

SEG2105[A] Final Report: Event Attendance Management System (EAMS)

Fall 2024
SEG2105[A] - Intro to Software Engineering
University of Ottawa
Group #2

Issac Jensen-Large,
Kunala Deotare,
Neelmani Bhardwaj,
Rachel Luo,
Shawn How,
and Veronica Nartatez

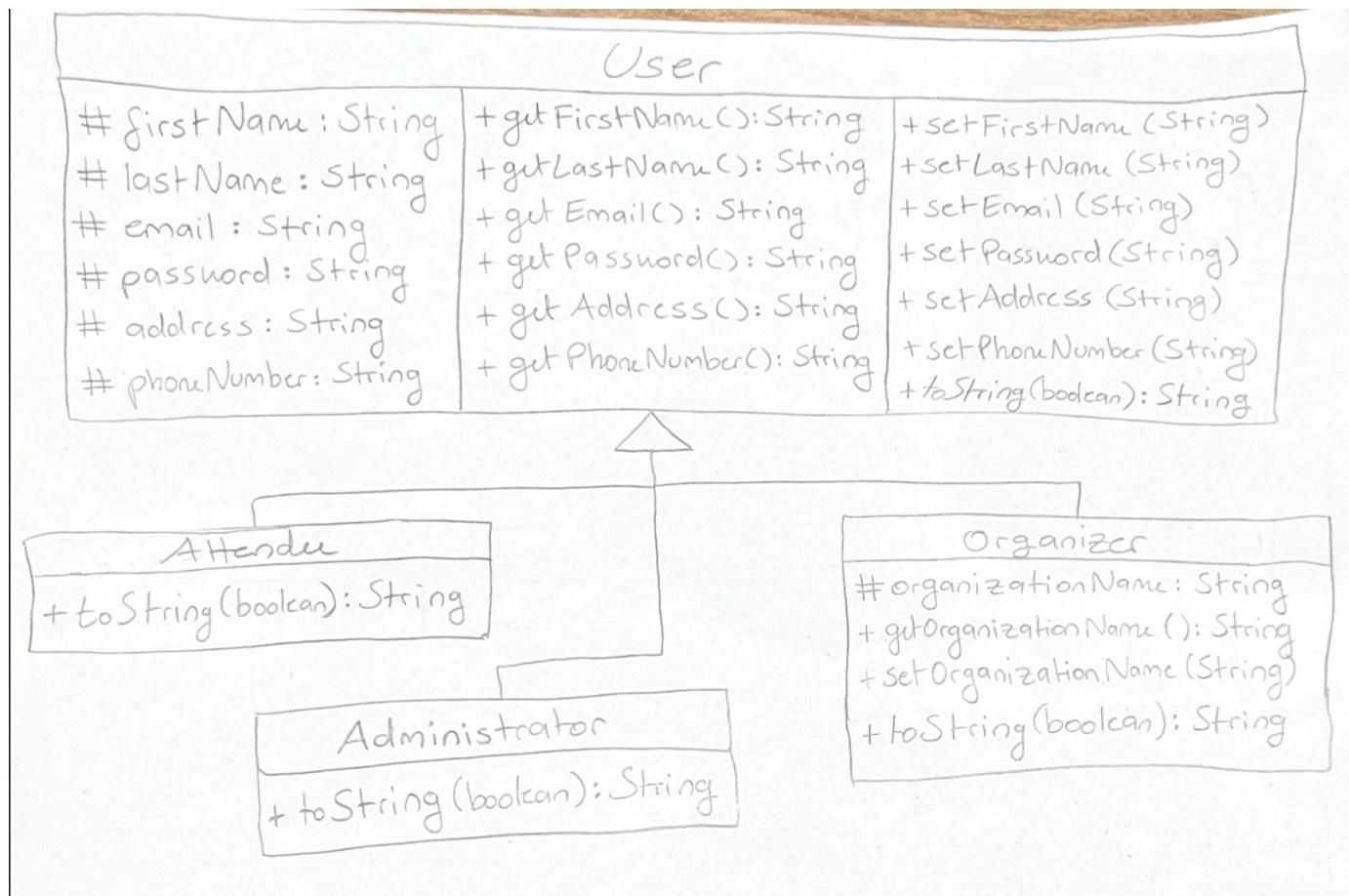
SEG2105[A]
Professor Hussein Al Osman
December 4th, 2024

