Android Project: HAMS

SEG 2105 A – Introduction to Software Engineering
Fall 2023
University of Ottawa

Professor: Hussein Al Osman
Teaching Assistants: Luitel, Ali, Awad

Section: A01
Project Group: 17

Kevin Govier- 300282040
Henry Li – 300283722
Martin Patrouchev – 300286634
Kylan Thurairajah - 300295362

Submission Date: December 4th, 2023

Short Introduction

This project involved the design, implementation, and testing of a Healthcare Appointment Management System for a telehealth clinic using Android Studio and Firebase as our group's chosen database. This project is significant as it allowed our group to gain a deeper understanding of the software development process and it provided us with hands-on experience with databases, UML class diagrams, user interface design, and team-based problem solving. The purpose of this report is to provide the reader with a deeper understanding of the development process of our application and to provide more context regarding the functionality of the application. This report will include the UML class diagram of our application, a table containing the contributions of each team member for each deliverable, screenshots of the different pages of the application, and the lessons that our group learned while working on the project.

Updated UML Class Diagram

Team Member Contributions

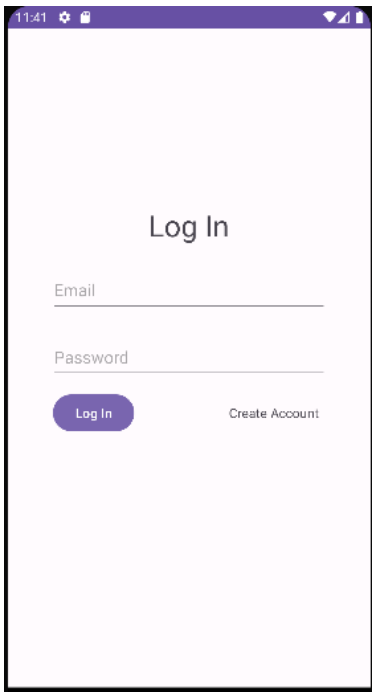
Deliverable	Group Members			
	Henry	**Kevin**	**Martin**	**Kylan**
1	**UI**: - Login page - Welcome page. **Firebase**: - Setting up Firebase and linking it to the project. - Uploading input data to Firebase from patient and doctor registry, including specialties. **Other:** - Field validation for doctor specialties. - Displaying your role when logged in. - Created UML Diagram	**UI:** - Patient Registration Page - Doctor Registration Page - Doctor Specialty Selection Page - User Selection Page **Other:** - Page routing - Preventing users from advancing in the registration process without providing necessary information	**Firebase:** - Check login credentials with ones in firebase **Other:** - Create validationUtils that validates all input fields for every page - Create alert communicating why the user cannot proceed with their inputs	**Other:** - Basic java classes with getters/setters (Doctor, Patient, Admin)
2	**UI** - Design of the 'Denied Requests' and 'Pending Requests' pages. - Design of the account info page for pending and denied users.	- Display patient/doctor info in the info page - show the list of doctors and patients in pending requests and denied requests page	**Firebase:** - Added registrations to firebase - When user registers it adds a pending registration **Other:** - Update account class adding	**UI:** - Pending/ denied admin with user info

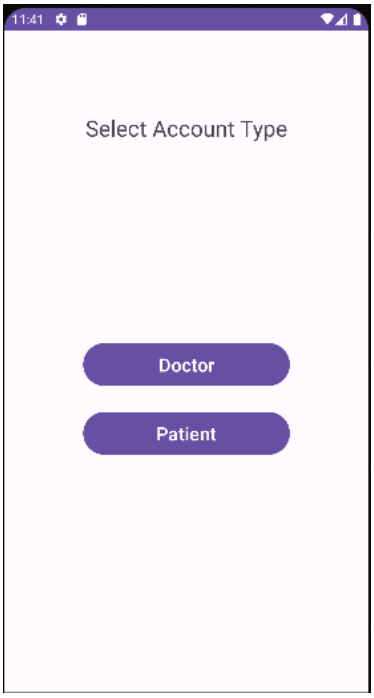
	- Design of the row in the adapter. Rows of both 'denied' and 'pending' request lists. **Email Notification** - Tried to implement an email sent to the user's given email when accepted/denied but unsuccessful. - Created UML Diagram	- send a pending request to denied requests if the admin rejects it - removes a request from pending/denied requests when a request is accepted	registrationStatus and type(which gets overridden) - Send Account object of who is logging in to welcome page - Login denies and alerts the user if they are pending or denied.	
3	**UI** - Pages of requested, accepted and past appointments. (3 pages total) - Patient info page when an appointment is clicked. - Adapter rows for each appointment. **Firebase:** - Retrieved appointment and patient info to be displayed. - Implemented functionality to make 'requested' appointments 'accepted', or 'rejected'. Also	- Shift creation page and its functionality - List of shifts page and its functionality - Updated firebase to show doctor shifts - Helped with appointment functionality and error handling	**UI** - Create doctor info page with an option to turn off and turn on automatic appointment accepting **Firebase:** - Set up firebase for appointments and shifts for each Doctor **Other:** -Create appointment class - Fix up Account class - Depending on type of account send to proper page when logging in	**Other:** - made a function that ensures shifts don't overlap during their creation

