



DANGERLESS

An application for providing travelers with real-time
neighborhood safety ratings and local safety news alerts.

Background



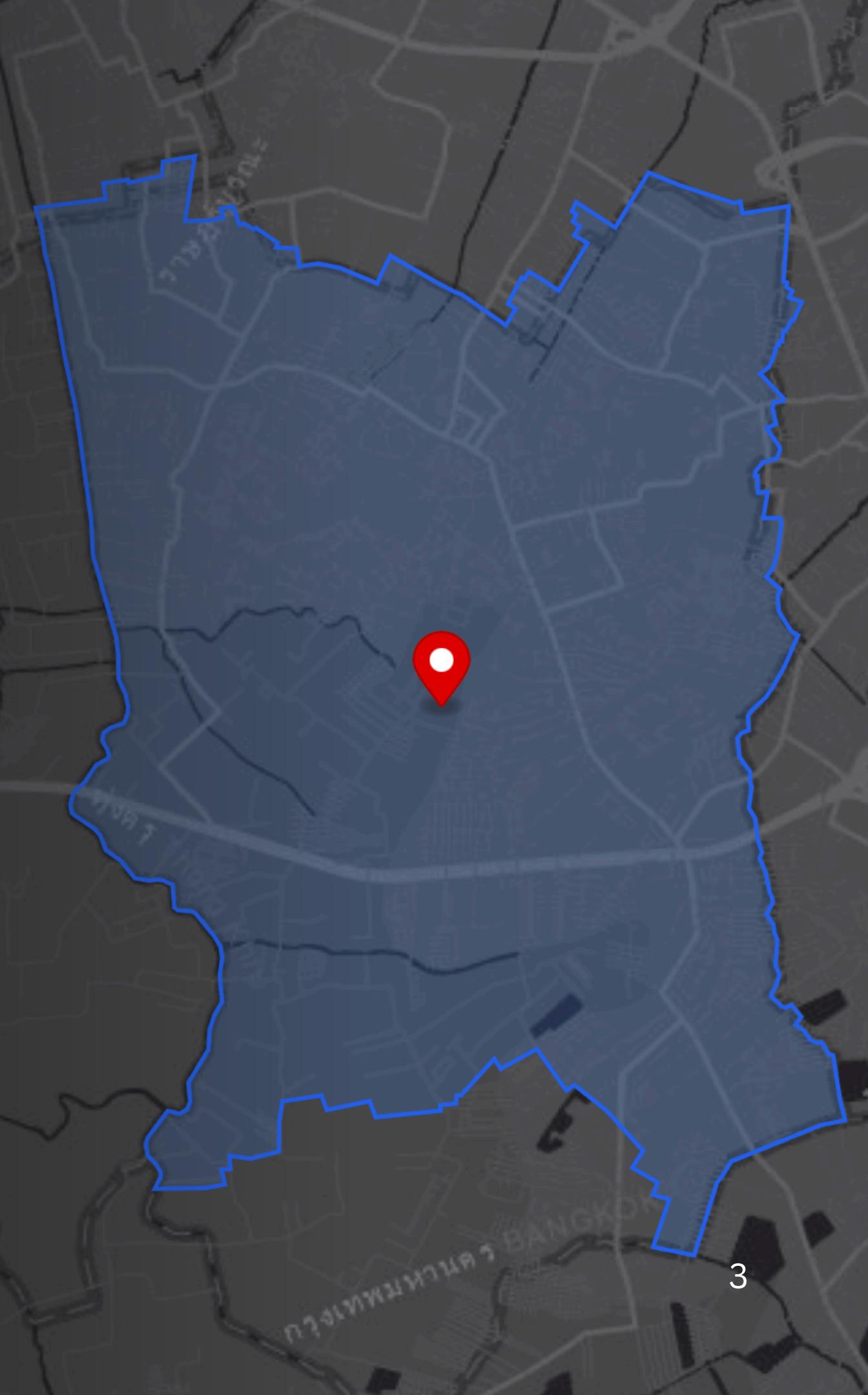
Where to land ?



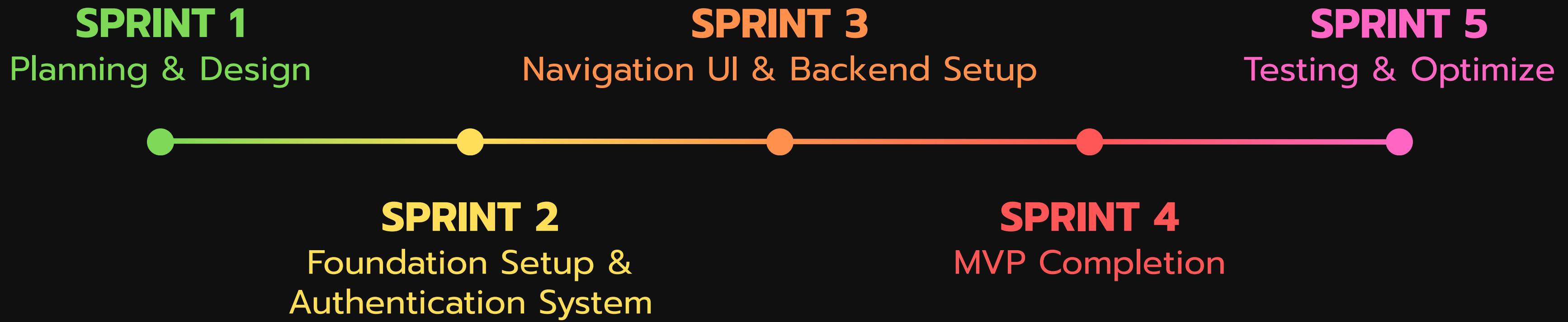
Overview

Dangerless is a location-based platform that helps users check the risk levels of different districts before traveling.

