Final Report: Event Attendance Management System (EAMS)

Project Title: Event Attendance Management System (EAMS)

Course: SEG2105 – Introduction to Software Engineering Semester: Fall 2024

Team Members:

Rachel Suriawidjaja (300332168)

James Attia (300353040)

Amani Mohamed (300349984)

Anima Mehraj Mehrin (300278018)

The Event Attendance Management System (EAMS) is an app for Android that helps make managing university events easier. It has three main roles: Attendee, Organizer, and Administrator. The system helps with event registration, approval, and tracking attendance. We used Firebase as the backend to make sure it's secure and works in real-time. EAMS is designed to handle different situations, like registration conflicts, admin approvals, and managing participants, while being easy to use. This report explains how we designed and built the EAMS project, covering the main features and how our team worked together.

Project Objectives

The main goal of EAMS is to improve how university events are managed. Specific objectives include:

- Automation: Replace manual registration and attendance tracking with an automated system.
- Scalability: Make a system that can handle lots of users and events at the same time.
- Security: Use strong authentication and role-based access control with Firebase.
- Usability: Design a simple and easy-to-use interface for all types of users.

We met these goals by using software engineering methods, working as a team, and testing thoroughly.

Features Implemented

Administrator Features

The Administrator manages user accounts and keeps the system running smoothly:

- Account Management: Administrators log in with special credentials saved in Firebase, making sure only the right people can access important features.
- Registration Approval: Administrators can approve or reject registration requests from attendees and organizers.
- Rejected Request Management: Administrators can look at previously rejected requests and approve them if needed.
- Registration Status Feedback: The system sends messages to users about their registration status, so everyone knows what is happening.

Organizer Features

Organizers manage events. Main features include:

- Event Creation: Organizers can create events by giving details like title, description, date, time, and location. They can choose if attendee approval is automatic or manual.
- Validation Rules: Event dates cannot be set in the past, and times have to be in 30-minute intervals.
- Attendee Management: Organizers can approve or reject requests from attendees. There is also an "Approve All" option to save time for big events.
- Event Deletion: Organizers can delete events unless attendees have already registered, to avoid accidental loss of data.
- Event History: The app shows lists of past and upcoming events so organizers can keep track of everything.

Attendee Features

Attendees are the main users who interact with events:

- Event Search: Attendees can search for events by keywords in the title or description.
- Event Registration: After finding an event, attendees can request to join. Once approved, the event is added to their list.
- Conflict Prevention: The system stops users from registering for events that overlap with their current schedule.
- Cancellation: Attendees can cancel registrations if the event hasn’t started within the next 24 hours.

