

Android Project: EAMS

(Event Attendance Management System)

SEG 2105[A] – Intro to Software Engineering

Fall 2024

School of Engineering and Computer Science

University of Ottawa

Course Coordinator: Hussein Al Osman

Teaching Assistants: Ammar Rashed, Hassan Awad, Subhashri Mohan, Nima Meghdadi

Group # 18

Shiraz Atamna (300354084)

Hugo Cayen (300294277)

Enah Cheng (300312056)

Keno Obuwoma (300345036)

Alina Shayakhmetova (300355615)

Shaghayegh Sheibani (300388357)

Submission Date: 4/12/2024

Introduction:

For this project, we created an Event Attendance Management System (EAMS) — a mobile application that manages university event creation and registration. The application has 3 user types: Administrator, Organizer, and Attendee.

The Attendee is anyone who might want to participate in the school events. They must first request registration to the app. Once their account is approved, they can view, search, and request events (made by the organizer) and cancel event registration. The Attendee cannot cancel event registration within 24 hours before the start of the event.

The Organizer is an event organizer. Like the Attendee, they also must first request registration to the app. Once their account is approved, they can create events and approve and reject any event registrations requested by Attendees. The Organizer can view upcoming events as well as past events (which are no longer available for registration) they have created.

Lastly, the Administrator is responsible for approving and rejecting application registration requests from Organizers and Attendees. The Administrator cannot reject requests they have already approved.

This project was assigned in early October 2024. We continuously improved our system architecture, applying relevant concepts learned in class to increase efficiency and reusability. There are 4 deliverables in total designed to assist us in completing the project. For each deliverable, we submit our work-in-progress through GitHub, along with a UML diagram and demo video to report what we have completed. The Project Description document outlines each deliverable:

1. Deliverable 1 focuses on the building foundations of the app, such as creating a Login and Welcome page and a Logout button.
2. Deliverable 2 focuses on Administrator functionalities.
3. Deliverable 3 focuses on Organizer functionalities.
4. Deliverable 4 is the most significant deliverable. Not only does it focus on Attendee functionalities, but we are also to fix and refine any app features and summarize our work in this report.

This report includes our finalized UML diagram, deliverable task divisions, screenshots of the finished application, and a section describing what we have learned.

Final UML Diagram:

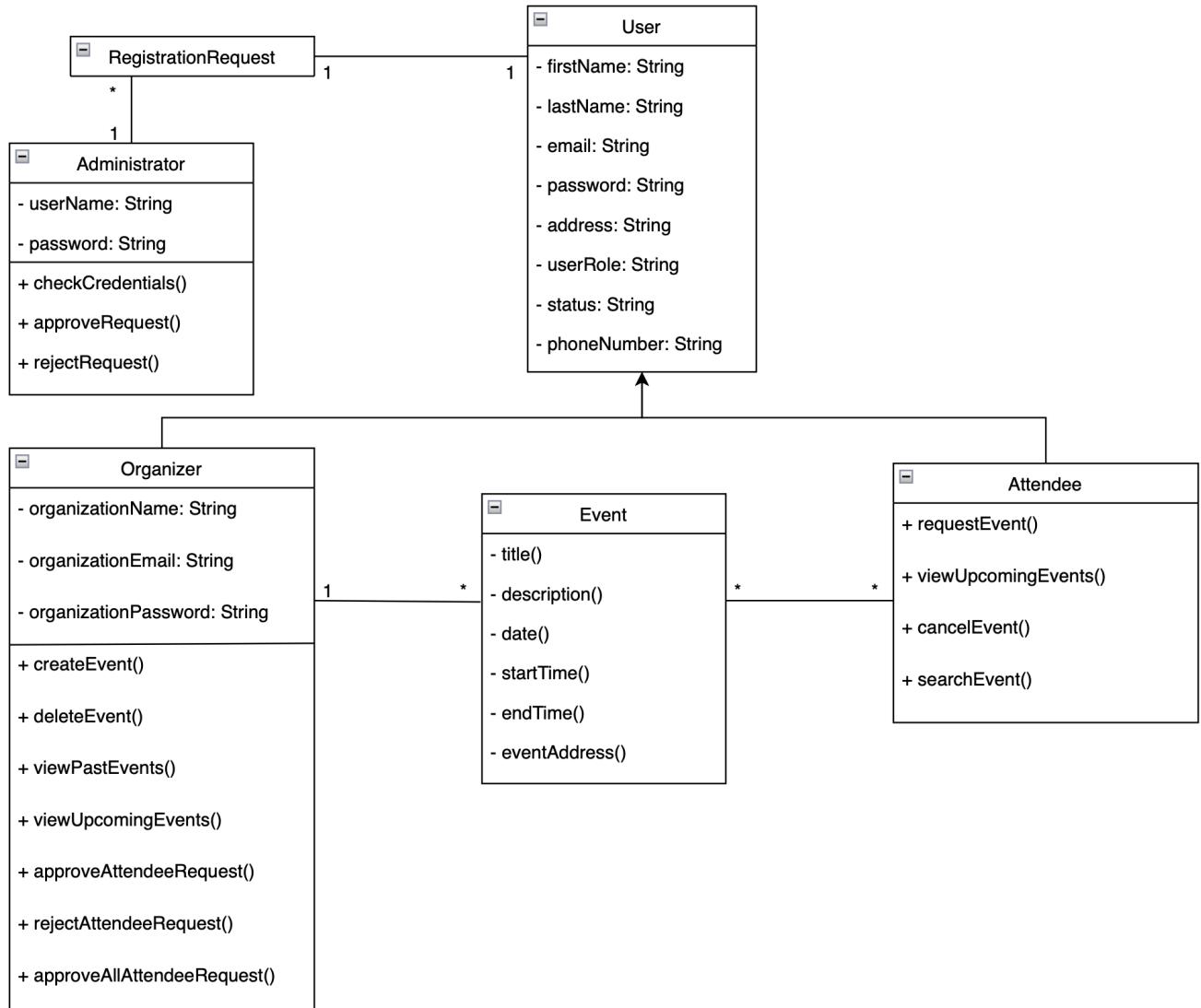


Table Specifying the Roles of Each Teammate for Each Deliverable:

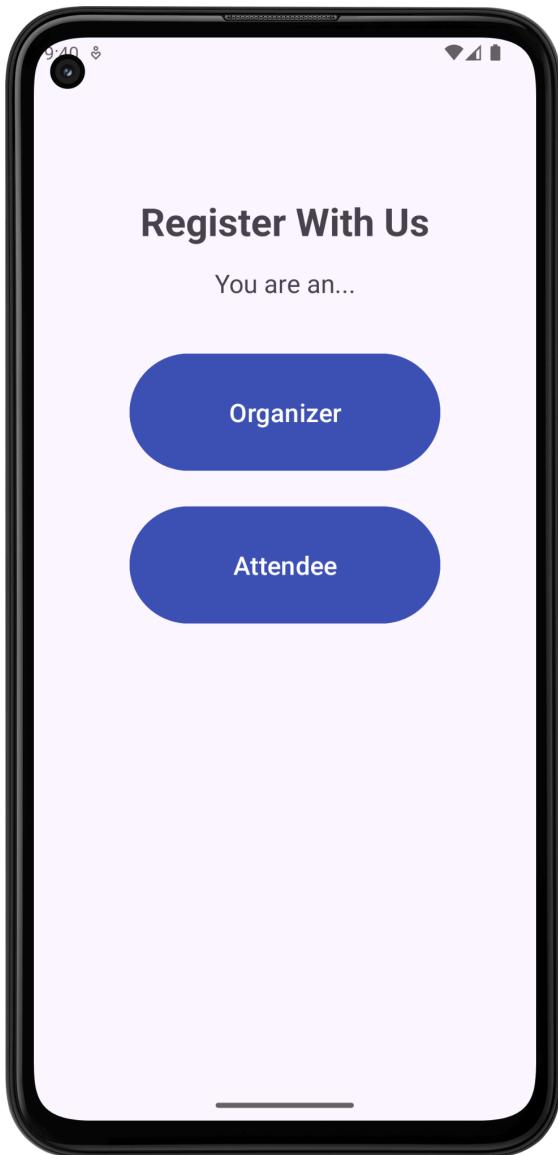
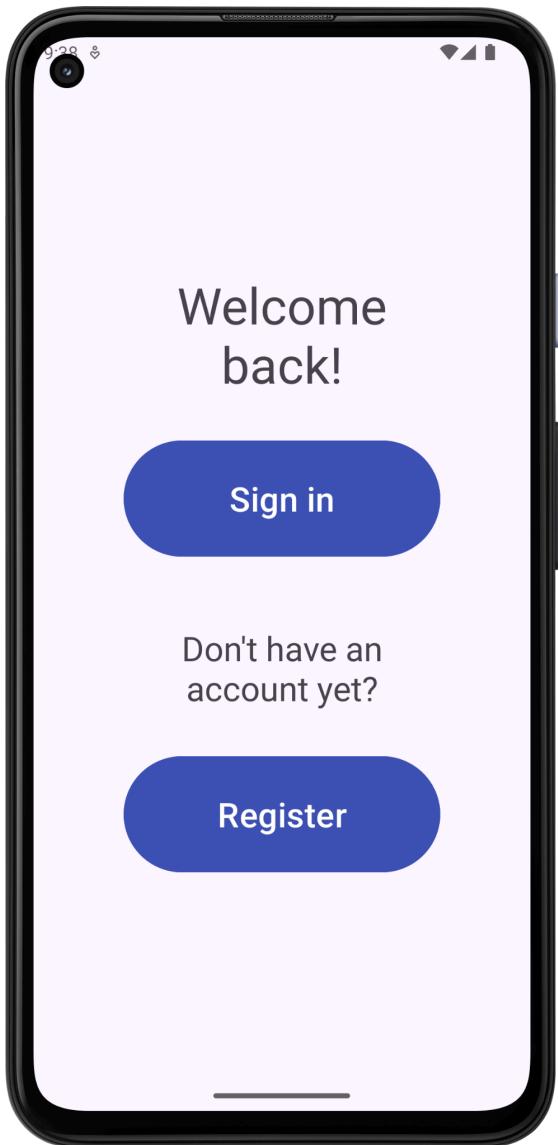
Deliverable One	
GitHub Classroom Setup	All members
Github Repository/Group Creation	Alina
Demo video Submission	All members
UML class diagram	All members
APK Submission	All members
Account Creation	Shiraz, Shaghayegh
Welcome Screen	Keno, Alina
Log Off Functionality	Enah, Hugo
Field Validation	Shaghayegh, Shiraz