1. Introduction

The purpose of this project was to develop an Event Attendance Management System (EAMS) that allows users to attend or organize events in their community. Our team approached this project using an agile management framework called Scrum, and we modified the process to fit an academic setting. Our overall goal was to reach MVP, the minimum viable product, to ensure we meet all deadlines and all requirements of each deliverable. We assigned project lead to Isaac Jensen-Large.

This final report will outline the overall UML class diagram of our project, the contributions of each team member of each deliverable, the screenshots of our app, and the lessons learned.

2. Final UML Diagram



UML Diagram #1: UML Diagram of Users and the Child Classes Attendee, Organizer, and Administrator

Account Registration Request

- firstName : String	+ get FirstName() : String
- lastName : String	+ get LastName() : String
- email : String	+ get Email() : String
- password : String	+ get Password() : String
- address : String	+ get Address() : String
- phoneNumber : String	+ get Phone Number() : String
- organizationName : String	+ get OrganizationName() : String
- status : Registration Request Status	
+ getStatus() : Registration Request Status	
+ setStatus(Registration Request Status)	

<<enum>>
 Registration Request Status
 PENDING
 REJECTED
 APPROVED

UML Diagram #2: UML Diagram of Object Account Registration Requests

Event	
<ul style="list-style-type: none"> - eventID : String - title : String - description : String - eventAddress : String - organizerEmail : String - dateMillis : long - startTimeMillis : long - endTimeMillis : long 	<ul style="list-style-type: none"> - registrationRequestsAreAutoApproved : boolean + getRegistrationRequestsAreAutoApproved() : boolean
<ul style="list-style-type: none"> + getEventID(): String + getTitle(): String + getDescription(): String + getEventAddress(): String + getOrganizerEmail(): String + getDateMillis(): long + getStartTimeMillis(): long + getEndTimeMillis(): long 	<ul style="list-style-type: none"> + getLocalDate(): LocalDate + getLocalStartTime(): LocalTime + getLocalEndTime(): LocalTime
<ul style="list-style-type: none"> + setEventID(string) 	<ul style="list-style-type: none"> - approvedRequests : HashMap<String, String> - pendingRequests : HashMap<String, String> - rejectedRequests : HashMap<String, String> + getApprovedRequests(): HashMap<String, String> + getPendingRequests(): HashMap<String, String> + getRejectedRequests(): HashMap<String, String> + addRequest(String, RegistrationRequestStatus)

UML Diagram #3: UML Diagram of Object Event

3. The Team's Contribution

The group was separated into 3 teams:

- View (frontend) → Neel and Veronica
- View Model (connects frontend and backend) → Isaac and Shawn
- Modal (backend) → Kunala and Rachel

3.1 Deliverable One

Deliverable One Requirements	Contributors
Github Classroom Setup	Isaac
Repository Contributions	Everyone
Demo Video Submission	Kunala
UML Class Diagram	Kunala
APK Submission	Isaac
Account Creation	20%Veronica 40% Kunala 40%Rachel
Welcome Screen	30%Kunala
Logout Functionality	Shawn
Field Validation	20% Kunala 80%Isaac

3.2 Deliverable Two

Deliverable Two Requirements	Contributors
UML Class Diagram	Kunala
APK Submission	Isaac
Admin Feature #1: The Administrator can view the list of registration requests.	50% Veronica 50% Neel
Admin Feature #2: The Administrator can view the information associated with each request	Veronica
Admin Feature #3: The Administrator can approve or reject a registration request.	30% Veronica 40% Isaac 15% Kunala 15% Rachel
Registration Request Management Feature #1: If a registration request is approved, it disappears from the list of registration requests.	30% Veronica 40% Isaac 15% Kunala 15% Rachel
Registration Request Management Feature #2: If a registration request is rejected, it is added to the list of rejected registration requests.	30% Veronica 40% Isaac 15% Kunala 15% Rachel
Rejected Request Management Feature #1: If a registration request is approved, it disappears from the list of registration requests	50% Neel 50% Veronica
Rejected Request Management Feature #2: If a registration request is rejected, it is added to the list of rejected registration requests	30% Veronica 40% Isaac 15% Kunala 15% Rachel
Login Functionality	Isaac
Data Storage	20% Shawn 40% Kunala

