

# Whats-Up MVP Specifications



By: Bennett Free

12/28/2025

# Table of Contents

---

Overview .....2

MVP Scope .....3

System Components .....3

Providers and Configuration (Global Constraints) .....4

Phase 1: Navigation Semantics and De Instagram Profile and Home .....4

Phase 2: Map Backed Events and Places plus Explore (Ticketmaster and Google Places) .....5

Phase 3: Lightweight Reviews with Photos (Persistent and Public).....5

Phase 4: OpenAI Powered “I Want To Do This” Planner (Real AI).....6

Phase 5: Saved and RSVP (“Plans”) - Persistent State .....6

Phase 6: Social Graph (Friends and Followers) - Core MVP .....6

Phase 7: Polish, Testing, and Launch Readiness.....7

Future Roadmap - High-Value Differentiating Features (Post-MVP).....8

Proposal: How I’d Like to Structure My Involvement .....8

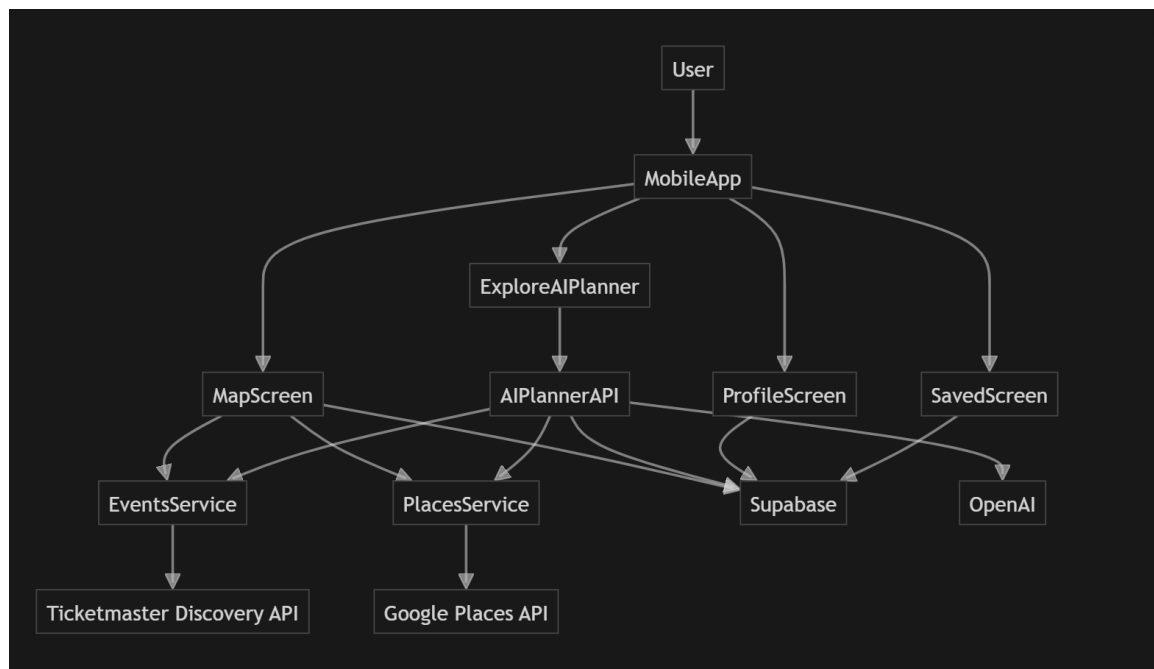
# Whats-Up MVP Refactor & AI Planner Plan

## Overview

Transform the existing Instagram-like social feed into a focused IRL events and places app with the following:

- A fully functional map powered by Ticketmaster Discovery API (events) and Google Places API (venues)
- A simple dating app style profile opposed to classic Instagram grid
- A lightweight review system with photos
- Reviews that are fully public and appear as posts on a user's profile
- An AI-assisted planner that adds human-style reasoning on top of real events and venue data
- A basic social graph so users can follow each other and see relevant context
- An Explore page to search and view nearby events and venues by current location or zip code

All work is organized into sequential phases with strict Definitions of Done (DoD). Do not start a later phase until the previous phase's DoD is fully met.