To reduce safety uncertainty, the system provides real-time danger updates and verifies user-submitted incident reports before publishing them. The main objective is to improve safety awareness and help users make better travel decisions through a reliable, easy-to-use interface.



Project Timeline



SPRINT 1 - SRS Document

Defined the system scope, key features, user stories, and technical requirements to align the team on the project direction.

Software Requirement Specification

Application Name	Dangerless - An application for providing travelers with real-time neighborhood safety ratings and local safety news alerts.
Member	<ol style="list-style-type: none">1. Parunchai Kaewkhampa2. Punnawat Mungkalarungsi3. Peerawish Channukool4. Pakkawat Kaolim5. Veerathach Rattanarojt6. Kamil Pattanasakulloy7. Raphael Canu8. Gabin Dalibard

Document Version:

Date	Author	Details	Version
------	--------	---------	---------

SPRINT 1 - Core Requirements & User Flows

Interactive Safety Map

- Show users a clear, real-time map of district safety levels.

Reliable Safety Information

- Provide trustworthy danger updates and verified incident reports.

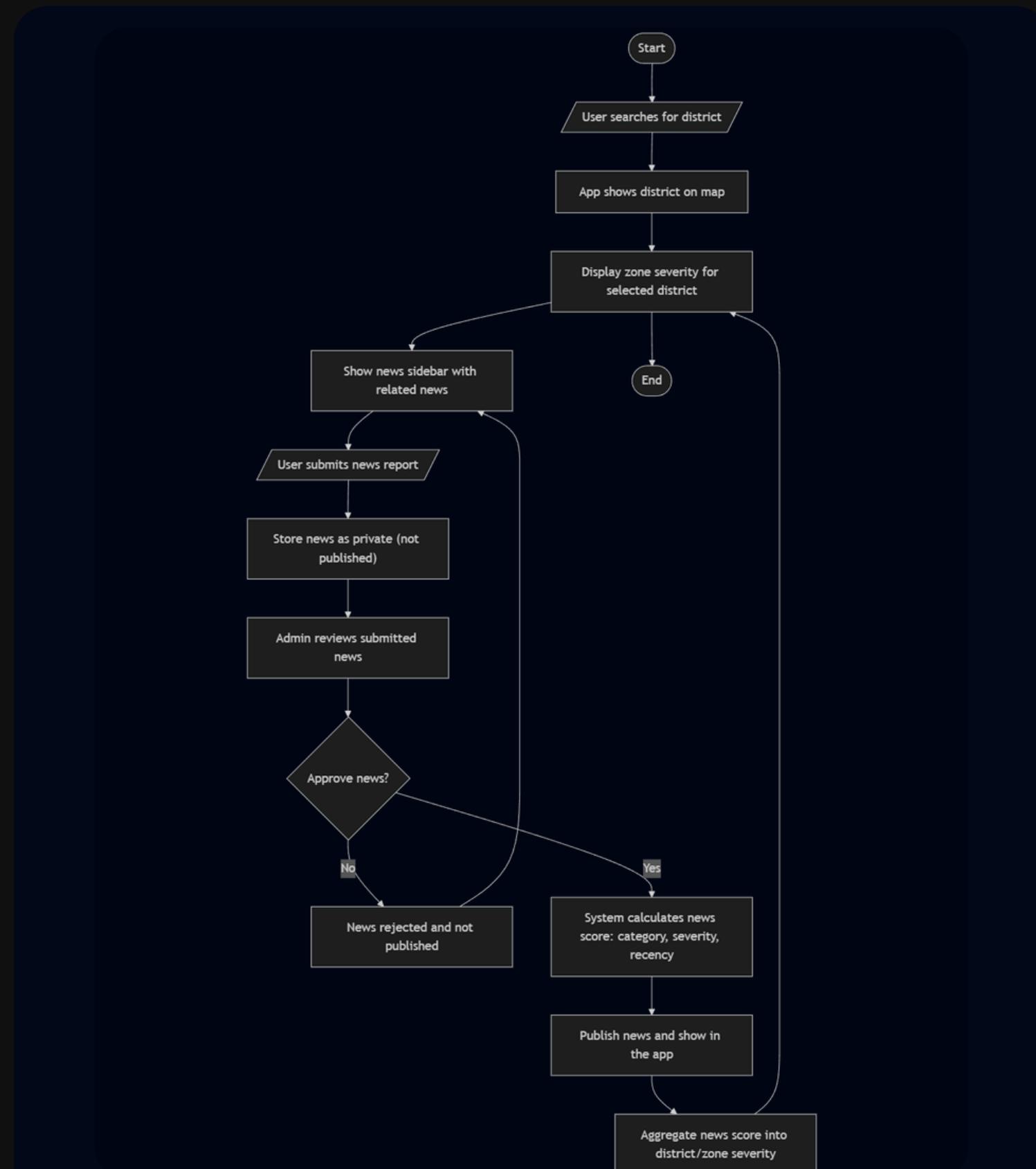
Risk Awareness Tools

- Let users view district profiles, recent incidents, and risk categories.

Smart Filtering & Navigation

- Help users find relevant safety information quickly (type, severity, time).

SPRINT 1 - Core Requirements & User Flows



SPRINT 1 - Wireframes

Created basic layout wireframes for core pages
(Map, Posting, News, History, Scouting)

The image displays three wireframes of a mobile application interface for 'SafeMap'.

