Android Project: HAMS

SEG 2105

Introduction to Software Engineering

Professor: Hussein Al Osman

Sedra Hamida	300286361
Anika Pathak	300306654
Connor Chamoun	300236627
Harleen Sandhu	300283944
Megan Palczak	300301072
Shazia Dewan	300287458

Date: December 4th 2023

T.A.: Mahtab Haj Ali, Dipeeka Luitel, Hassan Awad

Project Group 15

Introduction

The HAMS project (Healthcare Appointment Management System) is built on a mobile application using Android Studio, the official IDE for developing Android apps. The aim of this project is to create a simple, effective, and organized system for healthcare appointment scheduling and management. The user can sign in as a doctor, patient, or an administrator. These three types of users play various, vital roles that help implement a smooth integration of the system's objective. The patient can book appointments, cancel appointments, and rate doctors they had appointments with. The doctor handles their schedule by posting shifts and accepting or declining appointment requests. The administrator manages the patient and doctor registration requests to approve or deny access to the system. With the support of Firebase, the application can store and retrieve all the data entered by users.

Contributions

Table 1

The Team Members Contributions to the HAMS Project

Team Member Name	Contribution to the Project
Sedra Hamida	● Created the "Welcome Screen" for the app ● Displayed the information/details of the selected registration request and rejected registration request ● Created and implemented the autoapprove switch for the doctor. ● Created and implemented the delete shift function for the doctor. ● Created/Updated the UML Diagram for Deliverable 3 and 4 ● Wrote the report (title page, introduction paragraph, screenshots of the app, contribution to the project table). ○ All members participated in the "Lessons Learned" paragraph
Anika Pathak	● Created and implemented patient registration form ● Created and implemented the Patient and User class ● Created the view list of rejected requests ● Implemented an attribute status for user ● Implemented a display method in the Patient and Doctor class that overrides the abstract one in the User class to display info accordingly

	● Implemented Appointment, Shift, Doctor, User, Registration Status classes ● Displayed doctor's available time slots by showing a calendar (with days that are selectable only if the doctor has an available time slot) ● Divided shifts into slots that are either available or not
Connor Chamoun	● Created the sign up screen where the user chooses to sign up as a patient or doctor. ● Upon signing in ○ if request was rejected, display message informing the user to contact admin ○ if request has not been approved, display message that their registration has not been approved (pending) ● Created the display for the doctor's shifts ● Created all the test cases for the application
Harleen Sandhu	● Created and implemented the "Sign In" screen ● Created the inbox for registration requests ● Display upcoming appointments for the patient, where once clicked, information is displayed ● Display past appointment information for the doctor ● Created upcoming appointments list view for patient. ● Display past appointments where patients are able to rate doctor (1-5 stars)
Megan Palczak	● Created the account page that appears after sign in, containing a message and log off button ● Editing firebase database based on approval and rejection requests from the administrator ● Created and implemented the add shift functionality and calendar view for the doctor with time/date constraints ● Implemented comparator for chronological shift organization when displaying shifts in list view ● Implemented the delete appointment functionality with time constraints and layout inflation
Shazia Dewan	● Created and implemented Doctor registration form and Doctor class

	● Send an email and notification to user when request is approved/rejected (bonus) ● Created and updated the UML Diagram for deliverable 2 and 3 ● Created the “Welcome Screen” for administrator ● Displayed upcoming and past appointments for Doctor in listView ● Divided and organized all java classes into appropriate packages ● Implemented the functionality that a Patient can book appointments by specialty and selecting time slot (30-min period in doctor’s shift. ○ Bonus: patients can also search by name (first name and last name), see project instructions for specifications.