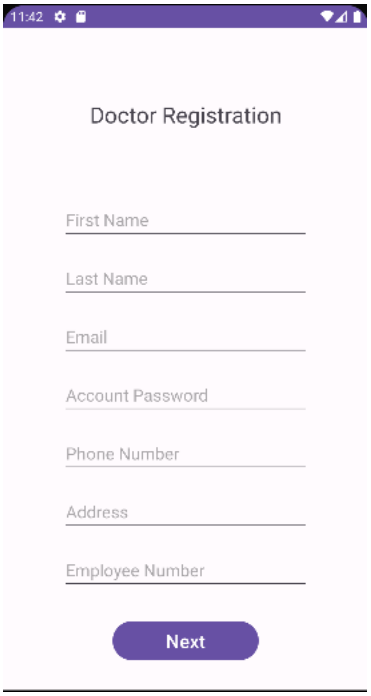
	for accepted appointments to be made ‘past’ or ‘cancelled.’ - Created UML Diagram			
4	**UI** - Pages that display a list of a patient’s upcoming and past appointments (2 pages) - Info pages to display information about those appointments (1 for upcoming and 1 for past) **Firebase** - Deleting a patient’s appointment if they cancel. Removes appointment if it doesn’t start in the next 60 mins only. - Created UML Diagram	-Wrote the final report -Created the AvailableAppointments table in Firebase -Modified Shift and Appointment classes -Added functionality to create 30-minute appointments that span a doctor’s shifts when new shifts are created -Created the Appointment Booking page and implemented all of its functionality, including moving available appointments to the requested appointments table on Firebase -Set up the testing functionality for the project and created the PatientRegister test	**UI** - Create rating bar and allow patients to rate appointments **Firebase** - Update appointment to update isRated, and update numRatings and rating to Doctor class **Other:** -fix up appointment class - add variables to Doctor class	-Made three of the tests to test the different aspects of the application -Integrated functionality to prevent doctors from cancelling shifts with corresponding appointments

