EAMS App Report
Group 12
SEG2105[A]

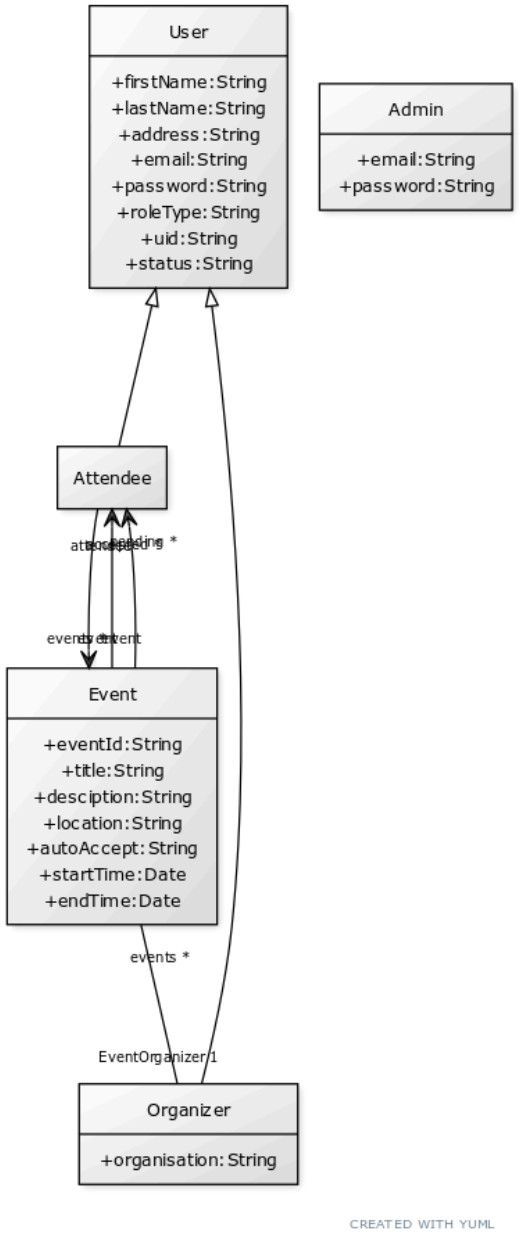
Professor: Hussein Al Osman
TAs: Ammar Rashed, Hassan Awad,
Subhashri Mohan, Nima Meghdadi

Names: Aourz, Badr
Camara, Ibrahim
Chen, Zixian
Radar, Alexander
Rahman, Yusuf
Wu, Mengzhi

Short Introduction:

This report documents the progress, challenges, and solutions undertaken during the development of the SEG2105 EAMS project. It includes the UML diagram, key contributions of each team member, application screenshots, and the evolution and lessons learned throughout the development of the project. The goal was to ensure a robust and efficient application aligned with the project requirements while ensuring clear communication and efficient teamwork.

UML class diagram:



Contributions Table:

Members	**Deliverable 1: Contributions**
Aourz, Badr	- Added a welcome page and made it the main activity which redirects to either the login or the registration page - Added a login page - Added RadioGroup to choose either attendee or Organizer upon registration - Fixed a number of UI bugs - Implemented the login page logic so that it actually checks for email/password - Implemented the welcome page logic so that it effectively shows the role of the user after logging in - Added the organization name field animation for organizer signup and its restrictions
Camara, Ibrahim	- Login and logout functionality - Configured DB + init android studio app - last minute testing and debugging with alex - recorded demo video
Chen, Zixian	- Connected firebase, added dependencies (Team decided later to use another firebase so the changes are discarded) - Helped setup and configure firebase - Helped debugging android studio (issues with gradle)
Radar, Alexander	- Initialize app on github. - Basic startup xml layout for fields. - Created User/Attendee/Organizer classes. - Multiple bug fixes for the app crashing. - Helped to fix bugs to get firebase working. - Helped work on logic for user login with the db. - Connected pages with onClicks. - Testing app: Getting the app ready for the v0.1 release. - Design planning + Helping with instructions for the group. - UML Diagram deliverable 1.
Rahman, Yusuf	- README.txt
Wu, Mengzhi	- UI optimization, like button position, constraints, font size, color, background etc - Added the Back button on registration page. Now it's able to return to sign page during registration - Added password visibility toggle, (The truck would be red when it's off and green when it's on - Deleted "main activity". The launch page will now be the login page - Fixed a error that the registration and login buttons were mixed

	up - Changed some variable's default name, it will be easier to identify and use - Added Icons for email and password entry on login page - Added field validation
	Deliverable 2: Contributions
Aourz, Badr	- Added Admin credentials in readme file - Fixed Org name field bugs - Added roles to DB and fixed multiple issues related to it - Modified method of verifying status of the user and showing/redirecting them accordingly - Fixed a small null pointer bug when retrieving the role of the user. - Removed unnecessary Snackbar methods - Reverted back for the role to be stored in "userType" for all users instead of "role" in DB - Changed all references from role to userType in the code to ensure consistency in the app and the DB - Fixed some UI and xml bugs
Camara, Ibrahim	- Registration Page - Project restructure - last minute testing and debugging with alex - recorded demo video

Chen, Zixian	- Attempted to implement the email notification function in the “email” branch (wasn’t able to make it fully work because ran out of time for deliverable 2)
	  - Created the MailSender class
	  - implemented the SMTP Email sending using the Outlook SMTP server
	  - Used ExecutorService to send emails asynchronously
	  - Managed a single threaded executor for handling email sending tasks.
	  - Created an outlook account and attempted to configure it to send emails on behalf of the app.

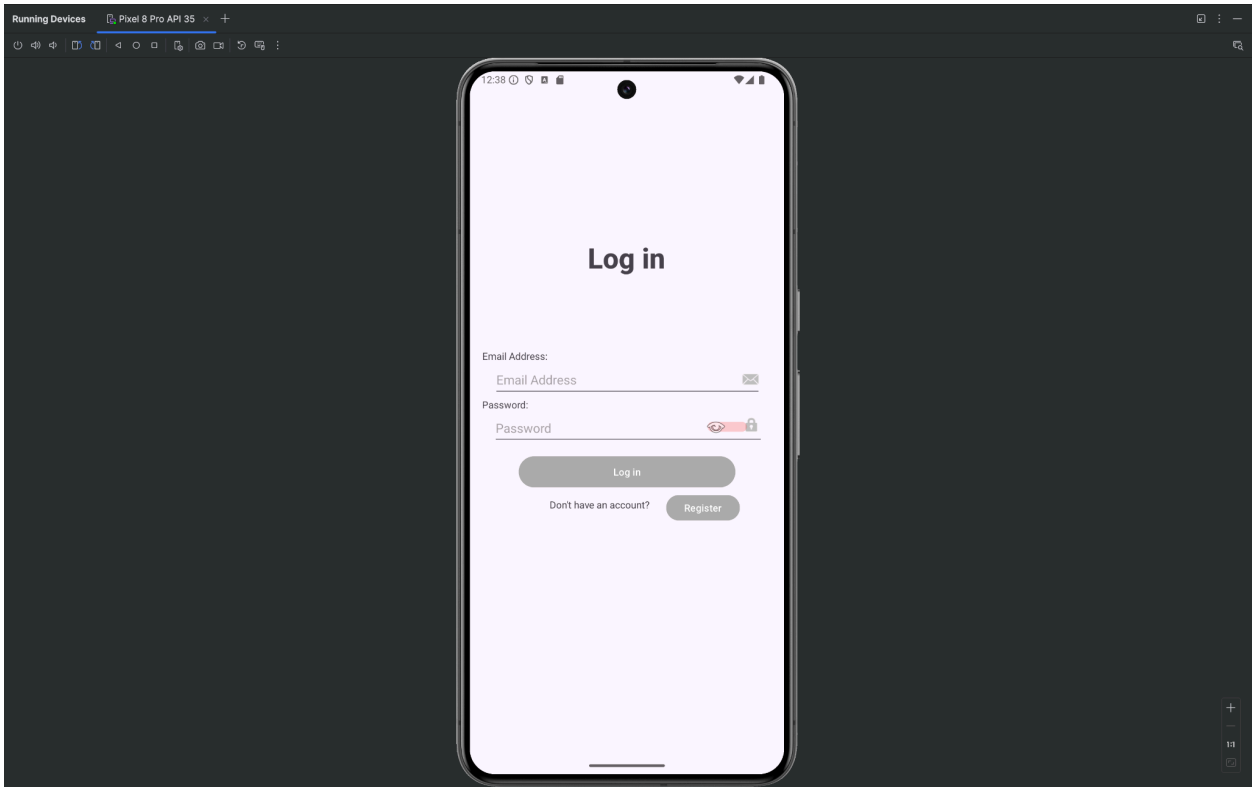
Radar, Alexander	- Created Admin class. - Created admin adapter and page. - Created Rejected list adapter, accepted list adapter. - Xml for adapters and pages . - Added logic for the buttons to remove from the page when clicked. - Helped with some firebase implementation. - Created toString for organizer and attendee. - Created status variable for users. - Helped work on login page redirects depending on role status. - Refresh page + back buttons. - More bug fixing.

