# Android Project:
# Event Attendance Management System
# Final Report

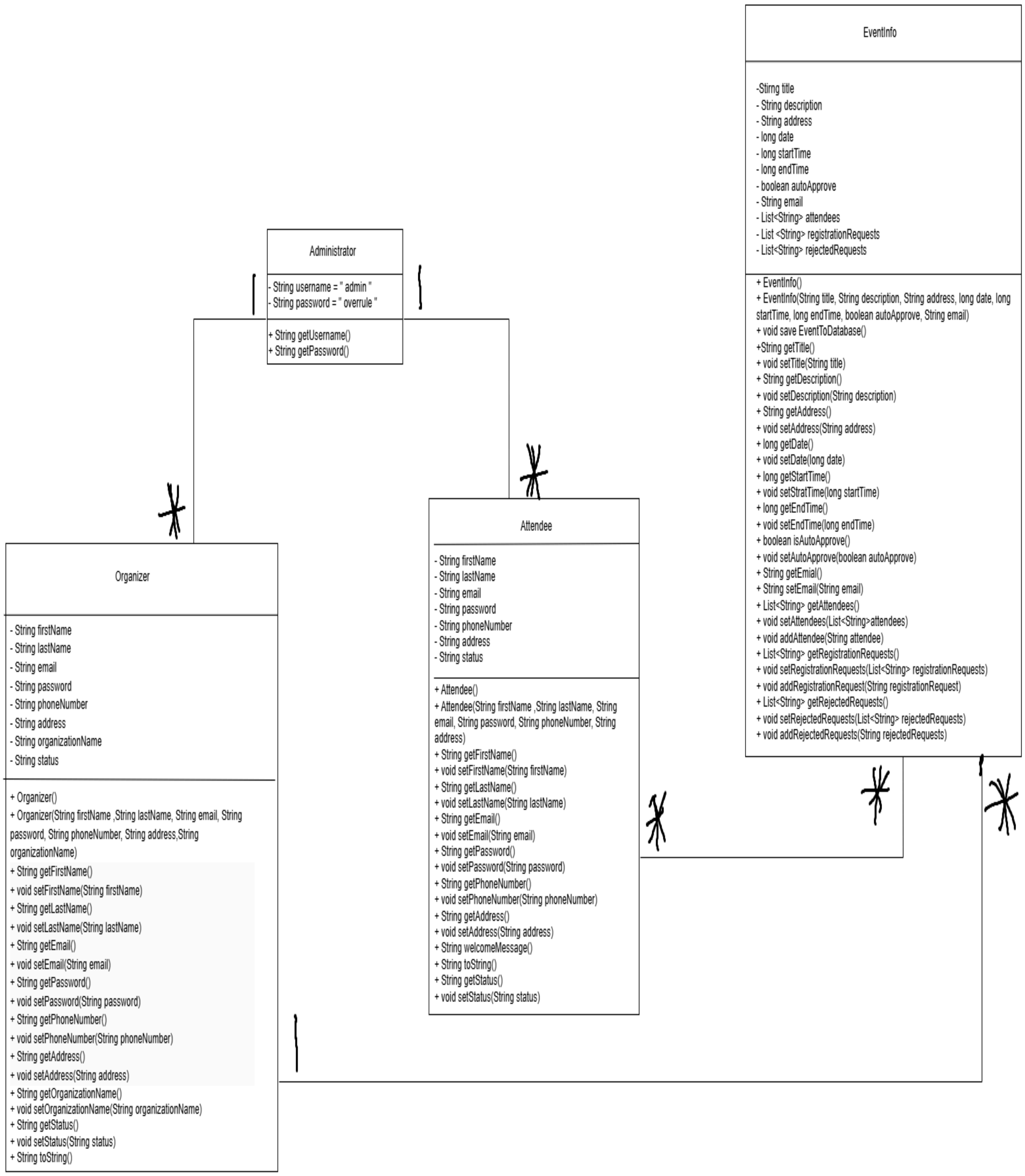
**Project Group: 08**
**Name: Mahmud Hossain(300371112)**
**Name: John Ghai Manyang(300272467)**
**Name: Ruqyyah Hassan(300383524)**
**Name: Elias Maouche(300355645)**
**Name: Batoul Rahal(300384487)**
**Name: Daniel Yang(300348682)**

**Course: SEG 2105**
**Instructor: Hussein Al Osman**
**Submission Date: December 4, 2024**

## Introduction:

The purpose of this project was to develop an android application called Event Attendance Management System (EAMS). This mobile application is designed to streamline the process of event registration and attendance tracking for events organized by a university. The app has three types of users, namely: Attendee, Organizer, and Administrator. Attendees can view available events to register for and manage their event participation. Organizers can manage event details and respond to registration requests. Finally, administrators approve account registration of attendees and organizers.

# UML Class Diagram:


EventInfo
-Stirng title
- String description
- String address
- long date
- long startTime
- long endTime
- boolean autoApprove
- String email
- List<String> attendees
- List <String> registrationRequests
- List<String> rejectedRequests
+ EventInfo()
+ EventInfo(String title, String description, String address, long date, long startTime, long endTime, boolean autoApprove, String email)
+ void save EventToDatabase()
+String getTitle()
+ void setTitle(String title)
+ String getDescription()
+ void setDescription(String description)
+ String getAddress()
+ void setAddress(String address)
+ long getDate()
+ void setDate(long date)
+ long getStartTime()
+ void setStratTime(long startTime)
+ long getEndTime()
+ void setEndTime(long endTime)
+ boolean isAutoApprove()
+ void setAutoApprove(boolean autoApprove)
+ String getEmial()
+ String setEmail(String email)
+ List<String> getAttendees()
+ void setAttendees(List<String>attendees)
+ void addAttendee(String attendee)
+ List<String> getRegistrationRequests()
+ void setRegistrationRequests(List<String> registrationRequests)
+ void addRegistrationRequest(String registrationRequest)
+ List<String> getRejectedRequests()
+ void setRejectedRequests(List<String> rejectedRequests)
+ void addRejectedRequests(String rejectedRequests)
Administrator
- String username = " admin "
- String password = " overrule "
+ String getUsername()
+ String getPassword()
1
1
*
*
Organizer
- String firstName
- String lastName
- String email
- String password
- String phoneNumber
- String address
- String organizationName
- String status
+ Organizer()
+ Organizer(String firstName ,String lastName, String email, String password, String phoneNumber, String address,String organizationName)
+ String getFirstName()
+ void setFirstName(String firstName)
+ String getLastName()
+ void setLastName(String lastName)
+ String getEmail()
+ void setEmail(String email)
+ String getPassword()
+ void setPassword(String password)
+ String getPhoneNumber()
+ void setPhoneNumber(String phoneNumber)
+ String getAddress()
+ void setAddress(String address)
+ String getOrganizationName()
+ void setOrganizationName(String organizationName)
+ String getStatus()
+ void setStatus(String status)
+ String toString()
Attendee
- String firstName
- String lastName
- String email
- String password
- String phoneNumber
- String address
- String status
+ Attendee()
+ Attendee(String firstName ,String lastName, String email, String password, String phoneNumber, String address)
+ String getFirstName()
+ void setFirstName(String firstName)
+ String getLastName()
+ void setLastName(String lastName)
+ String getEmail()
+ void setEmail(String email)
+ String getPassword()
+ void setPassword(String password)
+ String getPhoneNumber()
+ void setPhoneNumber(String phoneNumber)
+ String getAddress()
+ void setAddress(String address)
+ String welcomeMessage()
+ String toString()
+ String getStatus()
+ void setStatus(String status)
*
*
1
*
1