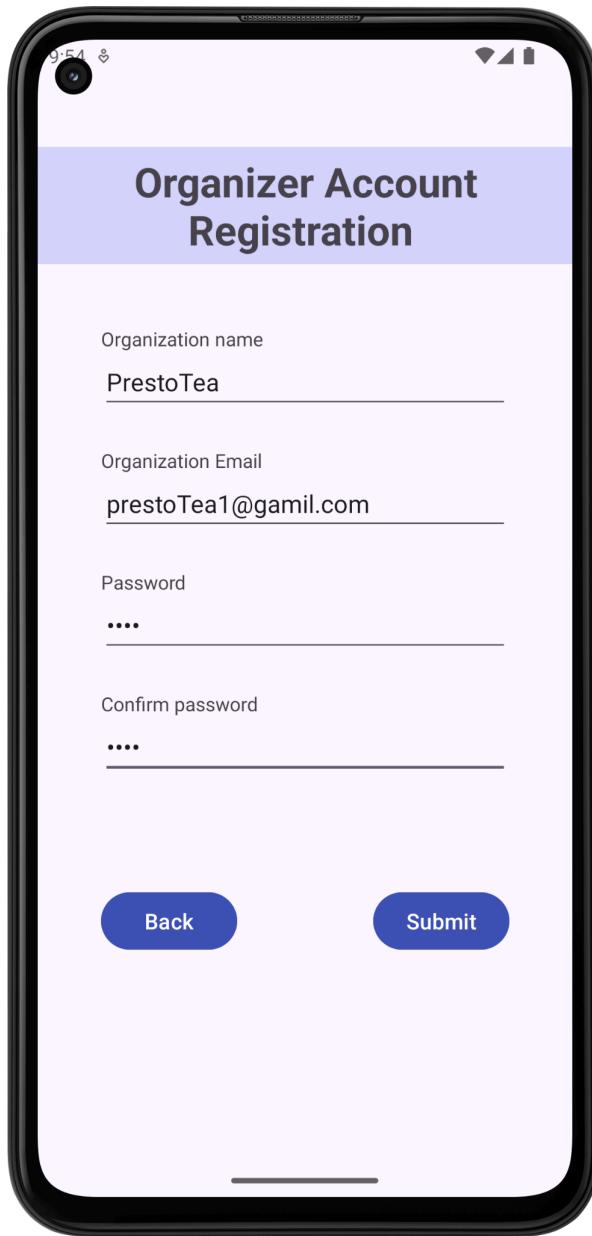
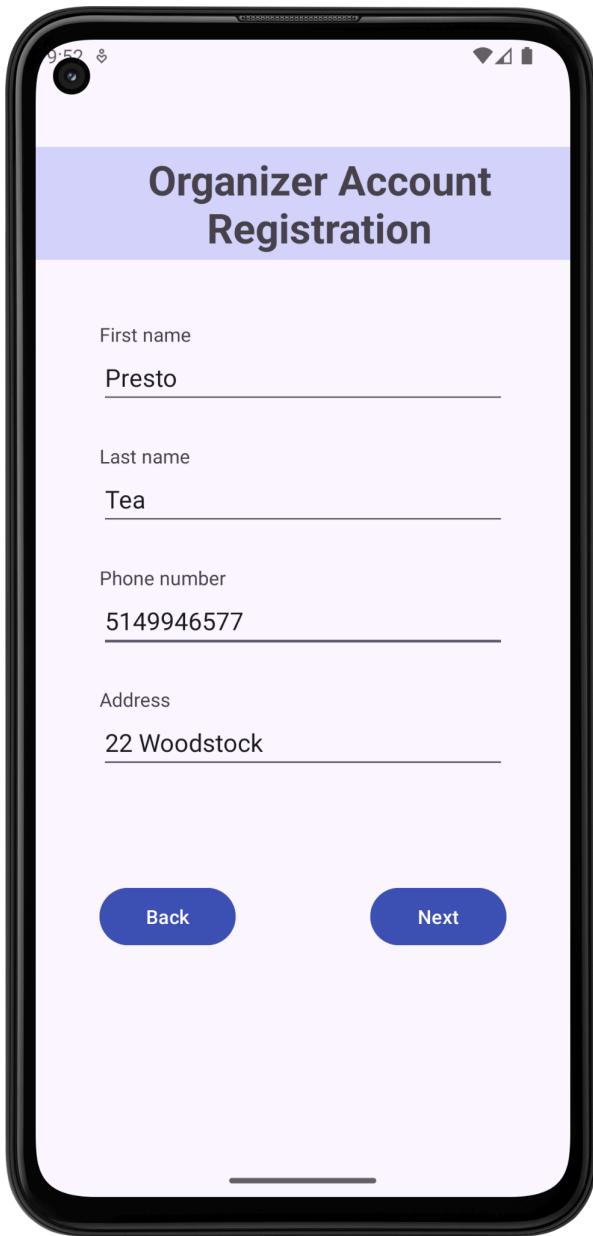
Deliverable Two	
UML Class Diagram	Enah, Keno
APK Submission	All members
Demo Video Submission	All members
The Administrator can view the list of registration requests.	Shaghayegh
The Administrator can view the information associated with each request.	Keno
The Administrator can approve or reject a registration request.	Shaghayegh, Keno
If a registration request is approved, it disappears from the list of registration requests.	Enah
If a registration request is rejected, it is added to the list of rejected registration requests.	Enah
The Administrator can view the list of previously rejected registration requests.	Hugo
The Administrator can approve a previously rejected request, which removes it from the rejected list.	Hugo
Login functionality based on registration status	Alina, Shiraz
Database Storage for Registration Requests	Shiraz, Shaghayegh