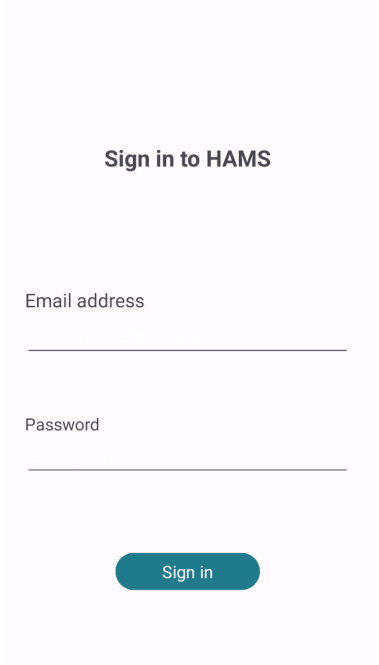
UML Diagram

Link to view the UML Diagram of the HAMS Project:

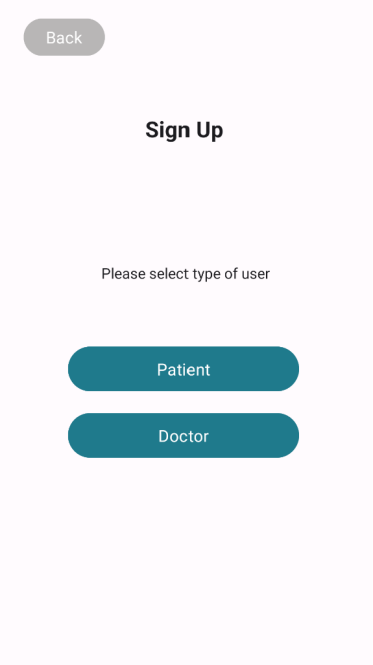
https://drive.google.com/file/d/1tg70-6z976bPVP2i8veAKO7TCHkz60Xd/view?usp=drive_link