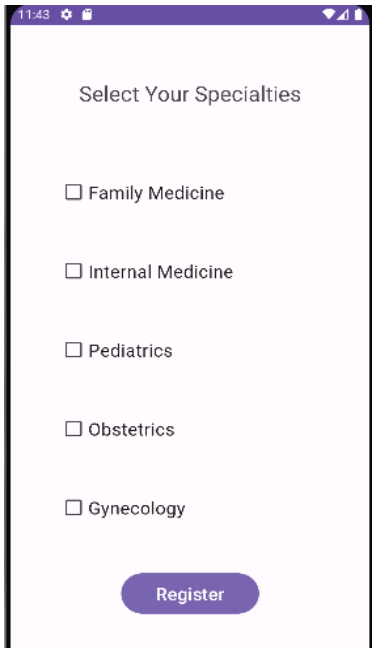
Screenshots

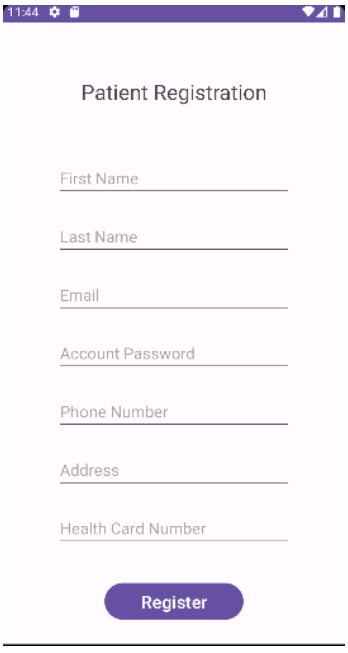
- **Login and Registration Pages**



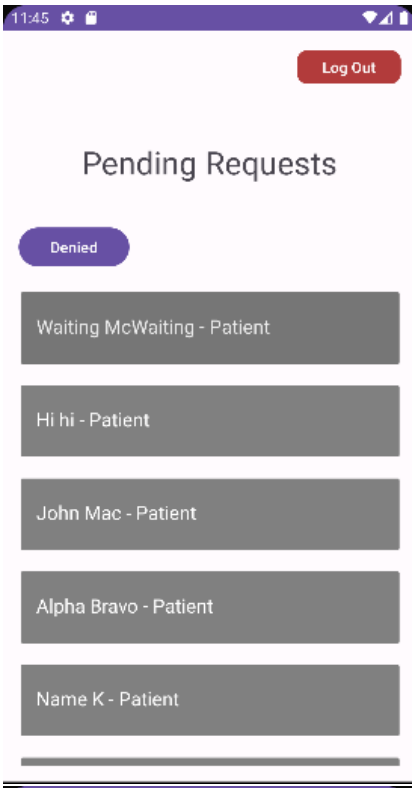


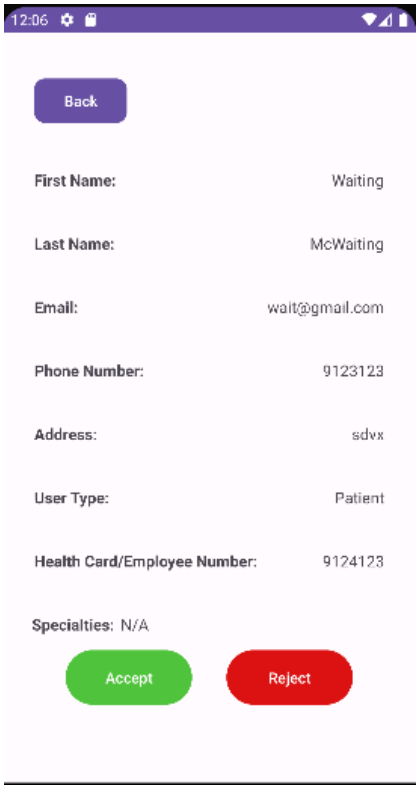


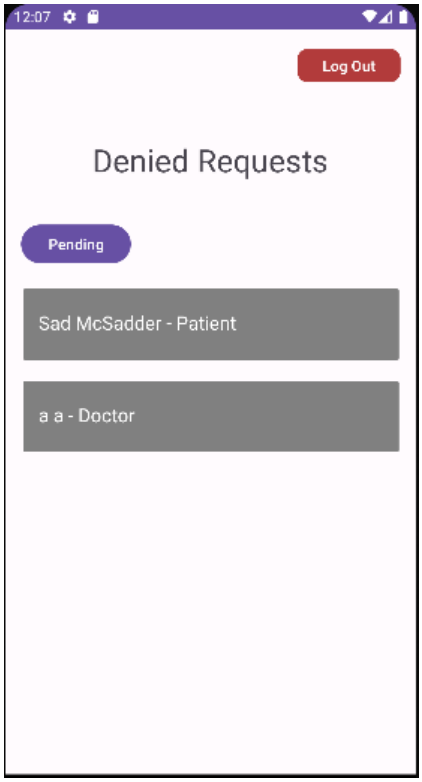


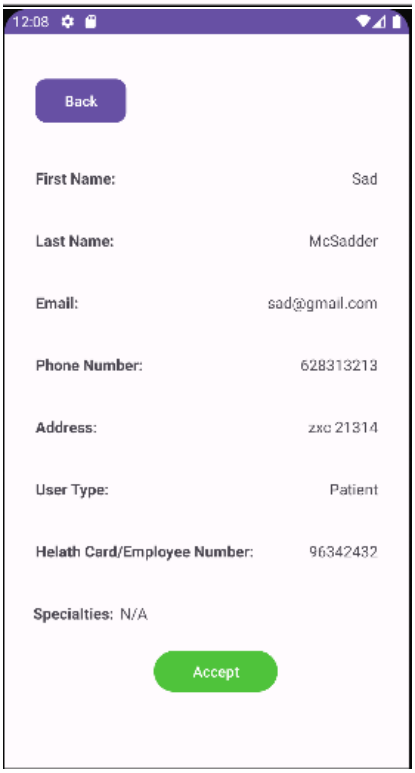


- **Admin Pages**

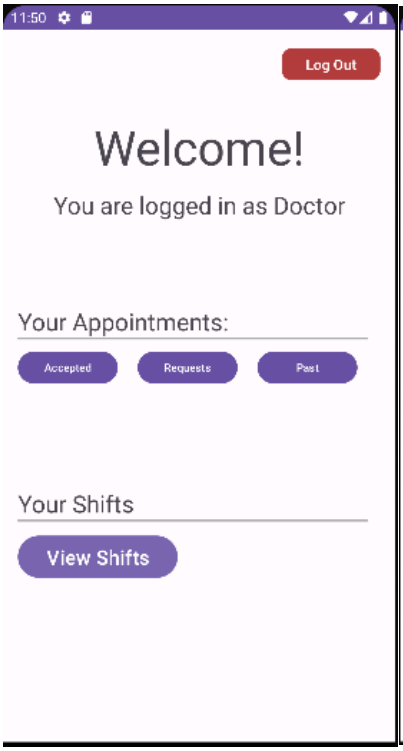

11:45
