

## **Seshy: College Events**

13-SWEF25: Alex Shaw (**11883919**), Eniola Akinola (**11809831**), Quinn Bankhead (**11650795**),  
Tomas Stoker (**11786010**)

Professor Kumar

CPTS 322: Software Engineering Principles I

Sunday, October 26, 2025

**Abstract**

College students today face a growing challenge: staying connected in an increasingly digital and disconnected world. Finding campus events, study sessions, or social gatherings often depends on scattered posts, knowing the right people well enough, or short-lived stories that fade before students can act. Existing platforms fail to engage, leaving campus life fragmented and participation low. Seshy offers a modern, tech-driven solution. Designed as an easy-to-use, intuitive, and iOS-native event discovery and management app, it helps students create, join, and share events through one trusted platform (a single source of truth). Smart recommendations, map-based filters, and verified sign-in ensure that every interaction feels personal, safe, and relevant. A built-in reputation system fosters trust and encourages genuine engagement between students. Beyond convenience, Seshy serves a deeper purpose—rebuilding connection in a decade defined by loneliness and social isolation. The app transforms event discovery into community building, helping college towns and campuses thrive through meaningful, in-person interaction. By combining digital convenience and real-world connection, Seshy brings people together before they even realize they belong together—helping students find connection, purpose, and their place on campus.

## **Table of Contents**

### **1. Introduction**

1.1 Overview

1.2 Problem Statement

1.3 Objectives and Scope

1.4 Importance and Potential Impact

1.5 Team Collaboration

### **2. Requirements**

2.1 Functional Requirements

2.2 Non-Functional Requirements

### **3. User Stories and Scenarios**

### **4. Traceability Matrix**

### **5. UML Design Diagrams**

5.1 Use Case Diagram

5.2 Class Diagram

5.3 Collaboration Diagram

5.4 Sequence Diagram

5.5 Activity Diagram

5.6 Component Diagram

### **6. Reflection**

### **7. References**

## **1. Introduction**

### **1.1 Overview**

Seshy is an all-in-one event discovery and management application designed for college campuses, starting with Washington State University in Pullman. It enables students to organize, discover, and attend study sessions, social gatherings, and parties—all within a single, user-friendly mobile app. Unlike fragmented event boards or temporary social media posts, Seshy acts as a trusted source of truth for both academic and social events.

### **1.2 Problem Statement**

Current event-sharing systems on college campuses are ineffective and decentralized. Platforms, such as Presence and university calendars, are poorly maintained, difficult to navigate, and fail to capture student interest. Social media, though popular, is ephemeral and unstructured—making it easy for many to miss opportunities. As a result, study groups are hard to form unless participants already know each other, and off-campus events lack coordination, exclusivity, and reliability.

### **1.3 Objectives and Scope**

Our primary objectives are to:

- Centralize event creation, discovery, and management into a single mobile application;
- Enable personalized recommendations based on class schedules, interests, and social reputation;
- Support both study-focused and recreational events with privacy and trust as core principles; and
- Provide a scalable architecture capable of integrating with university systems (links to room reservations and more).

The project scope focuses on an iOS prototype built with SwiftUI, Core Data, and a cloud backend, emphasizing performance, usability, and data security.

### **1.4 Importance and Potential Impact**

Seshy aims to redefine how college students connect and participate in campus life. By encouraging spontaneous, localized, and trusted real-world interactions, the platform promotes healthier social engagement and combats feelings of isolation and disconnection common among university students. Over time, Seshy can expand to other universities, functioning as a reputation-driven social utility for real world connection.

### **1.5 Team Collaboration**

This project was a collaborative effort among four team members who contributed to different areas of development and documentation. Alex Shaw, as team leader, authored the abstract and introduction, formatted and compiled the submission, and prepared the component diagram. Eniola Akinola contributed to system and UML diagrams. Quinn Bankhead and

Tomas Stoker assisted with the traceability matrix, document organization, and diagram creation. Together, the team ensured that the Seshy project was designed with both technical rigor and real-world usability in mind.