	40% Rachel
--	------------

3.3 Deliverable Three

Deliverable Three Requirements	Contributors
UML Class Diagram	Kunala
APK Submission	Isaac
Demo Video	Kunala
Event Creation	10% Neel 10% Shawn 40% Kunala 40% Rachel
Date Validation	Shawn
Manual and Automatic Registration Approval	33% Neel 67% Rachel
View Events Feature #1: The Organizer can view a list of upcoming events.	50% Neel 25% Isaac 25% Shawn
View Events Feature #2: Organizer can view a list of past events.	50% Neel 25% Isaac 25% Shawn
Attendee Registration Management features #1 & #2: The Organizer can view a list of Attendees who requested registration for an upcoming event and the Organizer can view the information for each Attendee.	33% Veronica 66% Neel
Attendee Registration Management Feature #3: The Organizer can approve or reject each Attendee's registration individually or select "approve all" to approve all registrations at once.	Isaac
Event Deletion	55% Kunala

	35% Rachel 10% Shawn
Field Validation	Shawn

3.4 Deliverable Four

Deliverable Four Requirements	Contributors
UML Class Diagram	Kunala
APK Submission	Isaac
Demo Video	Kunala
Final Report	Veronica
View Registered Events	25% Neel 25% Veronica 50% Kunala
Registration Status Indicator	Veronica
Cancel Registration	33% Shawn 67% Rachel
Search Events	25% Neel 25% Veronica 50% Kunala
Request Event Registration	33% Shawn 67% Rachel
Conflict Prevention	25% Shawn 75% Rachel
Event Deletion	75% Isaac 25% Kunala
Field Validation	Isaac
Unit Testing	75% Kunala 25% Rachel