Log Out
Pending Requests
Denied
Waiting McWaiting - Patient
Hi hi - Patient
John Mac - Patient
Alpha Bravo - Patient
Name K - Patient

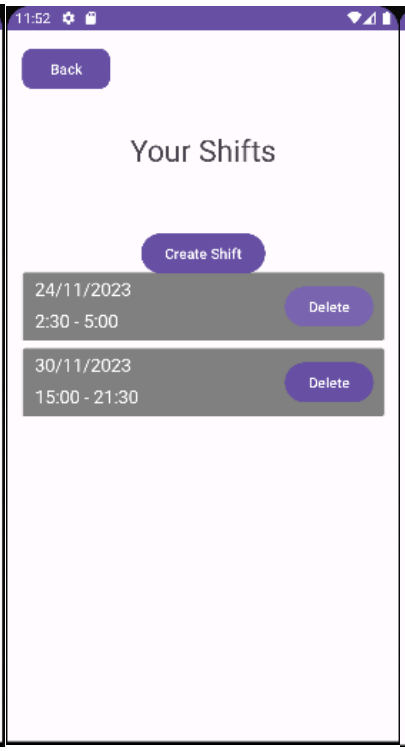

12:06
Back
First Name: Waiting
Last Name: McWaiting
Email: wait@gmail.com
Phone Number: 9123123
Address: sdvx
User Type: Patient
Health Card/Employee Number: 9124123
Specialties: N/A
Accept
Reject