## 2. Requirements

### 2.1 Functional Requirements

- **FR-01:** The system shall allow users to register using a valid phone number and shall send a one-time verification code to complete registration.
- **FR-02:** The system shall send time-based notifications for upcoming events, such as reminders as an event approaches.
- **FR-03:** The system shall allow users to schedule study events on or off campus, including reserving study spaces (when available) and notifying invitees.
- **FR-04:** The system shall allow a group leader to add or remove members from a study group.
- **FR-05:** The system shall allow users to filter events by class, time, and distance.
- **FR-06:** The system shall allow users to enter class schedules, interests, and preferences to enable personalized recommendations.
- **FR-07:** The system shall provide search across events with both list and map views showing nearby results that match entered terms.
- **FR-08:** The system shall allow users to reset or change their phone registration through an email-based verification process.
- **FR-09:** The system shall allow users to enable or disable notification categories: invites, reminders, and announcements and manage push/email notifications.

### 2.2 Non-Functional Requirements

- **NFR-01:** The system shall load any view in under 2.0 seconds (95th percentile) under normal load.
- **NFR-02:** The system shall support at least 500 concurrent users without performance degradation beyond 3.0 seconds latency.
- **NFR-03:** The system shall be integratable with campus room reservation services via API or export/import to streamline study room bookings.
- **NFR-04:** All sensitive data at rest shall be encrypted with AES-256-GCM. Passwords shall be hashed. All chats shall support end-to-end encryption for anonymous postings.

- **NFR-05:** The system shall minimize collection of personally identifiable information (PII) and be compliant with privacy regulations: GDPR in the European Union and CCPA in California. Users shall be informed of data use and retention.
- **NFR-06:** The system shall be designed with sufficient color contrast, keyboard access, meaningful error messaging, and dark/light mode support.
- **NFR-07:** The system shall maintain 99.5% uptime during academic semesters, excluding scheduled maintenance.

### 3. User Stories and Scenarios

**US-01:** As a registered user, I want to sign in with my phone number and a verification code, so that I can securely access my account without a password.

- **Scenario A:** Request Verification Code  
 Given I am on the sign-in screen  
 And I enter a valid phone number  
 When I tap "Send Code"  
 Then a one-time verification code is sent to my phone  
 And I see a confirmation that the code was sent.
- **Scenario B:** Successful Sign In with Valid Code  
 Given I have received a verification code  
 And I am on the code entry screen  
 When I enter the correct code before it expires  
 Then I am signed in  
 And I am redirected to the Home page.
- **Scenario C:** Invalid Code  
 Given I requested a verification code  
 When I enter an incorrect code  
 Then I remain on the code entry screen  
 And I see an error message indicating the code is invalid.

**US-02:** As a group leader, I want to control invitations, so I can decide who is invited to the study session.

- **Scenario A:** Invite Recommended Users  
 Given I am creating a new study session  
 When I open the "Invite Participants" view  
 And I view the recommended users (classmates and past attendees)  
 And I select specific users to invite  
 Then those users receive an invitation notification.
- **Scenario B:** Remove Invitee Before Sending

Given I have selected multiple users to invite  
 When I deselect a user before creating the event  
 And I click the “Create Event” button  
 Then the deselected user does not receive an invitation  
 And only the remaining selected users are notified.

**US-03:** As a group leader, I want to choose a convenient location so attendees can easily meet.

- **Scenario A:** On-Campus Reservation

Given that I am creating an event  
 When I select the “On Campus” button  
 And I browse the campus map to choose a building and room  
 Then the selected room and time are attached to the event  
 And if required, a room reservation request is created.

- **Scenario B:** Off-Campus Selection

Given I am creating an event  
 When I select the “Off Campus” button  
 And I search or pick from popular/past locations on a map  
 Then the chosen address appears in the event details and on the map view.

**US-04:** As a study group member, I want timely notifications so I can plan for upcoming events.

- **Scenario A:** Receive Reminder for Subscribed Event

Given I accepted an event invitation  
 When the event is starting soon  
 Then I receive reminder notifications according to my settings

- **Scenario B:** No Reminder for Declined Event

Given I declined an event invitation  
 When reminders are sent for that event  
 Then I do not receive any notification for that event.