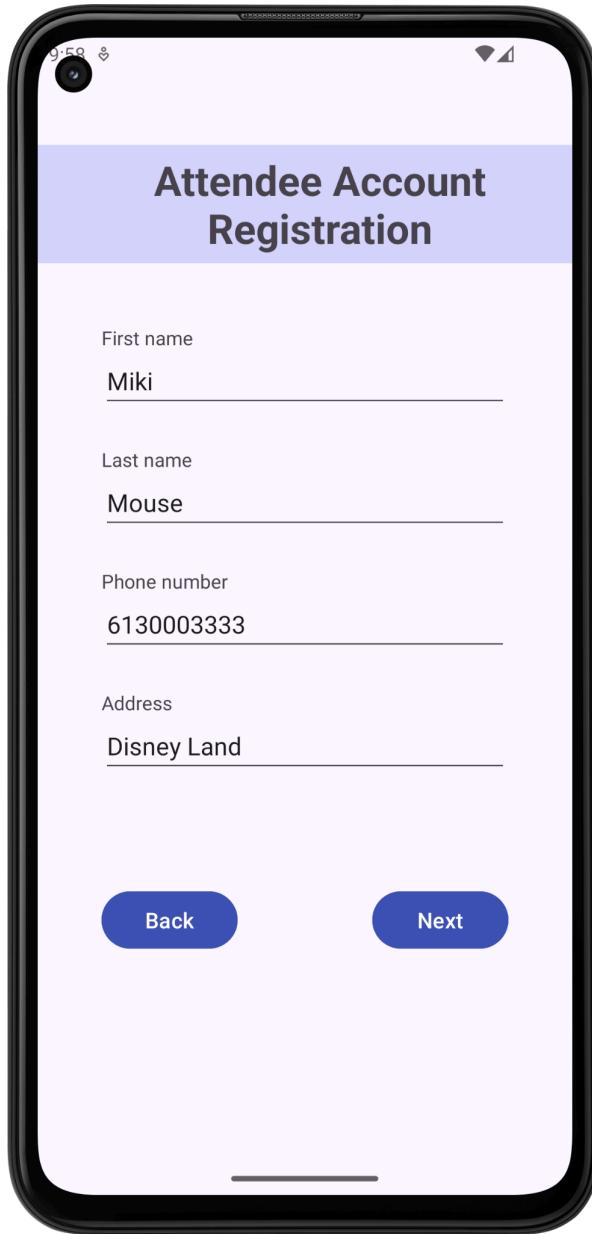
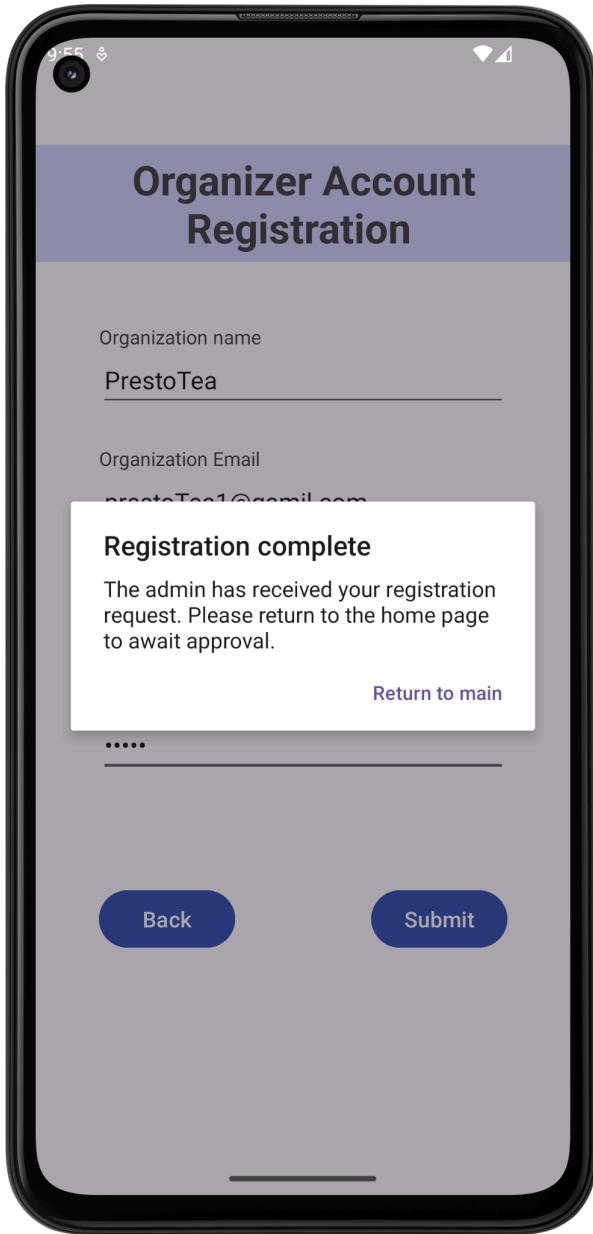
Deliverable Three	
UML Class Diagram	Enah, Keno
APK Submission	Alina
Field Validation and Error Messages	All members
Event Creation	Shiraz
Date Validation	Shiraz
Manual and Automatic Registration Approval	Alina
View Events	Shiraz
Attendee Registration Management	Keno, Shaghayegh
Event Deletion	Enah, Shiraz

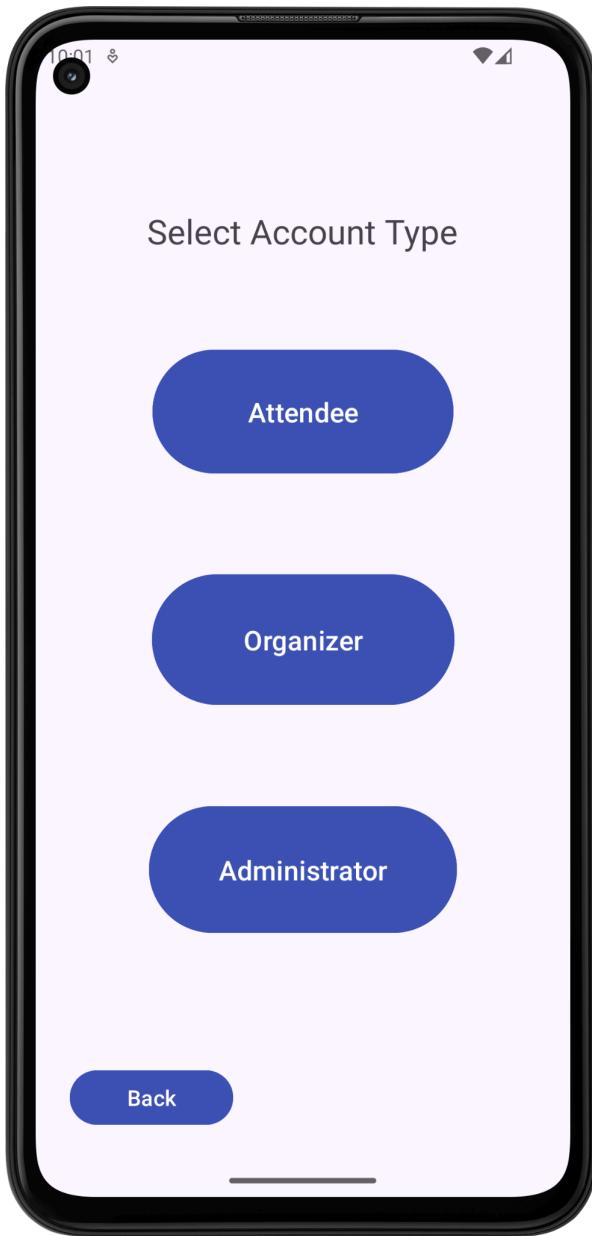
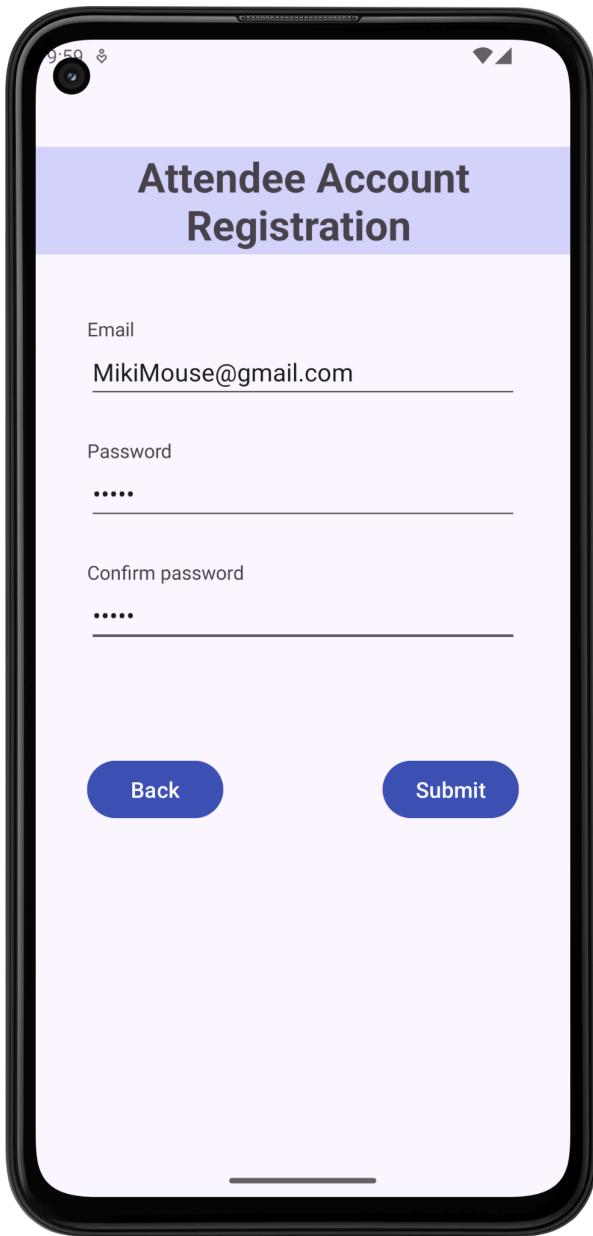
Deliverable Four	
UML Class Diagram	Enah, Keno
APK Submission	Alina
Demo Video Submission	Shaghayegh, Shiraz, Alina, Enah
Final Report	Shaghayegh, Shiraz, Alina, Enah
View Registered Events: The Attendee can view a list of events they have requested to register for, with the newest events displayed at the top.	Shaghayegh
Registration Status Indicator: There is an indicator showing whether the Attendee's registration is approved, rejected, or not processed yet by the Organizer.	Shaghayegh
Cancel Registration: The Attendee can cancel their registration for an event if it has not been processed or approved, provided the event is not scheduled to start within 24 hours.	Shiraz
Search Events: The Attendee can search for events by title or description, by specifying a keyword. Once a list of events is displayed in the search results, the Attendee can tap on any event to view its (title, description, date, start time, end time, and event address).	Keno
Request Event Registration: After finding an event, the Attendee can request registration. Once requested, the event is added to their list of events and disappears from search results.	Enah, Shiraz
Conflict Prevention: The system prevents the Attendee from registering for events that conflict with events they have already registered for or requested registration for.	Alina
Event Deletion	Hugo
Unit Test Cases	Shiraz
Field Validation	All members