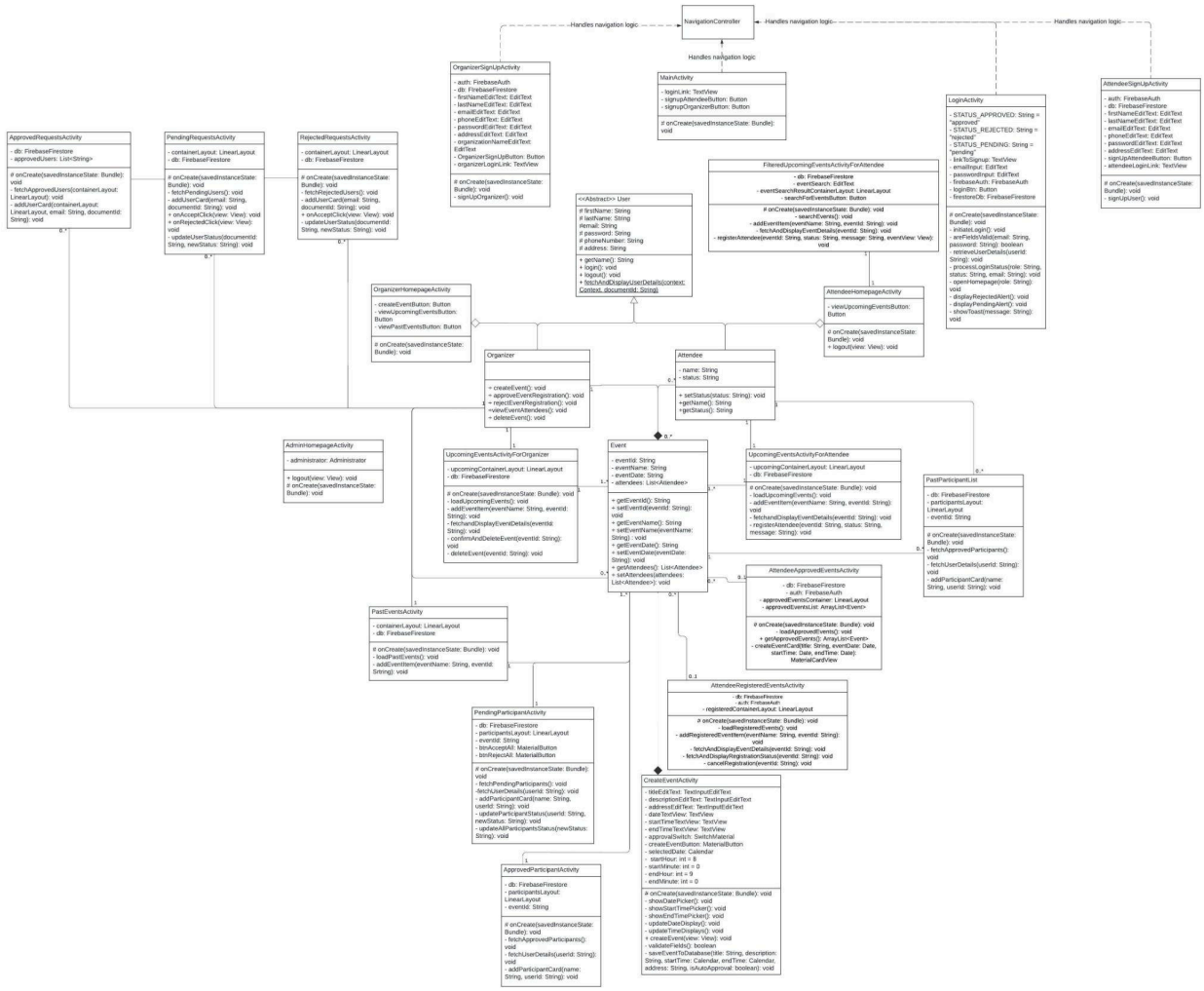
- Registration Status Tracking: Attendees get updates on whether their registration is approved, rejected, or still waiting.

Updated UML Class Diagram

The UML class diagram is a key part of the project. It shows how different classes, like User, Event, and the role-specific ones (Attendee, Organizer, Administrator), connect with each other. Activities like login, event creation, and managing participants are modeled to follow the Model-View-Controller (MVC) design. This helps keep business logic separate from the user interface, making the app easier to maintain and expand.

Key components include:

- User Class: The base class with shared information like name, email, and role.
- Event Class: Holds event details and has methods for validation and registration.
- Activity Classes: Handle interactions between users and the system (e.g., LoginActivity, CreateEventActivity).



Technical Overview

Development Environment:

- IDE: Android Studio
- Languages: Java for coding logic, XML for user interface layouts

Backend:

- Database: Firebase Firestore for real-time data storage and syncing
- Authentication: Firebase Authentication for secure logins and role management

Design Principles:

- Field validation with real-time error messages
- Separation of user roles to keep everything secure
- Modular code structure to make future improvements easier

Collaboration and Contributions

Our team used an agile approach, dividing tasks based on each person's strengths and availability. We planned and completed each part of the project carefully.

Deliverable 1			
	GitHub Classroom Setup: The team created in GitHub classroom contains all members of the group	N/A	All
	Repository Contributions: Each member of the group has made at least one commit to the repository	N/A	All
	Demo Video Submission: The team has submitted a demo video showcasing the app's functionality	N/A	Anima
	UML Class Diagram: The UML Class diagram of your domain model is valid	N/A	James
	APK Submission: The APK file is submitted correctly with the release.	N/A	Anima
	Account Creation: A user can create an Attendee or Organizer account with the required fields.	James	Anima/James
	Administrators Information: Create the firestore database, pre-fill it with the administrators email and password, so when the user goes to log in, if they enter the administrators email and password, it will take them to admin page.	Amani	Amani
	Welcome Screen: The Administrator, Organizer, or Attendee can see the "welcome screen" after succesful authentication, which specifies the user's role.	James	Anima
	Log Off Functionality: The user can log off after logging in	Rachel	Rachel
	Field Validation: All fields are validated, with appropriate error messages displayed for incorrect inputs.	James	James/Anima
	Database Usage: The group uses a database (e.g., Firebase, SQLite, or another similar technology) to store user data.	N/A	Anima