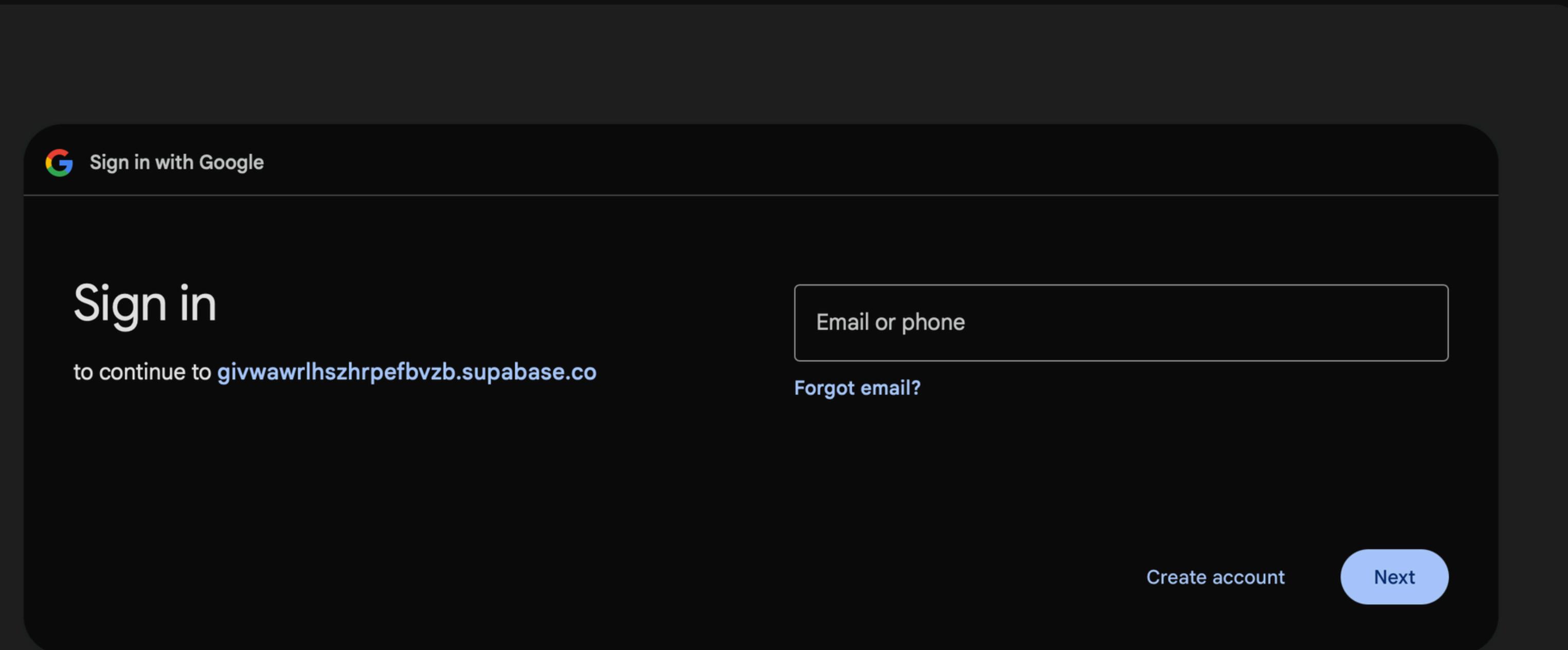
Left Wireframe: A map view showing a large orange polygon representing a 'Safe Zone'. The map includes street names in Thai, such as 'ถนนสุขุมวิท' (Sukhumvit Road) and 'ถนนเพชรบุรี' (Phetchaburi Road). On the left, there's a sidebar with navigation tabs: 'Scouting', 'News', and 'History'. Below the tabs is a search bar for announcements. Underneath are three news cards, each with a title, timestamp, location, and a small image. The first news card is titled 'News 1 Critical' and describes a robbery at a restaurant.

Middle Wireframe: A modal window titled 'Event' for posting new events. It contains fields for 'Title*', 'Link Source*', 'Location*', 'Image', 'Description', and 'Date*'. A 'Post' button is at the bottom. The background shows a map of a city area with roads labeled '3418', '347', '346', '351', and '3233'.

Right Wireframe: A sidebar titled 'News & Events' with a 'Post' button. It lists several items: a 'Road Closure Alert' from the 'City Traffic Dept' about a main street closure due to construction, a 'Community Safety Meeting' notice from the 'Community Board', a 'Weather Alert' from the 'Weather Service' about heavy rain, and a 'News Image' section with a thumbnail of a flooded street.