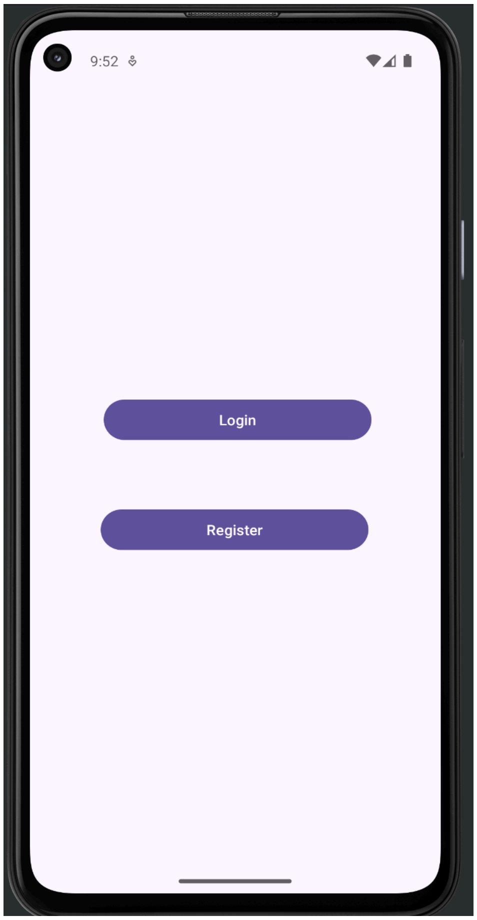
# Team Contributions:

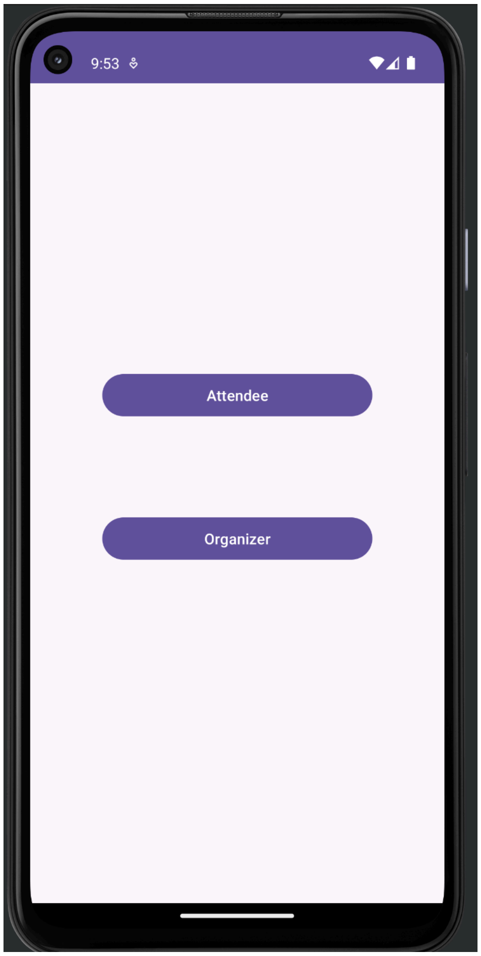
| Name | Deliverable 1 | Deliverable 2 | Deliverable 3 | Deliverable 4 |
|---|---|---|---|---|
| Mahmud Hossain | MainActivity created. Login page created.Registration page created. Administrator added. Logout functionality added. Made slight changes to AttendeeActivity and OrganizerActivity. Demo video added. | LoginActivity and AdministratorActivity updated. InboxActivity added. Updated InboxActivity to show pending requests. Updated AdminFunctions. Admin approve/reject function fixed. Demo video added. | OrganizerActivity created. CreateEventActivity, UpcomingEventsActivity, and PastEventsActivity created. CreateEventActivity updated so that an organizer is associated when creating events. UpcomingEventsActivity fixed. Added AttendeeEventsInboxActivity and AttendeeEventRegisterActivity so that attendees can see available events and register for them. EventRegistrationRequestActivity updated so that organizers can registration requests now. EventRegistrationRequestAttendeeDetailActivity added so that details of an attendee | AttendeeMyEventsActivity added. Rejected Request List for events added. Updated AttendeeMyEventsInboxActivity so that it shows the events associated with the attendee. Added AttendeeMyEventDetailActivity so that attendees can click on an event now. Made updates so that an event disappears from events inbox after being added to my events. Made updates so that attendees can cancel their registration for events they previously registered. Made updates so that newest events are |

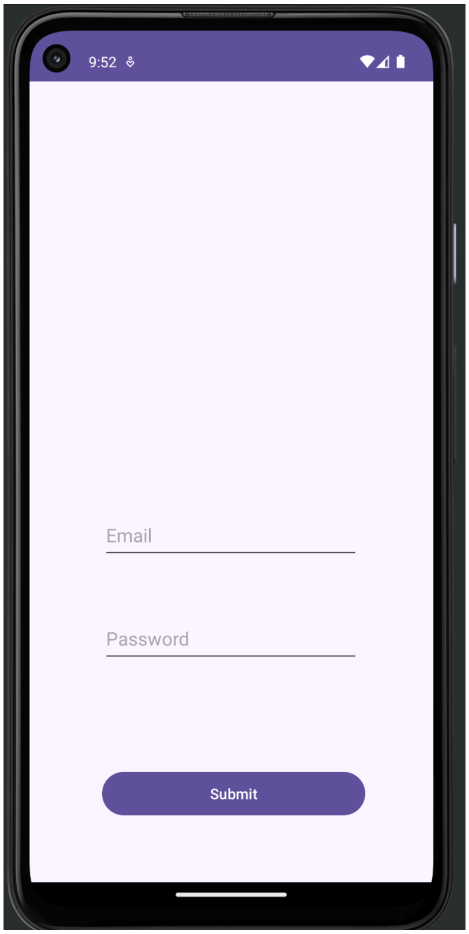
| | | | | |
|---|---|---|---|---|
| | | | can be seen by an organizer. Delete function added for Upcoming Events. Approve, reject, and approve all function added. UML diagram added. Demo video added. | displayed at the top in My Events Inbox. Demo video added. Final report added. |
| John Ghai Manyang | Worked on OrganizerRegistration | Updated the registration status in attendee, updated the login activity, Organizer.java, OrganizerRegistration, and created AdministratorActivity | worked on the Upcoming events activity | Cancel Registration logic for the attendee |
| Ruqyyah Hassan | Created the attendee object class | Modified the adminfunctions to try to approve/reject requests. | Implemented the past events class and modified the createevent activity so that organizers can choose event time in 30 minute intervals and so that they can’t create an event before | Added 4 unit test cases and they all passed successfully |