Deliverable 2			
	UML Class Diagram: The updated UML Class diagram of your domain model is valid and includes relevant classes from Deliverables 1 and 2.	N/A	Anima
	APK Submission: The APK file is submitted correctly with the release.	N/A	Anima
	Demo Video Submission: The team has submitted a demo video showcasing the app's functionality.	N/A	Anima
	The Administrator can view the list of registration requests.	James	James
	The Administrator can view the information associated with each request (the information the user entered during registration).	Rachel/James	James
	The Administrator can approve or reject a registration request.	Rachel/James	Anima/Amani
	If a registration request is rejected, it is added to the list of registration requests.	Anima	Anima
	The Administrator can view the list of previously rejected registration requests	Anima	Anima
	The Administrator can approve a previously rejected request, which removes it from the rejected list.	Anima	Anima
	Login attempt : if their request is approved, they are directed to the welcome screen.	James	James/Rachel
	Login attempt : If their request was rejected, they receive a message informing them of the rejection and displaying a phone number to contact the Administrator	James	James/Rachel
	If their request has not been processed yet, they receive a message informing them that pproval is pending. (Handling of all three scenarios is required.)	James	James
	Database Storage for Registration Requests: All registration requests, along with their statuses, are stored in the DB.	N/A	Anima
	Bonus: Notifications: When a user's registration request is approved or rejected they receive an email or phone notification	N/A	N/A
Deliverabl			

e 3			
	UML Class Diagram: The UML Class diagram of your domain model (including relevant classes from Deliverables 1, 2, and 3) is valid and complete	N/A	Rachel
	APK Submission: The APK file is submitted correctly with the release (v0.3)	N/A	Anima
	Demo Video Submission: The team has submitted a demo video	N/A	Anima
	Event Creation: The Organizer can create new events by specifying the title, description, date, start time, end time, and event address.	James	James
	Date Validation: The Organizer cannot select a date that has already passed.	Rachel	Rachel
	Manual and Automatic Registration Approval: The Organizer can choose during event creation whether to manually approve Attendee registration requests or approve all automatically.	James	Anima
	The Organizer can view a list of upcoming events	Amani	Amani
	Organizer dashboard and approve all attendee sign up requests	Amani	Amani
	Organizer can view a list of past events.	Amani	Amani
	The Organizer can view a list of Attendees who requested registration for an upcoming event.	Anima/Amani	Anima/Amani
	The Organizer can view the information for each Attendee.	James/Anima	Anima
	The Organizer can approve or reject each Attendee's registration individually or select "approve all" to approve all registrations at once.	Anima	Anima
	Event Deletion: The Organizer can delete any event they ahve previously created.	Anima	Anima
	Field Validation and Error Messages: All fields are validated, with appropriate error messages displayed for incorrect inputs.	N/A	James/Rachel
	Bonus - CircleCl Integration: The group integrates with	N/A	N/A
Deliverable 4			
	UML Class Diagram: The UML Class diagram of your domain model (including all relevant classes from previous deliverables) is valid and complete.	N/A	Rachel

