

6.0 Software Design Description (SDD) (Architecture Section)

Plot

Campus Events Application

Team Members: Julian, Chris, Donovan, Carson, Jeron

6.1 Introduction

This Software Design Description outlines how we plan to build Plot, an iOS app that gives college students an easier way to keep up with what's happening on campus. Plot puts events on a map, lets users RSVP directly from their phones, and keeps everything tied to their student accounts. This document explains how the app is organized behind the scenes: what parts exist, how they talk to each other, and what data they share. It acts as a shared reference for the entire team as development moves forward.

6.1.1 System Objectives

The objective of this application is to give college students a simple and reliable way to discover what is happening around campus. Events from various campus organizations are displayed together in one place and are easy to browse, whether the user prefers a map or a traditional list. A search bar allows students to quickly locate specific events or filter by interests such as sports, academic programs, or social groups. Most importantly, each event page provides key details including time, location, and hosting organization, along with an RSVP feature so students can plan ahead and know what to expect. Hosts are provided with a streamlined interface to create and update their own events. Plot ensures that students stay connected to campus life without needing to check multiple apps or social media pages.

6.1.2 Hardware, Software, and Human Interfaces

6.1.2.1 Hardware Interfaces

Plot runs on iPhones with iOS 17 or newer. The device must support touchscreen input, GPS/location services, and either Wi-Fi or cellular network access. All interaction uses the iPhone hardware only, with no external devices required.

6.1.2.2 Software Interfaces

Plot relies on several software systems that work together during operation.

iOS – Provides system services such as location permissions and network access. Supabase Swift SDK – Manages secure authentication and database read/write requests using HTTPS. Supabase PostgreSQL Database – Stores user accounts, events, RSVP data, and organization information with row-level security. Google Maps SDK for iOS – Displays the map, user location, and visual event markers. Xcode and SwiftUI – Used during development to build the interface and deploy the app. Future updates may introduce Apple Push Notification Service and the Uber API once needed.

6.1.2.3 Human Interfaces

Users interact through a touch-based interface written in SwiftUI. The main screens include Sign In/Sign Up, Event Feed, Map View, Event Detail View, and a Host Dashboard for creating and editing events. Navigation uses standard mobile gestures such as tapping and swiping, following familiar iOS design patterns.

6.2 Architectural Design

This section explains how the main parts of Plot fit together and how they support the required features of the application. Our architectural design focuses on separating responsibilities clearly so that each part of the system can be developed and modified without creating unnecessary dependencies.

Plot is organized into a few major functional groups. The user interface is built with SwiftUI and handles all interactions the student sees directly on their phone. A set of data models represent events, users, RSVPs, and organizations. A central service component, called SupabaseManager, is responsible for communicating with the backend for authentication and database operations. The map module displays event locations and receives coordinated event data from the backend.

By keeping these areas separate, each module can be implemented or updated without interfering with the others. This reduces coupling and increases clarity for every team member working on different parts of the app.

6.2.1 Major Software Components

This section describes the primary subsystems that support the functional requirements defined earlier in the Software Requirements Specification. Each component contributes directly to the main goals of the Plot application, including event discovery, user authentication, and host event management.

6.2.1.1 Authentication Module

Responsible for creating and validating user accounts through Supabase authentication. It ensures that users can securely sign up, log in, and maintain their session while using the app.

6.2.1.2 Event Module

Handles storing, retrieving, and updating event information such as title, time, and location. This module supports browsing, searching, and filtering functionality for general users.

6.2.1.3 Host Module

Provides tools for organization accounts to create and manage their hosted events. Only approved host users have access to these functions, which include updating and deleting event details.

6.2.1.4 SupabaseManager

Acts as the main communication layer between the SwiftUI interface and the Supabase backend. It performs all database operations, including reading events and writing user actions like RSVPs.

6.2.1.5 Map Integration Module

Displays event locations and the user's current position using the Google Maps SDK. It converts stored coordinates into map annotations for easier visual discovery.