## MVP Scope

The first public MVP must ship with a cohesive, fully functional experience with no obvious stubs or disappearing state.

### Included in MVP

- Phase 1 and Phase 2 complete (navigation, de-Instagram profile, map with live Ticketmaster and Google Places data, Explore page for location or zip based search, with graceful mock fallback)
- Phase 3 complete with persistent reviews that are fully public and appear as posts on a user's profile
- Phase 4 complete with a real OpenAI backed AI planner over live Ticketmaster and Google Places results, including reasons per recommendation
- Phase 5 complete with basic saved or plan state (saved places and events and Going or RSVP state) that persists across app restarts (AsyncStorage or Supabase)
- Phase 6 complete with a core social graph (follow and unfollow, follower and following counts, and basic social context in profiles and plan views)

### Post-MVP (Explicitly Out of Scope for first release)

- Push or email notifications and complex RSVP workflows
- Deep Supabase integration beyond review, saved, RSVP, and social persistence
- Cross device sync and sharing flows using Supabase
- Advanced social features like groups, DMs, stories, or algorithmic social feeds

Development scaffolding is allowed during development but must not remain in the shipped MVP build.

## System Components

- Mobile App
- Map Screen
- Explore Screen (search and list view around a location or zip)
- Explore AI Planner
- Profile Screen
- Saved Screen
- Events Service
- Places Service
- Social Graph Service
- AI Planner API
- Supabase (post-MVP enhancement and multi device sync)
- Ticketmaster Discovery API
- Google Places API
- OpenAI

## Providers and Configuration (Global Constraints)

### Map Layer

- Use react native maps
- Keep existing geolocation and recenter behavior

### Ticketmaster Discovery API - Events

- Primary events provider for MVP
- Query by lat, lng, radius and optional date
- Cap results to roughly 40 to 50 per query
- Environment variable: TICKETMASTER\_API\_KEY

### Google Places API - Venues

- Use Nearby or Text Search for venues
- Feed into internal Place type
- Environment variables: GOOGLE\_MAPS\_API\_KEY, GOOGLE\_PLACES\_API\_KEY

### OpenAI - AI Planner

- Used only for planner
- Must run server side
- Environment variable: OPENAI\_API\_KEY

## Phase 1: Navigation Semantics and De Instagram Profile and Home

### Goal

Confirm navigation semantics and remove Instagram-like patterns while keeping the app simple and performance focused.

### Definition of Done - Phase 1

- Profile tab uses clean “You” card layout (avatar, name, handle, bio, stats, interests)
- No Instagram style elements remain (no grids, story circles, etc.)
- Home feed cards use simple CTAs like “View details” and “Save”
- AI banner is purely visual (no AI calls yet)
- Top category row is compact and not vertically dominant

Note: Social stats may be visible, but follow logic is implemented in Phase 6.

## **Phase 2: Map Backed Events and Places plus Explore (Ticketmaster and Google Places)**

### **Goal**

Power the Map screen and Explore page with real data via a clean service layer while keeping safe mock fallback.

### **Definition of Done - Phase 2**

- Opening Map fetches live Ticketmaster and Google Places data
- Events and venues render together smoothly on the map
- Debounced refetch on pan and zoom
- Graceful fallback to mock data on failure
- Business detail modal shows key data and wired local actions for Save, Join, and Review
- Explore screen allows search by current location or entered city or zip, uses the same data layer as Map, includes a list view with ability to open detail modals, and uses the same debounce and fallback behavior as Map

## **Phase 3: Lightweight Reviews with Photos (Persistent and Public)**

### **Goal**

Users can leave reviews with ratings, text, and photos that persist across app restarts and are fully public.

### **Definition of Done - Phase 3**

- User can leave a review with rating, text, and photos
- Reviews appear immediately on the place or event
- Reviews are fully public and appear as posts on the reviewer's profile
- Reviews persist after app restart (AsyncStorage or Supabase)
- Profile shows recent reviews as public posts
- No "coming soon" review actions remain

## Phase 4: OpenAI Powered “I Want To Do This” Planner (Real AI)

### Goal

Build a real planner that recommends live events and places and explains the reasoning in plain language.