App Screenshots

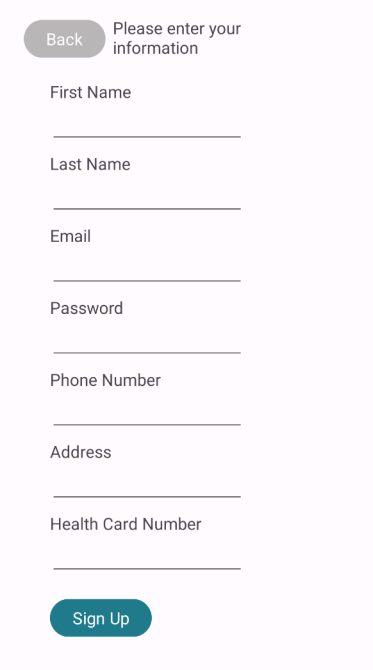
Welcome Screen



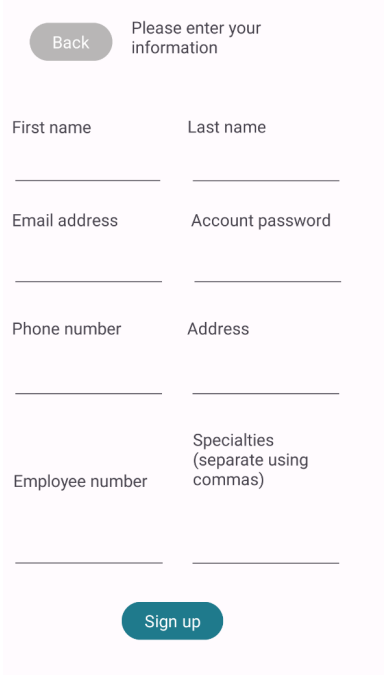
Sign In Screen



Sign Up Screen



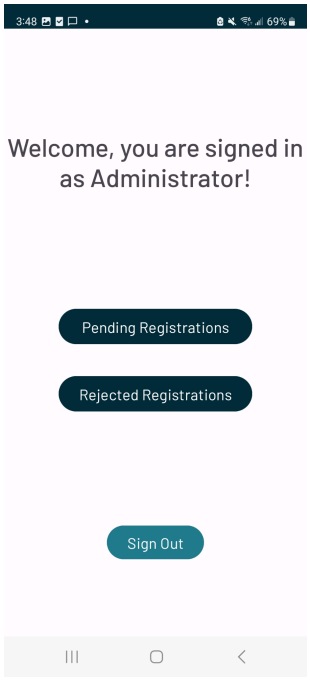
Sign Up as a Patient



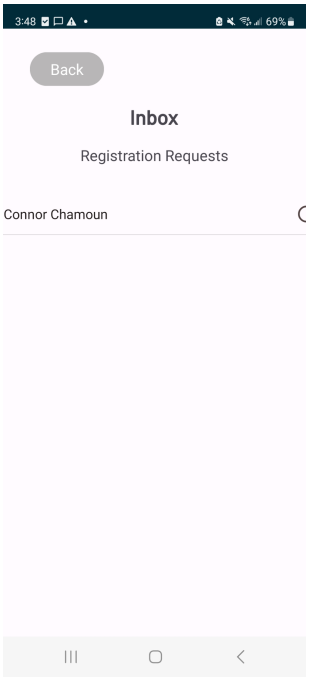
Sign Up as a Doctor

After the user successfully signed in or signed up, the resulting screen will depend on their user type account (Administrator, Doctor, Patient)