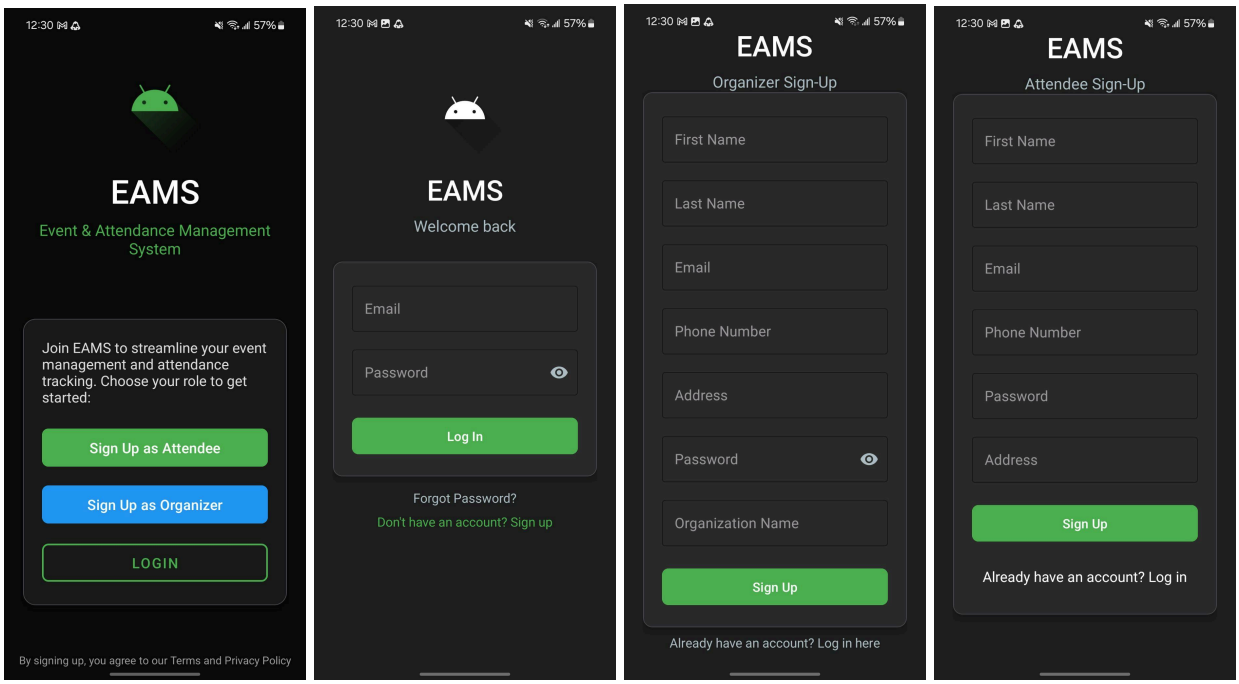
	APK Submission: The APK file is submitted correctly with the release.	N/A	Anima
	Demo Video Submission: The team has submitted a demo video showcasing the app's functionality.	Anima	Anima
	View Registered Events: The Attendee can view a list of events they have requested to register for, twith the newest events displayed at the top.	Amani	Amani
	Registration Status Indicator: There is an indicator showing whether the Attendee's registration is approved, rejected or not processed yet by the Organizer.	Anima	Anima
	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.	Anima	Anima
	Search Events: Attendees 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 to view its (title, description, date, start time, end time, and event address.	Rachel	Rachel
	Request Event Registration: After finding an event, the Attendee can request registration. Once requested, the event is added to their lis of events and disappears from search results.	Rachel	Rachel
	Conflict Prevention: The system prevents Attendees from registering for events that conflict with events they have already registered for or requested registration for.	Amani	Amani
	Event Deletion: If an Organizer attempts to delete an event associated with one or more approved Attendee registrations, the system displays a message preveting deletion.	Amani	Amani
	Approved Events for Attendee List: When an organizer approves an attendee's signup request for an upcoming event, the events details are added to this list. The list is used to keep track of upcoming events and to prevent scheduling conflicts for the attendee.	Amani	Amani

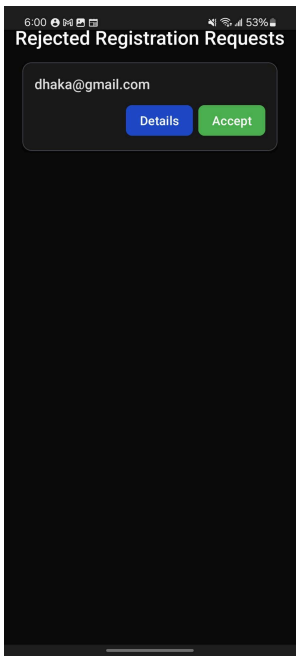
	Field Validation and Error Messages: All fields are validated with appropriate error messages for incorrect inputs.	N/A	James
	Unit Test Cases: At least four unit test cases are implemented	N/A	James
	Bonus: Attendee Event Reminder Notifications: the system sends push notifications or in-app reminders to Attendees 24 hours before an event starts, ensuring they dont miss it.	N/A	N/A
	Final Report	N/A	James
	Title Page	N/A	All
	Updated UML class diagram	N/A	Rachel
	Table specifying the contributions of team members for each Deliverables	N/A	All
	All the screenshots of your app	N/A	All
	Lessons learned	N/A	All