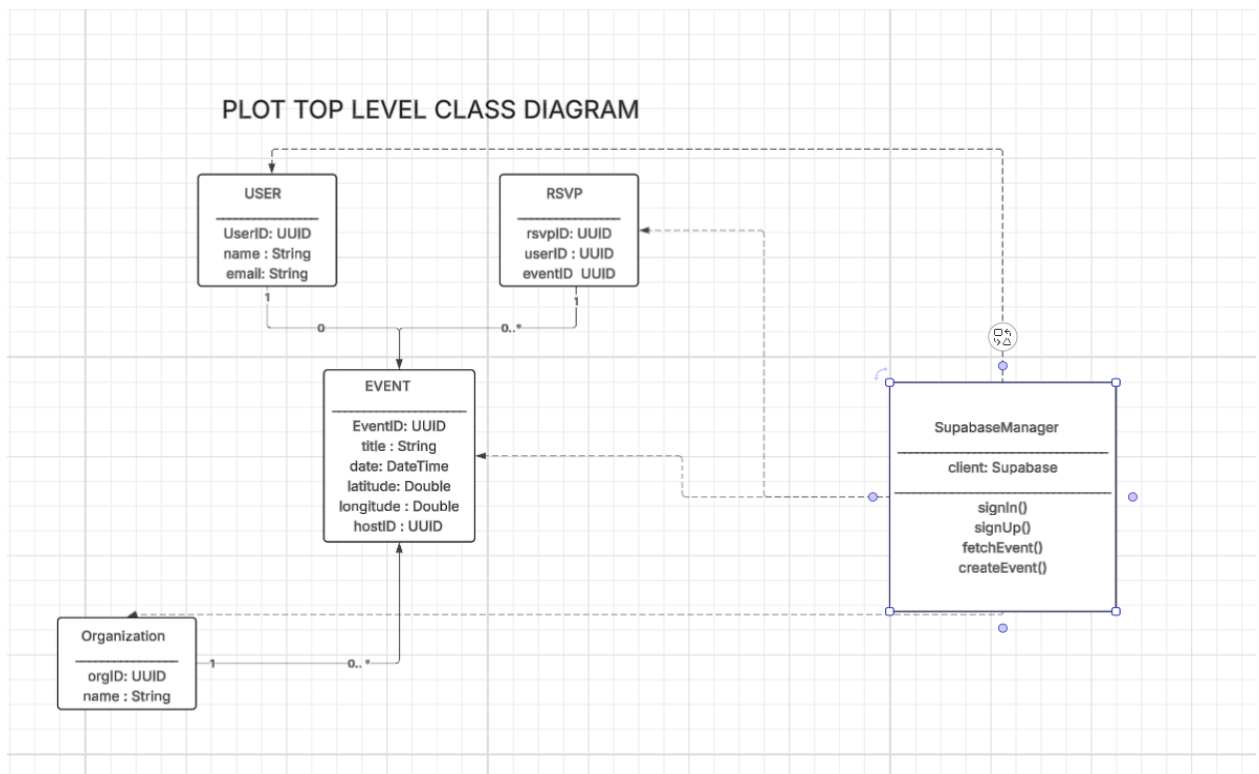
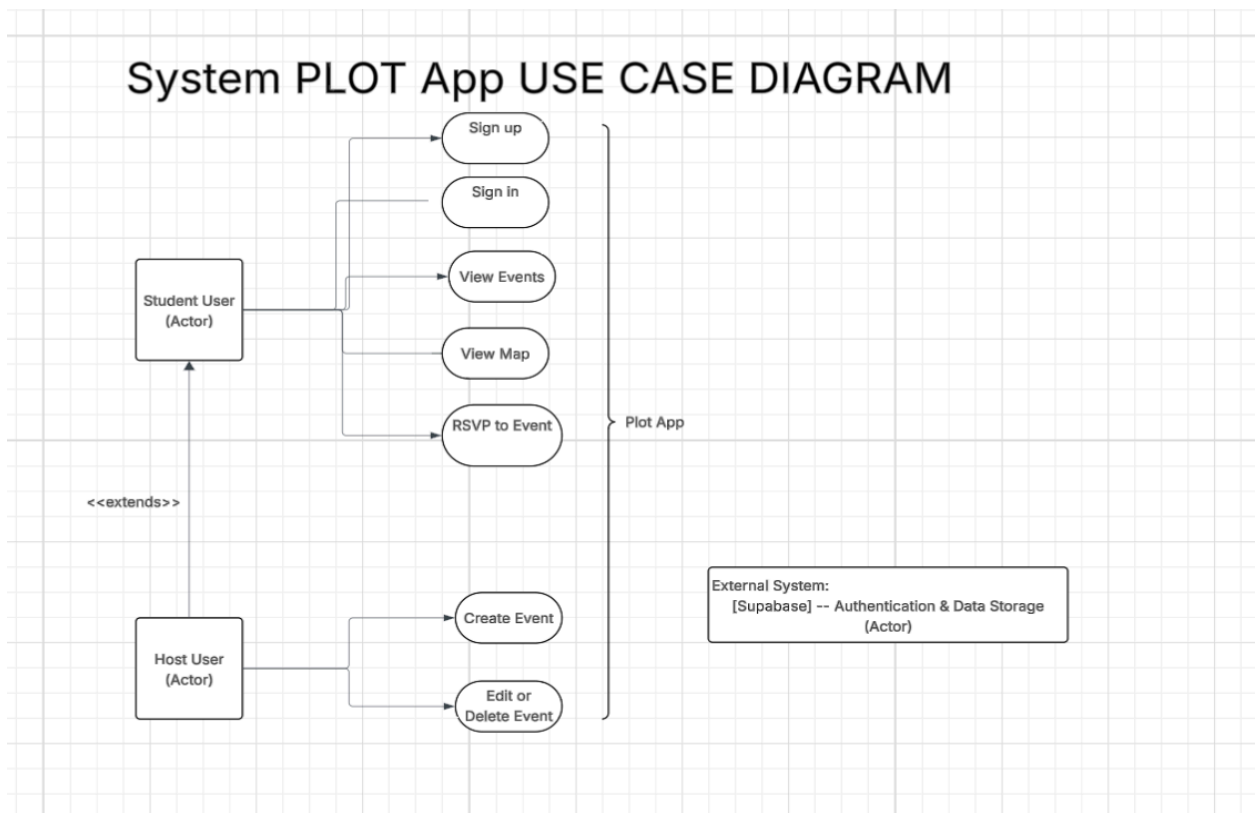
6.2.1.6 User Interface Components

SwiftUI screens such as the Event Feed, Map View, Event Details, and Authentication pages that make up the app's front end. These screens allow the user to navigate, view event information, and interact with all core features.

6.2.2 Major Software Interactions

This section explains how the main software modules communicate during normal app usage. The Authentication module sends login and account requests to Supabase to verify users and maintain sessions. All data operations flow through the SupabaseManager, which retrieves event information or submits updates on behalf of the user interface. The Event module passes geographic data to the Map Integration module to display event pins on the map. Host users interact with the database by accessing event creation and editing tools, which again call SupabaseManager to complete the requests. Future additions such as push notifications will allow APNS to notify users automatically when events they are attending are updated.

6.2.3 Architectural Design Diagrams



PLOT DEPLOYMENT DIAGRAM