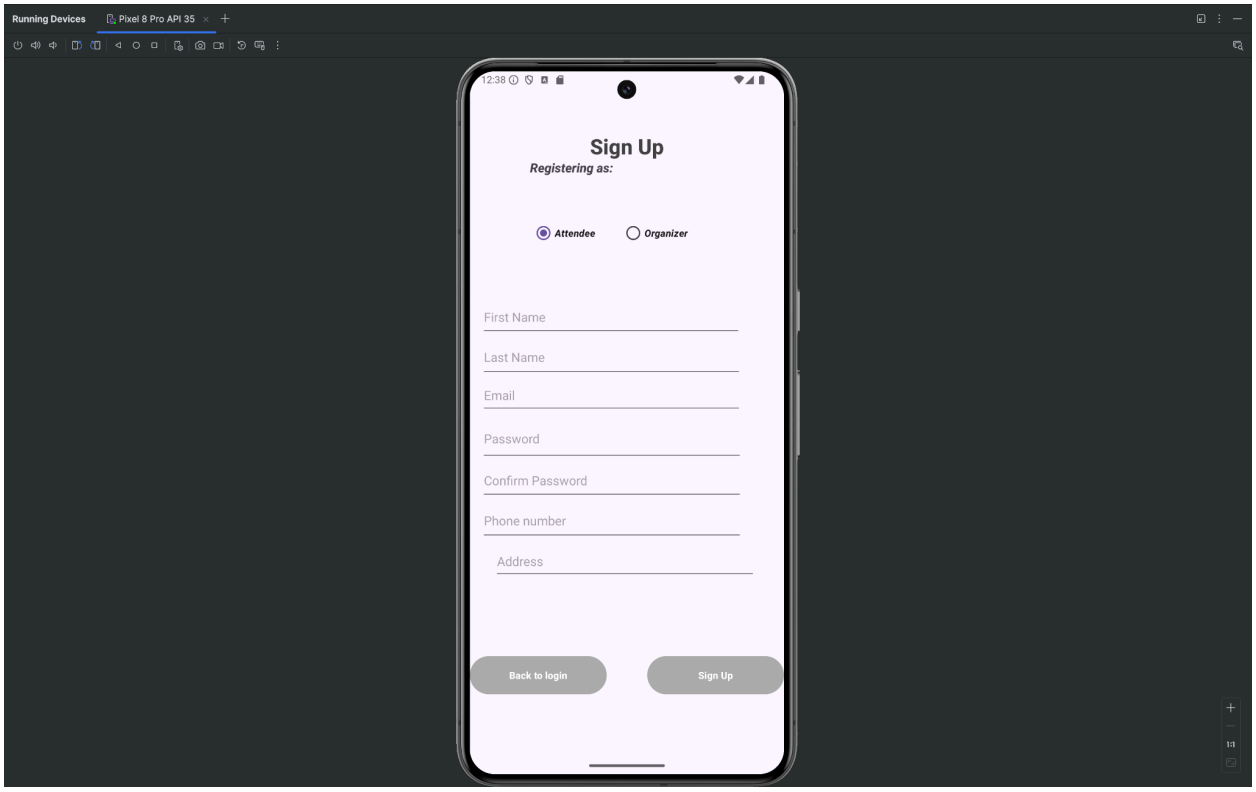
	- Testing app: Getting the app ready for the v0.1 release video. - Design planning + Helping with instructions for the group. - Design planning + Helping with instructions for the group. - UML Diagram for deliverable 2
Rahman, Yusuf	
Wu, Mengzhi	- Rejected method and some small changes - Report bugs - App tests - UML Diagram demo for deliverable 2
	Deliverable 3: Contributions
Aourz, Badr	- Few changes in xml. Added ApproveAll button in the layout and scrollbar to see past and current events on organizer page - Added ApproveAll button logic - Added the logic for both delete and update features - Some xml code changes as not everything was declared - Added the animations for each of the delete and update buttons - Fixed some event fetching bugs in between pages - The DB now reflects any changes made to the events whether that be deletion/update - Added two private List<Attendee> (for pending and accepted) for the events - Added passing and fetching the events to EventRequestPage and fetch the two lists for the event under the correct adapters - Synchronized attendee lists for events from and to DB - Added sample data for the events' lists of attendees for future testing - Changed some of the DB structure for better organization and more efficient retrieval of information - Fixed several inconsistencies in the attendee's adapters to ensure correct retrieval of objects
Camara, Ibrahim	- Attendee page - Last minute testing and debugging with alex - record demo video
Chen, Zixian	- Created the EventCreationPage - Added an eventCreationButton which gatherers the data from user inputs - Added a time picker for start time and end time - Added a date picker for start date and end date - Implemented the DatePickerDialog and TimePickerDialog
Radar, Alexander	- Created event request page (pending and rejected adapters for request page + xml) and basic button functionality. - Added viewable details about attendees to the request page on