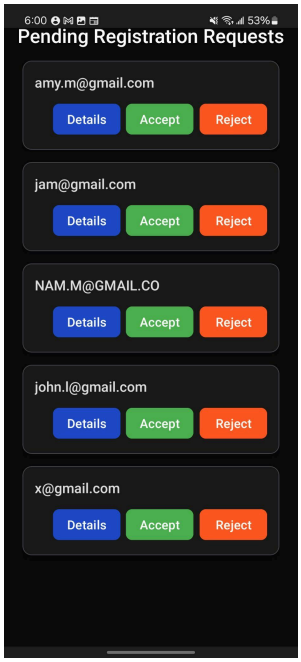
Screenshots

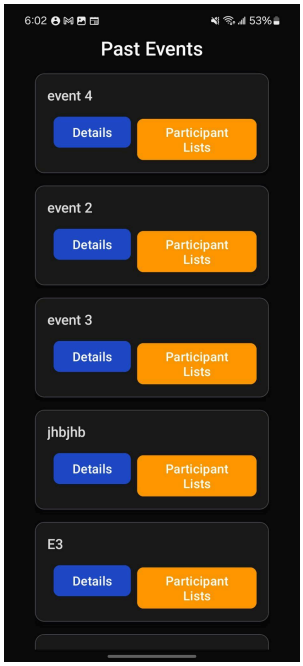
Screenshots showcasing key functionalities are included in the appendix. These highlight:

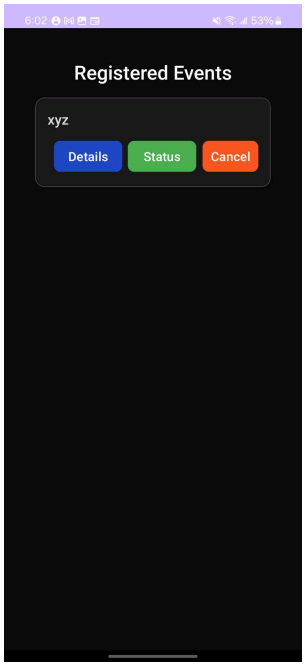
- The login interface with role-based redirection.
- Event creation and validation screens.
- Administrator dashboards for request management.
- Attendee views for registration and status updates.

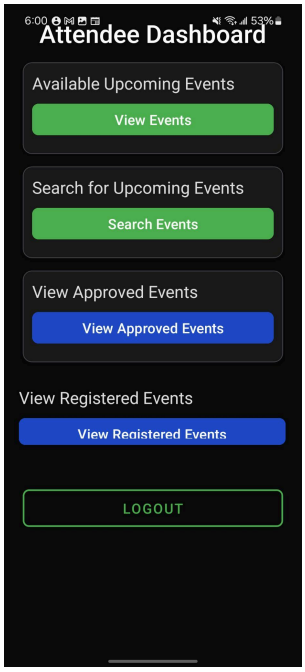

12:30
57%
EAMS
Event & Attendance Management System
Join EAMS to streamline your event management and attendance tracking. Choose your role to get started:
Sign Up as Attendee
Sign Up as Organizer
LOGIN
By signing up, you agree to our Terms and Privacy Policy
12:30
57%
EAMS
Welcome back
Email
Password
Log In
Forgot Password?
Don't have an account? Sign up
12:30
57%
EAMS
Organizer Sign-Up
First Name
Last Name
Email
Phone Number
Address
Password
Organization Name
Sign Up
Already have an account? Log in here
12:30
57%
EAMS
Attendee Sign-Up
First Name
Last Name
Email
Phone Number
Password
Address
Sign Up
Already have an account? Log in