**US-05:** As a student, I want to sort and filter my events feed so I can find the most relevant sessions.

- **Scenario A:** Filtered By Class

Given that I am on the Events feed  
 When I open Filters  
 And I select class “CPTS 322”  
 Then only events tagged with “CPTS 322” are shown.

- **Scenario B:** Sorted By Time

Given that I am on the Events feed

When I choose to sort by time (“Start time (descending)”)

Then events are displayed in chronological order from soonest to latest.

- **Scenario C:** Combined Class and Distance Filters

Given that I am on the Events feed

When I filter by class “CPTS 322”

And I set the maximum distance to “2 miles”

Then only nearby “CPTS 322” events within 2 miles are shown.

#### 4. Traceability Matrix

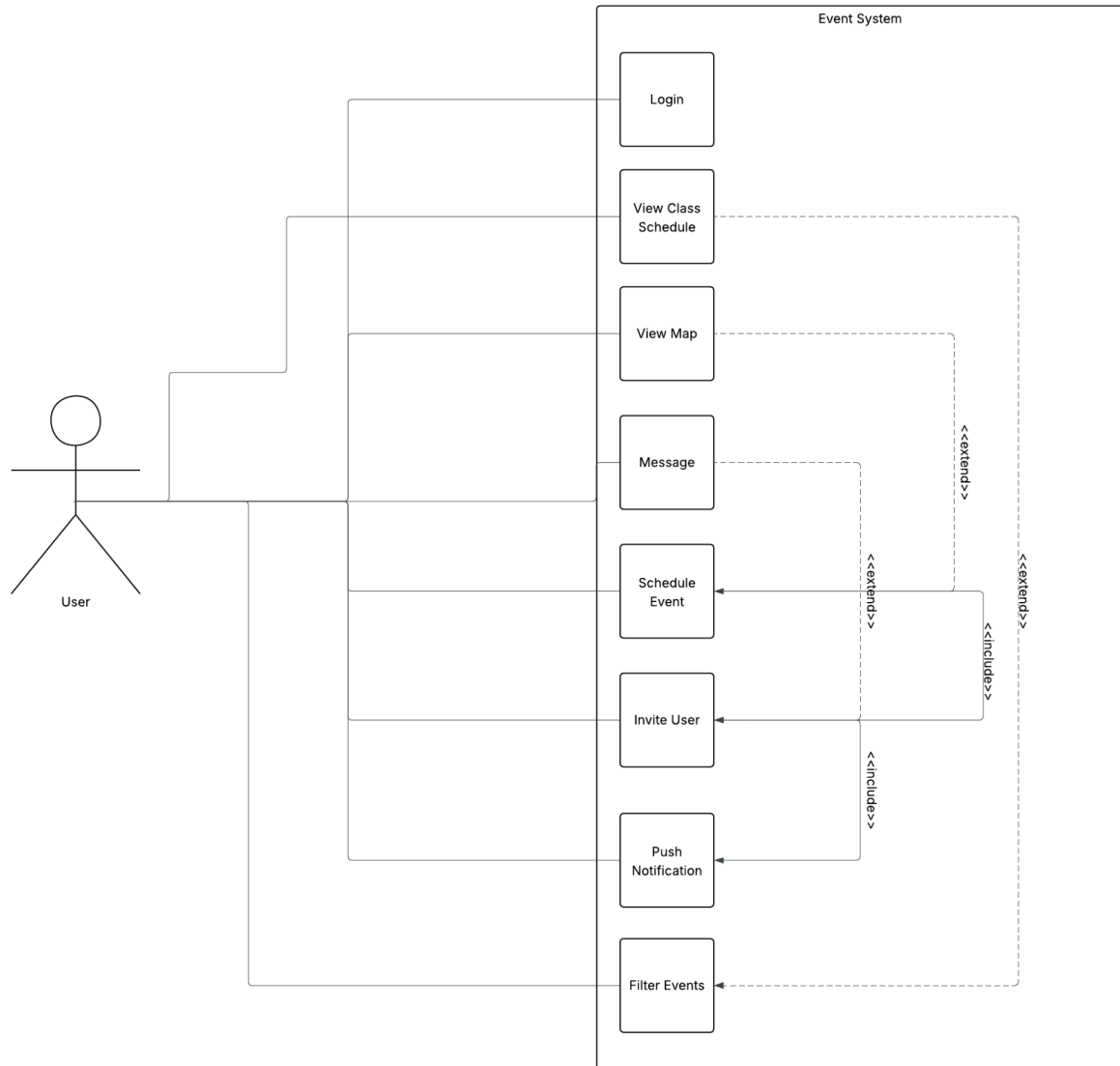
Requirement	Description	Type	User Story	Use Case
FR-01	The system shall allow users to register using a valid phone number.	Functional Requirement	As a registered user, I want to sign in with my phone number and a verification code so that I can securely access my account without a password.	Login
FR-02	The system shall send time-based notifications for upcoming events, such as reminders as an event approaches.	Functional Requirement	As a study group member, I want timely notifications so I can plan for upcoming events.	Push Notification
FR-03	The system shall allow users to schedule study events on or off campus, including reserving study spaces (when available) and notifying invitees.	Functional Requirement	As a group leader, I want to choose a convenient location so attendees can easily meet.	Schedule Event



