Requirements Specification Document
Plot
Campus Events Application

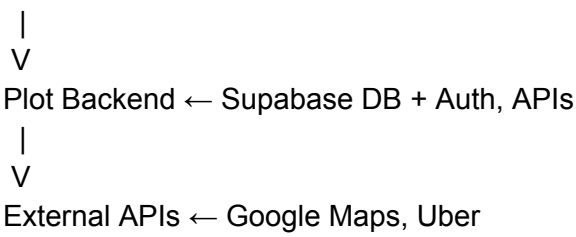
Sections

1. 5.1 Introduction
2. 5.2 CSCI Component Breakdown
3. 5.3 Functional Requirements by CSC
4. 5.4 Performance Requirements by CSC
5. 5.5 Project Environment Requirements

5.1 Introduction

Plot is a mobile application designed to help college students discover, share, and interact with campus events more effectively than current social media methods. Users will be able to view upcoming events, RSVP, see event details (time, date, location, capacity, cover/drink prices), and interact through features such as upvoting, downvoting, and filtering. The app will integrate Google Maps for location services and potentially Uber for ride-sharing information.

Diagram



5.2 CSCI Component Breakdown

CSCI: Plot Application

5.2.1 Client CSC (iOS App)

- 5.2.1.1 Main Screen CSU — Displays events feed and navigation.
- 5.2.1.2 Event Detail CSU — Shows full event info (time, date, place, capacity, host, etc.).
- 5.2.1.3 Map CSU — Built-in map showing event locations and nearby spots.
- 5.2.1.4 RSVP/Interaction CSU — Allows RSVP, upvote/downvote, and notifications.

5.2.2 Backend CSC (Supabase)

- 5.2.2.1 Authentication CSU — Handles login and user profiles.
- 5.2.2.2 Database CSU — Stores events, RSVPs, votes, organizations, and user data.
- 5.2.2.3 API CSU — Provides data access for events, filters, and notifications.

5.2.3 Integration CSC

- 5.2.3.1 Google Maps CSU — Displays event locations, routes, travel effort.
- 5.2.3.2 Uber API CSU — Provides ride price estimates (if implemented).

5.2.4 Notifications CSC

- 5.2.4.1 Alerts CSU — Push notifications for event reminders and followed orgs.

5.3 Functional Requirements by CSC

5.3.1 Client CSC (iOS App)

- 5.3.1.1 The client shall display a main feed of upcoming events.
- 5.3.1.2 The client shall allow filtering of events by type, date, and hosting organization.
- 5.3.1.3 The client shall display a detailed event page when an event is selected.
- 5.3.1.4 The client shall allow users to RSVP to events if the host allows it.
- 5.3.1.5 The client shall allow users to upvote or downvote events.
- 5.3.1.6 The client shall display a built-in map showing event locations.
- 5.3.1.7 The client shall allow verified “trusted hosts” to be highlighted in search and feeds.
- 5.3.1.8 The client shall display event capacity and entry requirements.

5.3.2 Backend CSC (Supabase)

- 5.3.2.1 The backend shall store all event data including name, description, date, time, location, capacity, and host.
- 5.3.2.2 The backend shall store RSVP information and associate it with user profiles.
- 5.3.2.3 The backend shall store upvote/downvote counts per event.
- 5.3.2.4 The backend shall authenticate users via secure login.
- 5.3.2.5 The backend shall support filtering queries for events.
- 5.3.2.6 The backend shall provide APIs for event listing, event creation, RSVP updates, and voting.

5.3.3 Integration CSC

- 5.3.3.1 The system shall use the Google Maps API to display event locations.
- 5.3.3.2 The system shall provide map-based navigation to event venues.
- 5.3.3.3 The system shall provide Uber ride estimates if the Uber API is integrated.

5.3.4 Notifications CSC

- 5.3.4.1 The system shall send notifications for events that a user RSVPs to.
- 5.3.4.2 The system shall allow users to follow organizations and receive alerts when they post events.
- 5.3.4.3 The system shall notify users when event details change (time, location, capacity).

5.4 Performance Requirements by CSC

- 5.4.1 The system shall return event feed results within 3 seconds of query.
- 5.4.2 The map subsystem shall render event locations within 2 seconds after data load.
- 5.4.3 The backend shall support at least 100 concurrent users without performance degradation.
- 5.4.4 The system shall process an RSVP or vote action within 1 second of user input.
- 5.4.5 Notifications shall be delivered within 30 seconds of triggering conditions.

`

5.5 Project Environment Requirements

5.5.1 Development Environment Requirements

- Programming language: Swift (iOS front end).
- Backend: Supabase (PostgreSQL DB + Auth + Functions).
- APIs: Google Maps API, Uber API (optional).
- Development tools: Xcode, GitHub repo, Postman for API testing.

5.5.2 Execution Environment Requirements

- Hardware: iOS devices (iPhone/iPad) with iOS 15 or newer.
- Connectivity: Internet connection required for event data, maps, and RSVP sync.
- Storage: At least 50MB local storage for app installation.

Diagrams

5.2.1 Client CSC (iOS App)

- 5.2.1.1 Main Screen CSU — Displays events feed and navigation.
- 5.2.1.2 Event Detail CSU — Shows full event info (time, date, place, capacity, host, etc.).
- 5.2.1.3 Map CSU — Built-in map showing event locations and nearby spots.
- 5.2.1.4 RSVP/Interaction CSU — Allows RSVP, upvote/downvote, and notifications.