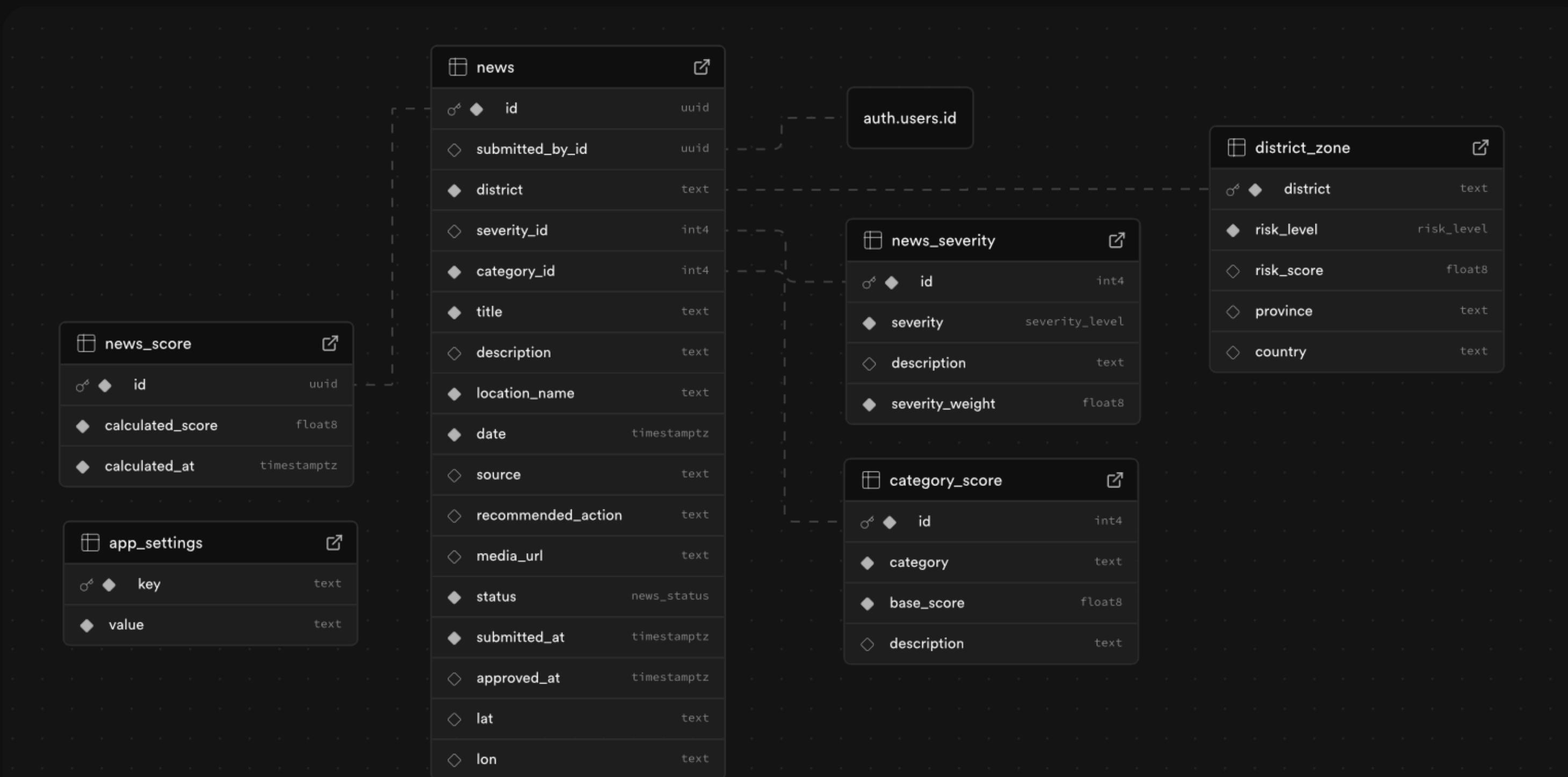
SPRINT 2 - Authentication

Implemented Google OAuth via Supabase Auth
to handle secure login and access control.