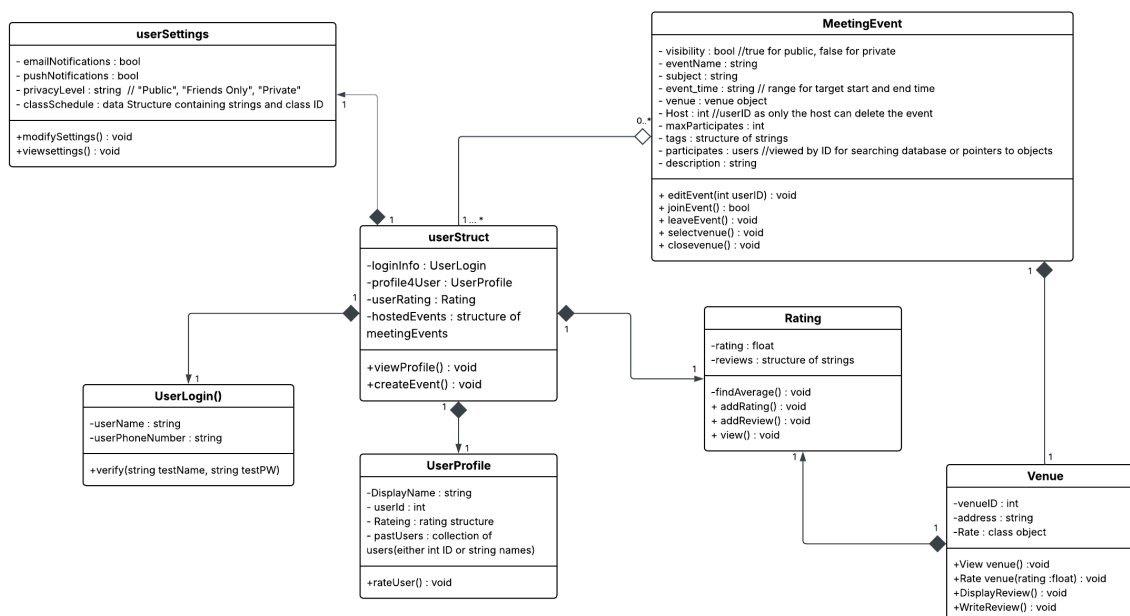
FR-04	The system shall allow a group leader to add or remove members from a study group.	Functional Requirement	As a group leader, I want to control invitations, so I can decide who is invited to the study session.	Invite User
FR-05	The system shall allow users to filter events by class.	Functional Requirement	As a student, I want to sort and filter my events feed so I can find the most relevant sessions.	Filter Events
FR-06	The system shall allow users to enter class schedules	Functional Requirement	As a student, I want to provide my class schedule so the app can recommend relevant events.	Enter Class Schedule
FR-07	The system shall provide a map view showing nearby results that match entered terms.	Functional Requirement	As a student user, I want to search for study events through a map view, so I can easily see events and where they are happening.	View Map
FR-08	The system shall provide chat threads for study groups and events, supporting both public and anonymous messages.	Functional Requirement	As a registered student user, I want an anonymous toggle in chats, so I can speak candidly when needed.	Message