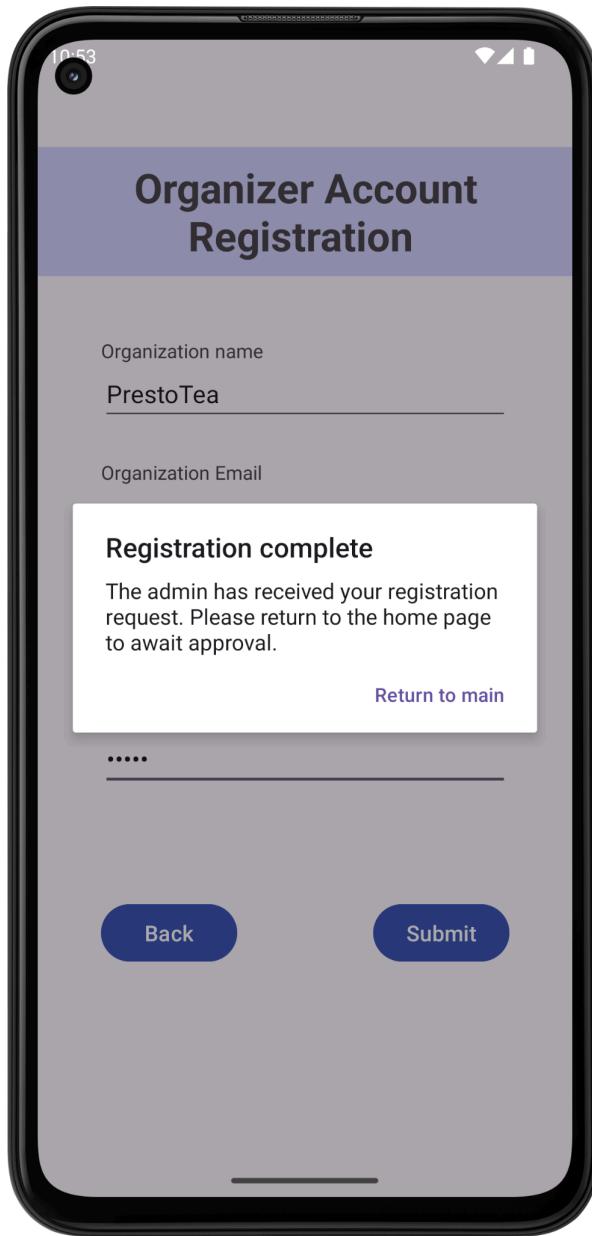
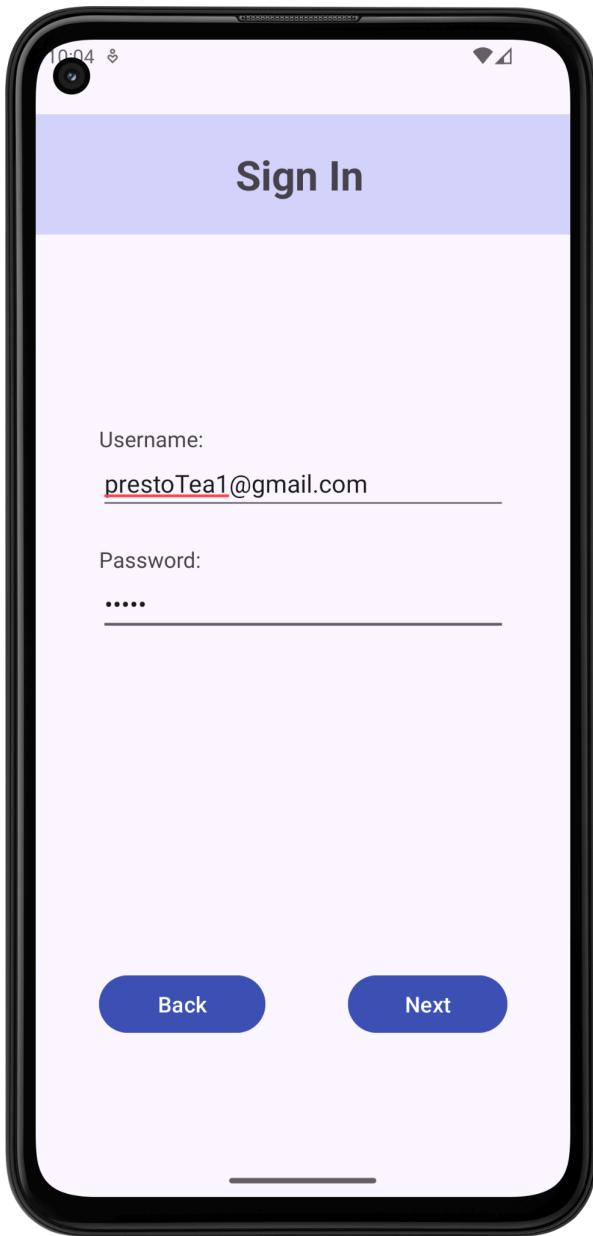
App screenshots

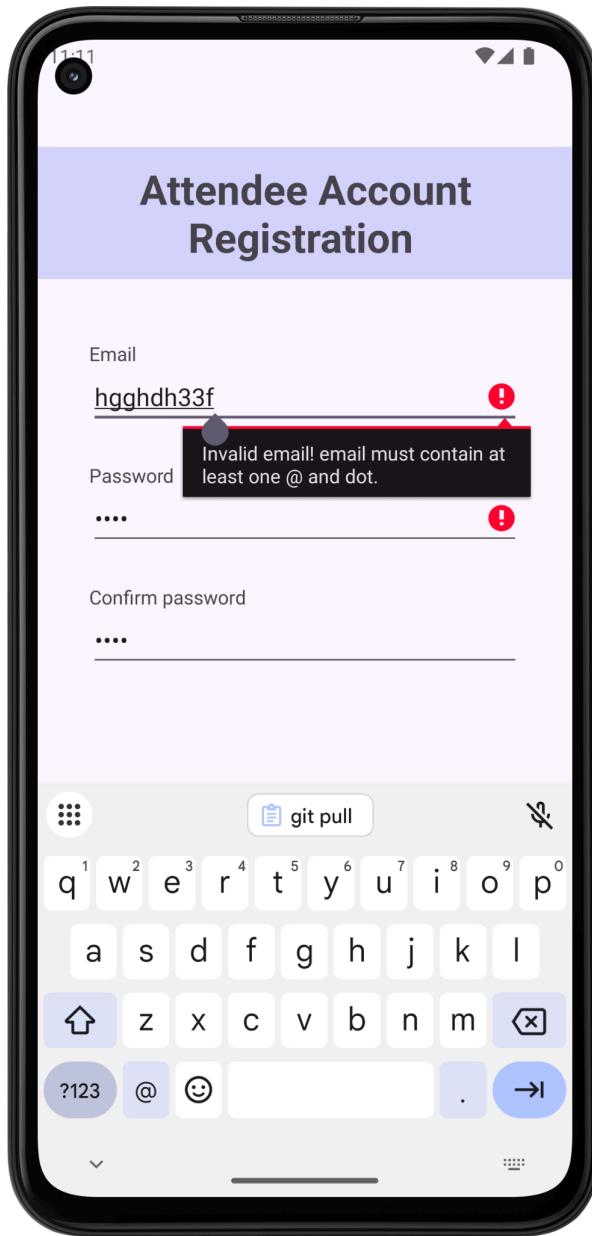
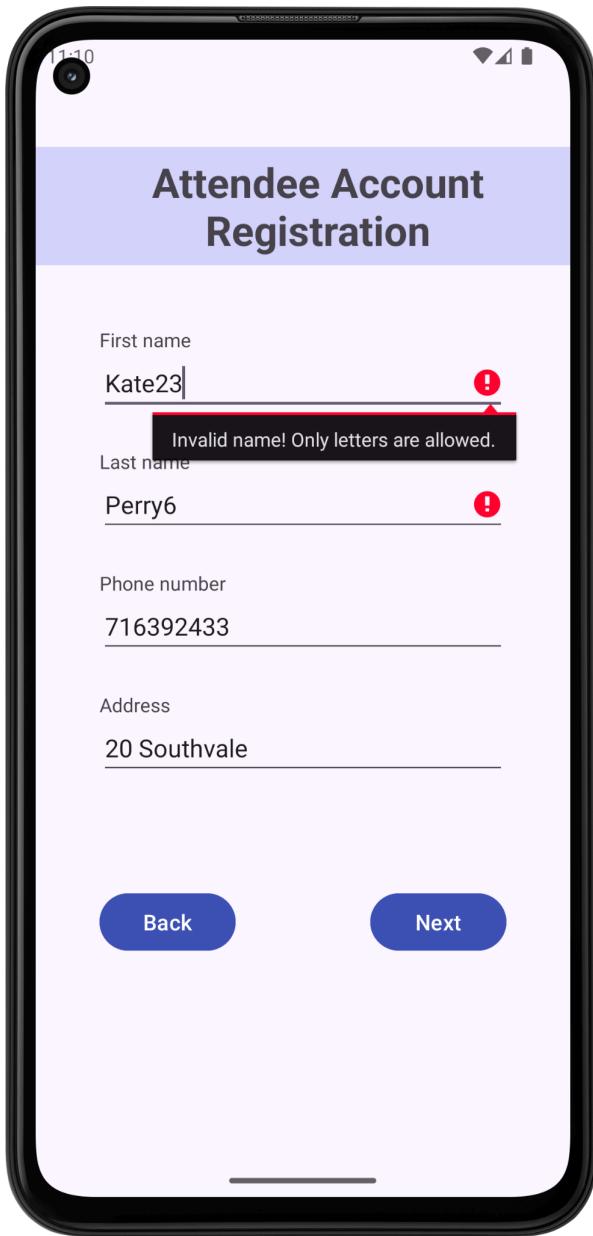