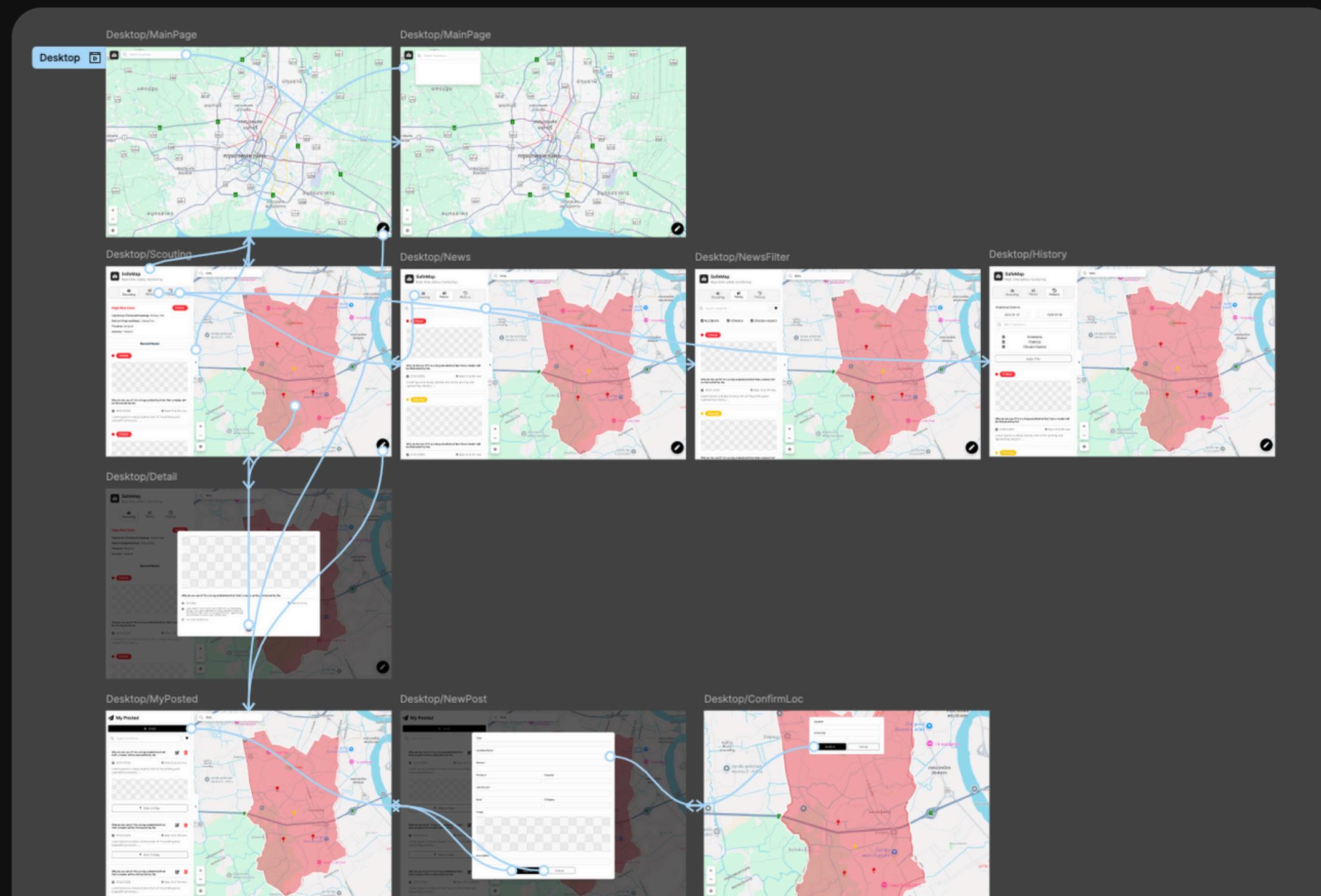
SPRINT 2 - ER Diagram

Designed the ER diagram for users, danger reports, districts, and admin approval to define the data model.



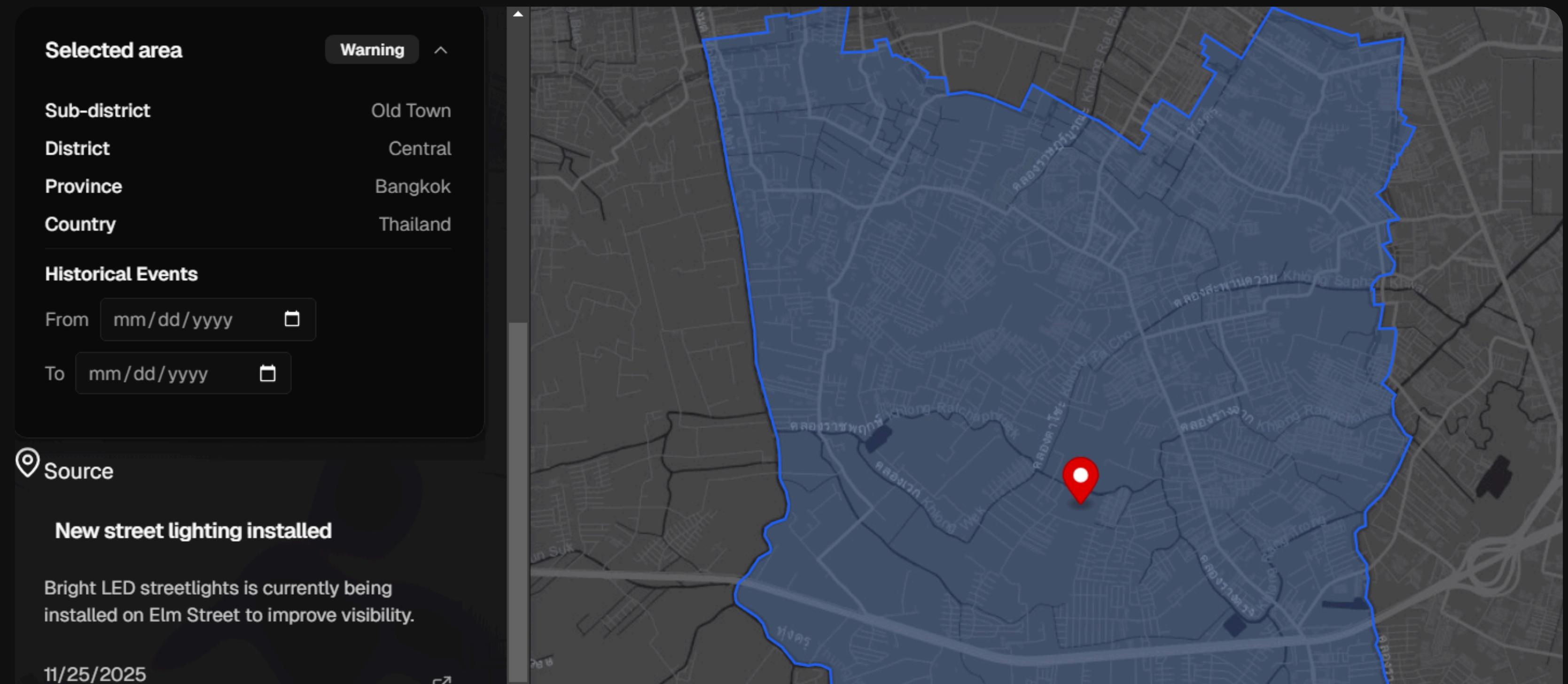
SPRINT 2 - Clickable Prototype

Integrated all main screens into a high-fidelity prototype, including the red-zone detail popup with mock API data for the Thung Khru area.