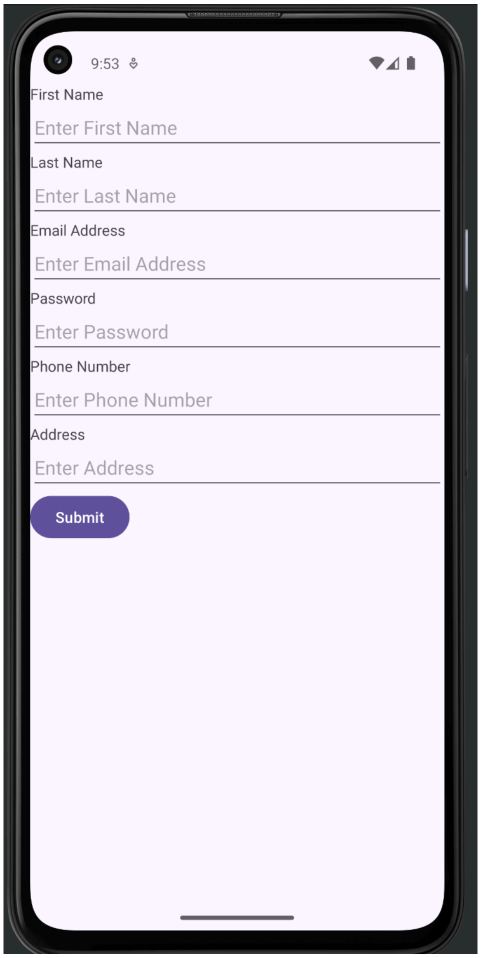
| | | | | |
|---|---|---|---|---|
| | | | the current time | |
| Elias Maouche | Firebase setup, attendee login page, fixed welcomepage, added Organizer logins, worked on firebase to save organizer info | Worked on inboxactivity, registeractivity, registrationrequest to not let users log in without approval and also database | Worked on eventinfo that stores a new event's info into the dabase and on createeventactivity | Worked on not allowing an organizer to delete an event that has attendees registered and setup the database so that attendees registered to an event show up under each event in firebase |
| Batoul Rahal | Welcome activity. UML Diagram added. | Working on the approve and reject functions for the admin. UML diagram added. | | Search method. UML diagram added. |
| Daniel Yang | Added organizer class. Added field validation for attendee and organizer registration | Created Admin approve and reject page. Created Admin rejected inbox | Added event page activity. Added delete function for event | Added time conflict detection to event registration page |

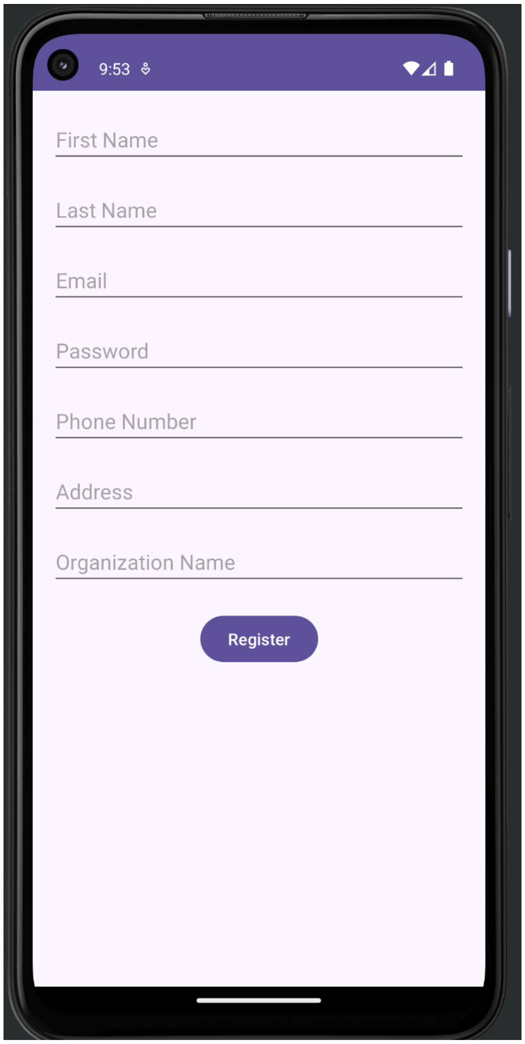
## App Screenshots:

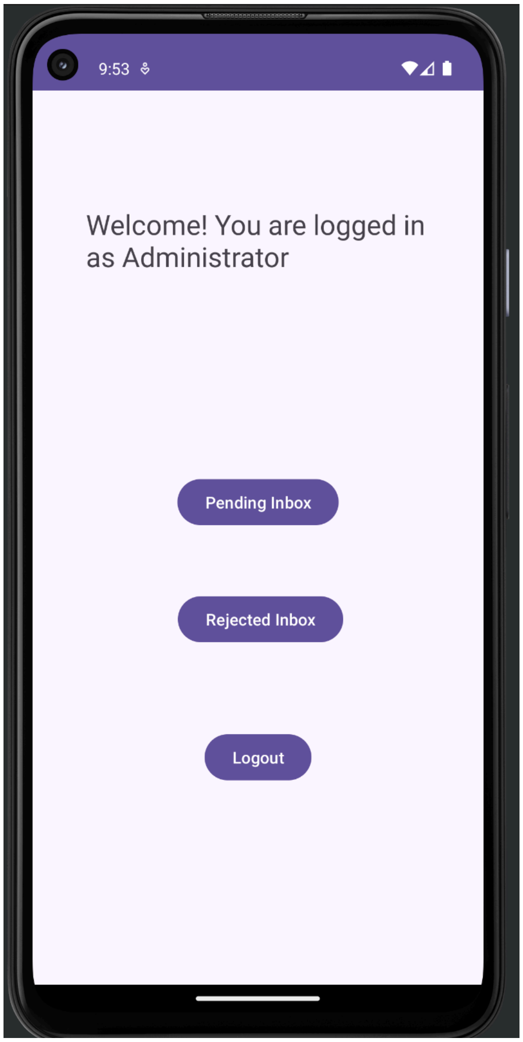



9:53
Attendee
Organizer

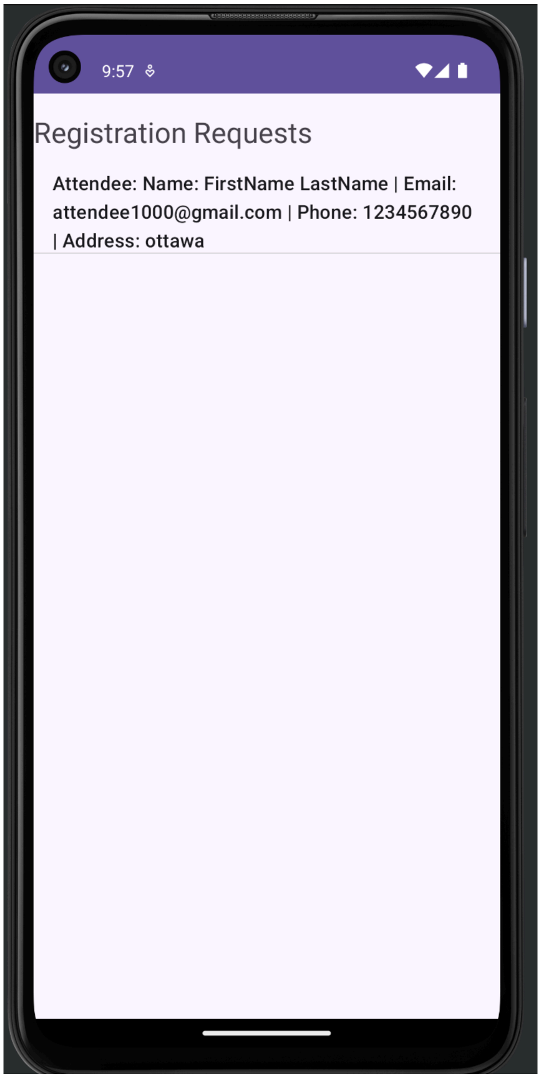

9:52
Email
Password
Submit


9:53
First Name
Enter First Name
Last Name
Enter Last Name
Email Address
Enter Email Address
Password
Enter Password
Phone Number
Enter Phone Number
Address
Enter Address
Submit


9:53
First Name
Last Name
Email
Password
Phone Number
Address
Organization Name
Register