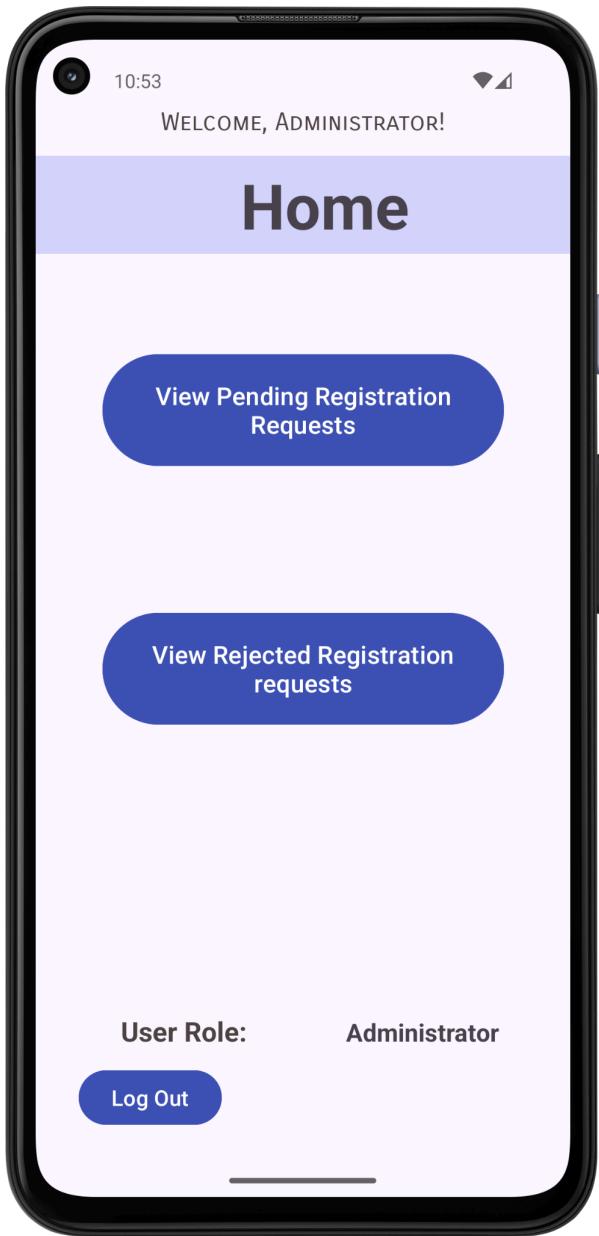
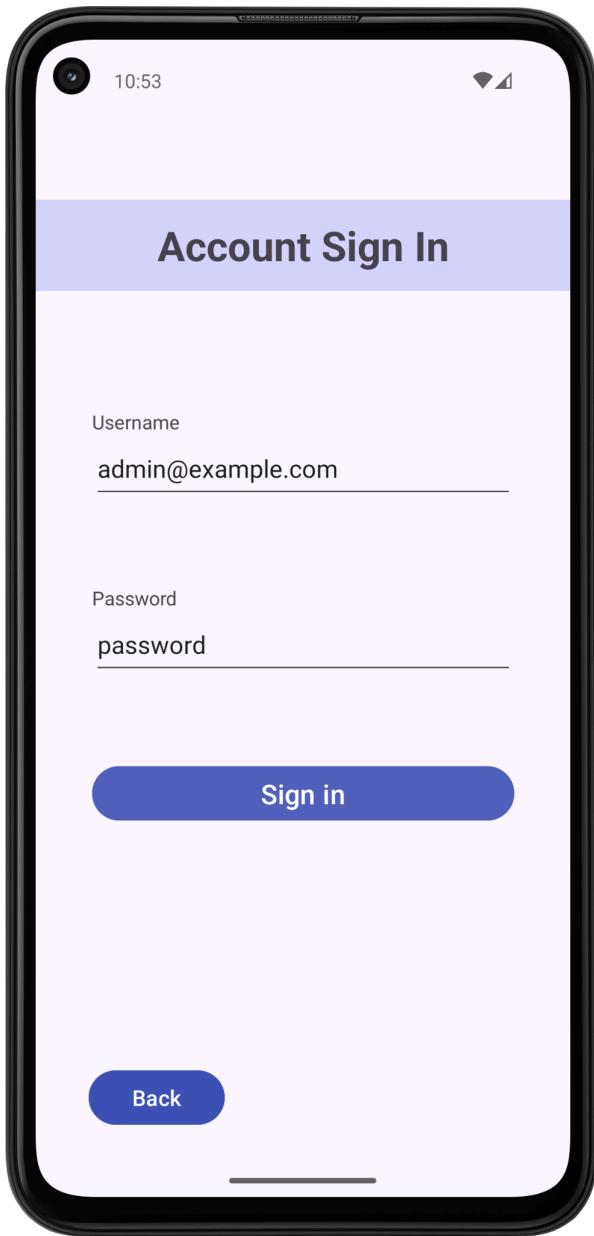