4. Screenshots of the App

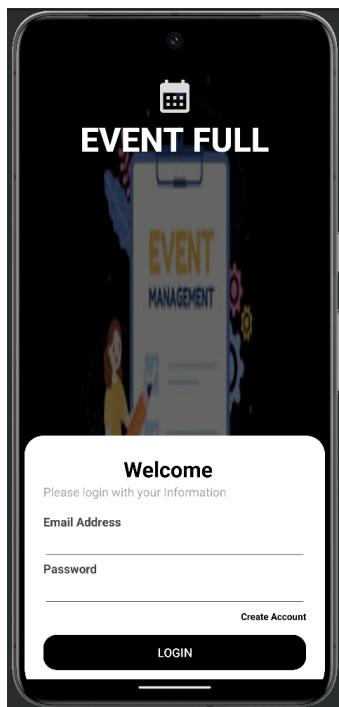


Figure 1: EAMS Login Page

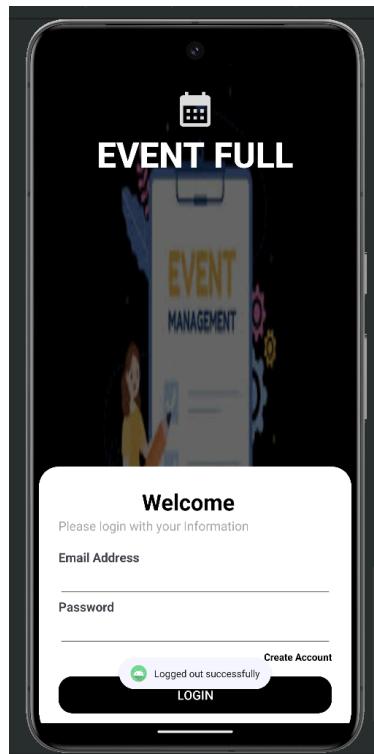


Figure 2: EAMS Logout Toast

4.1 Admin's Point of View

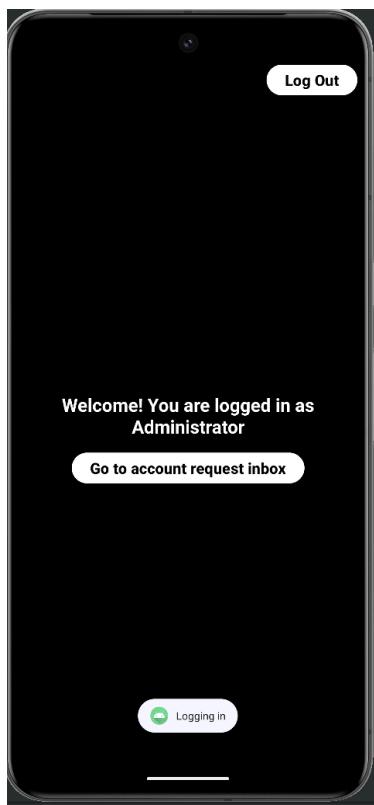


Figure 3: Admin Welcome Page

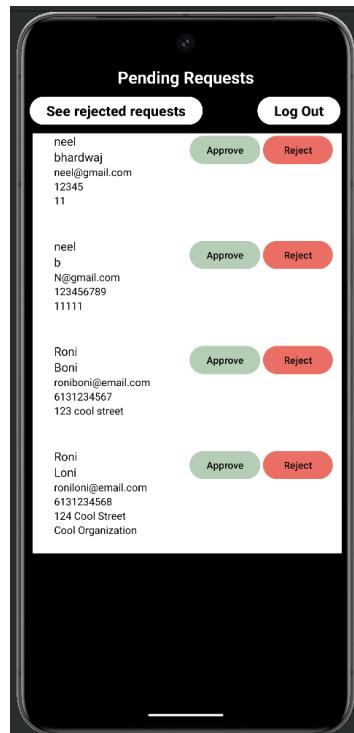


Figure 4: Admin's Screen of Pending Account Requests

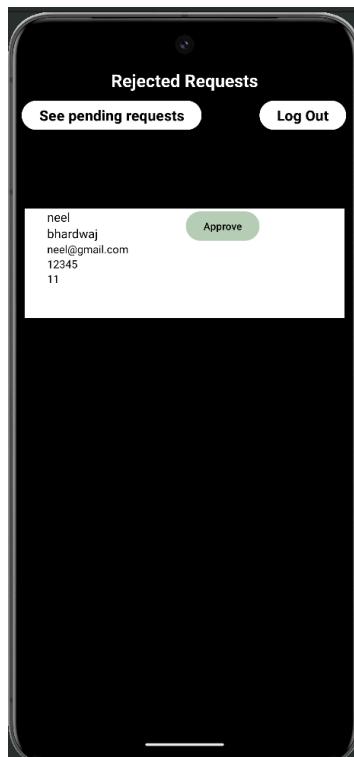


Figure 5: Admin's Screen of Rejected Account Requests

4.2 Organizer's Point of View

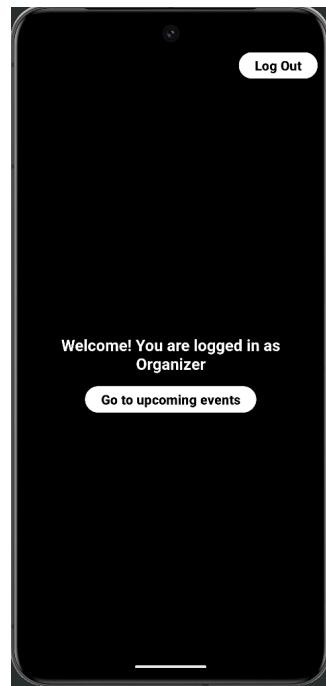


Figure 6: Organizer's Welcome Page

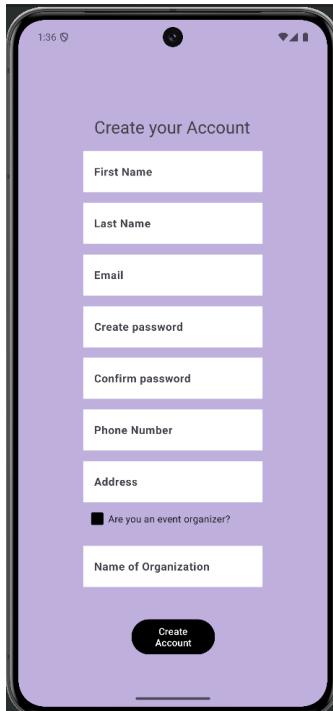


