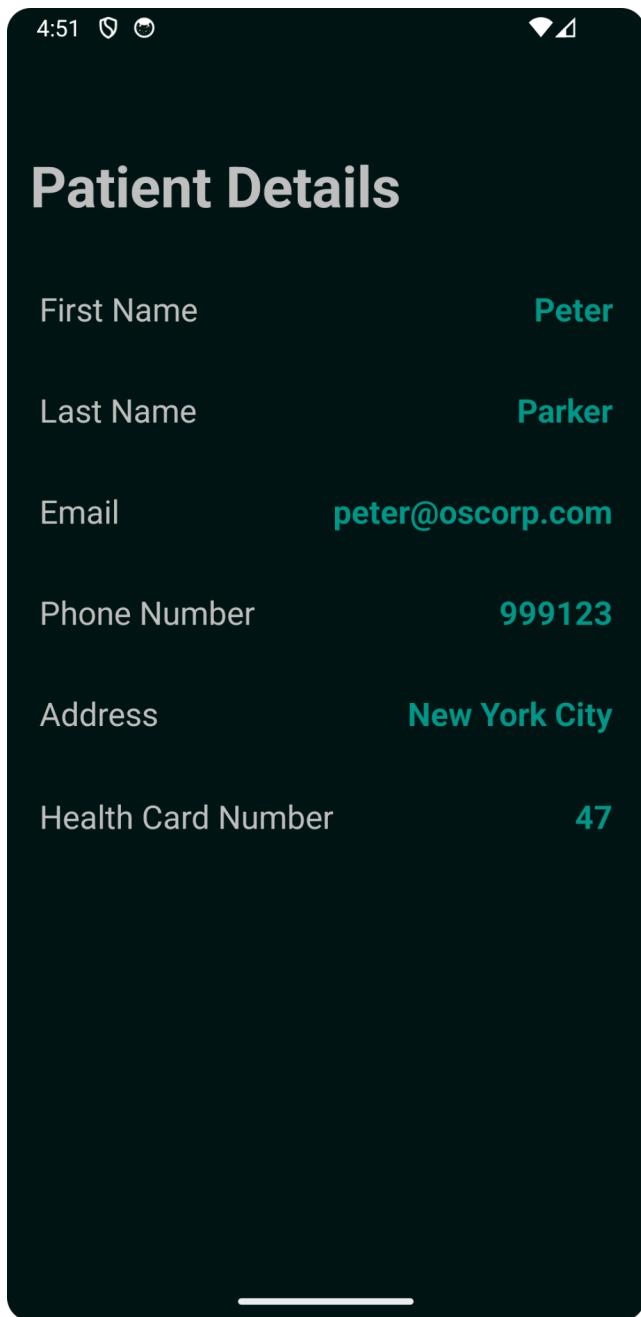
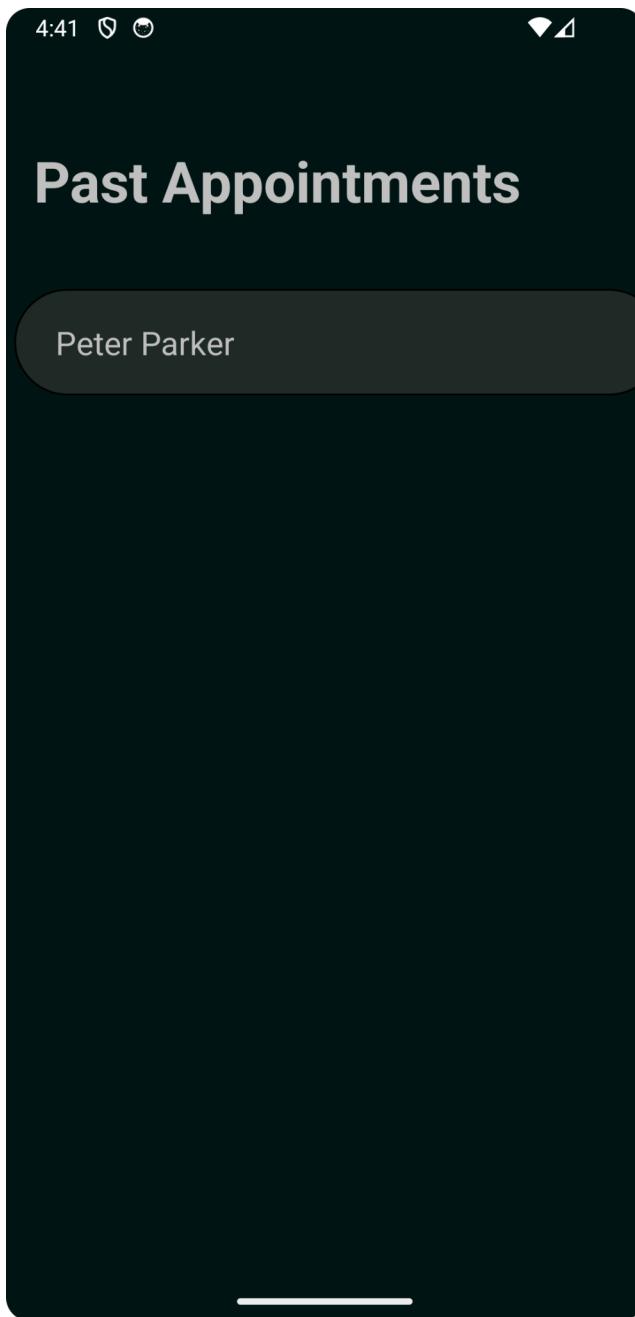
Add Appointment page (as a Patient):