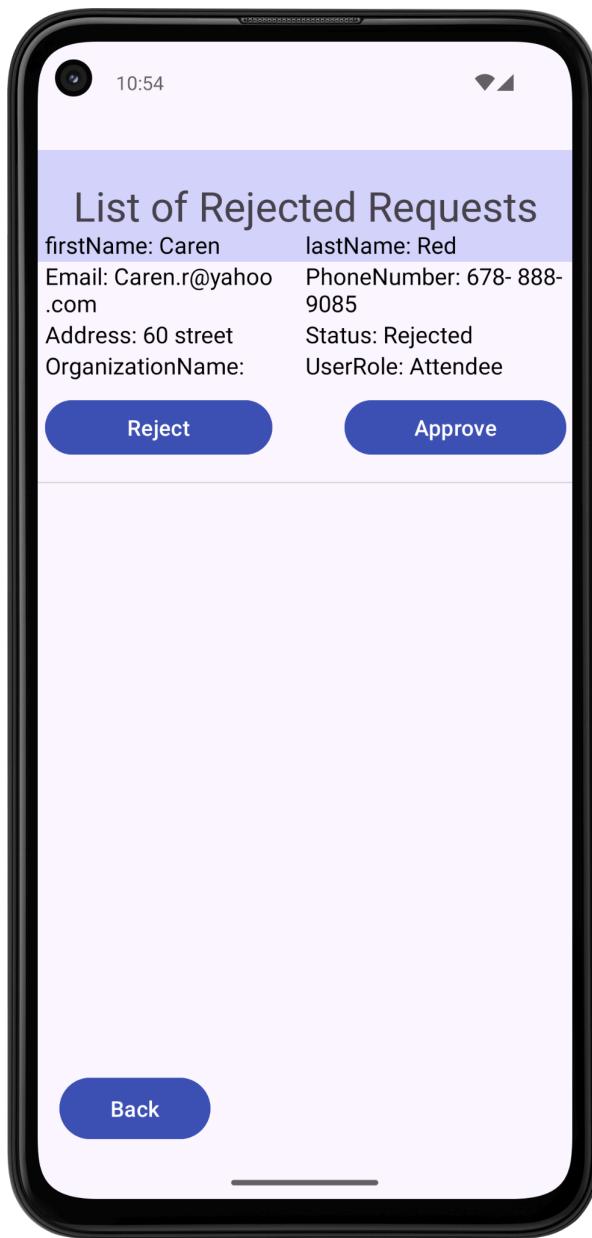
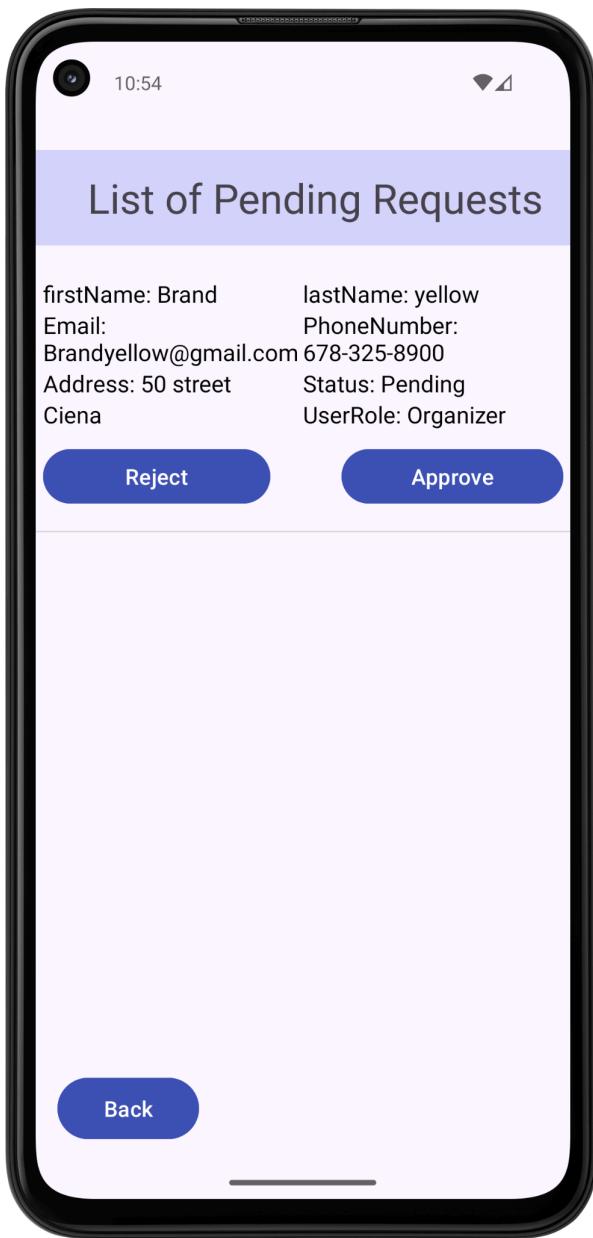