Figure 7: Organizer's Register Page

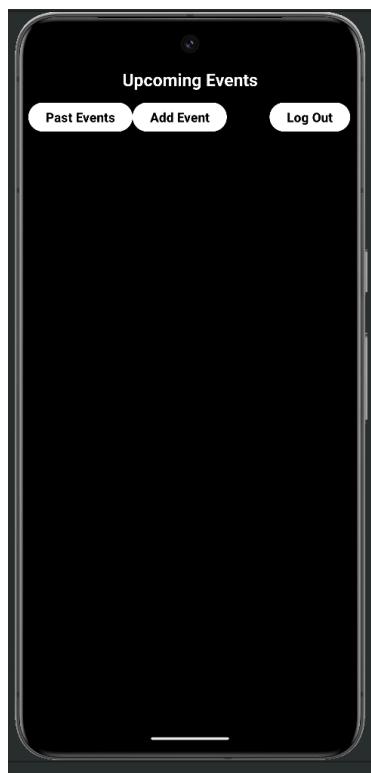


Figure 8: Organizer's Screen that Displays Their Upcoming Events Without Any Events

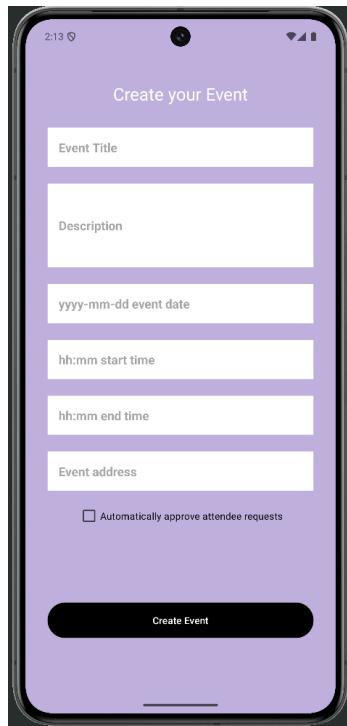


Figure 9: Organizer's "Create a New Event" Screen

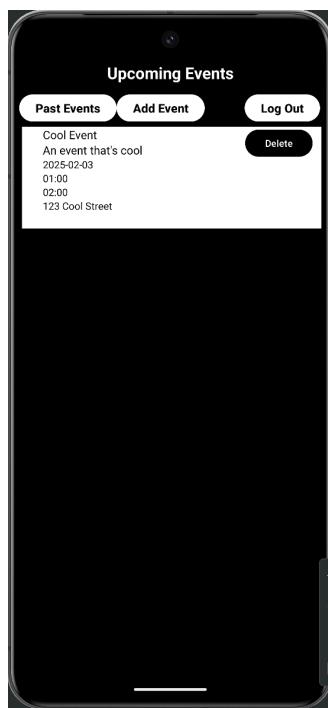


Figure 9: Organizer's Screen that Displays Their Upcoming Events Screen with an Upcoming Event

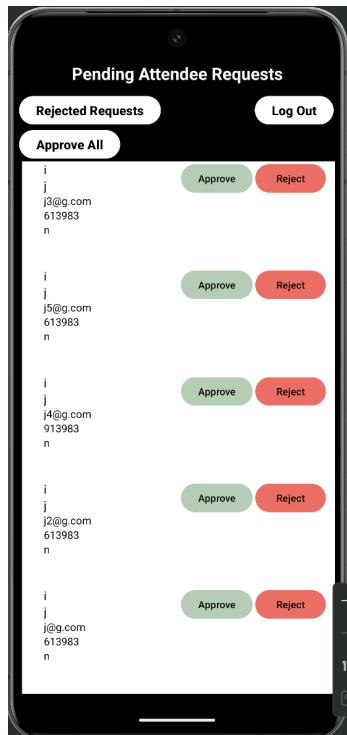


Figure 10: Organizer's Screen that Displays all Pending Attendee Requests to a Specific Upcoming Event, with the Option of Approving All Rejected Requests

