Android Project: HAMS

SEG2105 - Introduction to Software Engineering
Fall 2023

Course Coordinator: Hussein Al Osman
Teaching Assistant: Dipeeka Luitel

Group #26
Camilla Nunes, 300307447
Kiara Saahfon, 300290520
Somana Debnath, 300312973
Lara Debsi, 300281977
Upama Utpal, 300296883
Jeny Patel, 300289583

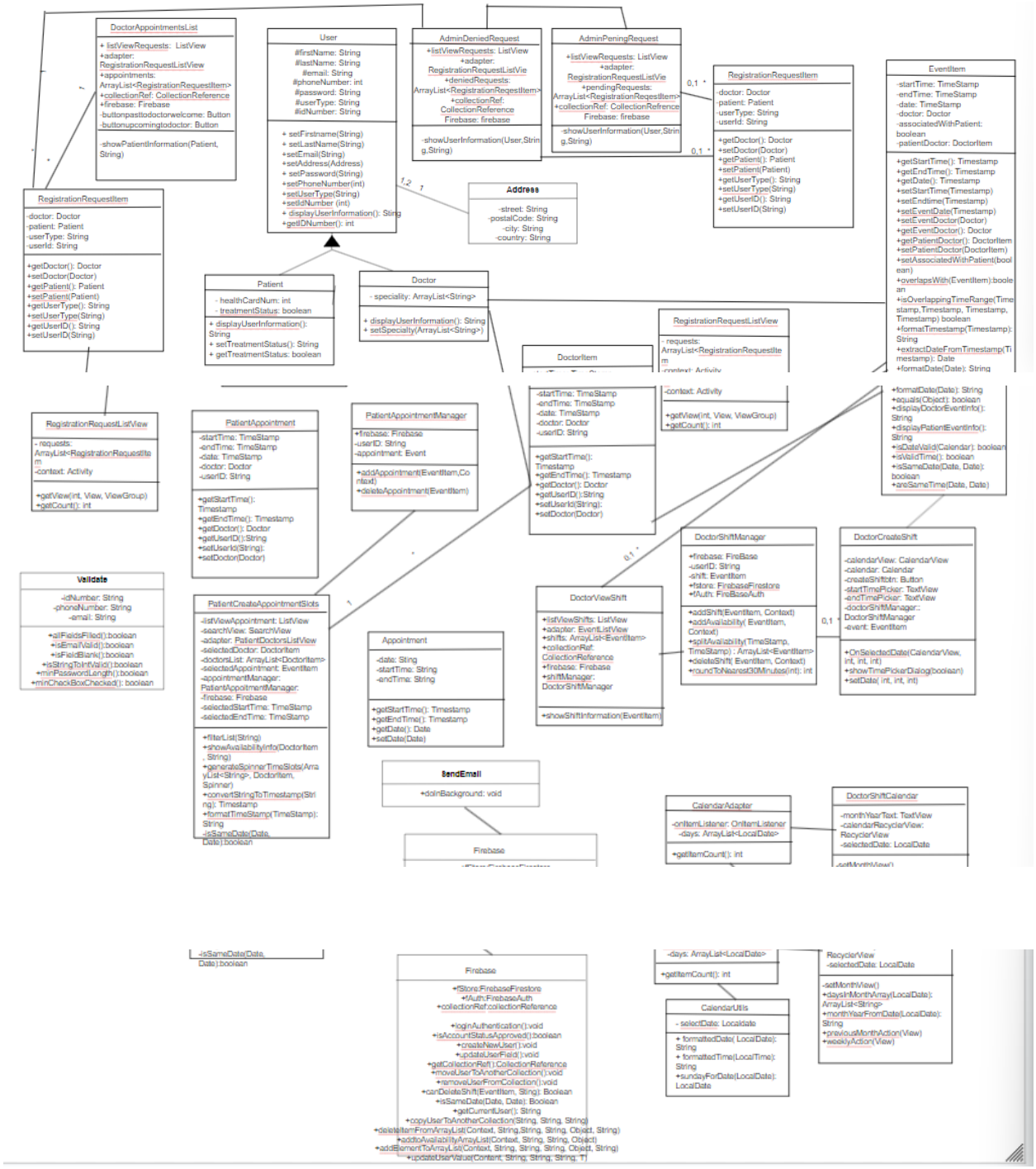
Date: December 4th, 2023

Introduction

With the use of a mobile application, the process of healthcare appointment scheduling and management is enhanced for both doctors and patients alike. The goal of this project is to develop a Healthcare Appointment Management System (HAMS) using Android Studio, which would establish a simple and effective method for organizing appointments. Using Firebase, the account information of all users can be stored and accessed when necessary. This application provides a variety of functionalities for three types of users: Patient, Doctor, and Administrator. Patients can book, view, and cancel appointments as well as rate their doctor. Meanwhile, the Doctor user manages their shifts and can view past or upcoming appointments. They are also able to reject or approve appointment requests and cancel appointments they had previously approved. Finally, the Administrator approves or rejects account registration requests of any users (Doctor or Patient) that make a new account and can change their decision on a previously rejected user.