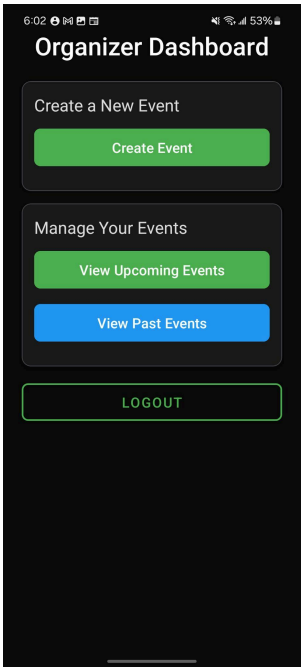

Rejected Registration Requests
dhaka@gmail.com
Details
Accept

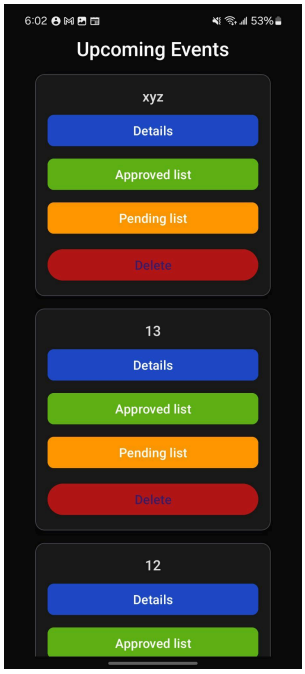

Pending Registration Requests
amy.m@gmail.com
Details
Accept
Reject
jam@gmail.com
Details
Accept
Reject
NAM.M@GMAIL.CO
Details
Accept
Reject
john.l@gmail.com
Details
Accept
Reject
x@gmail.com
Details
Accept
Reject

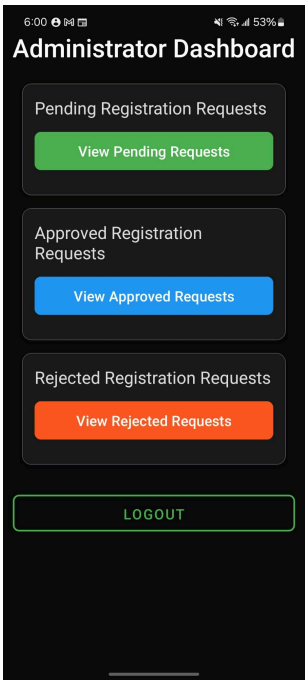

Past Events
event 4
Details
Participant Lists
event 2
Details
Participant Lists
event 3
Details
Participant Lists
jhbjhb
Details
Participant Lists
E3
Details
Participant Lists

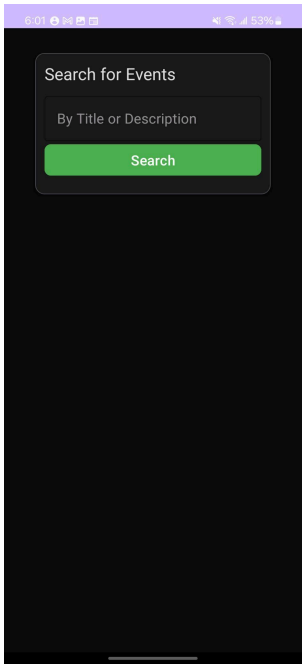

Registered Events
xyz
Details
Status
Cancel

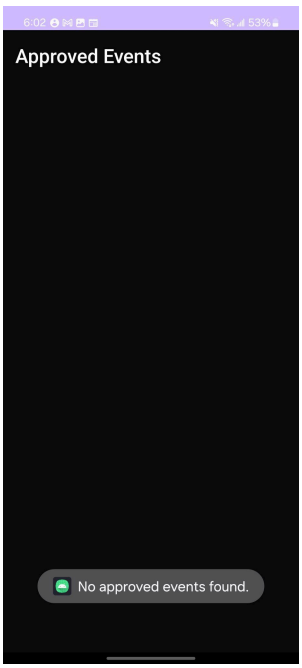

Attendee Dashboard
Available Upcoming Events
View Events
Search for Upcoming Events
Search Events
View Approved Events
View Approved Events
View Registered Events
View Registered Events
LOGOUT