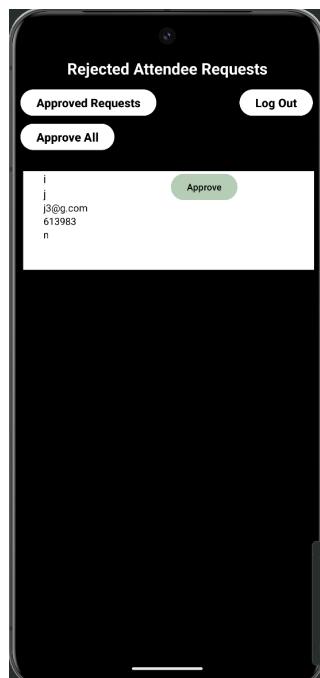


Figure 11: Organizer's Screen that Displays all Rejected Attendee Requests to a Specific Upcoming Event, with the Option of Approving All Rejected Requests

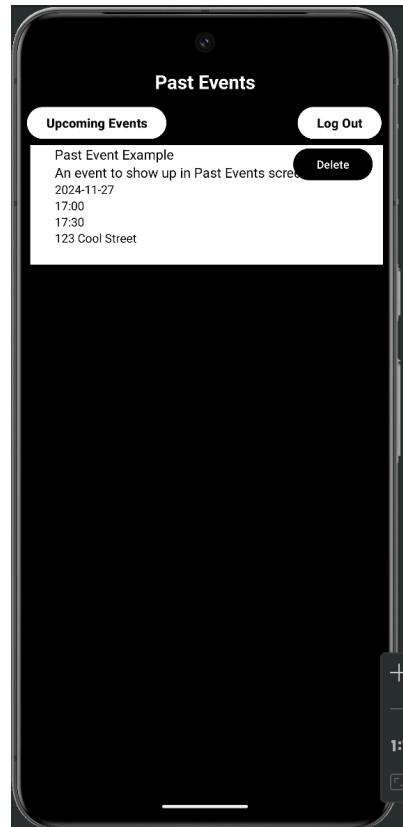


Figure 12: Organizer's Screen that Displays Their Past Events Screen with a Past Event