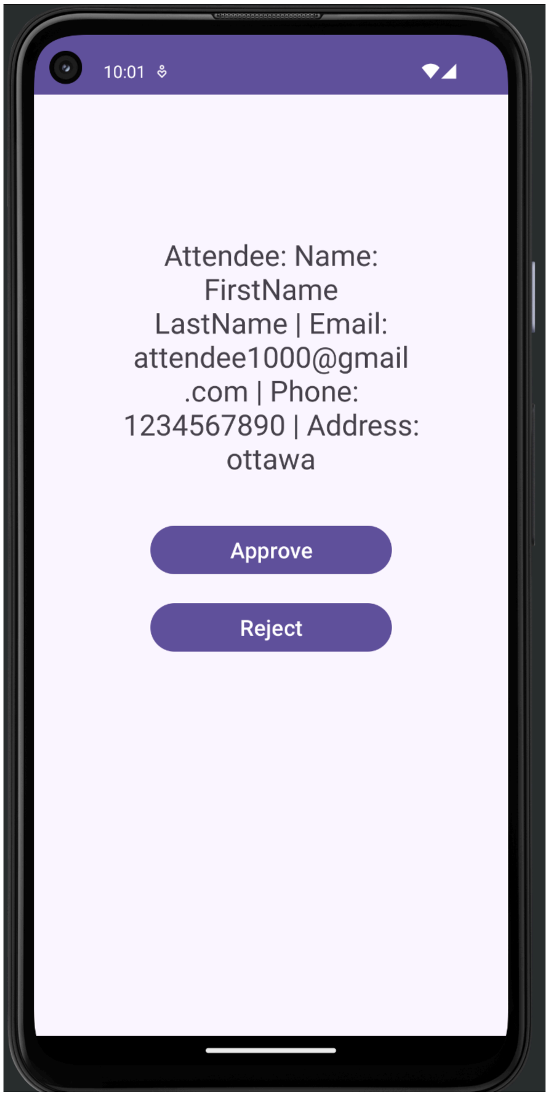


Welcome! You are logged in as Administrator

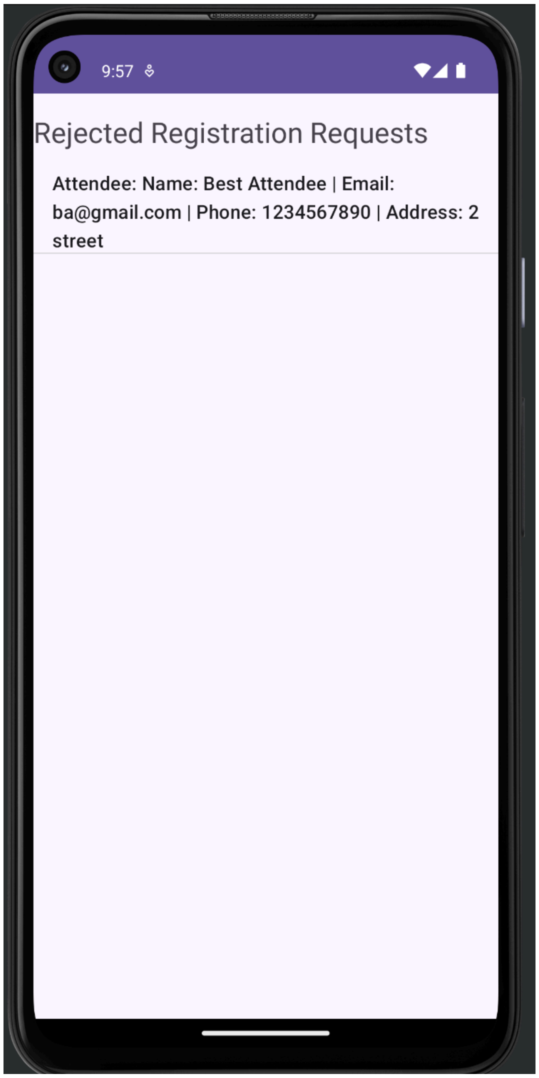


Registration Requests

Attendee: Name: FirstName LastName | Email: attendee1000@gmail.com | Phone: 1234567890 | Address: ottawa

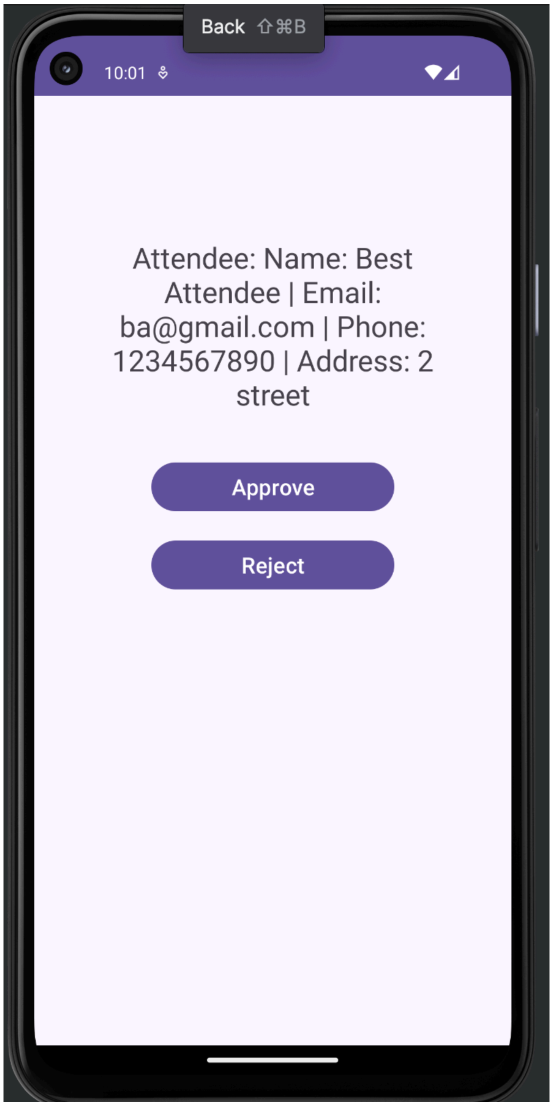


Attendee: Name: FirstName LastName | Email: attendee1000@gmail.com | Phone: 1234567890 | Address: ottawa