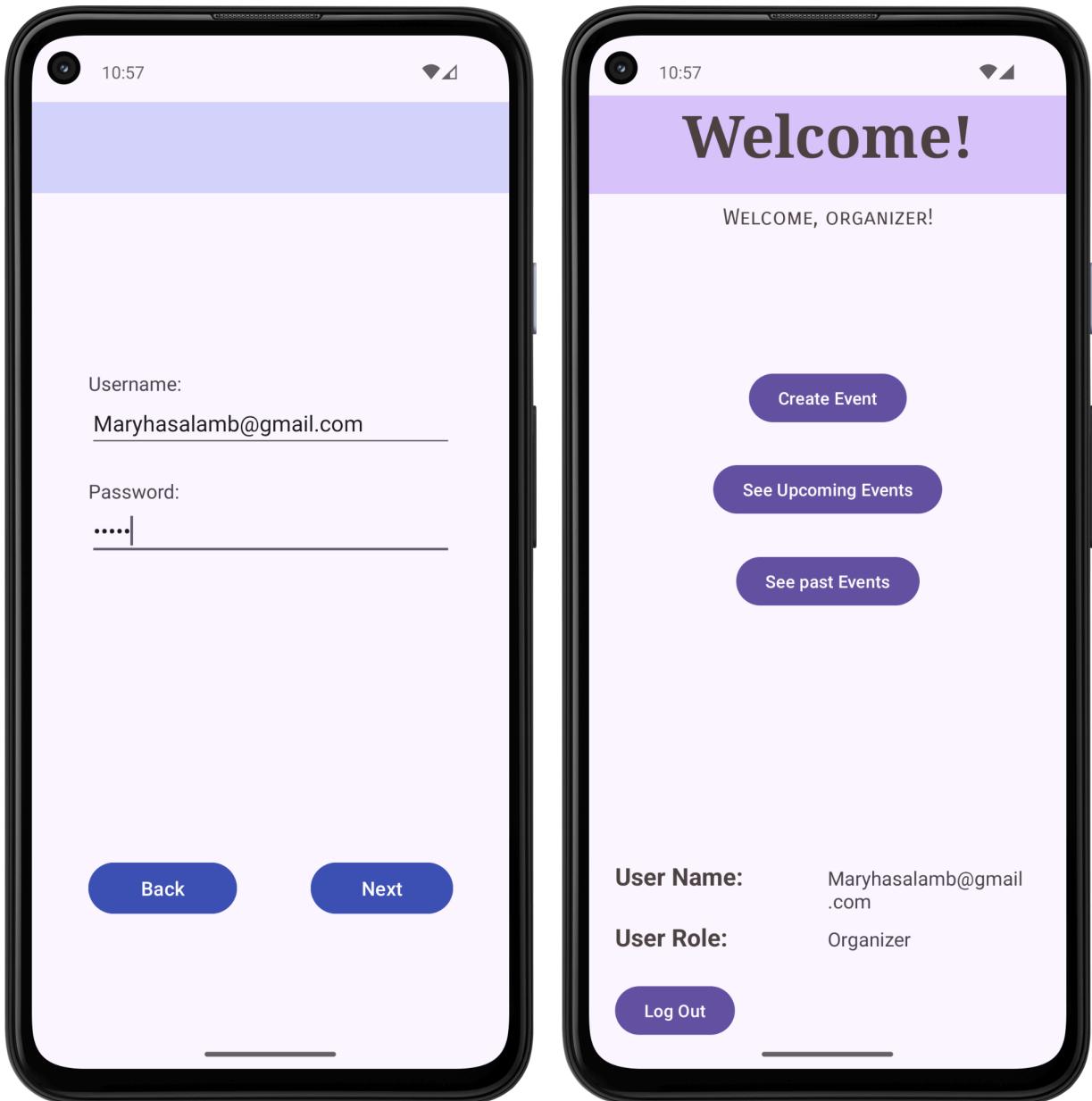


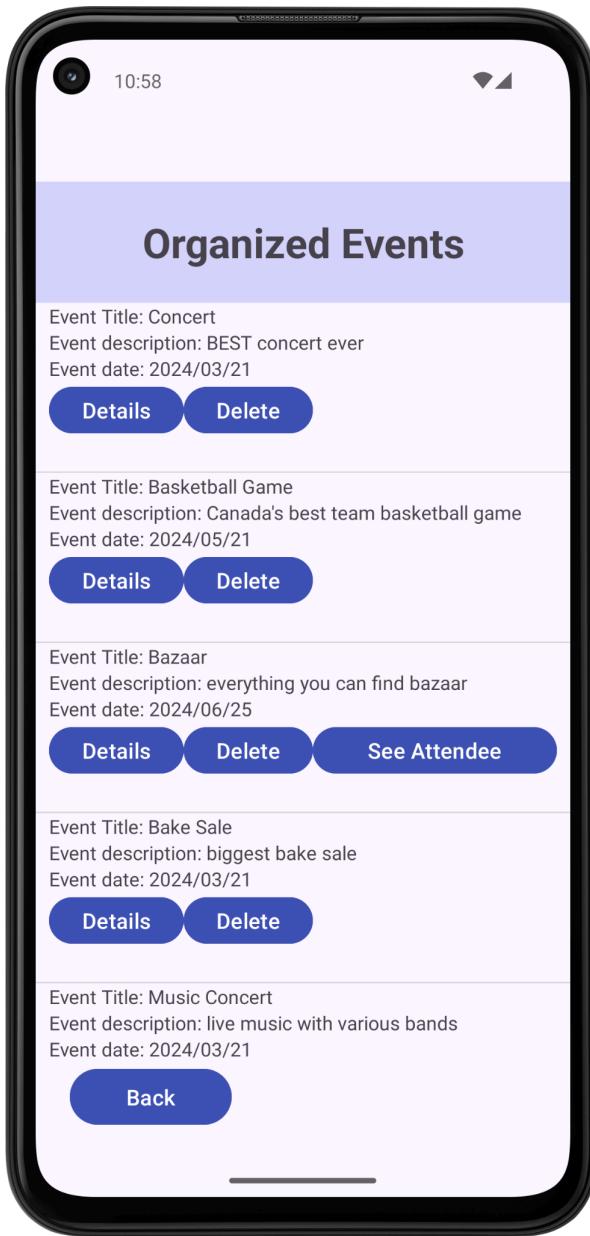
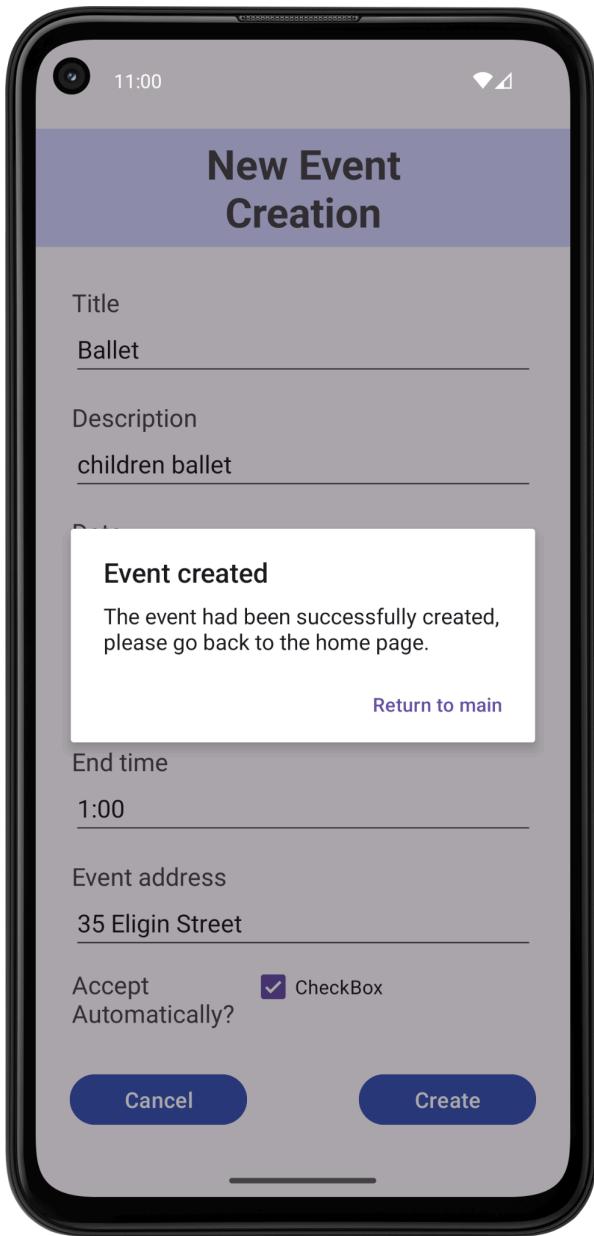