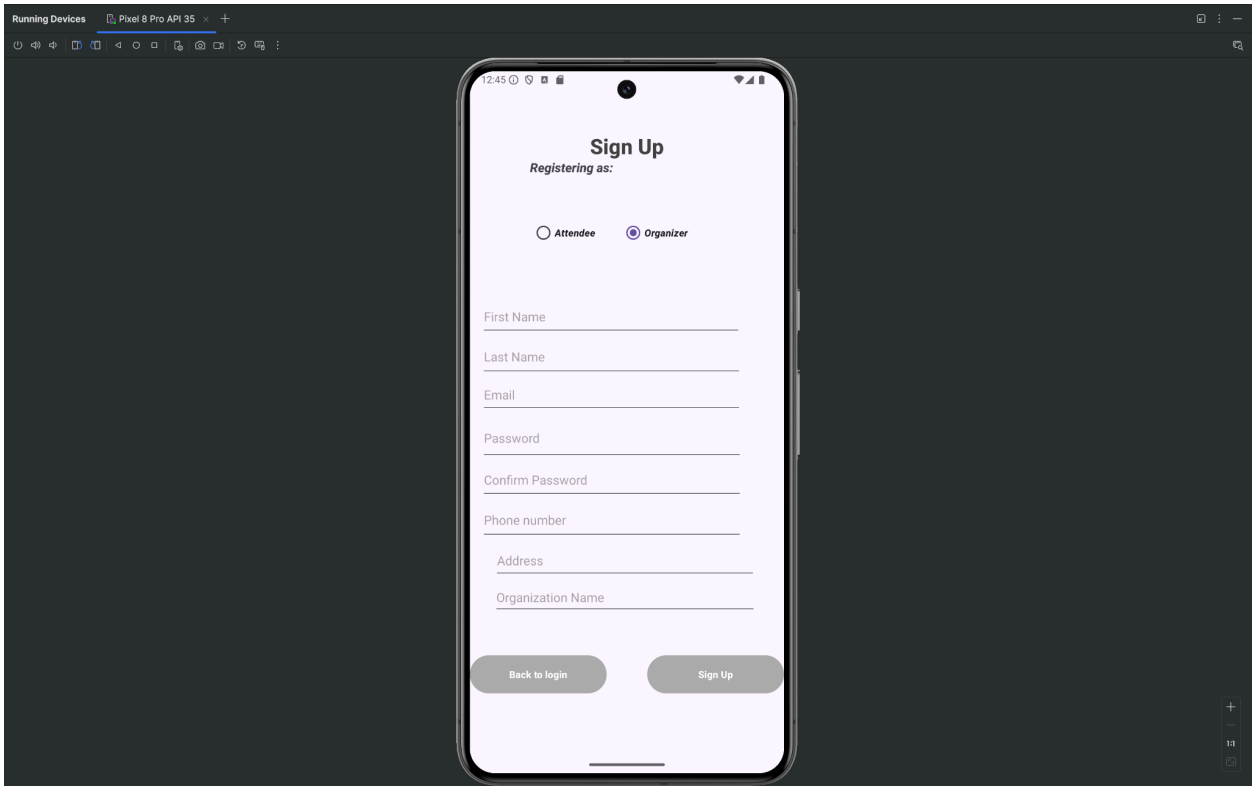
	click on each request. - Buttons to move around the event pages and attendee page. - Logic to send the user's id (intent) to the different event/attendee page(s) and not losing it. - Added logic to the information being sent in an intent to request page (can display event, description, location, date) (date converted to look nicer). - Added auto accept switch into event registration and logic for how it works in event class. - Logic to reconstruct the logged in attendee's object in the attendee page. - Added logic to the attendee page to fetch all the events and allow you to join them. - Logic to delete events. - Logic so only organizers can delete events they created. - Logic to send finished events to the past events. - Testing app: Getting the app ready for the v0.3 release video. - More bug fixing. - UML Diagram for deliverable 3.
Rahman, Yusuf	
Wu, Mengzhi	- Created event request page UI and half of functionality - Added validation when creating events(The limit of minute input(0 and 30) - Connect the event adding to db - allow the event adapter to display the event
	Deliverable 4: Contributions
Aourz, Badr	- Reverted some bugs from master - Worked with Alexander on the attendee adapters in preparation for upcoming tasks - Check for time conflicts when attendee tries to join event - Check to see which events the attendee hasn't joined yet before adding them to the adapter so that when an attendee joins an event it doesn't show up on their search area - Deleted some unnecessary lines of code that were crashing the app - Populating the attendee's list of events with their actual joined events from DB - Populating the events' list of attendees with their actual attendees from DB - Fixed how the events are fetched on "my events" from DB - Fixed how the date and time gets fetched for verification when joining events - Report writing
Camara, Ibrahim	- Unit test and event start less than 24hrs notificaion

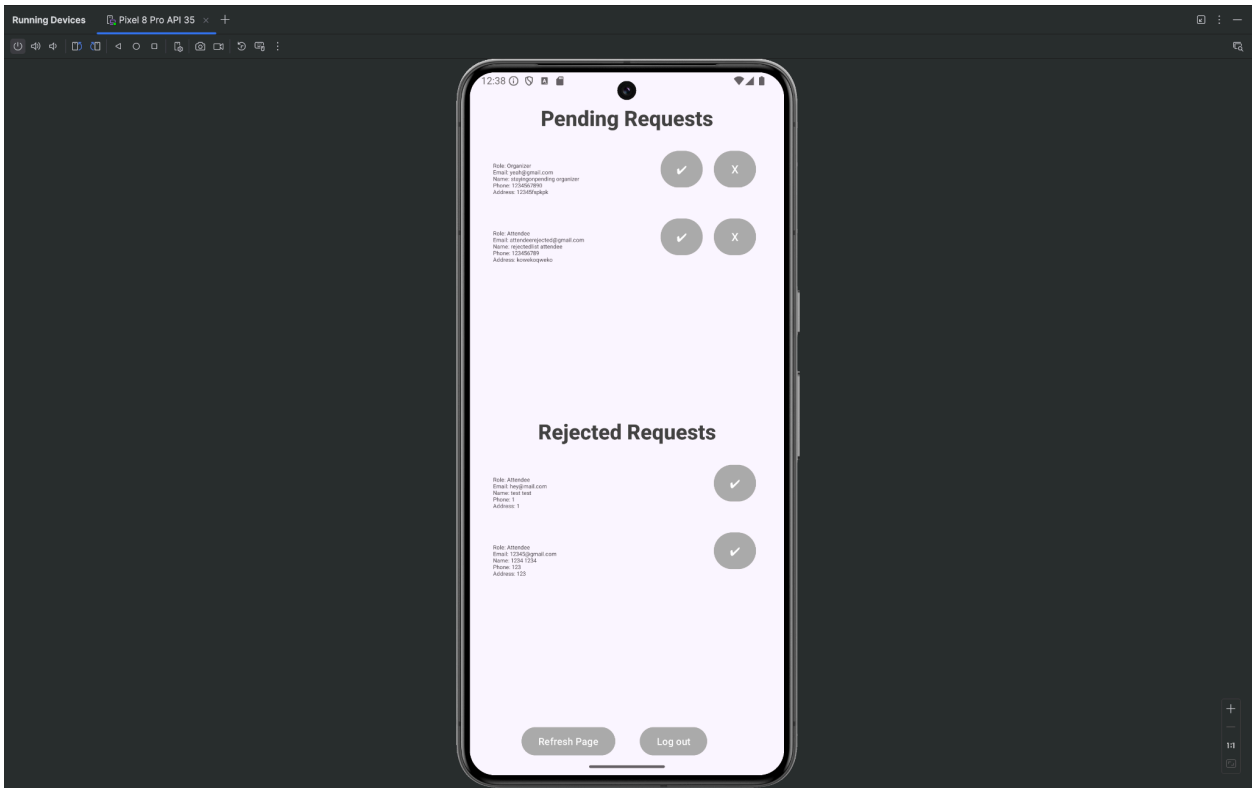
	- Record demo video - Uml diagramm - Last minute testing and debugging with alex
Chen, Zixian	- Implemented the search bar on the attendee page, which allows attendees to search for events available. - Implemented a filtering logic in the Adapter - Defined performFiltering to match search keywords with event titles and descriptions - Updated the filteredEvents list and notified the adapter to refresh the displayed results - Added separate lists (originalEvents and filteredEvents) in the adapter to maintain both the unfiltered and filtered datasets. - Ensured smooth integration of filtered results into the ListView by notifying the adapter ehenver the results changed.
Radar, Alexander	- Event deletion logic (can't delete if an attendee is accepted in the event). - Created EventsPage and ArrayAdapter (EventListActivity.java, MyEventAdapter.java, xml) - Updated Attendee: Added eventIds in Attendee as an ArrayList and fixed in all other locations. - Connected joined eventIds into Attendee and database. - Database functions for AttendeePage (events and attendee). - Database functions for Leaving an event (updating attendee (remove event id) and event (remove from pending and accepted event lists)). - Merging branches into main work together. - Fixing even more bugs that people pushed. - Testing app: Getting the app ready for the v0.4 release video. - Decision making on what code is used. - Eams report help.
Rahman, Yusuf	
Wu, Mengzhi	- Logic for 24hours leaving limit - User events status setting - Display status text to item - Function tests