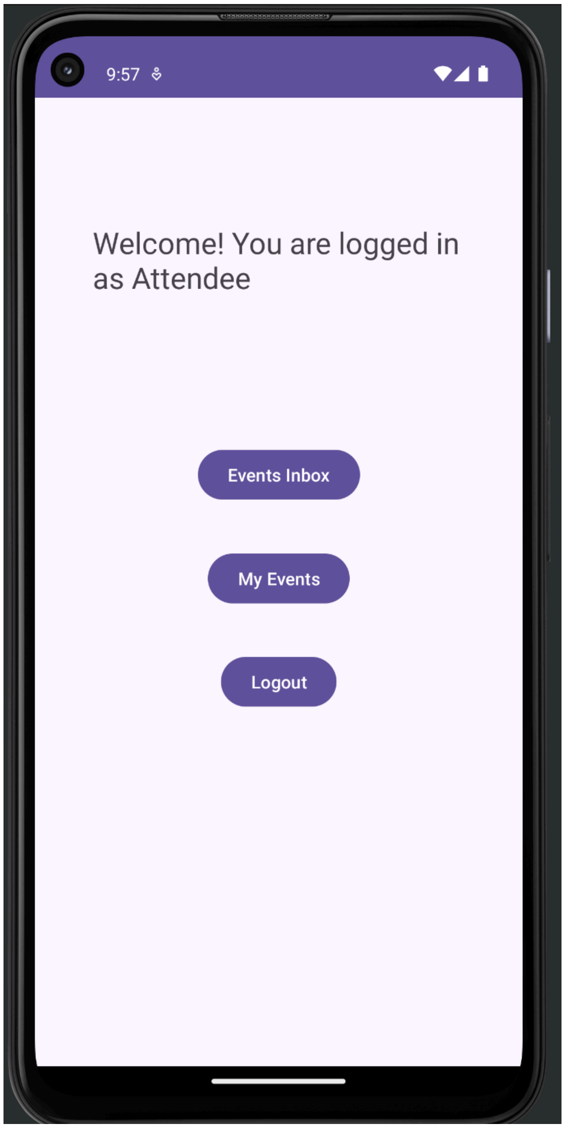


Rejected Registration Requests

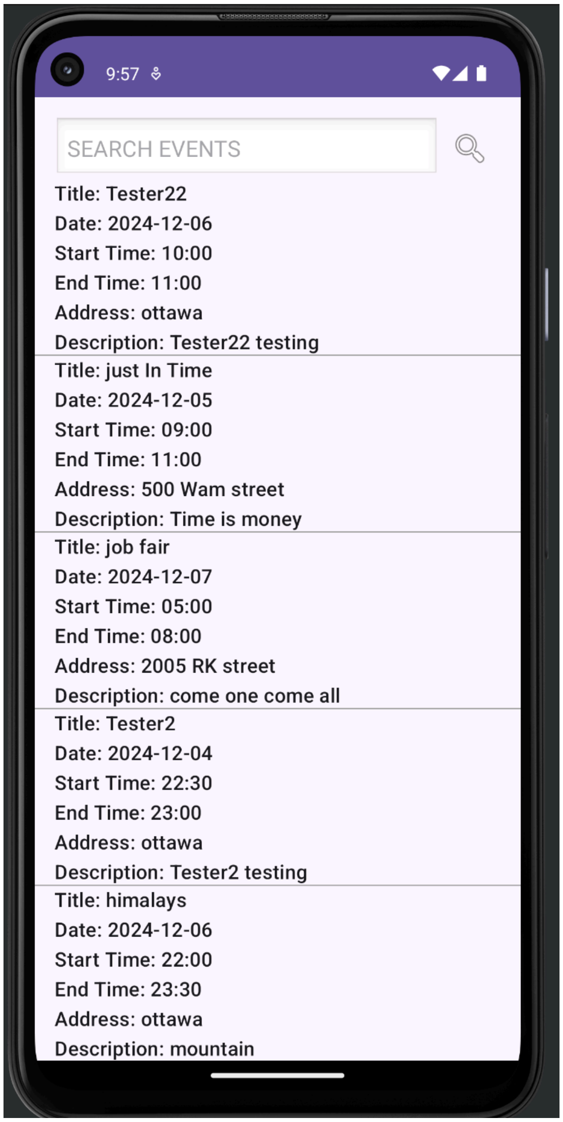
Attendee: Name: Best Attendee | Email: ba@gmail.com | Phone: 1234567890 | Address: 2 street



Attendee: Name: Best Attendee | Email: ba@gmail.com | Phone: 1234567890 | Address: 2 street



Welcome! You are logged in as Attendee



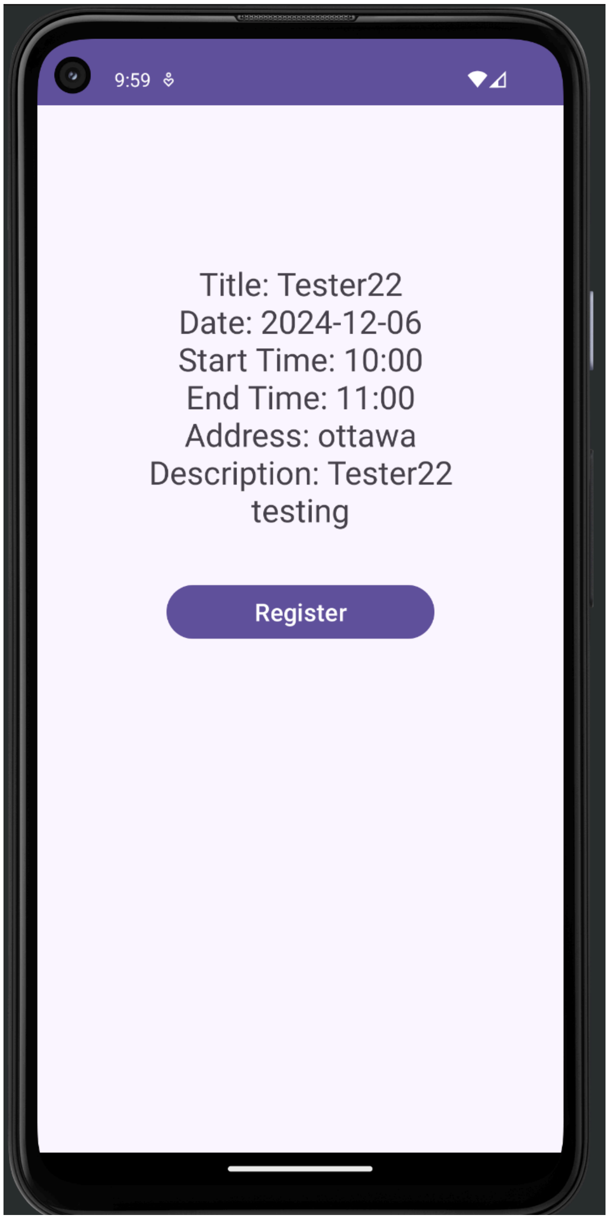
Title: Tester22
Date: 2024-12-06
Start Time: 10:00
End Time: 11:00
Address: ottawa
Description: Tester22 testing

Title: just In Time
Date: 2024-12-05
Start Time: 09:00
End Time: 11:00
Address: 500 Wam street
Description: Time is money

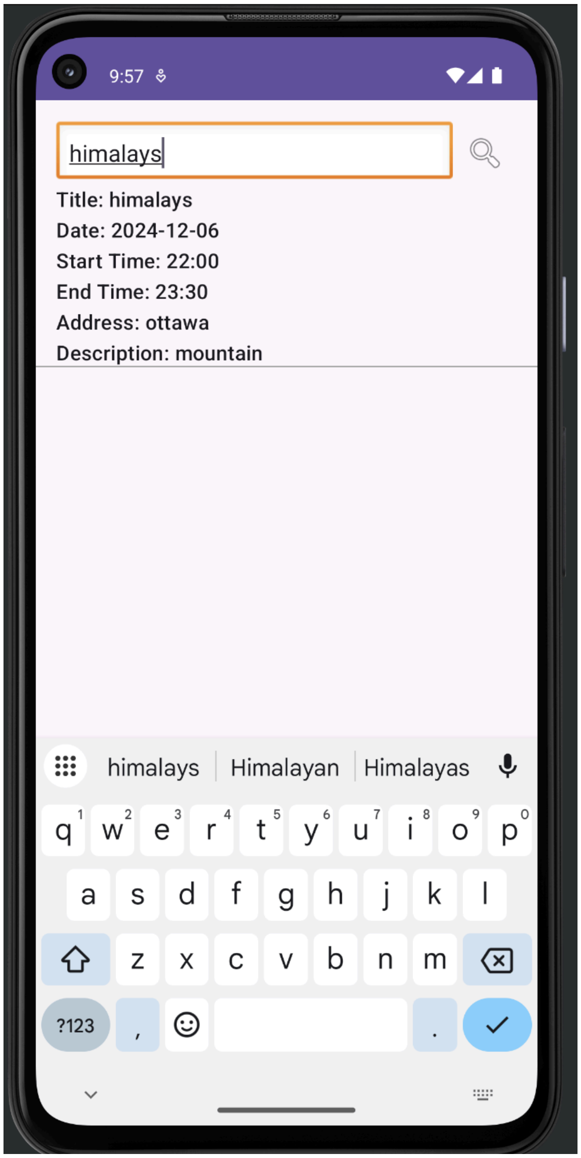
Title: job fair
Date: 2024-12-07
Start Time: 05:00
End Time: 08:00
Address: 2005 RK street
Description: come one come all

Title: Tester2
Date: 2024-12-04
Start Time: 22:30
End Time: 23:00
Address: ottawa
Description: Tester2 testing