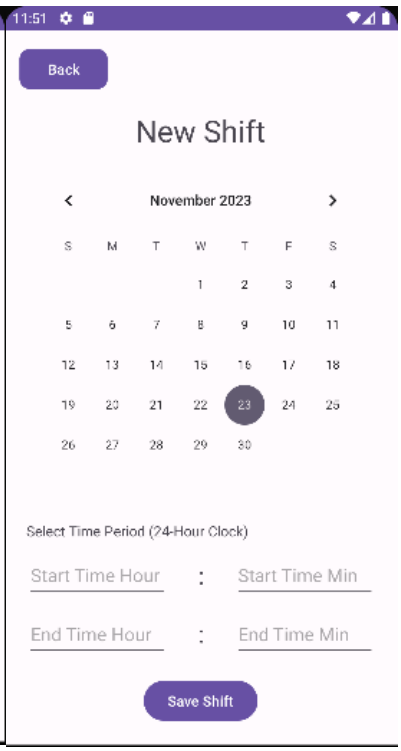

12:07
Log Out
Denied Requests
Pending
Sad McSadder - Patient
a a - Doctor

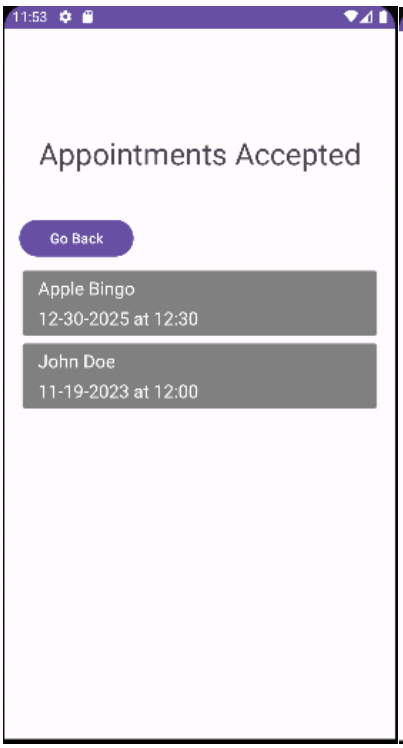

12:08
Back
First Name: Sad
Last Name: McSadder
Email: sad@gmail.com
Phone Number: 628313213
Address: zxc 21314
User Type: Patient
Helath Card/Employee Number: 96342432
Specialties: N/A
Accept