UML Diagram

*View GitHub for full clear image



Contributions

Deliverable 1:

Names	Contribution
Camilla	● Backend functionality of Doctor registration. ● Frontend UI design of Doctor registration page. ● Completed bonus with set up and integration of firebase. ● Created/modifed UML diagram ● Implemented Validate.class to validate inputted text fields ● Created and set up Doctor, Patient and User classes ● Helped implemented functionality that User can log in and redirects to respective Welcome page (After Firebase integration) ● Implement functionality that Users can log off
Jeny	● Backend functionality of Patient registration. ● Frontend UI design of Patient registration page.
Somana	● Frontend UI design of Login page. ● (Before Firebase integration) Implemented login functionality which directs user to Welcome Page
Lara	● Worked on frontend UI design of welcome page ● Added several of the needed classes into the manifest file ● Implemented the ability to get to the registration page or the log-in page
Kiara	● Frontend UI design doctor welcome page ● Frontend UI design admin welcome page ● Frontend UI design patient welcome page, ● Navigation between welcome screens
Upama	● Frontend UI design of register as patient or doctor page ● Backend button navigation for multiple pages

Deliverable 2:

Names	**Contribution**
Camilla	● Helped implement validation of Patient or Doctor's registration status before allowing successful login, display relevant token ● Completed bonus of sending email notification when registration request is approved or rejected ● Re-implemented backend functionality of Admin inbox view of pending registration requests and previously rejected registration requests ○ Implemented ListView to display requests in realtime from firebase collections ○ Upon selecting request, retrieves information from firebase and alert box shows user information (except password) ○ Admin can approve pending requests or previously rejected requests ○ Based on registration status, document moved to related Firebase collection (Approved Requests, Pending Requests, Denied Requests)
Jeny	● Worked on frontend UI design of Admin's pending registration requests page
Somana	● Updated UML diagram ● Helped implement functionality for validation of patient & doctor's registration status to allow login to account
Lara	● (Not complete, had to scrap) Backend of Admin Denied Request page ○ Implemented the approval of the request when it's been rejected already ● Worked on frontend UI design of Admin's approved registration requests page
Kiara	● Worked on frontend UI design of Admin's pending registration requests page
Upama	● (Not complete, had to scrap) Backend of Admin Denied Request page ○ Implemented the approval of the request when it's been rejected already ○ Implemented the functionality to see list of registration requests ● Worked on frontend UI design of Admin's approved registration requests page

Deliverable 3:

Names	**Contribution**
Camilla	● Completed/updated UML diagram ● **Reworked all Firebase usages and created a facade Firebase class** ● **Re-implemented and fixed incomplete work from deliverable 2** ● Frontend UI of list view of Doctor Shifts ● Frontend UI of Doctor create shift page ● Re-implemented backend functionality of Doctor creating shifts ○ Shifts are in 30 minute increments ○ Shifts must be a day that hasn't passed and does not conflict with other shifts ○ Start time must be before end time of shift ● Backend functionality of Doctor viewing and deleting shifts ○ Listview displays shifts in realtime from Firebase ○ When selected, AlertDialog appears showing shift information ○ Can delete shifts
Jeny	● Frontend UI design of Doctor's upcoming and past appointments pages ● Backend functionality of Doctor's past appointments list and upcoming appointments list ○ Implemented ListView to display past and upcoming appointments in real time based on Doctor's choice of past or upcoming appointments
Somana	● (Not complete, had to scrap) Frontend UI Design of Calendar used for Doctors Shift ● (Not complete, had to scrap) Backend functionality for calendar function
Lara	● (Not complete, had to scrap) Implemented the time and date picker for the creation of shifts
Kiara	● (Not complete, had to scrap) Frontend UI design of week view ● (Not complete, had to scrap) Frontend UI design of date, start time and end time ● (Not complete, had to scrap) Back end shifts validation, 30 minute increments, the doctor cannot add a shift that conflicts with one that has already passed.
Upama	● Backend functionalities of displaying a patients information, and appointment approval, rejection and cancellation ● Frontend UI design of the View for Patients information and the buttons