### Definition of Done - Phase 4

- Planner runs via a backend endpoint
- Backend retrieves real Ticketmaster and Google Places candidates based on user context (location, time, preferences, etc.)
- Backend applies baseline filtering and ranking (distance, relevance, timing, rating, price, deduping, etc.)
- OpenAI is used only to:
  - Refine final ranking
  - Generate natural-language reasons for each recommendation
  - Returned results must always map to real events or places with real metadata
  - No stub or fake AI in the shipped MVP

## Phase 5: Saved and RSVP (“Plans”) - Persistent State

### Goal

Allow users to save items and mark events as Going, with persistence across app restarts.

### Definition of Done - Phase 5

- Saved places and events persist across app restarts
- RSVP or Going state persists across app restarts
- Saved tab lists upcoming plans and saved items sensibly
- Profile stats include only: Plans Saved, Events Joined, Spots Reviewed
- Data structure is compatible with later social surfacing

## Phase 6: Social Graph (Friends and Followers) - Core MVP

### Goal

Introduce a basic social graph and simple messaging so users can coordinate plans directly, without turning the app into a traditional social feed product.

### Definition of Done - Phase 6

- Follow System
  - Users can friend and unfriend other users
  - Profiles show friend counts
  - Friend relationships persist across app restarts

- Profiles and Social Context
  - Public reviews appear as posts on user profiles
  - Saved or plans screens can show simple “friends going” or “friends saved this” indicators where appropriate
- User Search
  - Users can search for people by name or username
  - Tapping a search result opens that user’s profile
  - Search results only include public profiles
- Direct Messaging (1:1)
  - Users can send and receive private messages
  - Conversations persist across app restarts
- Group Messaging (Lightweight)
  - Users can create group chats with multiple participants
  - Group chats can be optionally tied to events/venues/plans
  - Messages persist across app restarts
- Intentional Constraints (MVP Boundaries)
  - Messaging is text-only (no media, reactions, voice, etc)
  - No public social feed
  - No story system
  - No broadcast or creator tools

## Phase 7: Polish, Testing, and Launch Readiness

### Scope

- Ensure UI is clean and consistent across Map, Explore, Profile, Saved, and Planner
- Remove any Instagram-like patterns that crept back in
- Add appropriate loading, empty, and fallback states
- Perform end to end manual testing across new user onboarding, Explore and Map search, Save, RSVP, review, planner flows, and follow and unfollow flows



## Future Roadmap - High-Value Differentiating Features (Post-MVP)

These features are intentionally not part of the MVP. Each should ship complete with no half states. They are focused on real world connection, trust, and utility.

- Group Meetups and Hosted Plans: Structured user hosted gatherings with RSVPs, invite links, host controls, and safety tools.
- Smart Social Layer ("People Like Me"): Optional privacy-first taste clustering that improves recommendations without influencer dynamics.
- Open Tables and Empty Seats: Allow users to open extra spots on real plans so others can join with host approval.
- Trusted Host Program: Lightweight tools and credibility for recurring community organizers without turning them into creators.
- Dynamic Itinerary Builder: AI-assisted but user-directed itineraries with budgets, timing, and bookings.
- Night Mode Safety Companion: Optional trust features like live route share and safe arrival check-ins.
- Local Culture Discovery: Community and host curated guides for authentic local experiences.

## Proposal: How I'd Like to Structure My Involvement

If I join this as the person building the full MVP, owning the product and tech, and continuing to maintain and improve it after launch, then I'm not just a developer on a project - I'm stepping in as a true partner in the business.

To make that fair and aligned for everyone, here's what I'd need:

- 33.33% ownership in the company once it's formed
- A simple written agreement confirming that equity, even before the company is officially created
- Standard vesting so we're all committing long-term
- An understanding that I'll continue supporting and maintaining the product, not just building v1

This reflects the reality that the business only exists if the product gets built, shipped, and improved - and that's the responsibility I'd be taking on. I want this to feel fair on both sides and set us up as real partners, not a client and developer.