SPRINT 3 - Map Interaction & Zone Overlay

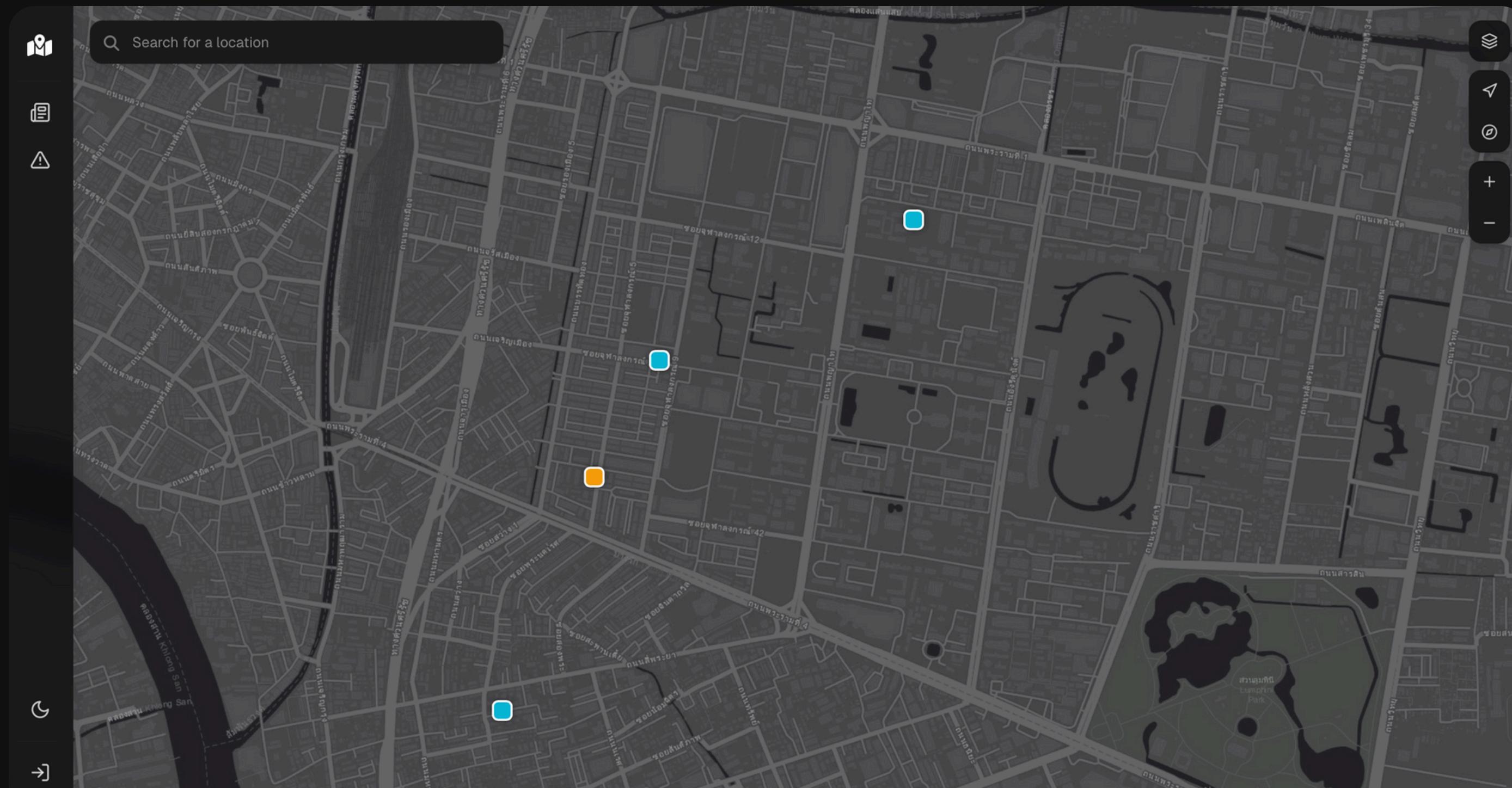
Dynamic Map API - Integrated the live MapLibre API to enable an interactive map features.
Live Red Zone Overlay - Fetched unsafe zone data and displayed red overlays on the map.



SPRINT 3 - News Display on Map

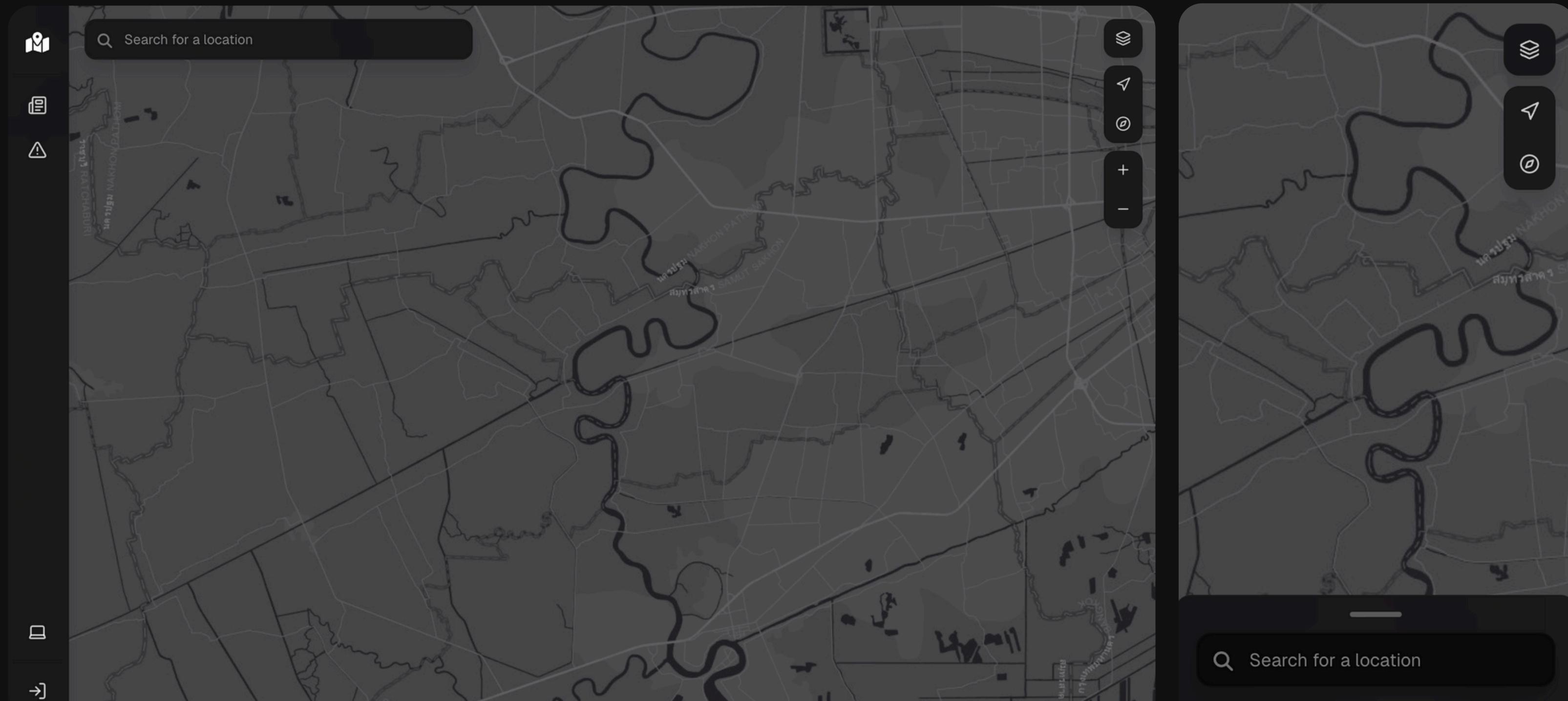
News Marker Integration - Displayed safety news on the map using markers linked.