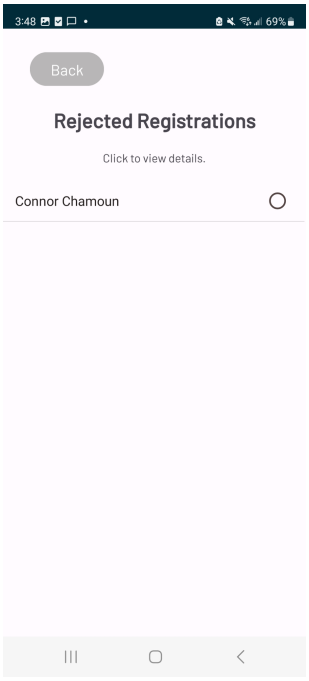
Admin Screens



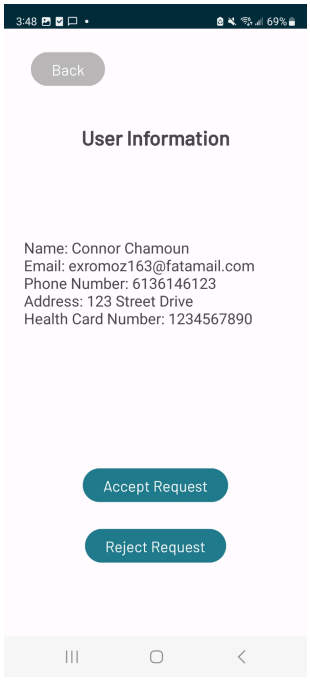
Welcome Screen



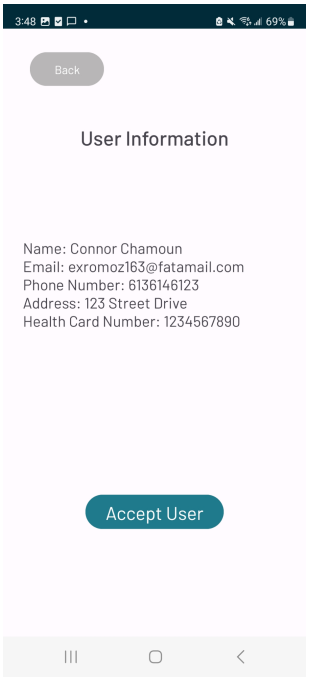
Pending Registrations Inbox



Rejected Registrations Inbox

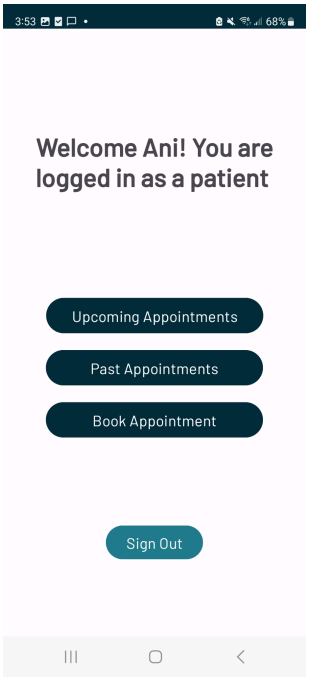


Display of User's Info Request

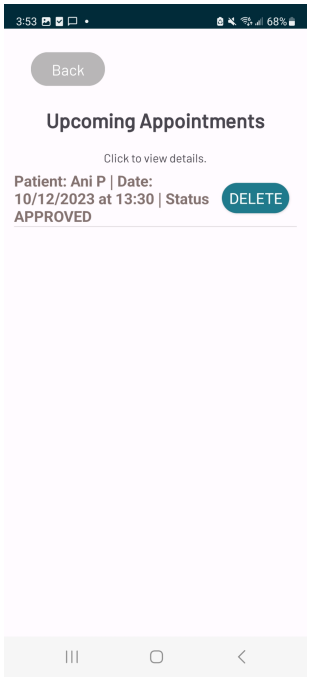


Display of Rejected User's Info Request

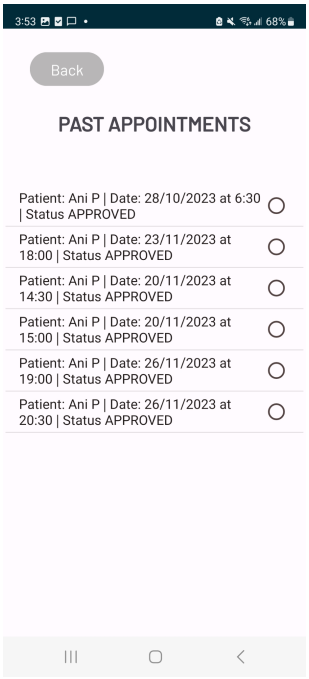
Patient Screens



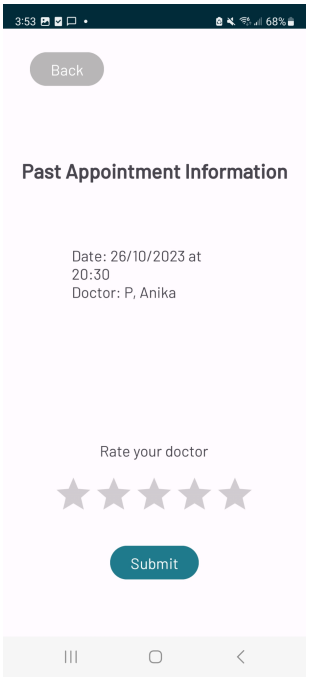
Welcome Screen



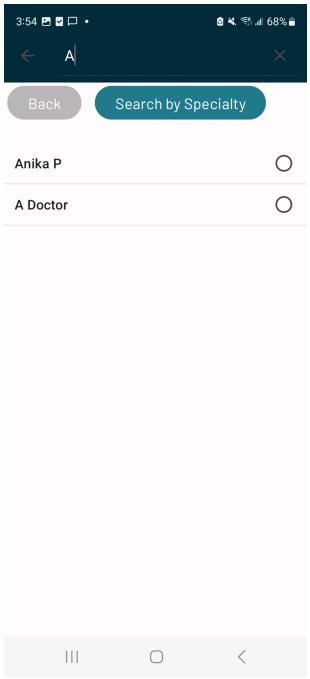
Upcoming Appointments Screen



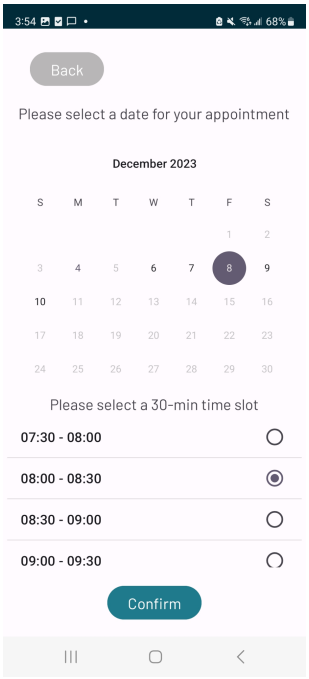
Past Appointments Screen



Display Patient's Past Appointment Information & Rate Doctor

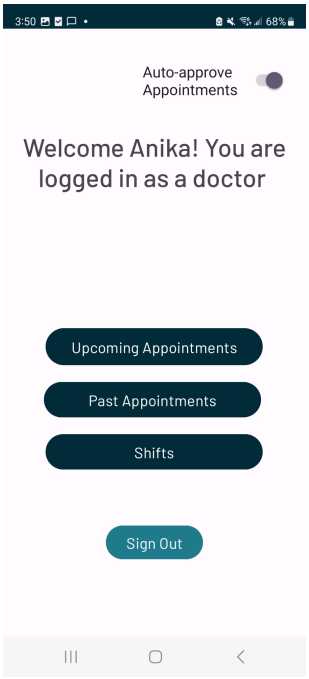


Book Appointment by Searching for Doctor