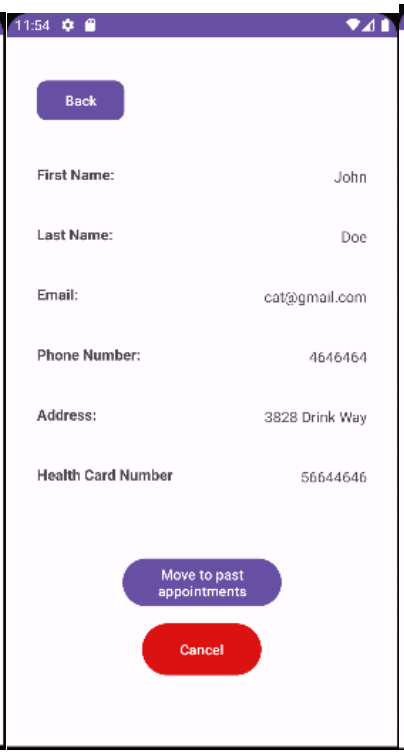
- **Doctor Pages**

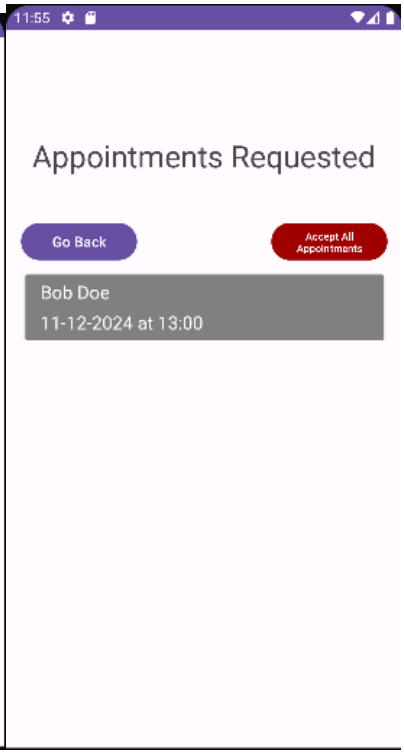


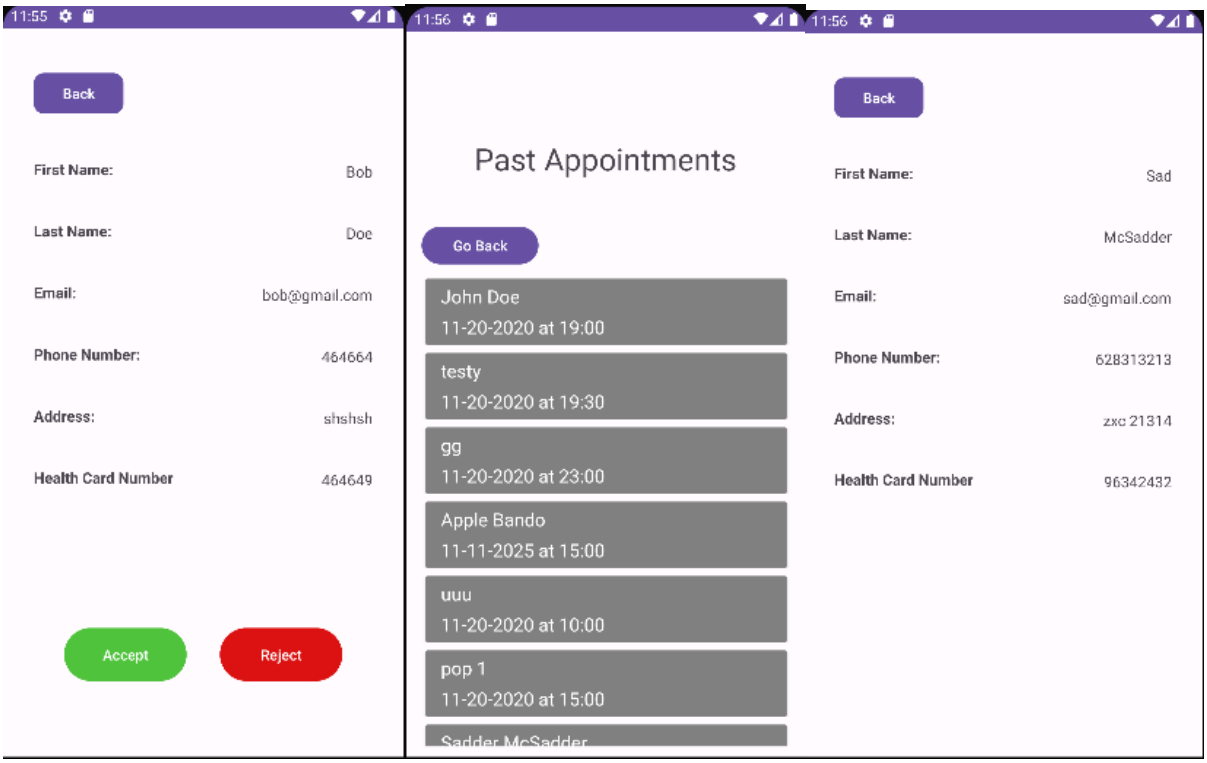




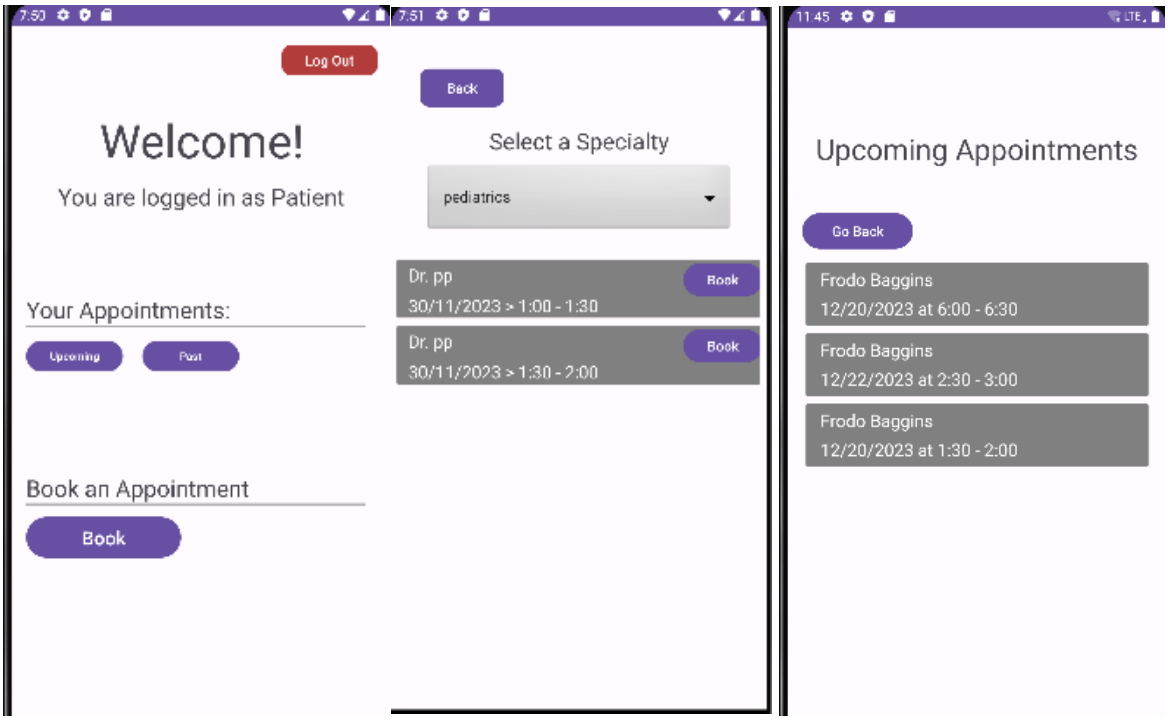


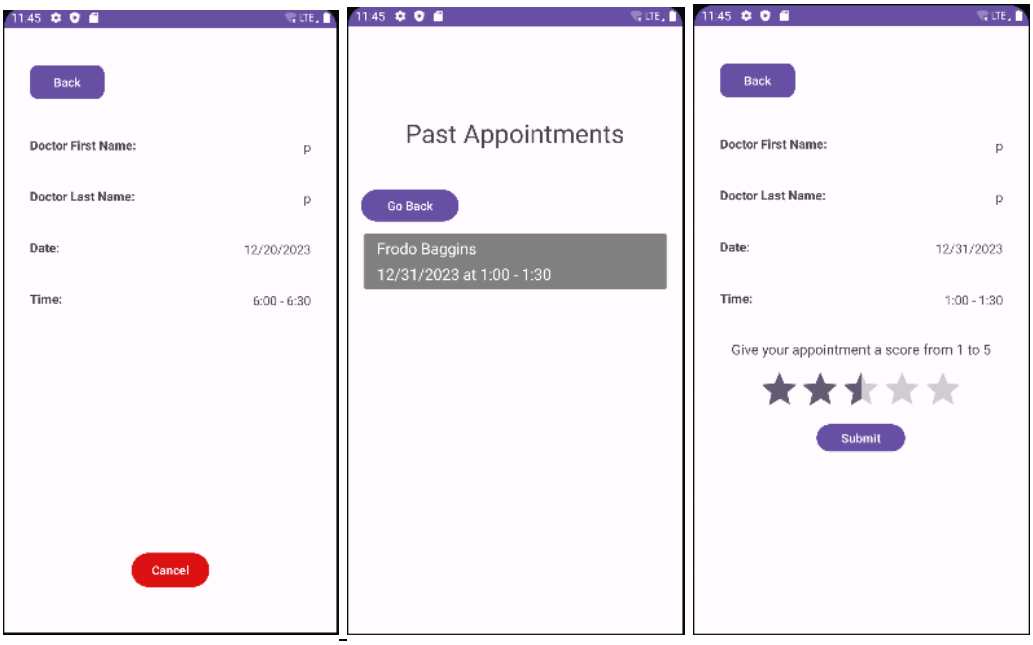






- **Patient Pages**





Lessons Learned

Throughout the development of our application, our group learned many valuable lessons and skills that will be useful in our future careers as software engineers. Firstly, we learned how to develop software in a group. Most programming jobs involve a large group of people working together on the same software to achieve a common goal, and this project gave us all experience in this regard. Moreover, this project allowed our group to gain skills in the development of a large software application. Software engineering is the process of developing large, high quality software systems to satisfy the needs of a customer, so the skills and knowledge that we obtained developing a large application of our own will undoubtedly be useful in our futures as software engineers. Another crucial lesson we learned was the importance of planning and communication throughout the software development process. It was through the use of a UML class diagram that our group was able to plan out the structure of our program. This greatly enhanced our group's efficiency which ensured that we were able to meet all of the project's deadlines. By having regular discussions regarding the state of our application, our group was able to manage our time more effectively and further increase our efficiency. In addition, this project taught us a lesson about the importance of databases and allowed our group to develop skills in the usage of databases. Our application uses Firebase as its database, and it is used to achieve much of the application's functionality. As such, our group developed knowledge in how to create a database and its tables, retrieve information from it in an application, and modify it from an application. Because databases are a crucial part of many software applications, these skills are very valuable for us to have. Finally, this project taught us the importance of resource management and dealing with constraints. Our group constantly had to manage our time to ensure that we met project deadlines. In addition to time constraints, we had to deal with other constraints as well including software constraints and constraints on the development process as a result of changing demands in the application's functionality. In sum, this project was very beneficial for our group's learning, as we gained a great deal of useful skills and insight which will help us become successful software engineers.