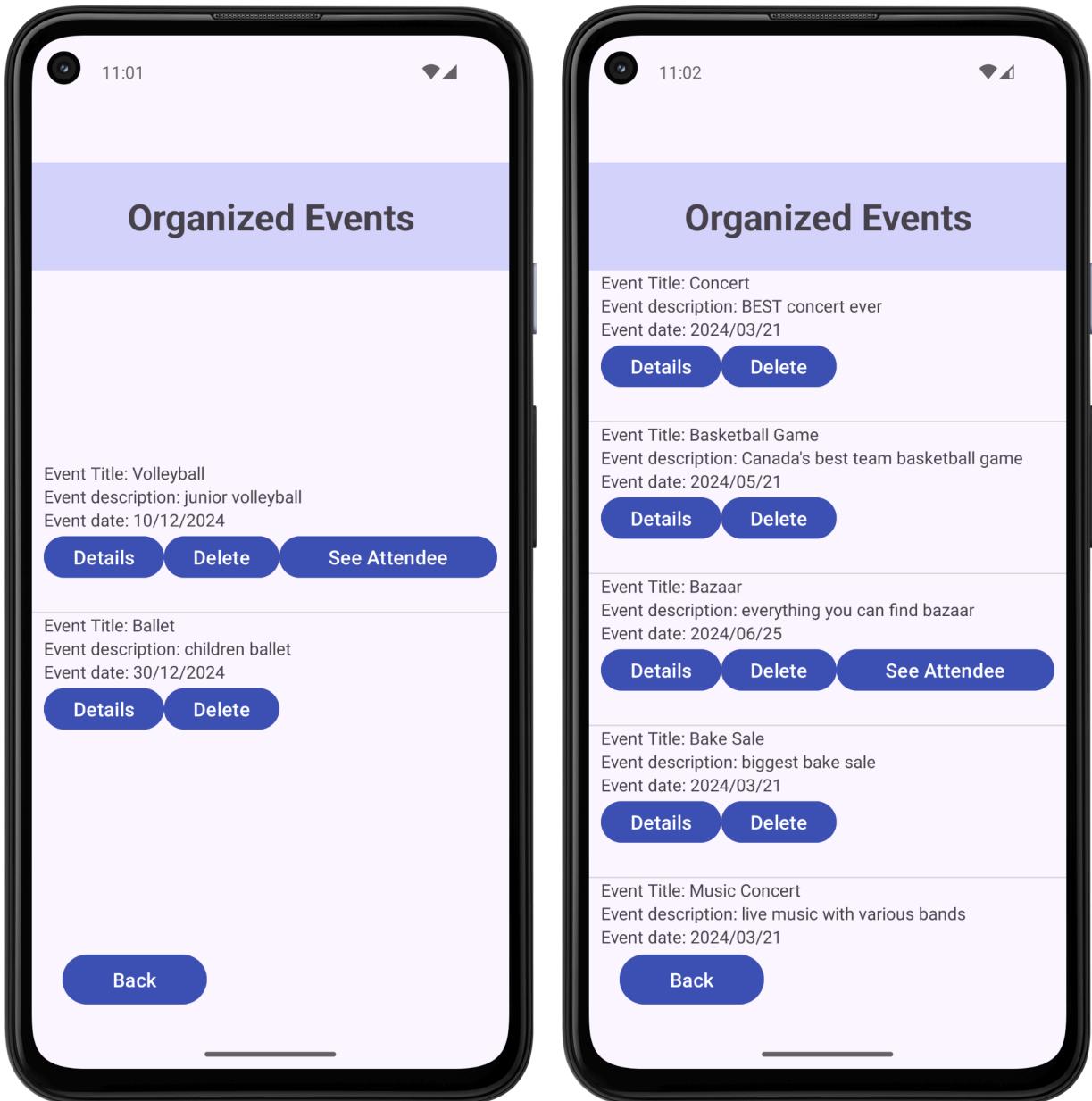


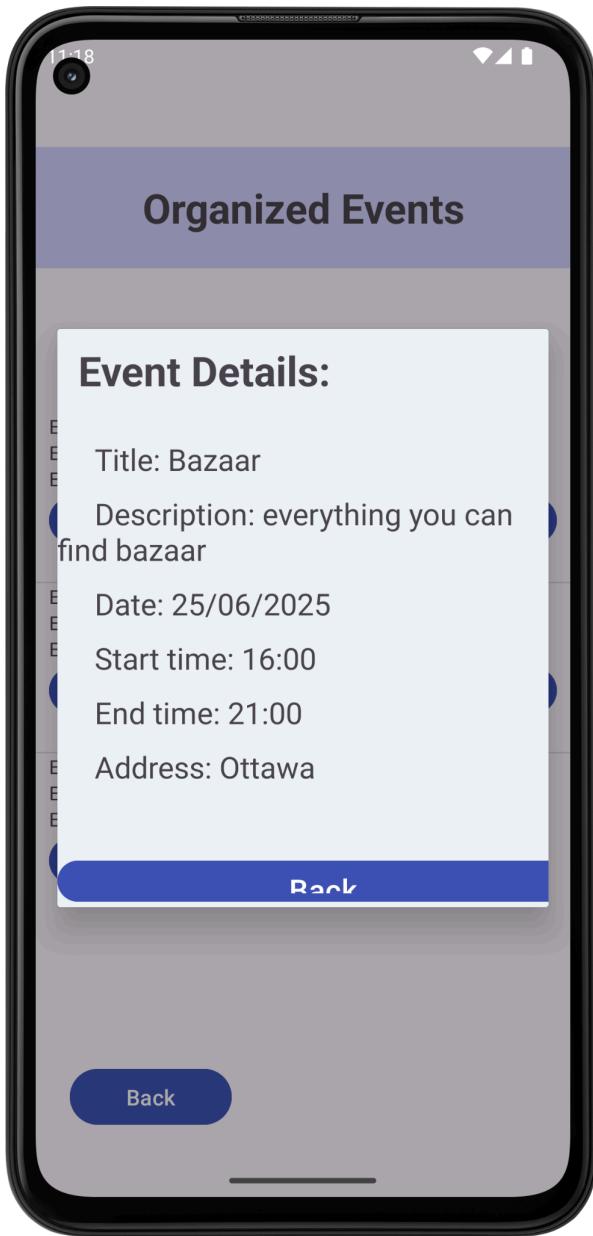


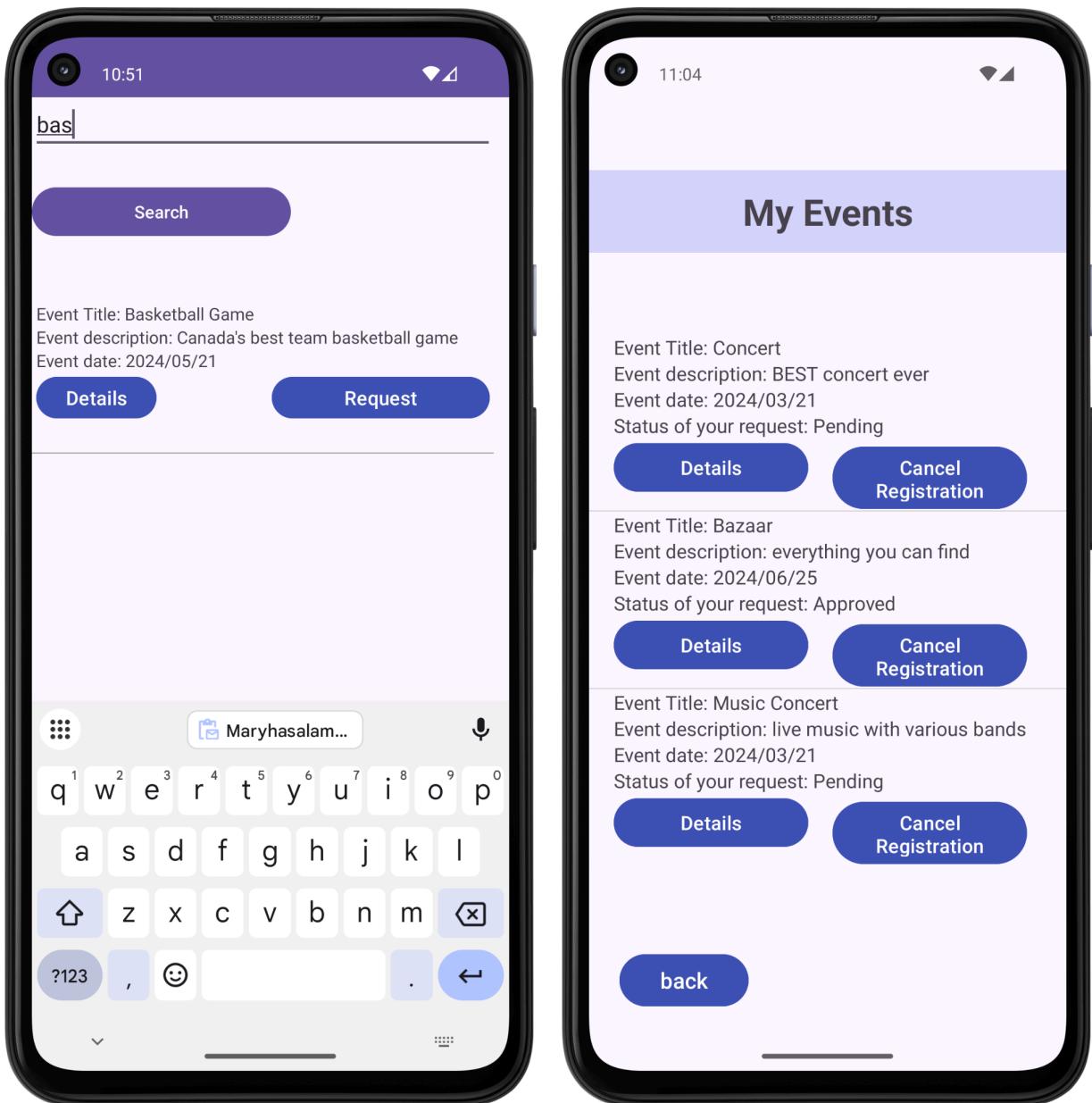


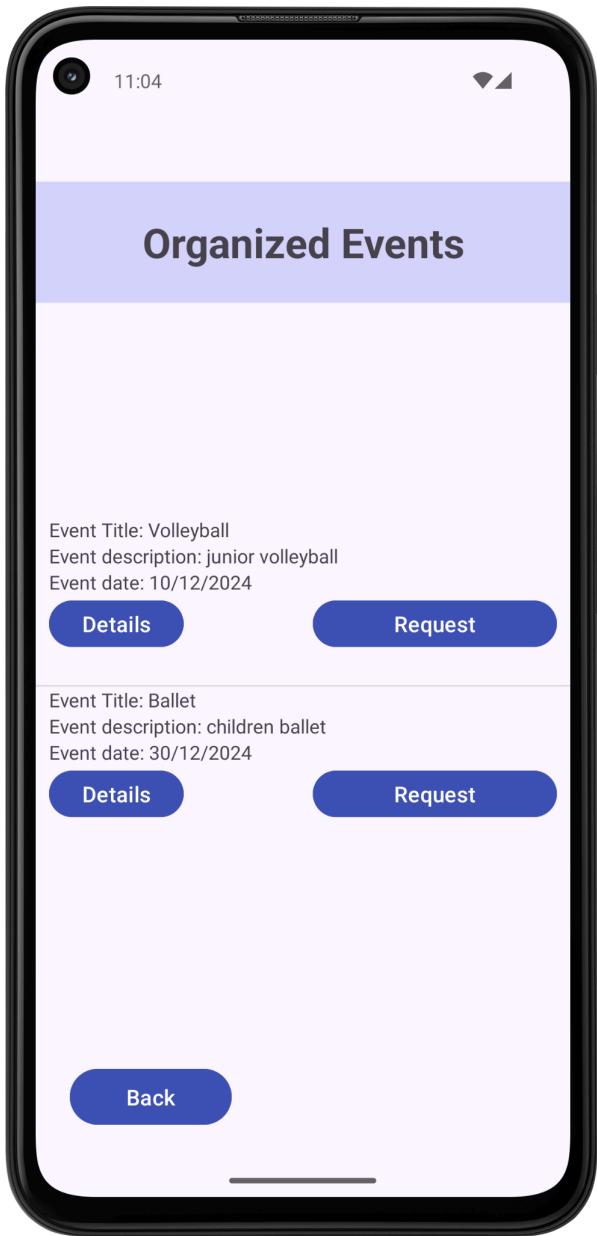












Lessons Learned:

- **Strengthening our Knowledge of Java:** Our Java programming skills improved not only through coding but perhaps even more from reading and understanding other people's implementations. This project allowed us to put our theoretical knowledge into actual use, letting us have a taste at what it is like to work in the technological industry.
- **Teamwork:** Whenever one was having trouble with their part, there would always be someone to help them and vice versa. Communication was key for smoothly completing our application. Providing clear descriptions and comments of methods and classes in the code was extremely helpful, as we can easily catch up to what others have been working on.
- **Time Management:** Scheduling regular group meetings and setting our own deadlines proved to be beneficial due to our different schedules and workloads. This allowed us to complete our deliverables on time — on the day of submission, all we had to do was review our code and film the video.
- **Operating Android Studio** Through Android Studio, we familiarize ourselves with all the necessary tools to build an app. This skill will be very beneficial for us when we tackle other android-based projects.
- **UI Design:** UI Design was a completely new concept to us. Fortunately, we learned and understood this skill quickly by putting it into practice in this project. By now, all team members are capable of creating pleasant UI designs and are capable of implementing them so that they fulfill their designated tasks.
- **Use of test cases:** Although we learned how to write test-cases in class, this project enabled us to finally put this skill into practice. It was confusing at first, but eventually, we were able to grasp the concept and write efficient test cases for our program. The test cases we wrote were very helpful in double checking each deliverable stage.
- **Efficiency:** Since our meetings were limited, we made sure to carefully address/fix any bugs we had in the code and delegate the next deliverable tasks. Meetings were efficient and straight-to-point so we made good progress. If any members were struggling, they would communicate via text messages. All team members chipped away at this project at their own pace, but self-imposed deadlines helped us in completing all the tasks.