## 5. UML Design Diagrams

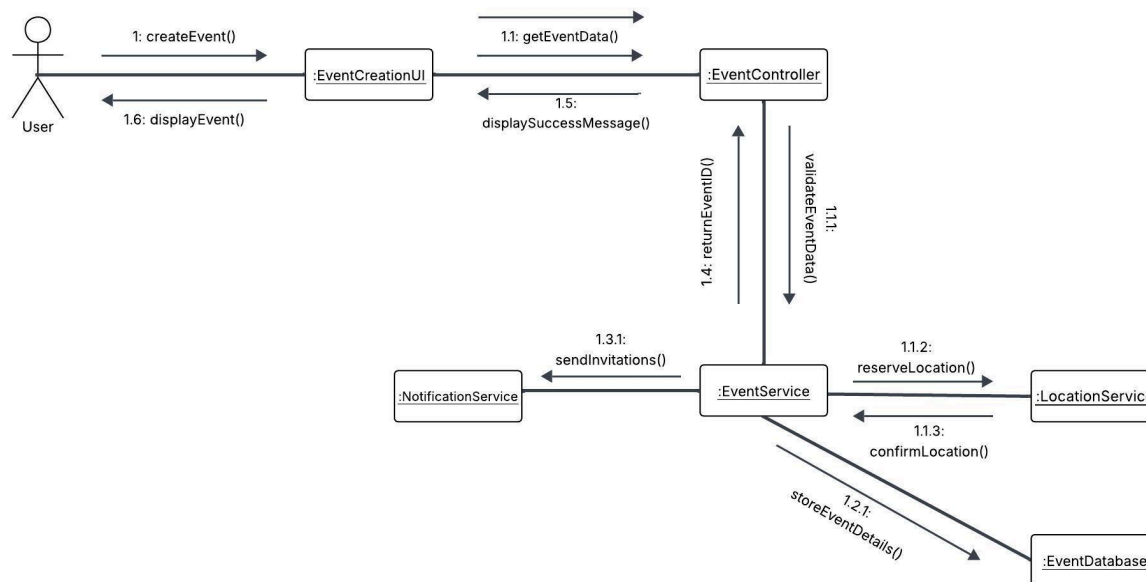
### 5.1 Use Case Diagram



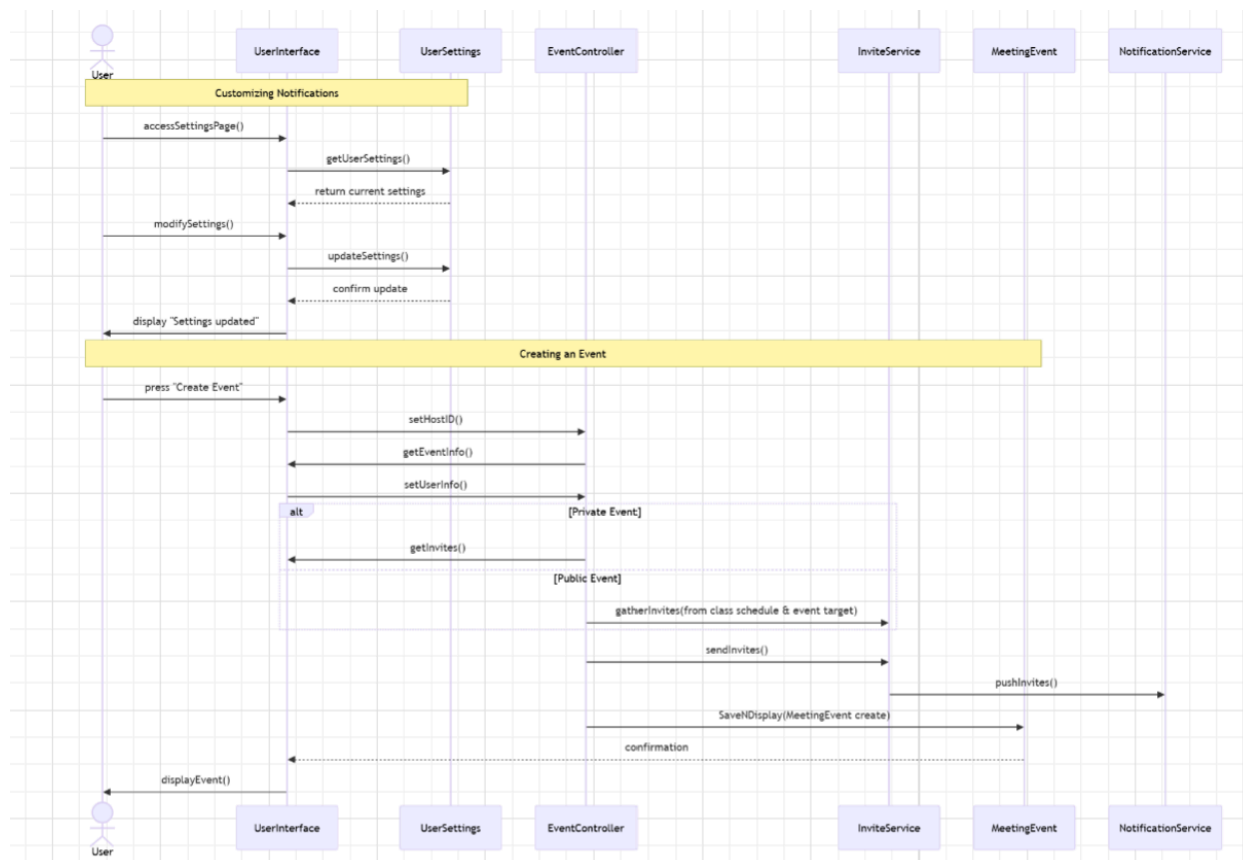
## 5.2 Class Diagram



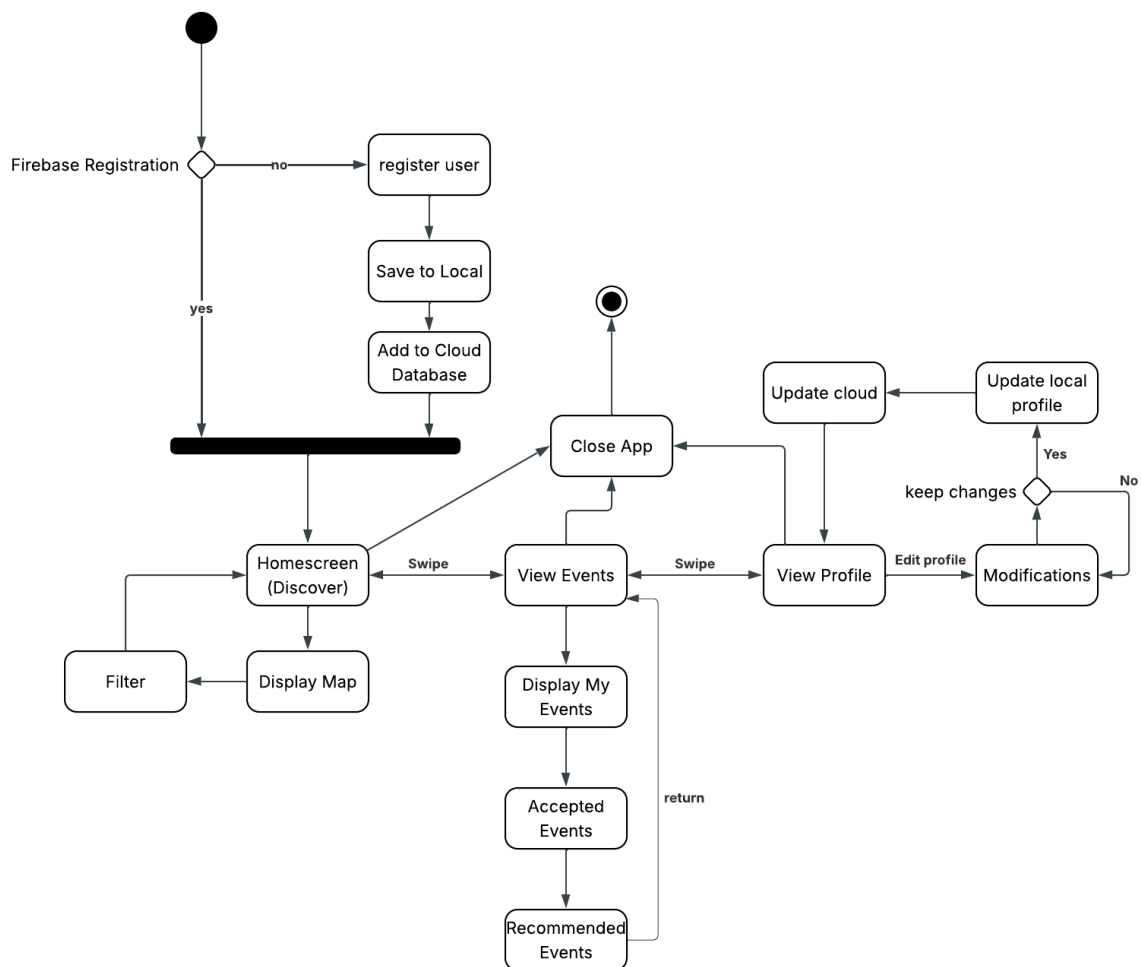
## 5.3 Collaboration Diagram



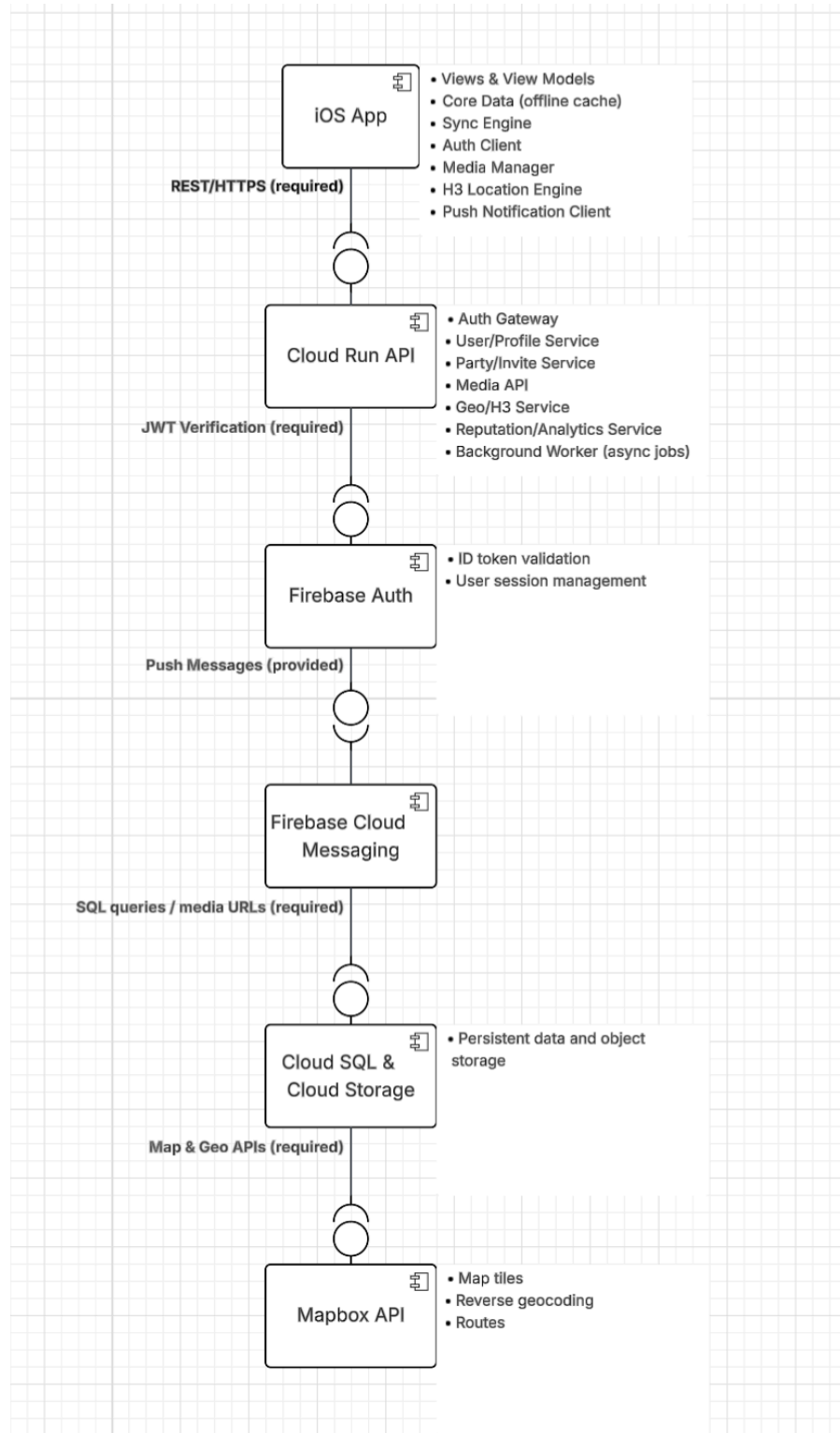
## 5.4 Sequence Diagram



## 5.5 Activity Diagram



## 5.6 Component Diagram



## 6. Reflection

During this phase of development, the team learned that AI-driven development can be a powerful yet limited tool. Its effectiveness depends heavily on clearly defining the problem, narrowing the scope, and focusing on one feature at a time. AI proved to be most useful for debugging isolated errors, producing step-by-step integration guidance, and assisting with backend feature implementation, such as event view models or user settings synchronization. However, it struggled to meet expectations when the work required a strong sense of design or a custom user experience. The team found that while AI accelerates repetitive or technical work, it lacks intuition for creative vision and emotional appeal—both of which are essential to our project's success. Despite initial hesitation, more team members are now becoming comfortable using AI to boost productivity, such as in research and task automation.