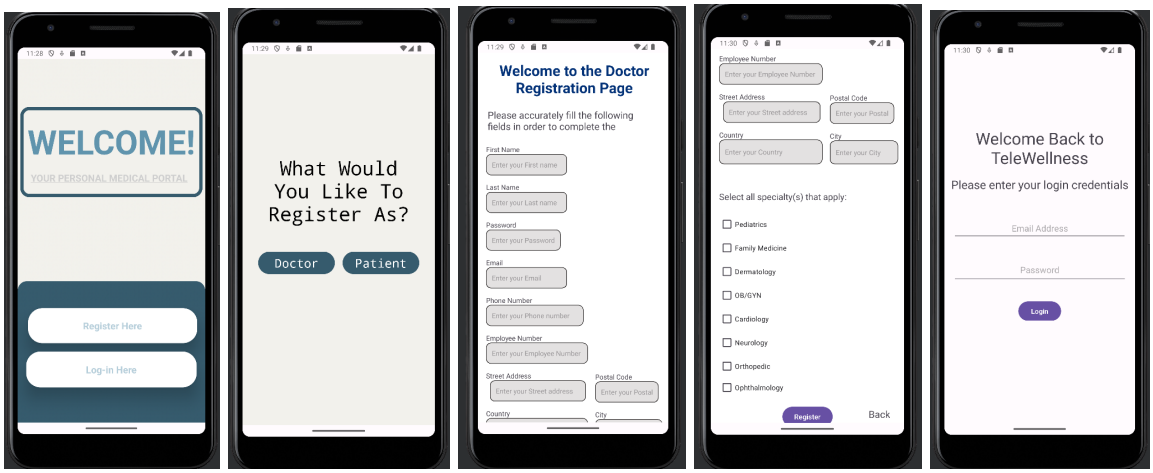
Deliverable 4:

Names	**Contribution**
Camilla	● **Re-implemented and fixed incomplete work from deliverable 3** ● Backend functionality for the patient searcing for doctors based on specialty ○ Implemented SearchView and ListView to display Doctors realtime from Firebase collection ○ When a Doctor is selected, AlertDialog appears to select date and time of appointment ○ Validated fields ■ Appointment date can't have passed ○ Appointment is added to Patients upcoming appointment list ○ That slot is no longer in Doctors availability for other Patients ● Modified Doctor deletion of shift functionality, can't delete shift if associated with Patient ● Frontend UI for searching for Doctors and displaying appointment slot information ● Added helpful shared methods in Firebase.class
Jeny	● Worked on implementing frontend and backend functionality for displaying patient's past and upcoming appointments
Somana	● Worked on Project report
Lara	● Worked on Project report ● Worked on 3/4 unit tests (patient, user, doctor)
Kiara	● Updated UML diagram
Upama	● Implemented rating functionality ● Fixed bugs in displaying patient appointment info

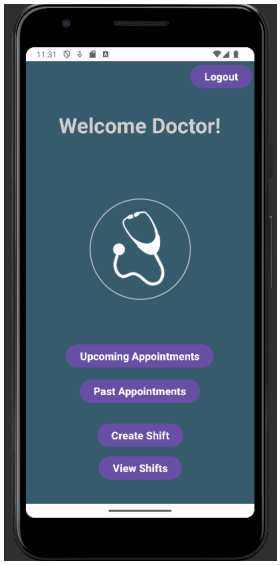
```
   return new Doctor(123456, specialties, "John", "Doe",
"john.doe@example.com",
           "password123", 987654321, address);
}
```

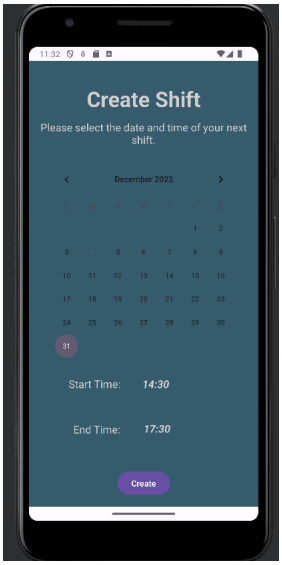
Screenshots of App

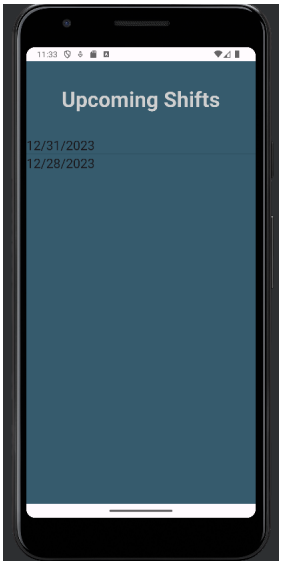
Launcher/General Activities:

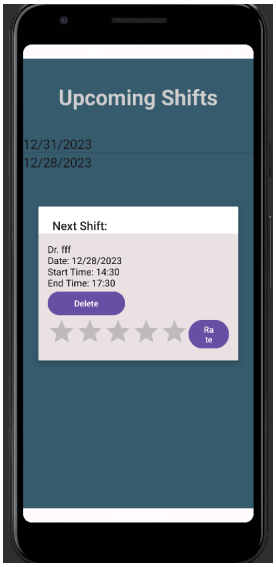


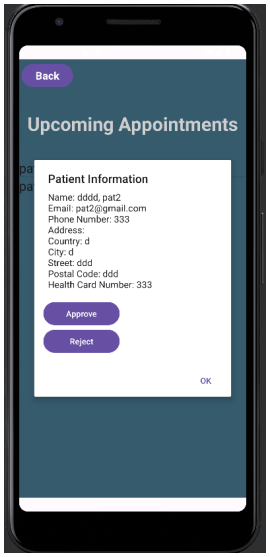
Doctor Activities:

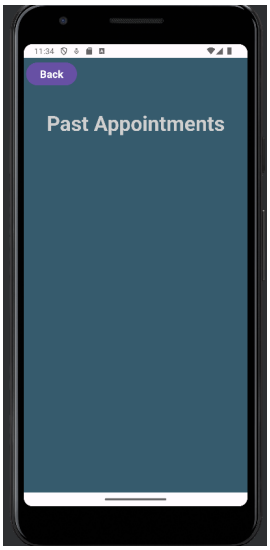




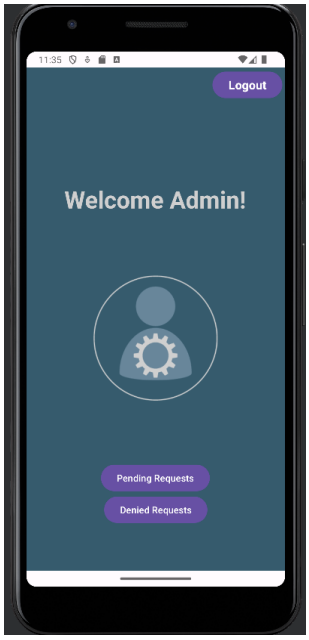


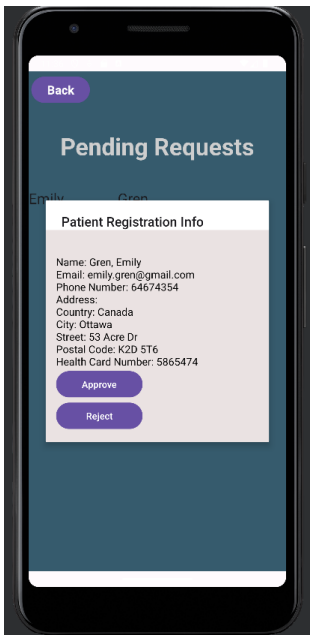


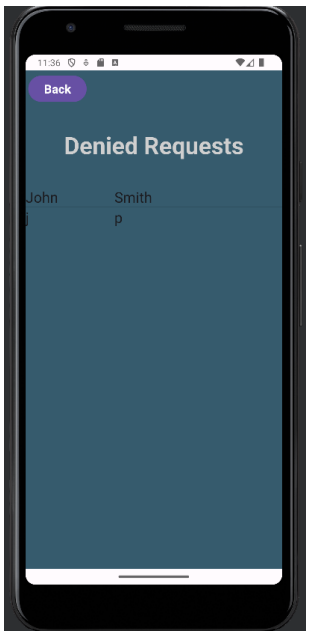


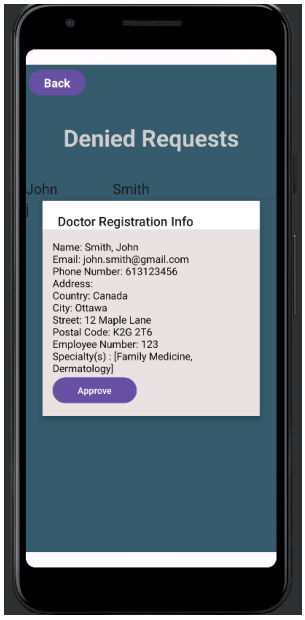


Admin Activities:

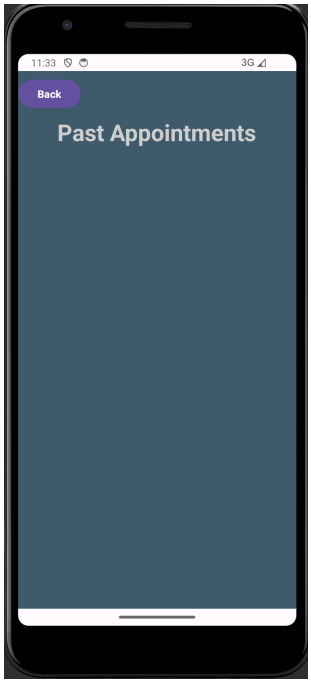


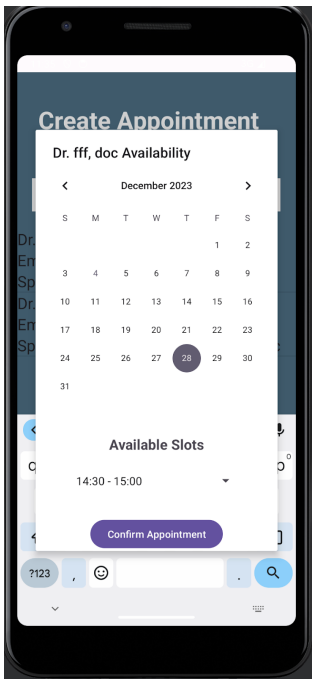


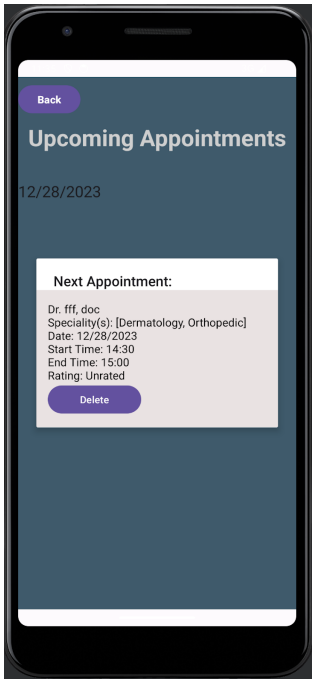


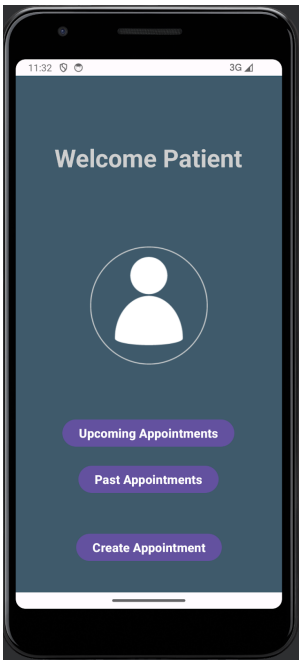


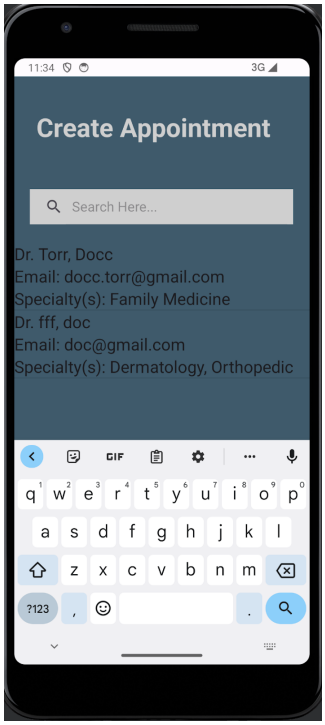
Patient Activities:











Conclusion (Lessons Learned)

Throughout the course of this semester, working on this project has taught us many skills and lessons that are applicable to our future careers. First, we were introduced to basic app development using Android Studio which taught us concepts such as designing UI, coding in the context of an application, and how to implement unit testing. Moreover, we learned how to store and retrieve user information in a database, emphasizing how it is helpful to use a separate class for the database when integrating it into a project. Another valuable lesson we learned is the importance of utilizing resources available to us to independently learn and research how to implement specific functionalities before they are formally taught to us in the course (e.g., implementing Firebase for the first deliverable). Finally, we learned the significance of planning our code to meet the requirements of each deliverable and organizing code, especially when working with numerous files that relate to different parts of the application.