News Detail Popup - Added tap-to-view popups showing news details.



SPRINT 3 - User Experience

Responsive UI - Adjusted layout and components to ensure smooth usability across desktop and mobile devices.



SPRINT 3 - Schema Setup

Database Setup - Tables for zones, news, events, and relationships needed for the platform
API Integration - Endpoints allow the front-end to request map data, zones, and news.

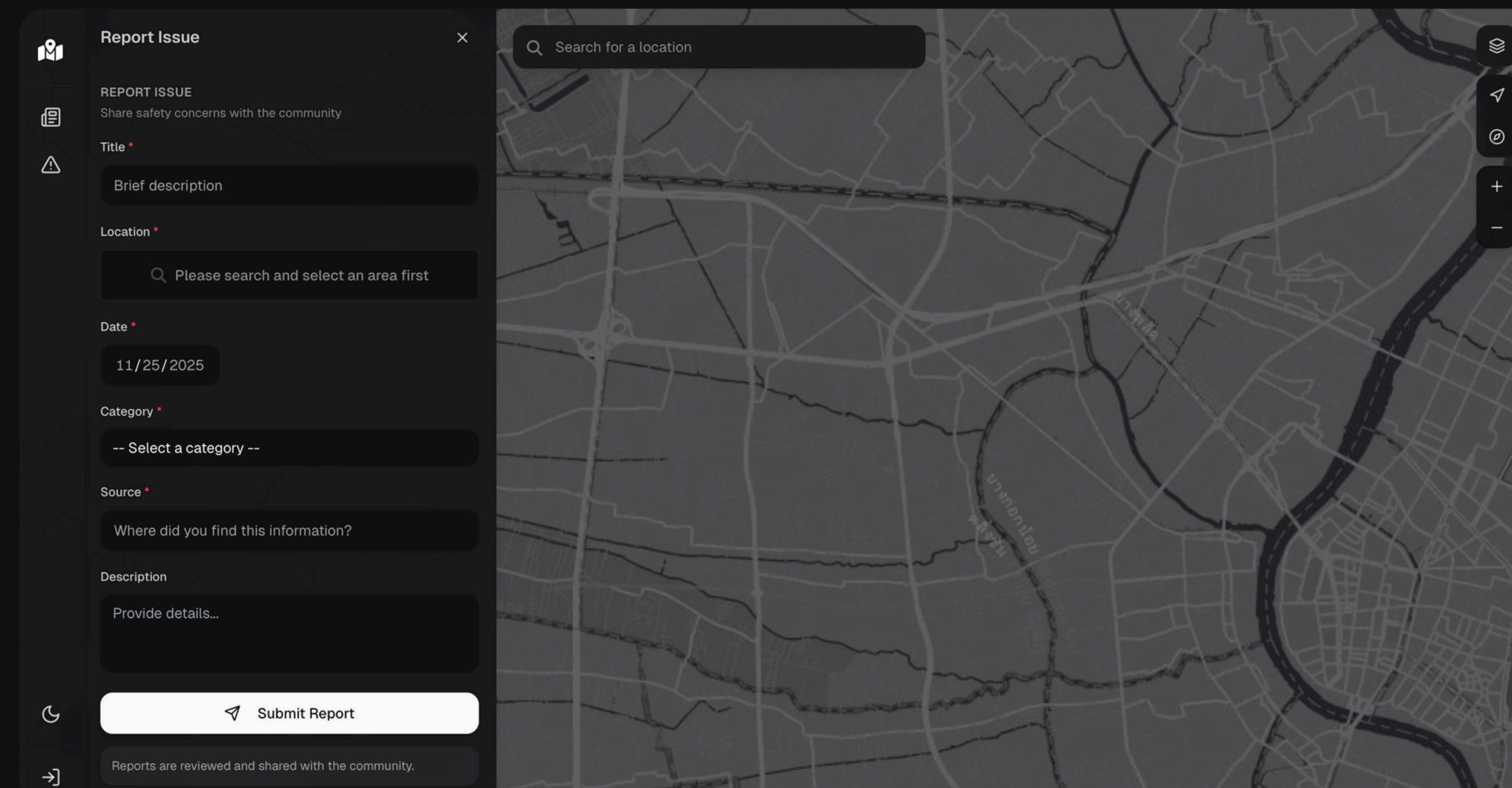
Database Tables

The screenshot shows a database management interface with a dark theme. At the top, there is a dropdown for 'schema public' and a search bar with placeholder text 'Search for a table'. A green button on the right says '+ New table'. Below this is a table with the following data:

NAME	DESCRIPTION	ROWS (ESTIMATED)	SIZE (ESTIMATED)	REALTIME ENABLED		
app_settings	No description	1	32 kB	X	2 columns	⋮
category_score	No description	4	32 kB	X	4 columns	⋮
district_zone	No description	50	80 kB	✓	5 columns	⋮
news	No description	10	72 kB	✓	17 columns	⋮
news_score	No description	10	56 kB	✓	3 columns	⋮
news_severity	No description	0	32 kB	X	4 columns	⋮

SPRINT 4 - Post News Feature and Validation Flow

Implemented a form allowing users to submit incident reports with title, description, location, and optional media reference.