Title: himalays
Date: 2024-12-06
Start Time: 22:00
End Time: 23:30
Address: ottawa
Description: mountain

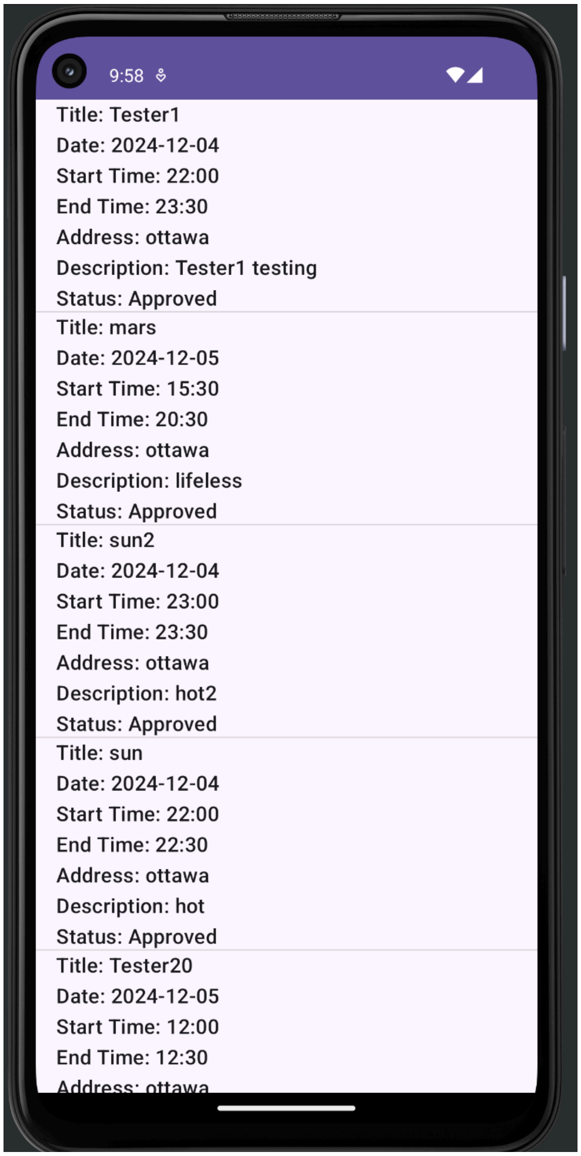


Title: Tester22
Date: 2024-12-06
Start Time: 10:00
End Time: 11:00
Address: ottawa
Description: Tester22
testing



himalays

Title: himalays
Date: 2024-12-06
Start Time: 22:00
End Time: 23:30
Address: ottawa
Description: mountain



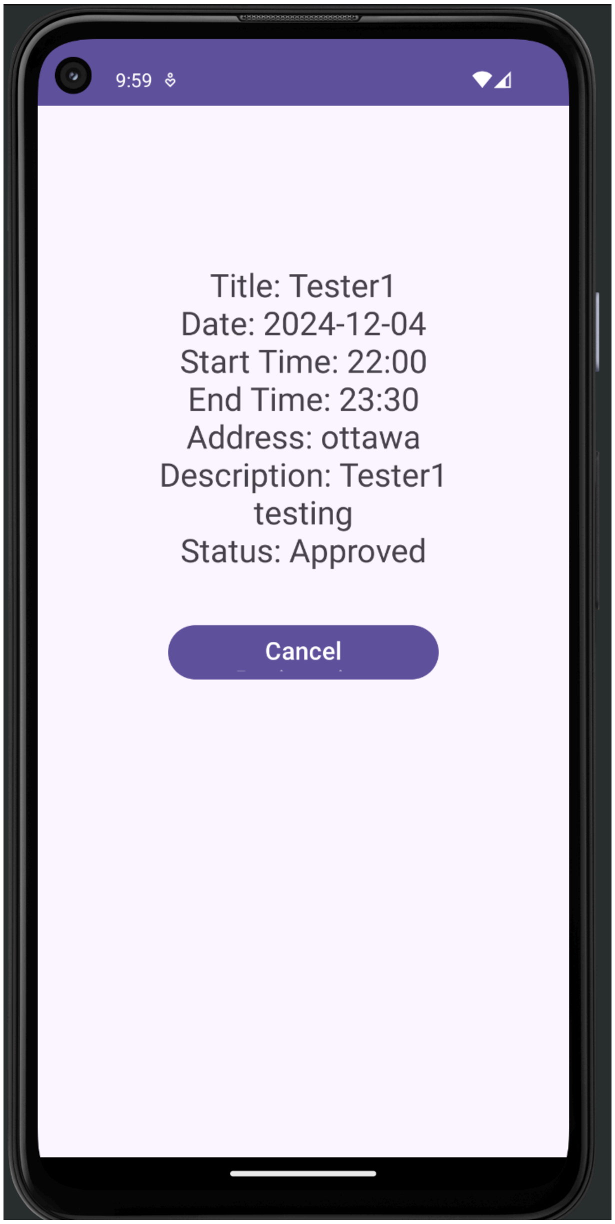
Title: Tester1
Date: 2024-12-04
Start Time: 22:00
End Time: 23:30
Address: ottawa
Description: Tester1 testing
Status: Approved

Title: mars
Date: 2024-12-05
Start Time: 15:30
End Time: 20:30
Address: ottawa
Description: lifeless
Status: Approved

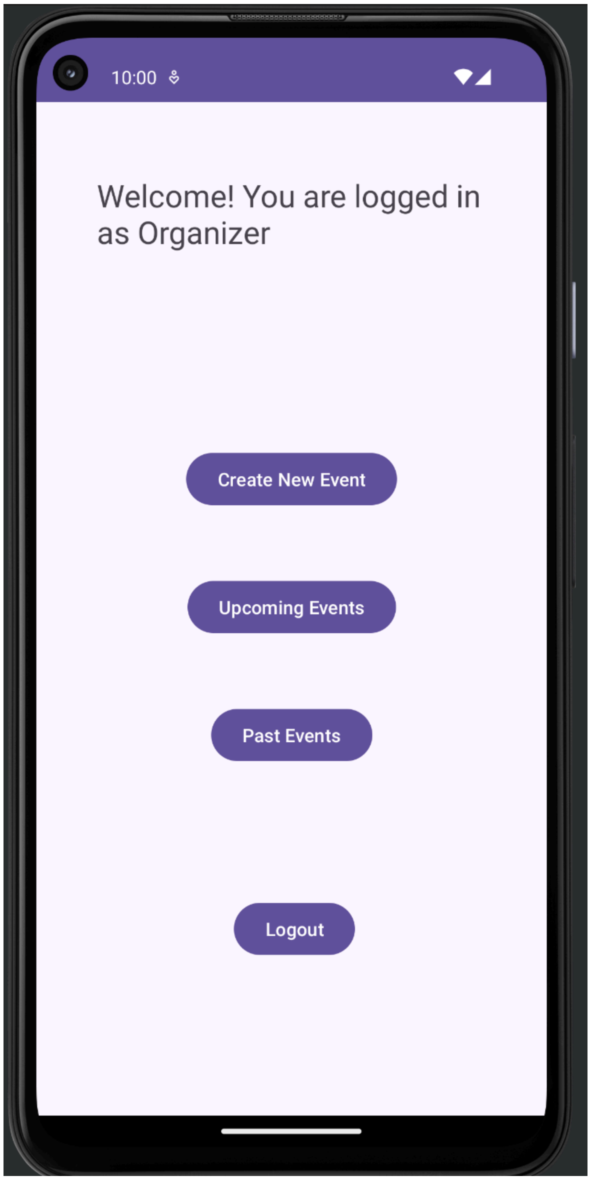
Title: sun2
Date: 2024-12-04
Start Time: 23:00
End Time: 23:30
Address: ottawa
Description: hot2
Status: Approved