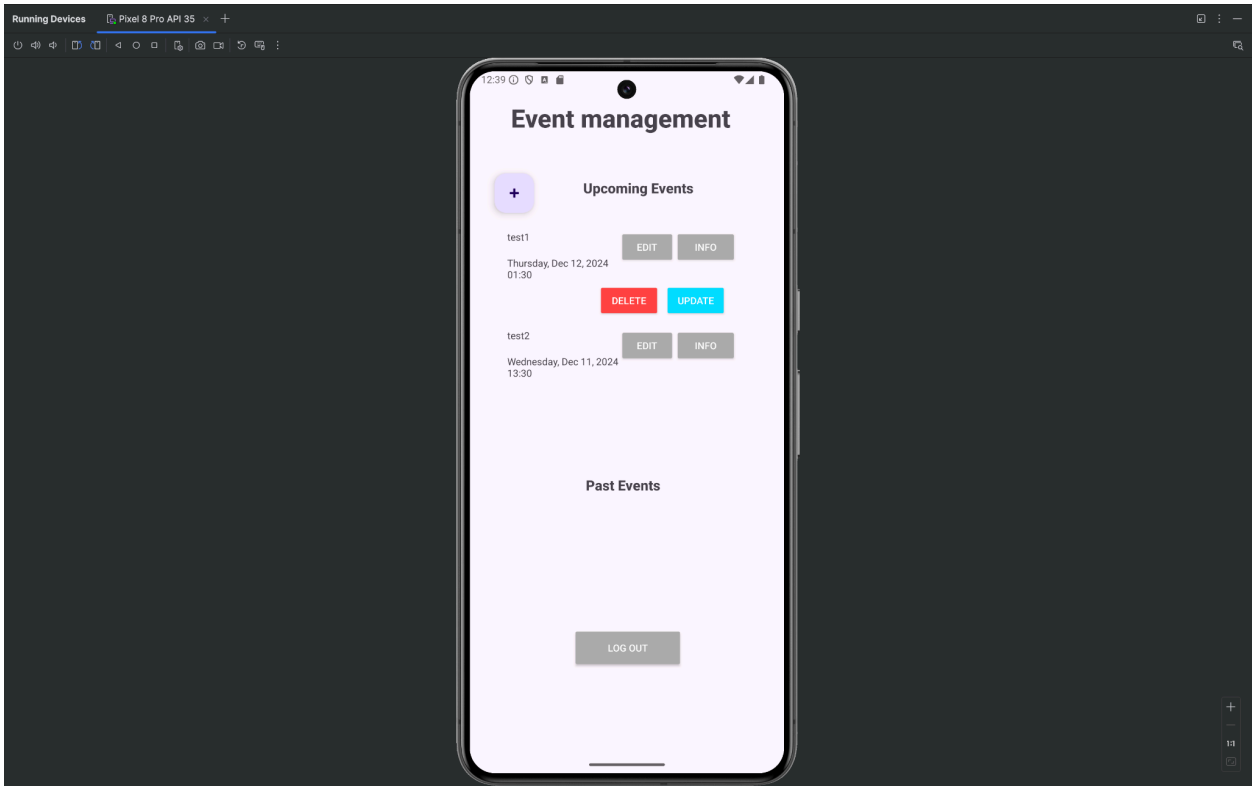
Pictures of App:

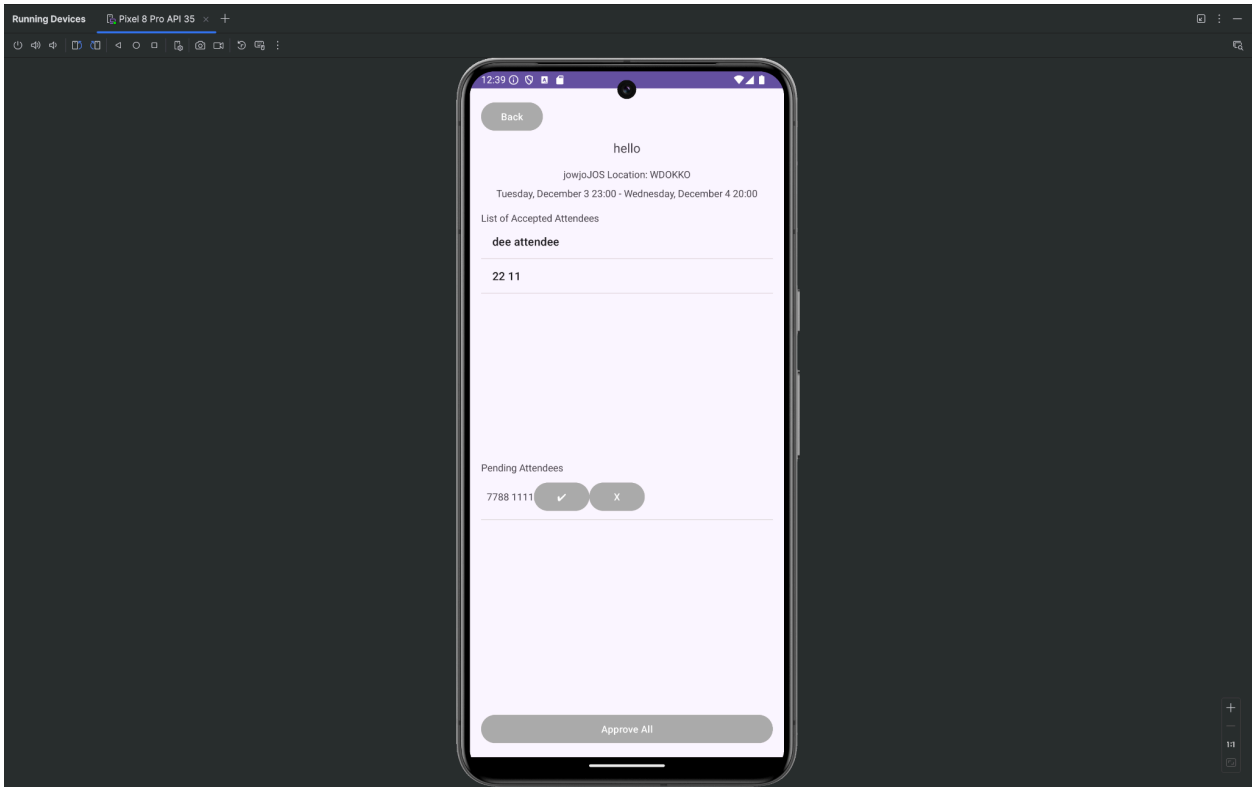


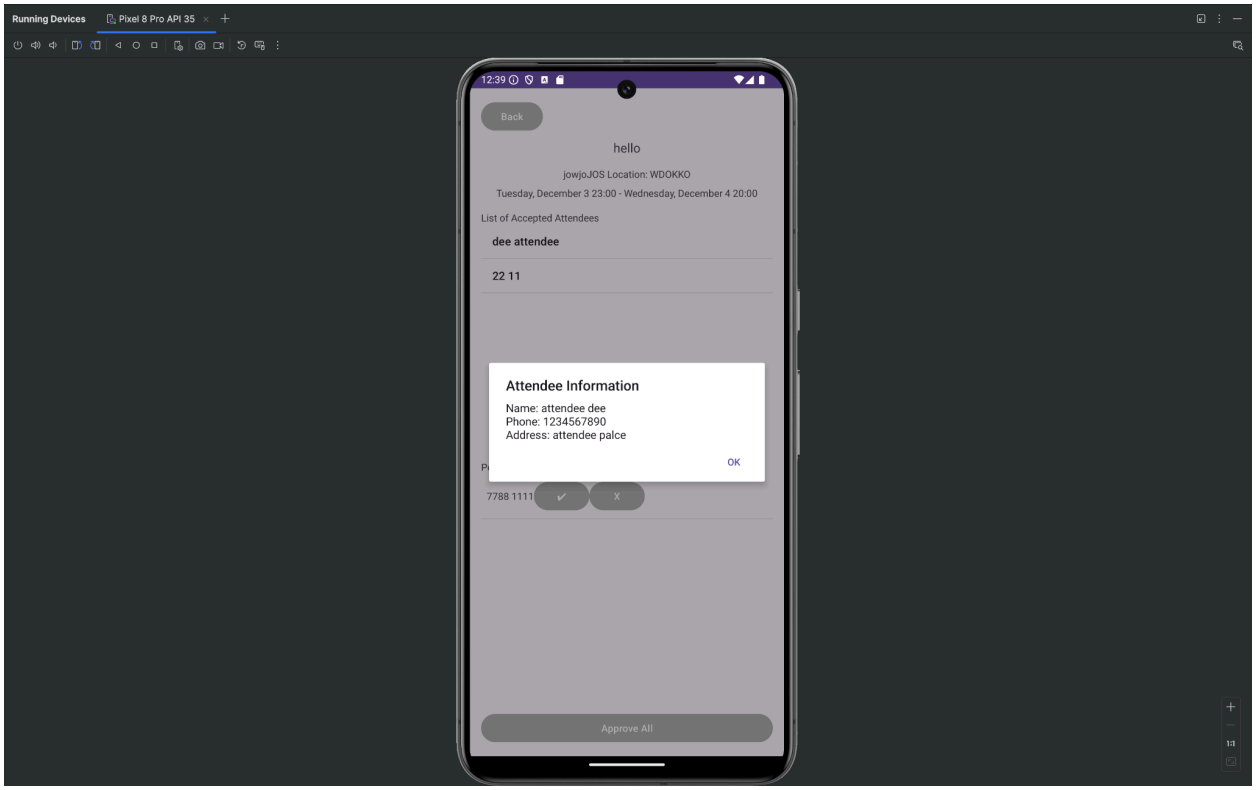


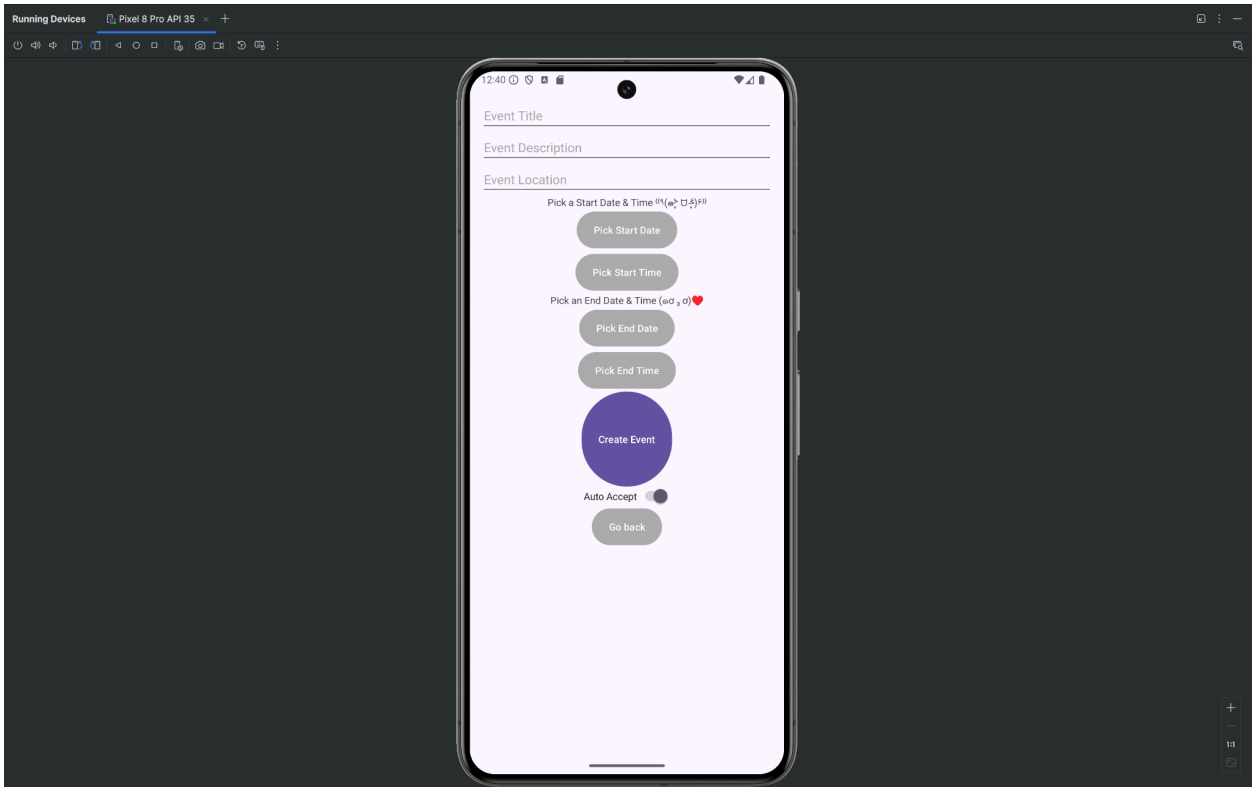

Running Devices
Pixel 8 Pro API 35
12:45
Sign Up
Registering as:
Attendee
Organizer
First Name
Last Name
Email
Password
Confirm Password
Phone number
Address
Organization Name
Back to login
Sign Up

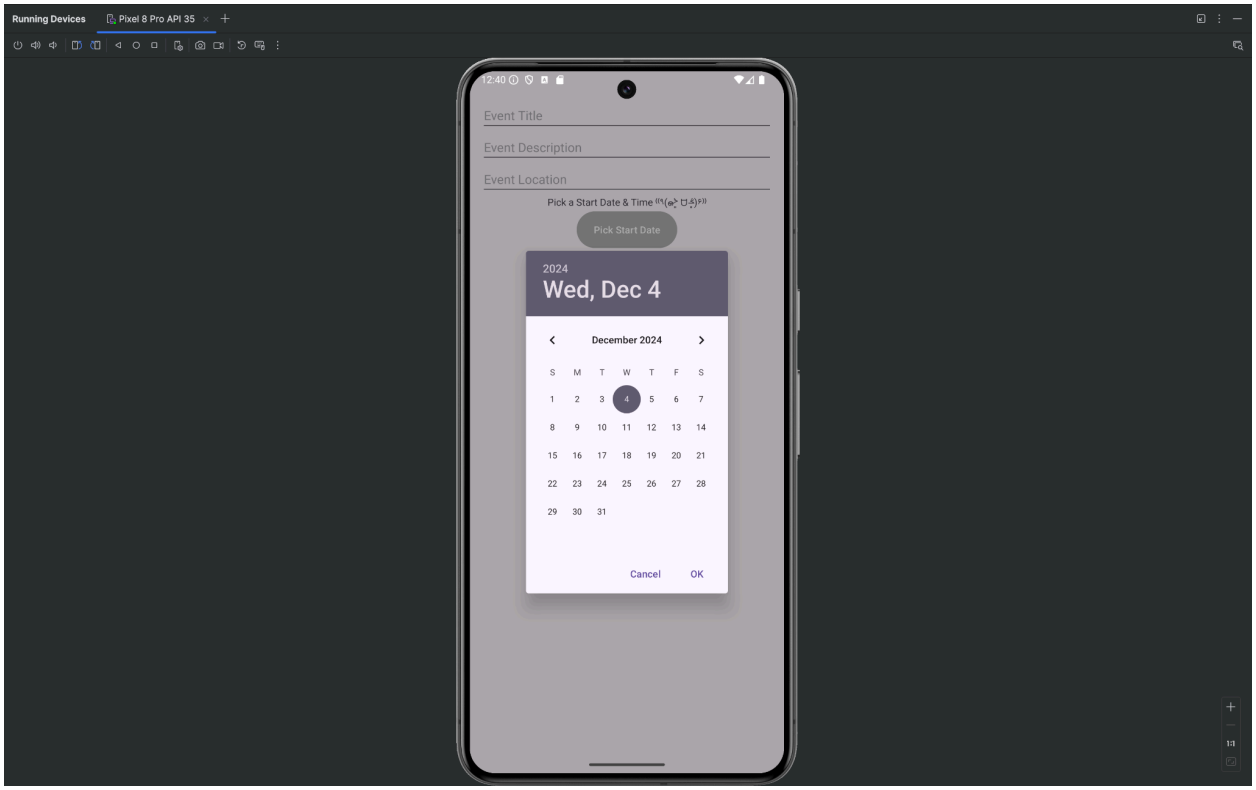

Running Devices
Pixel 8 Pro API 35
12:38
Pending Requests
Rejected Requests
Refresh Page
Log out

