**Software Development Plan**

**Plot**

**Campus Events Application**

**4.1 Plan Introduction**

This Software Development Plan provides the details of the development of the Plot mobile application, which helps college students discover, share, and interact with campus events more effectively than current social media platforms.

**4.1.1 Project Deliverables**

* **Project Proposal** – *Due: Weeks 2-3*
  + A document outlining the Plot app concept as an iOS app built with Swift, backed by Supabase, featuring map functionalities via MapKit, and ride requests/integration through the Uber API.
    - State the primary use-case (e.g., users plotting locations or stories, viewing them on a map, and potentially hailing rides), the rationale for these technologies, feature goals, known competitors, and anticipated technical challenges such as integrating MapKit and handling OAuth for Uber.
* **Requirements Specification Document** – *Due: Week 6*
  + Detailed functional and performance requirements for the Plot app.
    - Functional Requirements:
      * User authentication with Supabase.
      * Map display and interaction using MapKit (pinning plots, geolocation, route display).
      * Uber API integration for requesting rides to/from plotted locations.
      * User profile and CRUD operations for plots.
    - Performance/Non-Functional Requirements:
      * Fast, smooth map rendering on iOS devices.
      * Robust handling of real-time updates via Supabase.
      * Secure management of Uber API credentials and OAuth flow.
      * Compliance with iOS UI/UX guidelines.
* **Software Design Description Document [Architecture Section]**– *Due: Week 10*
  + Describe the high-level system layout:
    - Frontend: SwiftUI-based iOS app.
    - Backend: Supabase (PostgreSQL DB, Auth, Realtime).
    - Maps Module: MapKit framework for displaying and annotating maps, handling user location and navigation.
    - Uber Integration: Uber Ride Requests via SDK/API for booking, shown contextually within plots that support ride functionality.
    - Data Flow: Diagrams showing data moving between SwiftUI views, Supabase, MapKit, and Uber; address asynchronous networking and error handling strategies.
* **Software Design Description Document [Detailed Section]** – Due: *Week 12*
  + Breakdown of:
    - Data Models: Swift structs/classes for plots, users, ride requests; Supabase table mappings.
    - MapKit Integration: Code/data flow for adding/removing map pins, geocoding, overlays, and directions.
    - Uber API Integration: OAuth login flow, handling callbacks in AppDelegate, calling ride request endpoints, managing ride statuses/updates.
    - UI/UX Details: Wireframes for map screens, plot creation, and Uber ride screens; navigation logic across different features.
    - Security: Details of credential management for Supabase and Uber, secure storage, permission flow for location.
* **User's Manual Final Updates**– *Due: Week 12*
  + Include step-by-step instructions with screenshots for:
    - Installing the app
    - Creating/logging into an account (Supabase)
    - Pinning or plotting a location/story using MapKit
    - Viewing location details and requesting an Uber via in-app integration
    - Handling errors and privacy settings (location permission prompts etc.)
    - Managing user profiles and settings.
* **ALPHA/BETA/Critical Design Review** – Due: *Weeks 13 — 14*
  + Showcase a working alpha/beta of Plot running on an iOS device or simulator:
    - Users can sign up, plot points, view them on the map, and initiate an Uber ride.
    - Demonstrate flow from plot creation → map view → Uber ride request.
    - Discuss architectural decisions (Swift + Supabase, MapKit, Uber API) and how integration issues were handled.
    - Gather peer/instructor feedback, especially for usability and API failures.
* **FINAL PROJECT PRESENTATIONS** – *Due: Week 15*
  + Demonstrate all features, especially the interaction between MapKit and Uber ride requests, and discuss technical and design challenges (e.g., handling map errors, Uber API edge cases). Present lessons learned and future improvements for scaling or new features.
* **FINAL PRODUCT DELIVERY** – *Due: Week 16*
  + Deliver source code (documenting use of MapKit and Uber API), Supabase DB schema, instructions for deploying/running the app, all manuals, and test results. Package everything needed for instructors to review and run the app, ideally including a TestFlight beta invite or similar.

**4.2 Project Resources**

**4.2.1 Hardware Resources**

* MacBook to develop using XCode
* Wi-Fi router / Internet connection – Required for API testing and backend connectivity

**4.2.2 Software Resources**

* Xcode – iOS IDE
* Swift – Programming language for iOS app
* Supabase – Backend services
* MapKit – Event location mapping
* Uber API (optional) – Ride estimates
* GitHub – Version control

**4.3 Project Organization**

1. **Julian -** Project Lead and Full Stack Developer : Full Stack Development, Planning the work, managing the Jira tracker

2. **Chris -** Frontend Lead: Building the main structure and UI of the Swift iOS app, setting up a scalable code repository.

3. **Donovan -** Backend Lead: Designing the Supabase database, writing the API code for features like RSVP and voting, and securing all data with RLS.

4. **Carson** - Integrations Specialist: Setting up all external tools: push notifications (APNS), Google Maps, implementing notification triggers, and exploring tools like Uber.

5. **Jeron** - Full-Stack Developer: Implementing Authentication workflows and other assigned features across both the Swift app and the Supabase backend, and helping fix bugs, and reviewing code before merges to the main branch.

**4.4 Project Schedule**

**4.4.1 GANTT Chart**

| Task Category Group | Start | Finish | Duration | Assigned |  |
| --- | --- | --- | --- | --- | --- |
| [AUTH] Security & Login Flow | 09/22/2025 | 10/05/2025 | 14 days | Jeron |  |
| [DB] Core Schema & Read API | 09/22/2025 | 10/05/2025 | 14 days | Donovan |  |
| [UI] Core Screens (Feed & Detail) | 09/22/2025 | 10/05/2025 | 14 days | Chris |  |
| [Integrations] Google Maps Setup | 09/22/2025 | 09/28/2025 | 7 days | Carson |  |
| [DB] Interaction APIs (RSVP/Voting) | 10/06/2025 | 10/19/2025 | 14 days | Donovan |  |
| [UI] Interaction/Filtering Logic | 10/06/2025 | 10/19/2025 | 14 days | Jeron |  |
| [Integrations] Map Navigation | 10/06/2025 | 10/12/2025 | 7 days | Carson |  |
| [DB] Host APIs (Create/Follow Org) | 10/20/2025 | 11/02/2025 | 14 days | Julian |  |
| [UI] Host Features (Creation/Highlight) | 10/20/2025 | 11/02/2025 | 14 days | Chris |  |
| [Integrations] APNS Setup & RSVP Trigger | 10/20/2025 | 11/02/2025 | 14 days | Carson |  |
| [Integrations] Remaining Notification Triggers | 11/03/2025 | 11/16/2025 | 14 days | Julian |  |
| [Integrations] Uber API Research Spike | 11/03/2025 | 11/09/2025 | 7 days | Donovan |  |
| [UI] Final UI/UX Polish & Bug Fixes | 11/03/2025 | 11/16/2025 | 14 days | Chris |  |
| [Testing] Final Bug Fixes | 11/17/2025 | 11/30/2025 | 14 days | Jeron |  |
| [Integrations] Uber Implementation (If Prioritized) | 11/24/2025 | 11/30/2025 | 7 days | Julian |  |

**4.4.2 Task/Resource Table**

Task Table located at: [**https://jeronalford.atlassian.net/jira/software/c/projects/PLOT/list**](https://jeronalford.atlassian.net/jira/software/c/projects/PLOT/list)