4.3 Attendee's Point of View

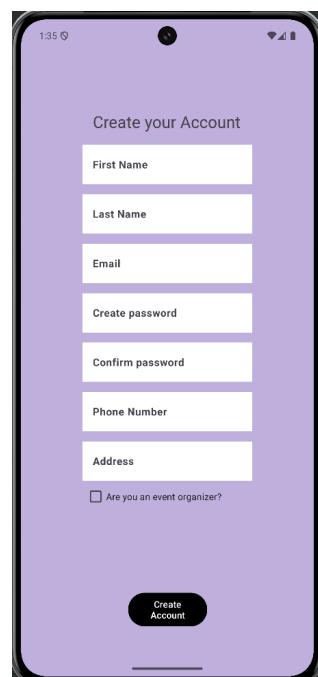


Figure 13: Attendee Register Page

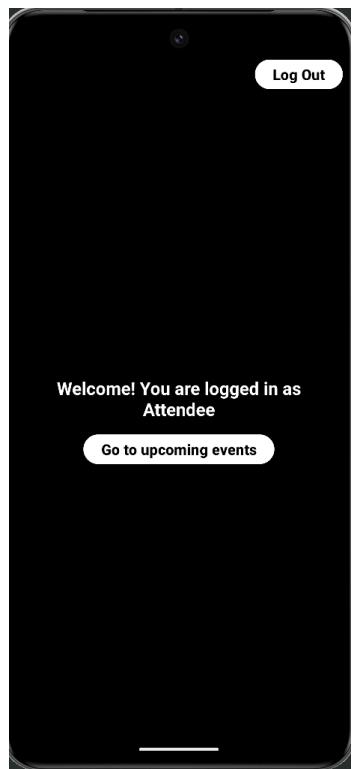


Figure 14: Attendee's Welcome Page

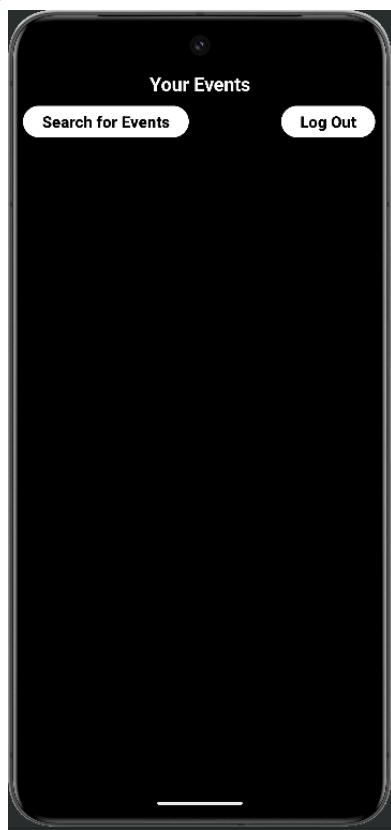


Figure 15: Attendee Screen of Upcoming Events (No Events)

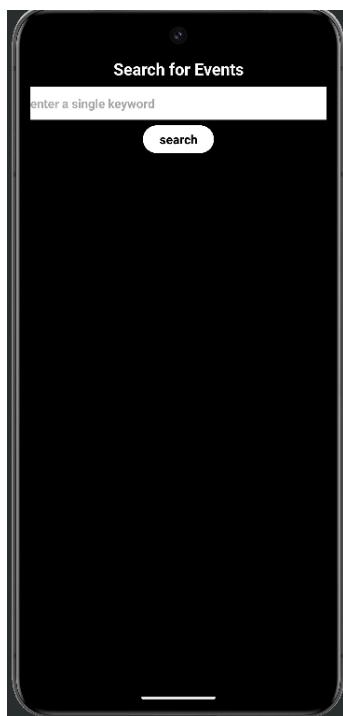


Figure 16: Attendee Screen to Search for New Events before Searching

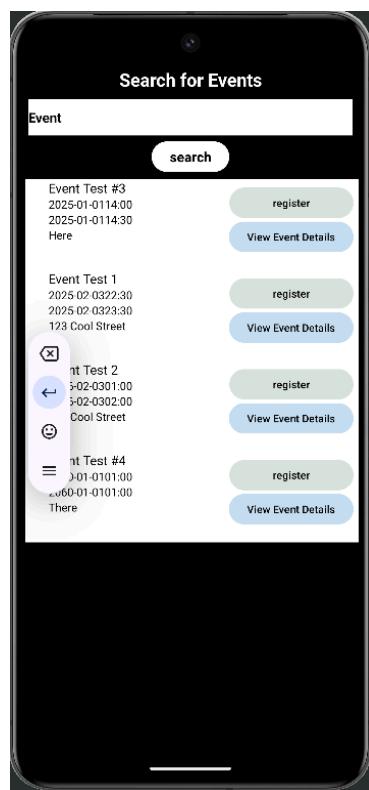


Figure 17: Attendee Screen to Search for New Events After Searching with a Keyword

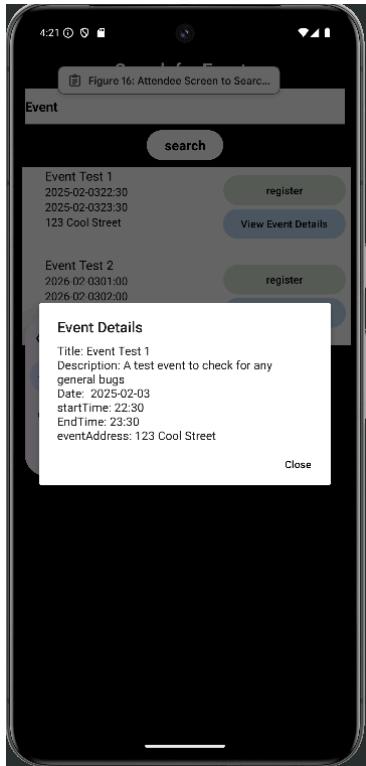


Figure 18: Attendee Search Screen with Dialogue that Displays Extra Information About an Event