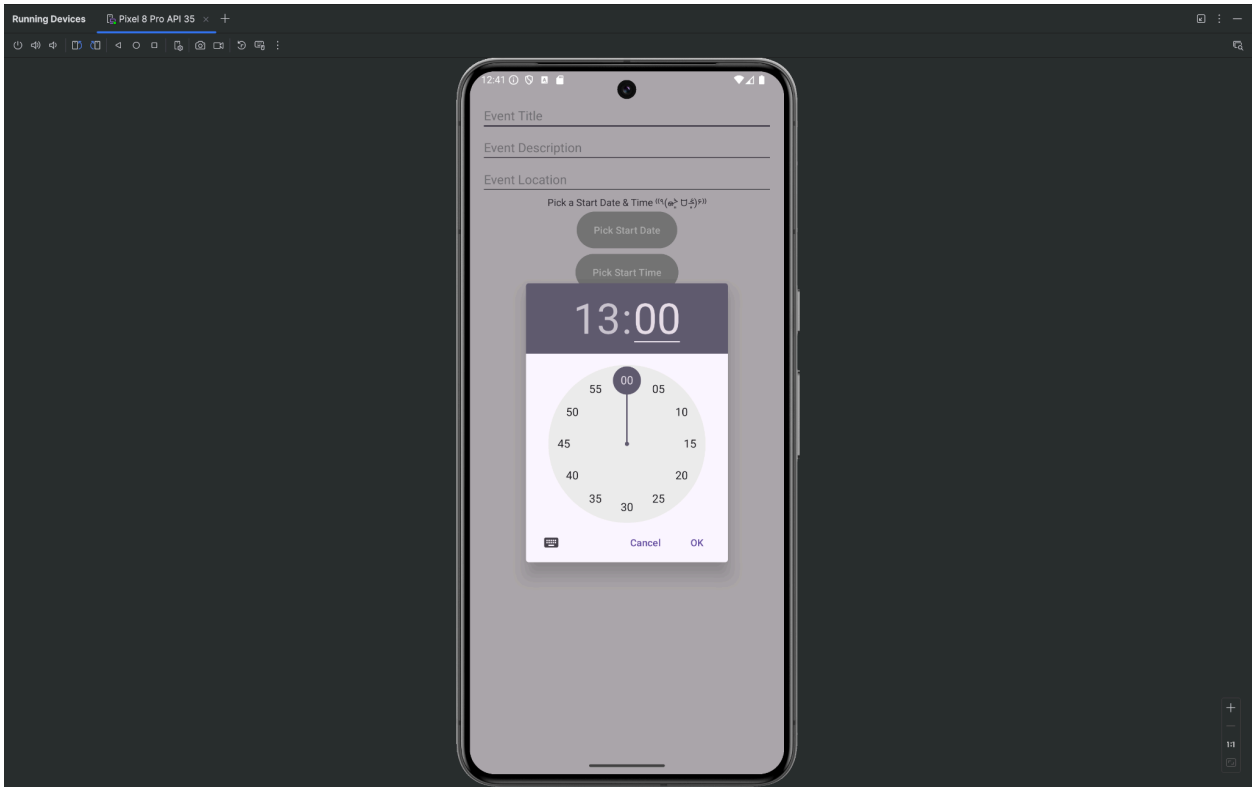

Running Devices
Pixel 8 Pro API 35
12:39
Event management
Upcoming Events
test1
EDIT
INFO
Thursday, Dec 12, 2024
01:30
DELETE
UPDATE
test2
EDIT
INFO
Wednesday, Dec 11, 2024
13:30
Past Events
LOG OUT

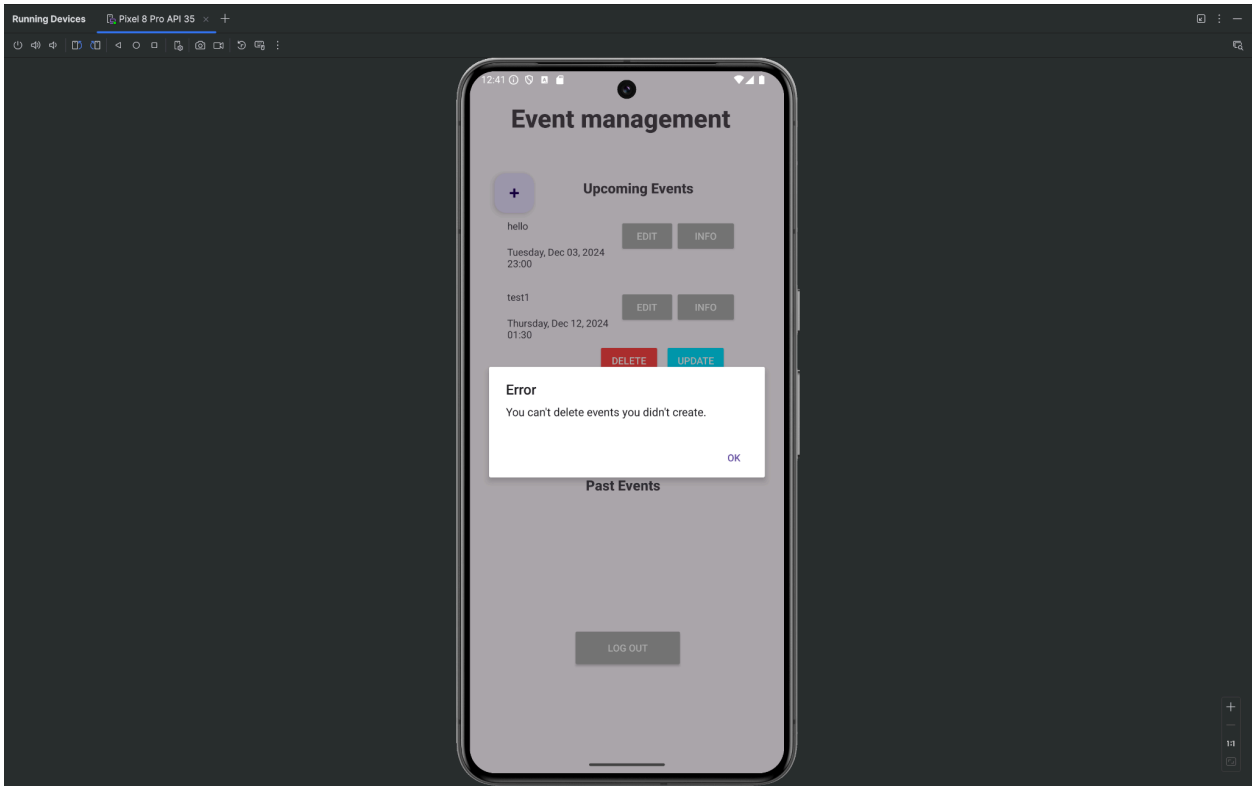

Running Devices
Pixel 8 Pro API 35
12:39
Back
hello
jowjoJOS Location: WDOKKO
Tuesday, December 3 23:00 - Wednesday, December 4 20:00
List of Accepted Attendees
dee attendee
22 11
Pending Attendees
7788 1111
✔
X
Approve All