Title: sun
Date: 2024-12-04
Start Time: 22:00
End Time: 22:30
Address: ottawa
Description: hot
Status: Approved

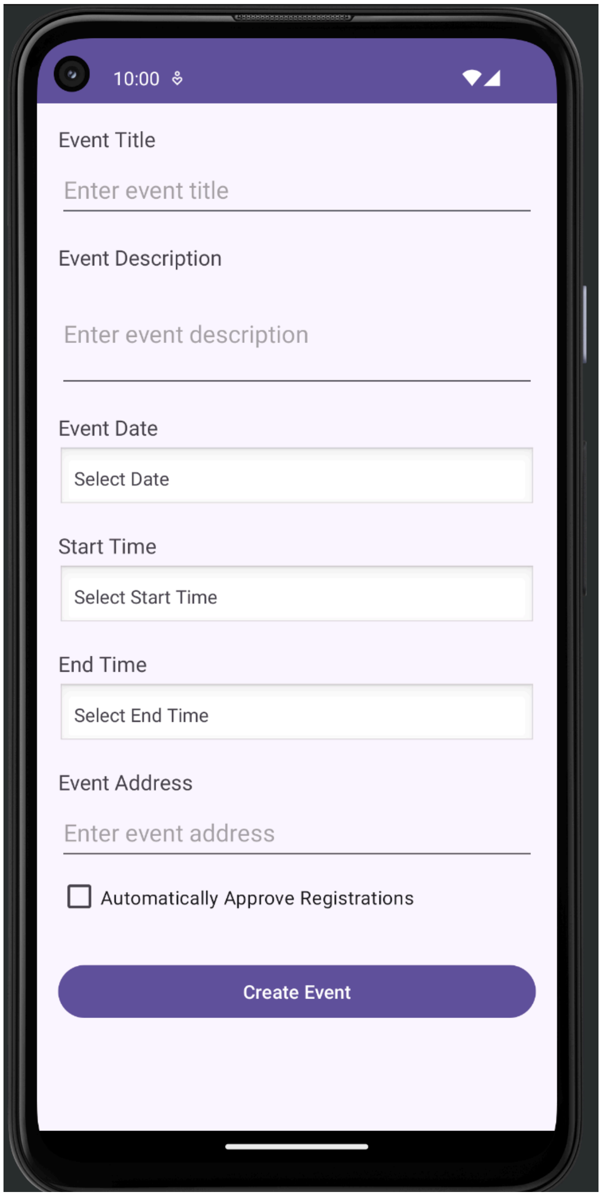
Title: Tester20
Date: 2024-12-05
Start Time: 12:00
End Time: 12:30
Address: ottawa



Title: Tester1
Date: 2024-12-04
Start Time: 22:00
End Time: 23:30
Address: ottawa
Description: Tester1 testing
Status: Approved



Welcome! You are logged in as Organizer



Event Title

Enter event title

Event Description

Enter event description

Event Date

Select Date

Start Time

Select Start Time

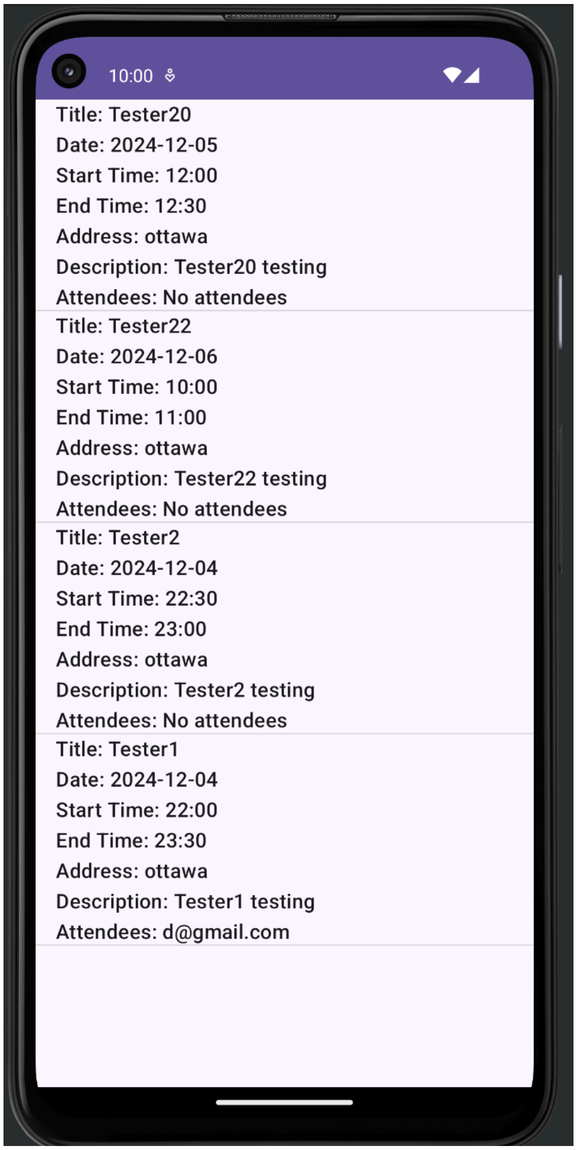
End Time

Select End Time

Event Address

Enter event address

☐ Automatically Approve Registrations

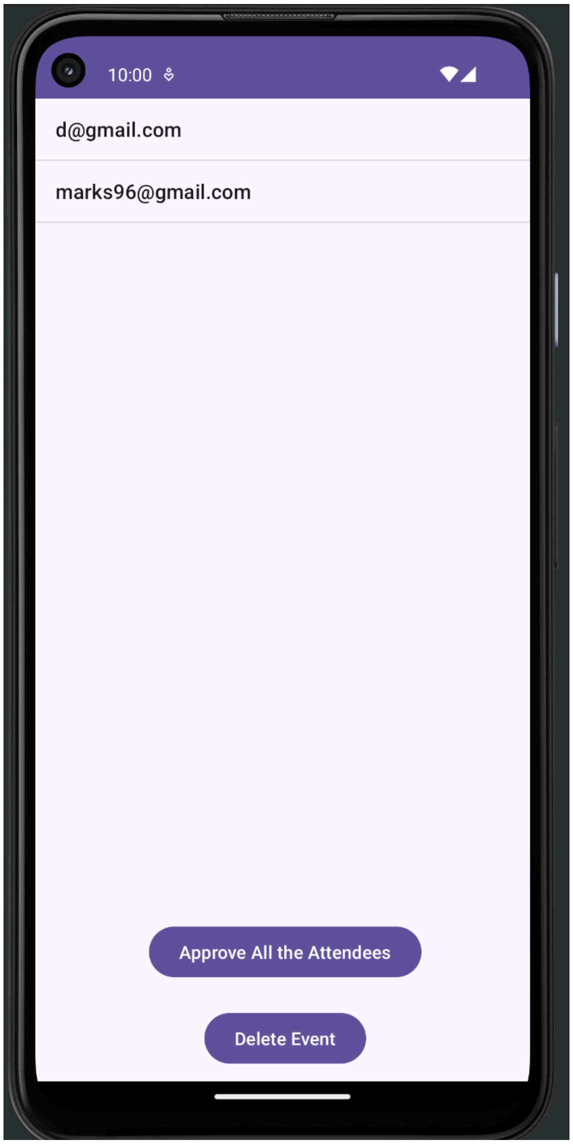


Title: Tester20
Date: 2024-12-05
Start Time: 12:00
End Time: 12:30
Address: ottawa
Description: Tester20 testing
Attendees: No attendees