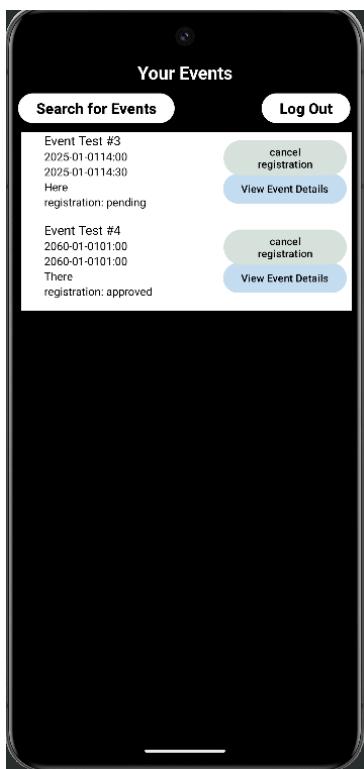


Figure 19: Attendee “Your Events” Screen with Events They Have Registered For

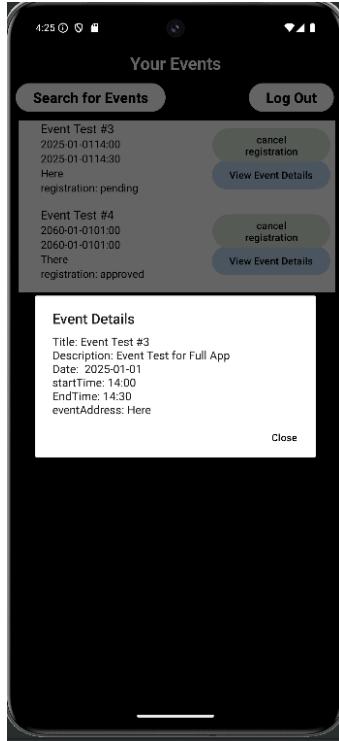


Figure 20: Attendee “Your Events” Screen with Dialogue that Displays Extra Information About an Event They’ve Registered F

5. Lessons Learned

As mentioned before, our team approached this project using scrum. During this process, we learned a lot of fundamental skills that came with working on a project with multiple people.

When making pull requests (PRs), we learned the importance of PR comments and commit comments. By being specific and explaining the changes you made, it helps the PR reviewer know exactly what features were added or what changes were made. Not only that, but it will also be easier to track any bugs in our code.

Having specific and detailed comments did not only apply to PRs, but to our documentation as well. Working on a big project means that not everyone will know the implementation of everyone’s code. We learned that proper documentation, such as javadocs, facilitates using each other’s code.

Working with multiple people taught us how important it was to keep naming conventions consistent. During deliverable 1-3, the UI-side had inconsistent naming conventions, which made it harder for other team members to code.

We also learned the importance of effective research and how to read documentation. Since we were new to Android Studio, some of us had to rely on the documentation throughout the development process. This experience helped us become more proficient in researching and utilizing documentation when exposed to new software tools.

Some of us were new to the scrum process and github. What we learned was the importance of git pull, and making sure you were working on the right branch *before* you start

coding. Doing this would prevent any merge conflicts and any interference between each other's work.

As project lead, Isaac said, "The importance of effective communication was very clear to me in this project. The ability to understand the expectations and concerns of other team members and to integrate them with my own expectations and concerns was highly valuable to the degree that I was able to do it. When doing difficult work that requires the coordination of multiple individuals' efforts, this ability to integrate perspectives allows the team to adapt to varying schedules, priorities, and capabilities. The strengths of teamwork become clear when effective communication is present".