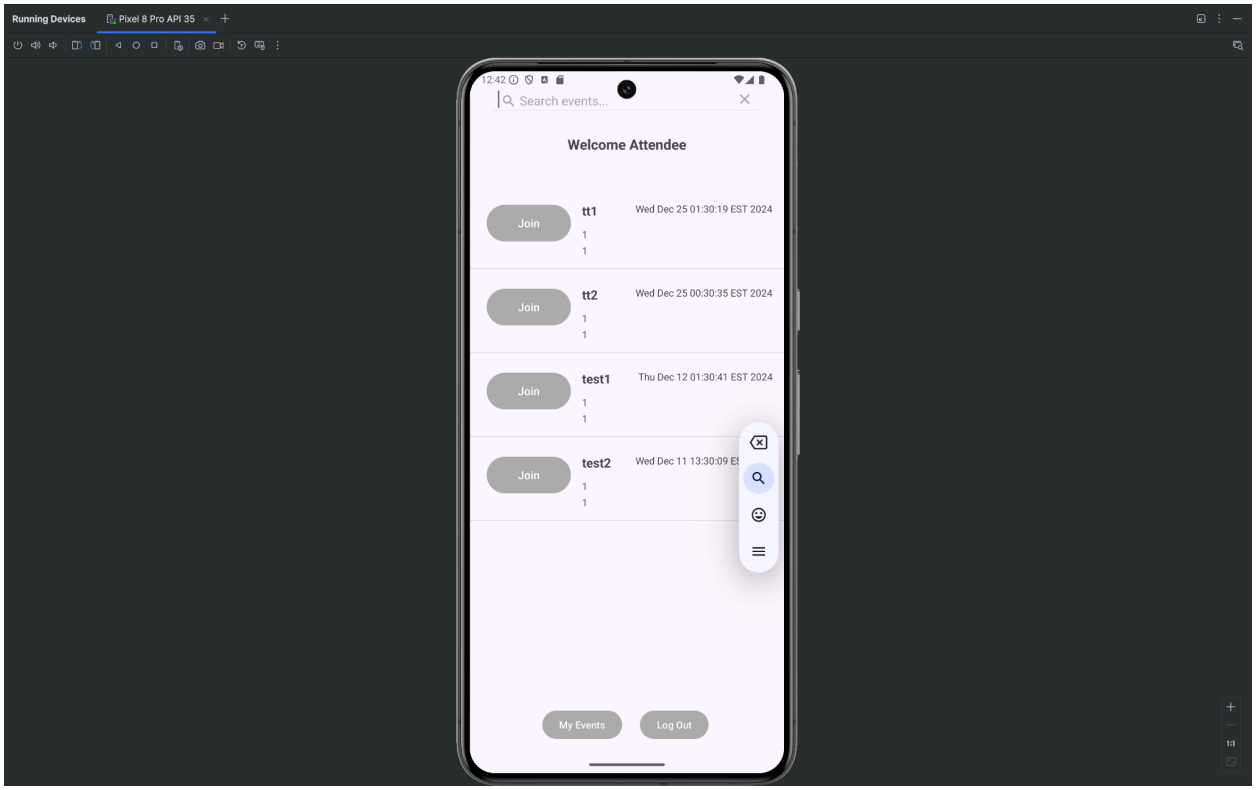

Running Devices
Pixel 8 Pro API 35
12:39
Back
hello
jowjoJOS Location: WDOKKO
Tuesday, December 3 23:00 - Wednesday, December 4 20:00
List of Accepted Attendees
dee attendee
22 11
Attendee Information
Name: attendee dee
Phone: 1234567890
Address: attendee palce
OK
7788 1111
Approve All

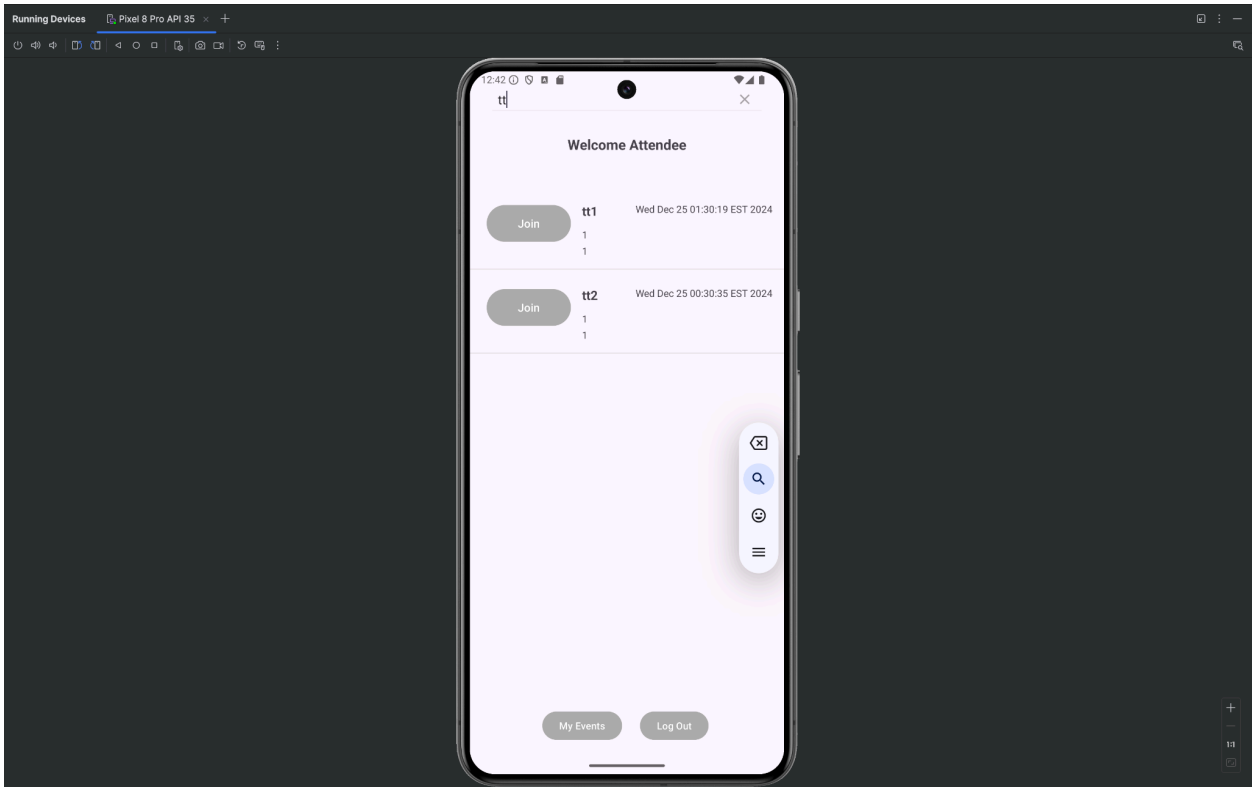

Running Devices
Pixel 8 Pro API 35
12:40
Event Title
Event Description
Event Location
Pick a Start Date & Time
Pick Start Date
Pick Start Time
Pick an End Date & Time
Pick End Date
Pick End Time
Create Event
Auto Accept
Go back