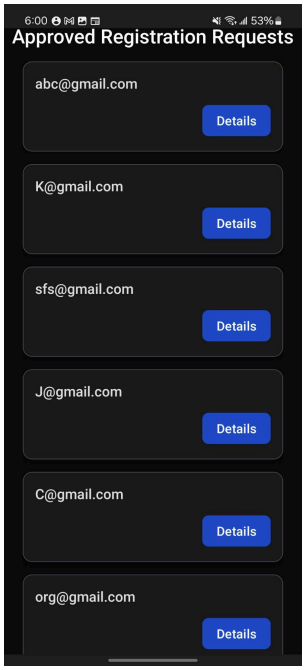

Organizer Dashboard
Create a New Event
Create Event
Manage Your Events
View Upcoming Events
View Past Events
LOGOUT

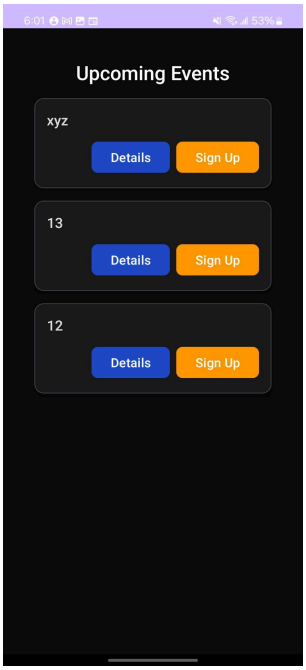

Upcoming Events
xyz
Details
Approved list
Pending list
Delete
13
Details
Approved list
Pending list
Delete
12
Details
Approved list

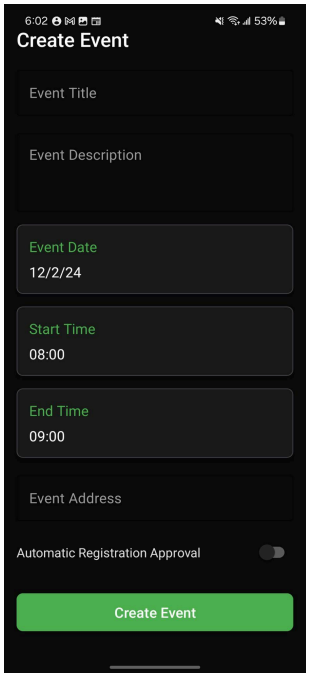

Administrator Dashboard
Pending Registration Requests
View Pending Requests
Approved Registration Requests
View Approved Requests
Rejected Registration Requests
View Rejected Requests
LOGOUT


Search for Events
By Title or Description
Search


Approved Events
No approved events found.


Approved Registration Requests
abc@gmail.com
Details
K@gmail.com
Details
sfs@gmail.com
Details
J@gmail.com
Details
C@gmail.com
Details
org@gmail.com
Details


Upcoming Events
xyz
Details
Sign Up
13
Details
Sign Up
12
Details
Sign Up


Create Event
Event Title
Event Description
Event Date
12/2/24
Start Time
08:00
End Time
09:00
Event Address
Automatic Registration Approval
Create Event

Testing and Quality Assurance

To ensure the application met all requirements:

- Unit Testing: Focused on core functionality like login validation, event creation, and role-based access.
- Manual Testing: Team members tested the app on different devices to identify and resolve UI inconsistencies.
- Firebase Integration Testing: Verified secure and accurate data synchronization across devices.

Lessons Learned

The project provided valuable insights into:

- Collaborative Development: Leveraging GitHub for version control and task assignments.
- Firebase Integration: Understanding real-time database functionalities and authentication mechanisms.
- Role-Based Access Control: Designing a secure, multi-role application architecture.
- UI/UX Design: Importance of user-friendly interfaces adhering to Android Design Guidelines.

Challenges faced included debugging Firebase interactions and ensuring consistent validation across the app.

Conclusion

The EAMS project successfully fulfilled all deliverables, demonstrating robust functionality and adherence to software engineering principles. The final product is a scalable, secure, and user-friendly application that meets the needs of its target audience.

Future enhancements could include:

- Push notifications for event reminders.
- Enhanced analytics for organizers to track event performance.
- Multi-language support to increase accessibility.

The project showcases the team's ability to design, develop, and deliver a comprehensive mobile application.