View Potential Appointments page (as a Patient):



View Past and Upcoming Appointments page (as a Doctor):



4:52

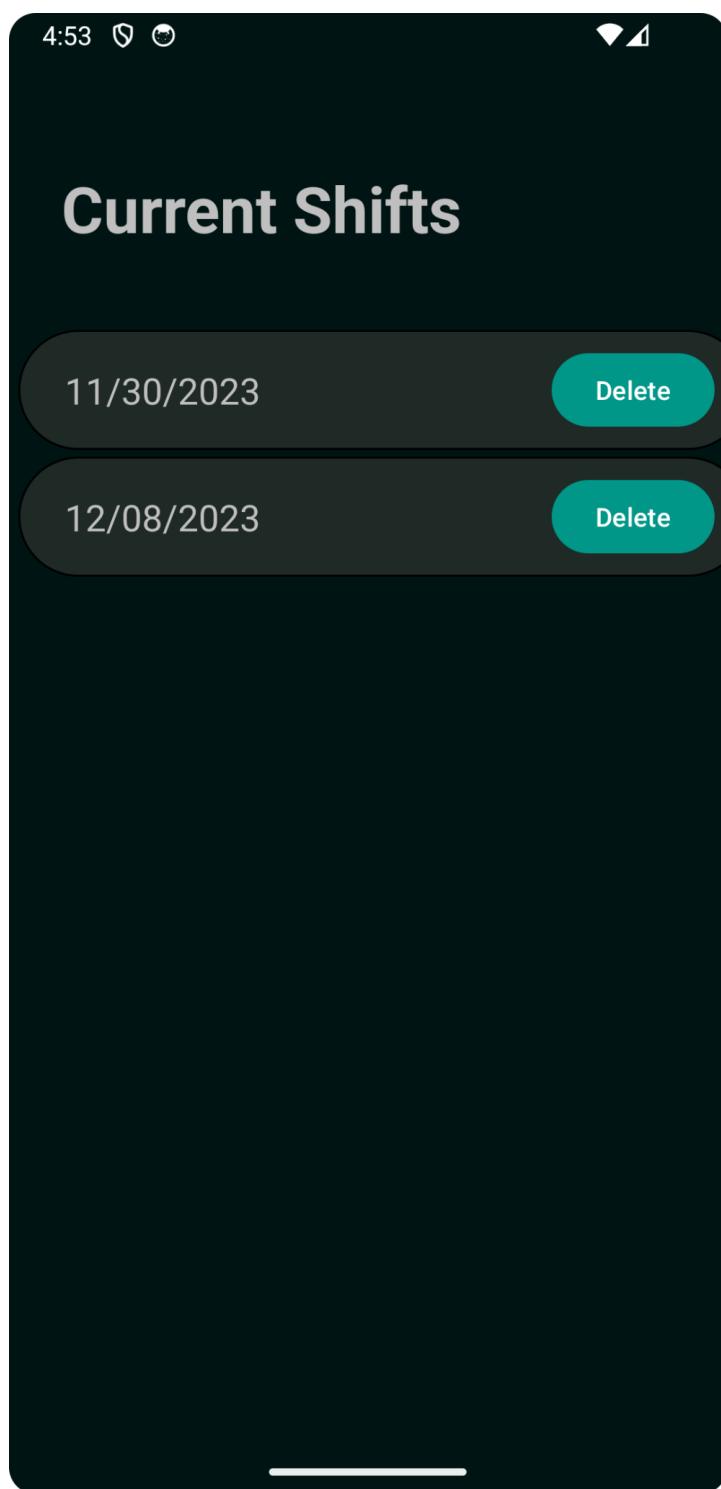


## Upcoming Appointments

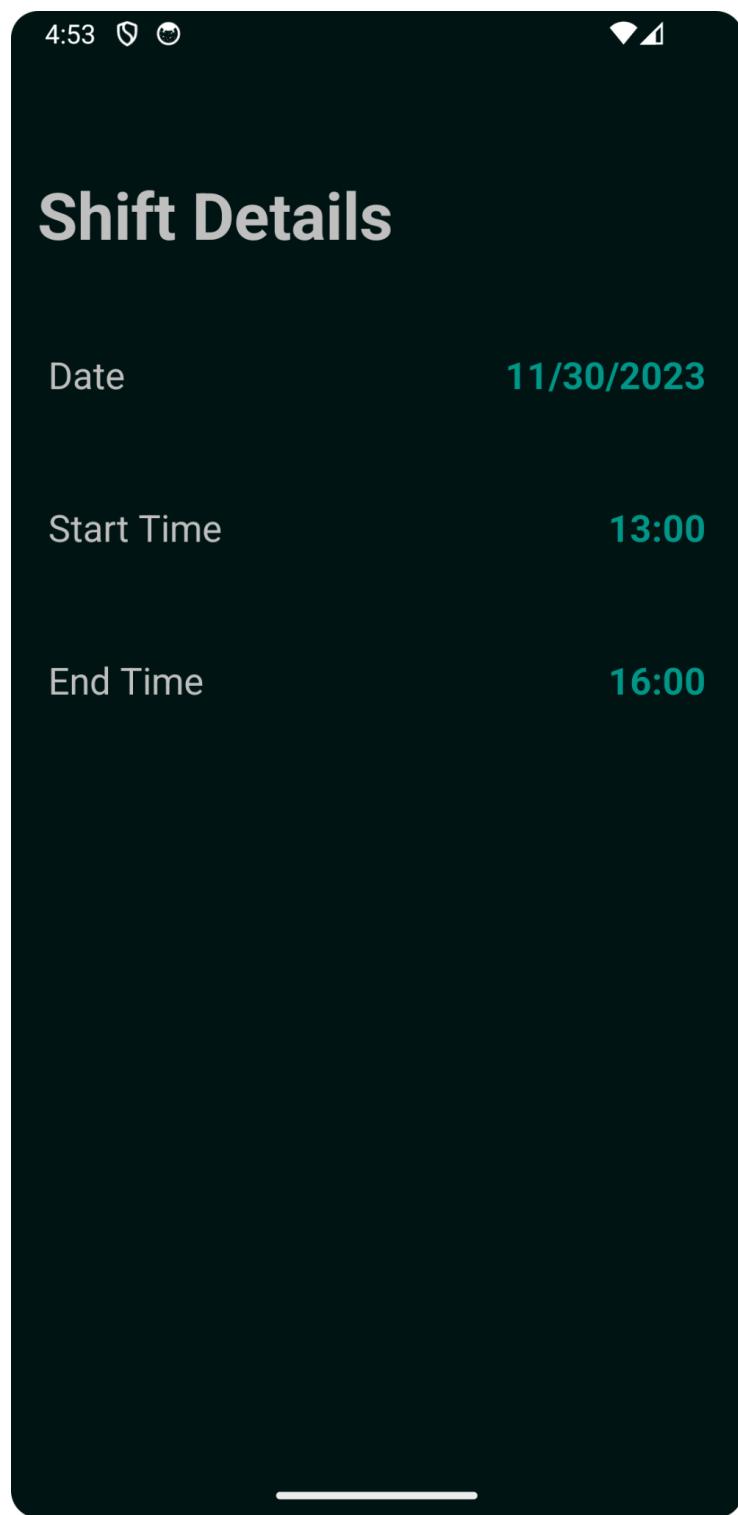
Peter Parker

Cancel

**View Current Shifts page (as a Doctor):**



**View Current Shift Details page (as a Doctor):**



Add Shift page (with sample input, date picker and time picker / only in 30-min blocks) (as a Doctor):