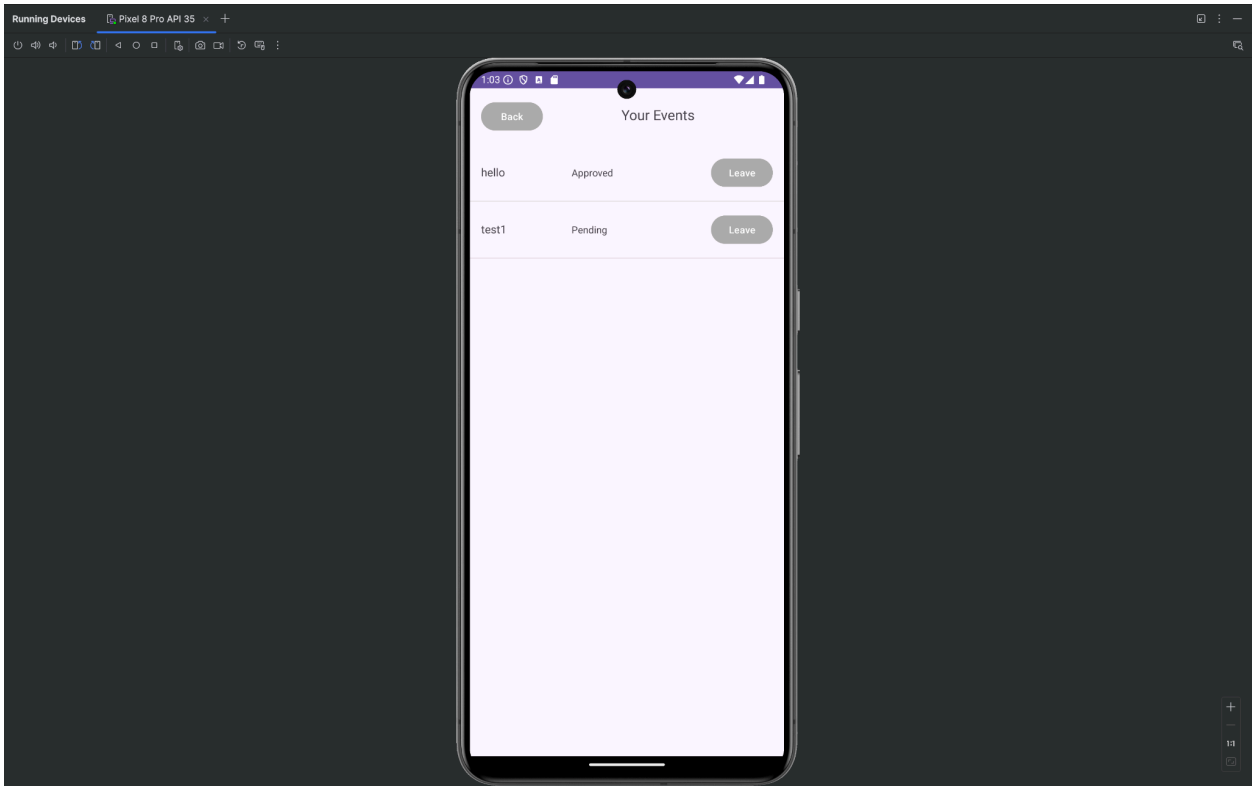

Running Devices
Pixel 8 Pro API 35
12:40
Event Title
Event Description
Event Location
Pick a Start Date & Time ヾ(≧▽≦)ﾉ
Pick Start Date
2024
Wed, Dec 4
December 2024
S M T W T F S
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
Cancel
OK

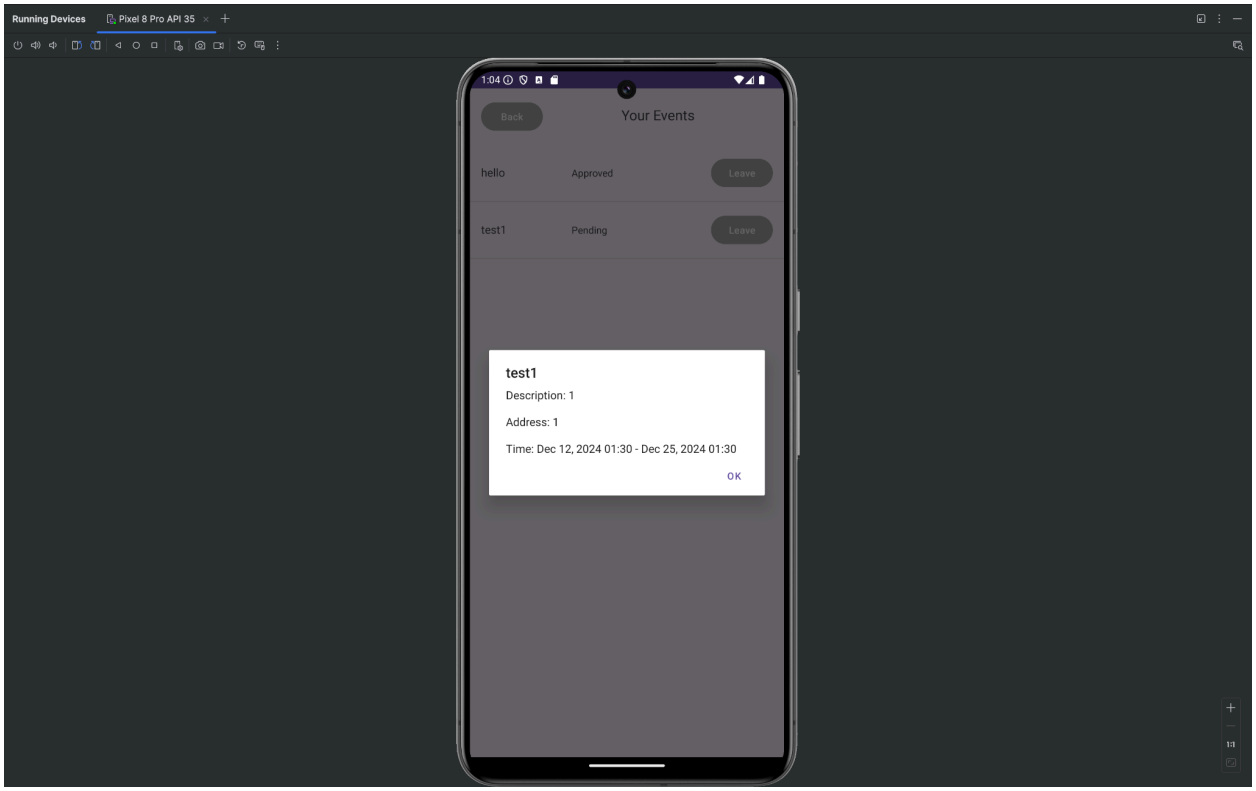

Running Devices
Pixel 8 Pro API 35
12:41
Event Title
Event Description
Event Location
Pick a Start Date & Time ヾ(≧▽≦)ﾉ
Pick Start Date
Pick Start Time
13:00
00
05
10
15
20
25
30
35
40
45
50
55
Cancel
OK


Running Devices
Pixel 8 Pro API 35
12:41
Event management
Upcoming Events
hello
EDIT
INFO
Tuesday, Dec 03, 2024
23:00
test1
EDIT
INFO
Thursday, Dec 12, 2024
01:30
DELETE
UPDATE
Error
You can't delete events you didn't create.
OK
Past Events
LOG OUT


Running Devices
Pixel 8 Pro API 35
12:42
Search events...
Welcome Attendee
Join
tt1
Wed Dec 25 01:30:19 EST 2024
1
1
Join
tt2
Wed Dec 25 00:30:35 EST 2024
1
1
Join
test1
Thu Dec 12 01:30:41 EST 2024
1
1
Join
test2
1
1
My Events
Log Out


Running Devices
Pixel 8 Pro API 35
12:42
tt
Welcome Attendee
Join
tt1
Wed Dec 25 01:30:19 EST 2024
1
1
Join
tt2
Wed Dec 25 00:30:35 EST 2024
1
1
My Events
Log Out


Running Devices
Pixel 8 Pro API 35
1:03
Back
Your Events
hello
Approved
Leave
test1
Pending
Leave


Running Devices
Pixel 8 Pro API 35
1:04
Back
Your Events
hello
Approved
Leave
test1
Pending
Leave
test1
Description: 1
Address: 1
Time: Dec 12, 2024 01:30 - Dec 25, 2024 01:30
OK

Lessons Learned:

In the creation of the EAMS App we learned several things about Software Engineering. Our project was created on Android Studio using Java, Xml, and Gradle. One of the main things we got to learn was how to use GitHub. At first we did not use branches, but later we realized it would be way better for the workflow to do this. During the deliverables, we first figured out how to use the features of android's xml, and were able to make a user interface for our project. We developed Java classes and pages which utilized databases, specifically Firebase, to store information that could be fetched when the app was reloaded. The main parts of the project involved following guidelines as well as field validation. Helping ourselves understand how to create a project based on given information. At the start our files were badly formatted and really nonoptimal at certain points like the admin page array adapter. We learned to implement design principals better, as well as understanding the basics of creating an android app. Then lastly, we got to experience how it is to work in a group for a coding project.