The primary challenges faced by the group stemmed from uneven technical experience among members. Some team members lacked access to MacBook hardware or familiarity with Swift and the Apple ecosystem, which limited their ability to contribute directly to iOS development. As the team recognized these limitations, it became clear that redistributing roles would be necessary to leverage individuals' strengths. Members who are concurrently learning Python are now planning to focus on the cloud backend, using technologies such as FastAPI, Docker, GitHub Actions, and Google Cloud Run. This alignment allows Alex to continue leading mobile development while ensuring that Tomas, Quinn, and Eni can meaningfully contribute to backend integration. The shift also broadens the team's technical diversity and positions the project for scalable, cross-platform growth.

Several key design decisions defined this phase and set the tone for development going forward. The team's central design philosophy emphasizes *powerful simplicity*—a focus on ease of use and clarity before complexity. The onboarding process was refined to reflect this philosophy, ensuring that users can begin interacting with the app immediately without cognitive overload. The group collectively decided that the core of the application should revolve around events as a unifying structure, whether for study sessions, parties, or personal scheduling. This led to a strategic pivot away from building an internal messaging system toward developing a dynamic, calendar-centric experience. In this design, even private user events share the same structure as public ones, simplifying data modeling and user flow. The app's differentiation lies in its emotional connection and intuitive usability: every interaction must feel natural, rewarding, and aligned with the Hooked model, a behavioral model for engagement. User interface, user experience, and user engagement design cannot be treated as afterthoughts but as core requirements of the system's architecture.

This stage has also been a period of consolidation and planning. The team now has a solid understanding of the system architecture, including how frontend and backend components will synchronize through cloud services. While some UI/UX details remain uncertain, the team's mission to make the app emotionally appealing and easy to use is now clearer than ever. Future development will focus on refining event creation: keeping the initial steps minimal with options, such as name, color, start time, and location, and progressively allowing for deeper customization. The understanding gained during this phase ensures that future coding efforts are grounded in deliberate simplicity rather than feature overload.

As the project advances, Alex will continue leading iOS development and integration and Quinn will contribute to UI/UX design and conceptual brainstorming. Tomas and Eni, with growing expertise in Python and cloud technologies, will collaborate on backend development and deployment automation. Collectively, this foundation transforms the project from a conceptual design into an actionable roadmap for implementation. The lessons learned around AI's role, balanced delegation, and disciplined simplicity will guide the team's approach to future phases, ensuring that Seshy continues to evolve into a product that is functional, accessible, and meaningful to its users.

## **7. References**

N/A