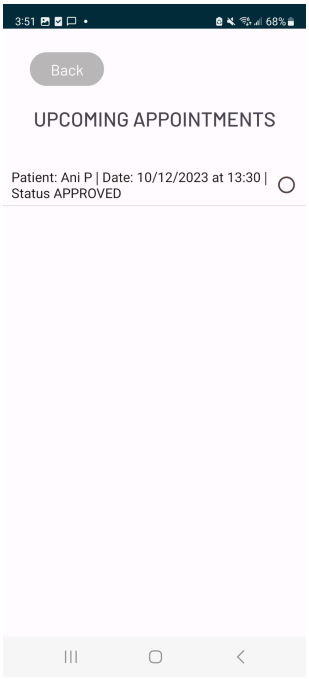


Select Appointment Date & Time

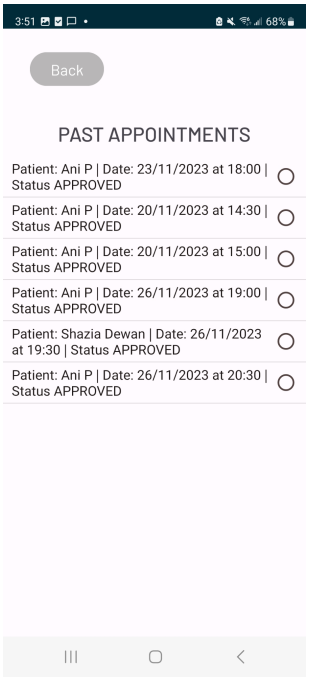
Doctor Screens



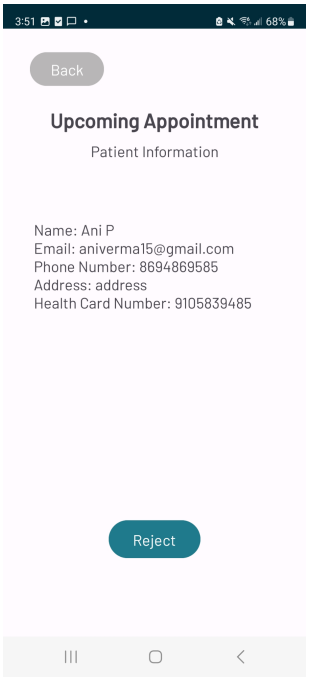
Welcome Screen



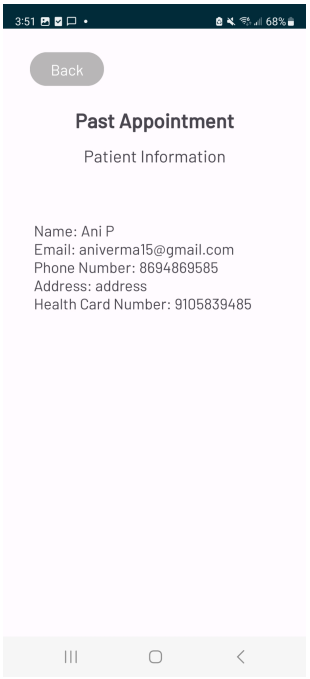
Doctor's Upcoming Appointments



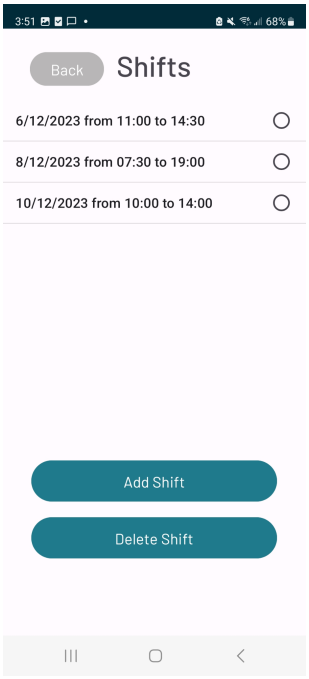
Doctor's Past Appointments



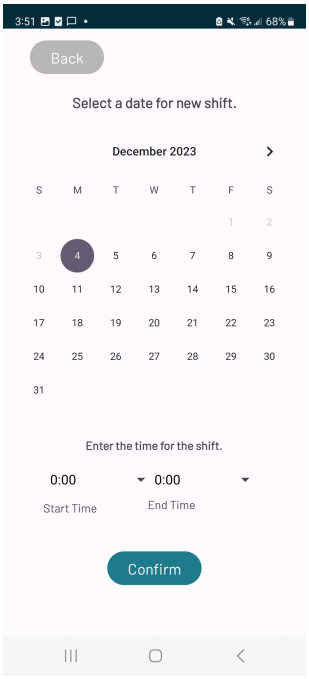
Selected Upcoming Appointment Information



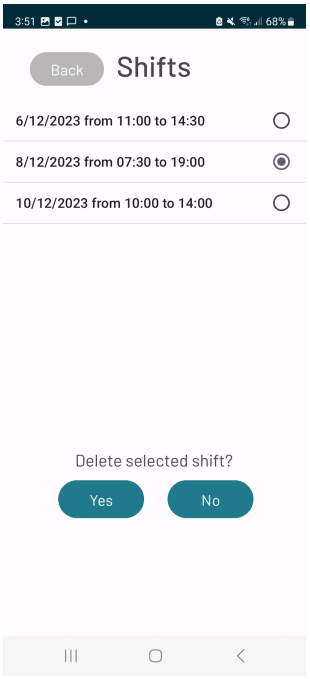
Selected Past Appointment Information



Doctor's Shifts



Add a Shift by Selecting Date & Time



Select a Shift to Delete

Conclusion (Lessons Learned)

During the process of developing the HAMS application, the team was exposed to and learned several new concepts. Working with Android Studio, the group was exposed to the UI development, unit testing, and the Firebase database. The UI demonstrated the ability to visually display the code and application, which was a new notion to the team. Reading through documentation to grasp how methods in packages and classes including Calendar, ArrayAdapter and Firebase work was a skill set that the group members developed. In addition to programming, the team also developed communication skills and learned how to fairly and efficiently allocate work for each deliverable. The team achieved this by separating the backend and frontend work, and distributing classes for each user – doctor, patient, and administrator. Conducting weekly meetings and constantly communicating allowed the team to plan and manage time to deliver the project within the deadlines.