Title: Tester22
Date: 2024-12-06
Start Time: 10:00
End Time: 11:00
Address: ottawa
Description: Tester22 testing
Attendees: No attendees

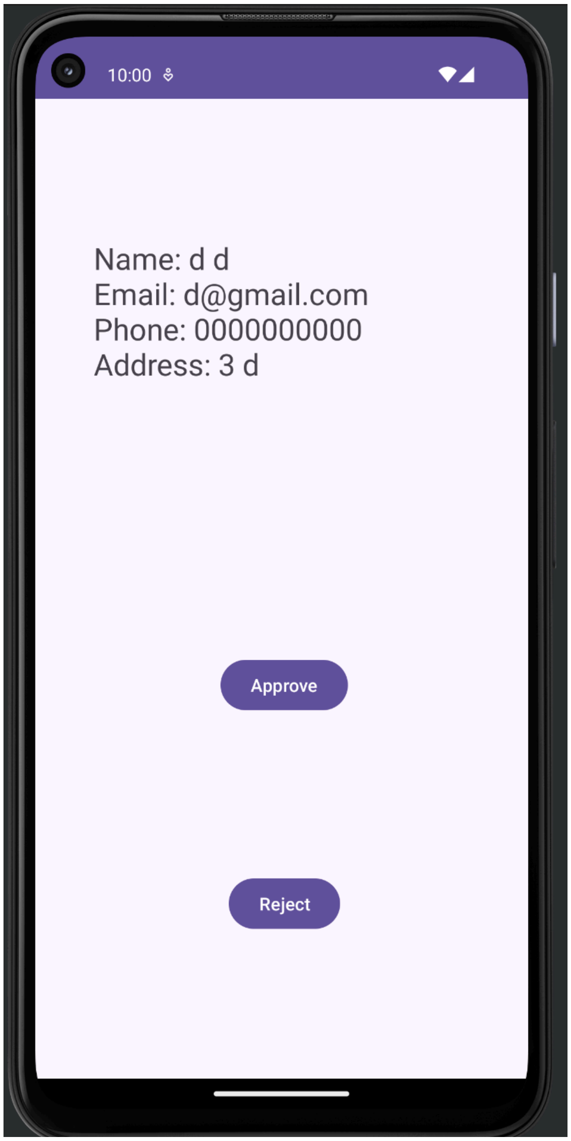
Title: Tester2
Date: 2024-12-04
Start Time: 22:30
End Time: 23:00
Address: ottawa
Description: Tester2 testing
Attendees: No attendees

Title: Tester1
Date: 2024-12-04
Start Time: 22:00
End Time: 23:30
Address: ottawa
Description: Tester1 testing
Attendees: d@gmail.com

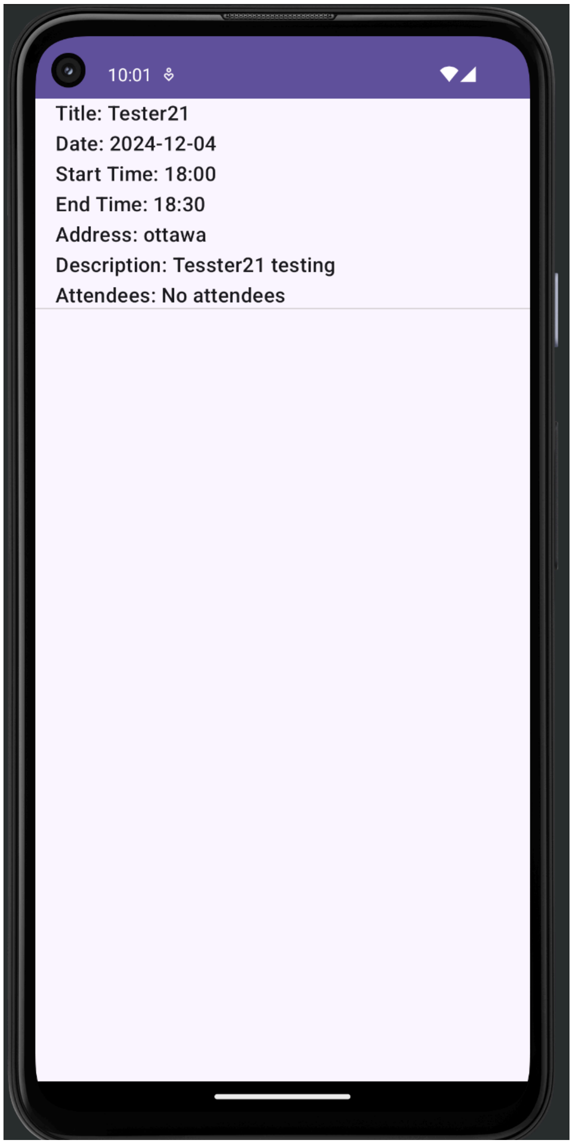


d@gmail.com

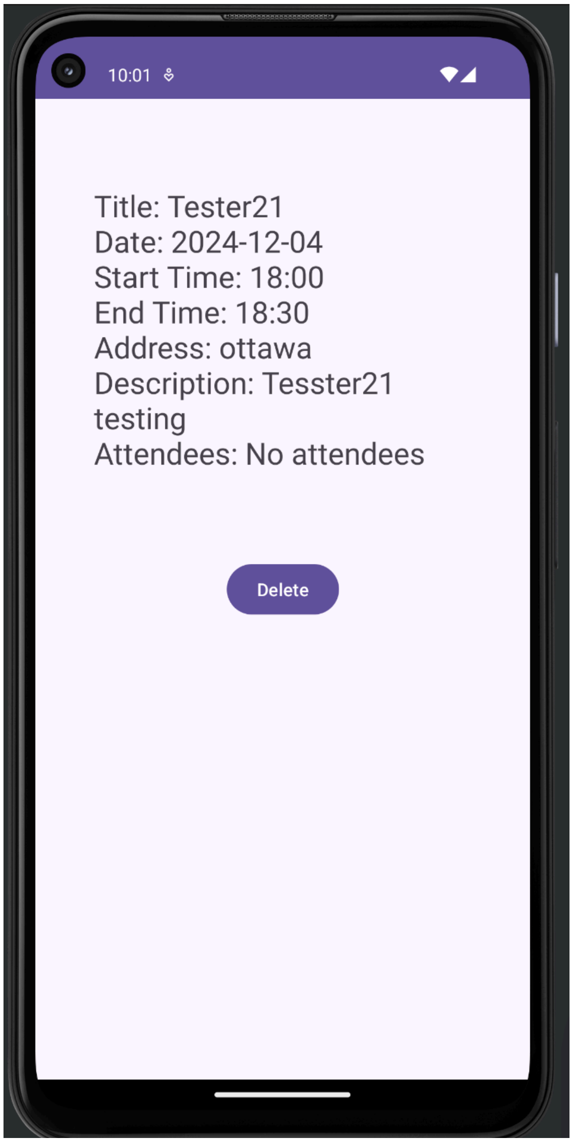
marks96@gmail.com



Name: d d
Email: d@gmail.com
Phone: 0000000000
Address: 3 d



Title: Tester21
Date: 2024-12-04
Start Time: 18:00
End Time: 18:30
Address: ottawa
Description: Tesster21 testing
Attendees: No attendees



Title: Tester21
Date: 2024-12-04
Start Time: 18:00
End Time: 18:30
Address: ottawa
Description: Tesster21 testing
Attendees: No attendees

## Conclusion:

This project was an excellent learning opportunity on how to develop android applications. Throughout the project, we learned how to create different activities, along with how to write the front and backend code for them. We also learned the use of databases and how crucial of a role they play in running the app smoothly. We encountered several issues along the way, but with collaboration among the teammates, we were able to solve them all.