SPRINT 4 - API Integration (CRUD)

All major features are now connected to backend endpoints for full functionality.

Create news – POST /api/news to submit incident reports.

Read news – GET /api/news and /api/news/{id} for map markers and detail pop-ups.

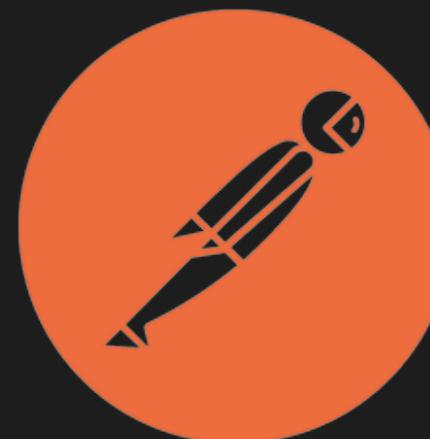
Update news – PUT /api/news/{id} for admin editing and approval.

Delete news – DELETE /api/news/{id} to remove invalid reports.

District risk API – GET /api/districts/{district} to show live risk level on the map.



Coding & debugging



API testing



Database & backend services

All endpoints are tested via Postman and connected to Supabase database & triggers.

SPRINT 4 - Filtering

Implemented simple filters allowing users to view zones or news based on type or severity.

The image displays three screenshots of a mobile application interface, likely a safety or emergency app, showing how users can filter news and alerts based on type and location.

Screenshot 1: Safety News Filter

Top Bar: Warning Alerts

Section: SAFETY NEWS

Text: Latest safety alerts and updates

Filters:

- Accidents
- Violence
- Caution
- Natural Hazard

Search: Search news by title

Bottom Buttons: Selected area (button), Warning (button)

Content Area:

- Source:** Car crash on Highway 50
- Description:** Caution advised due to a multi-vehicle accident on Highway 50 causing delays.
- Date:** 11/25/2025
- Location:** Highway 50, Exit 7

Screenshot 2: Safety News Filter

Top Bar: Warning Alerts

Section: SAFETY NEWS

Text: Latest safety alerts and updates

Filters:

- Accidents
- Violence
- Caution
- Natural Hazard

Search: Search news by title

Bottom Buttons: Selected area (button), Warning (button)

Content Area:

- Source:** Tornado Warning Issued
- Description:** A tornado warning has been issued for the area. Residents are advised to take shelter.
- Date:** 11/25/2025
- Location:** Northern District

Screenshot 3: Safety News Filter

Top Bar: Info Alerts

Section: SAFETY NEWS

Text: Latest safety alerts and updates

Filters:

- Accidents
- Violence
- Caution
- Natural Hazard

Search: Search news by title

Bottom Buttons: Selected area (button), Warning (button)

Content Area:

- Source:** Community safety patrol launched
- Image:** An illustration of a character with purple hair and a blue scarf.
- Description:** Local volunteers have started a nightly safety patrol in the downtown area.
- Date:** 11/25/2025
- Location:** Downtown Patrol HQ

SPRINT 4 - Scoring System in Database

Stores credibility, severity, and recency weights for scoring incidents, aggregate into district severity.

Database Triggers

Execute a set of actions automatically on specified table events

schema public + Docs

+ New trigger ⋮

NAME	TABLE	FUNCTION	EVENTS	ORIENTATION	ENABLED	⋮
news_global_recompute	news	trg_news_global_reco...	AFTER UPDATE AFTER INSERT AFTER DELETE	STATEMENT	✓	⋮
news_score_update_district	news_score	trg_news_score_updat...	AFTER UPDATE AFTER INSERT AFTER DELETE	ROW	✓	⋮
news_update_score	news	trg_news_update_score	AFTER DELETE AFTER UPDATE AFTER INSERT	ROW	✓	⋮

Database Functions

schema public + Docs

Create a new function ⋮

NAME	ARGUMENTS	RETURN TYPE	SECURITY	⋮
classify_risk_level	p_score double precision	risk_level	Invoker	⋮
compute_news_score	p_news_id uuid	double precision	Invoker	⋮

SPRINT 5 - Testing & Documentation Update

Re-ran E2E + unit tests and updated the User Manual and Deployment Guide for the v1.0 release.

The screenshot shows a CI pipeline summary for a job named "test". The job status is green with a checkmark, indicating success. The pipeline steps are listed as follows:

- > Set up job (1s)
- > Checkout repository (1s)
- > Setup Node.js (3s)
- > Install dependencies (46s)
- > Install Xvfb (10s)
- > Install Playwright browsers (18s)
- > Start local server (13s)
- > Run Playwright tests in headed mode (15s)
- > Post Setup Node.js (0s)
- > Post Checkout repository (0s)
- > Complete job (0s)

At the top right, there are buttons for "Re-run all jobs", "Latest #5", and three dots for more options. On the left, there's a sidebar with links for "Summary", "Jobs" (which is selected), "Run details", "Usage", and "Workflow file".

Project Details - Features and MVP

★ Interactive Safety Map ★



Safety News Feed



Zone scoring system

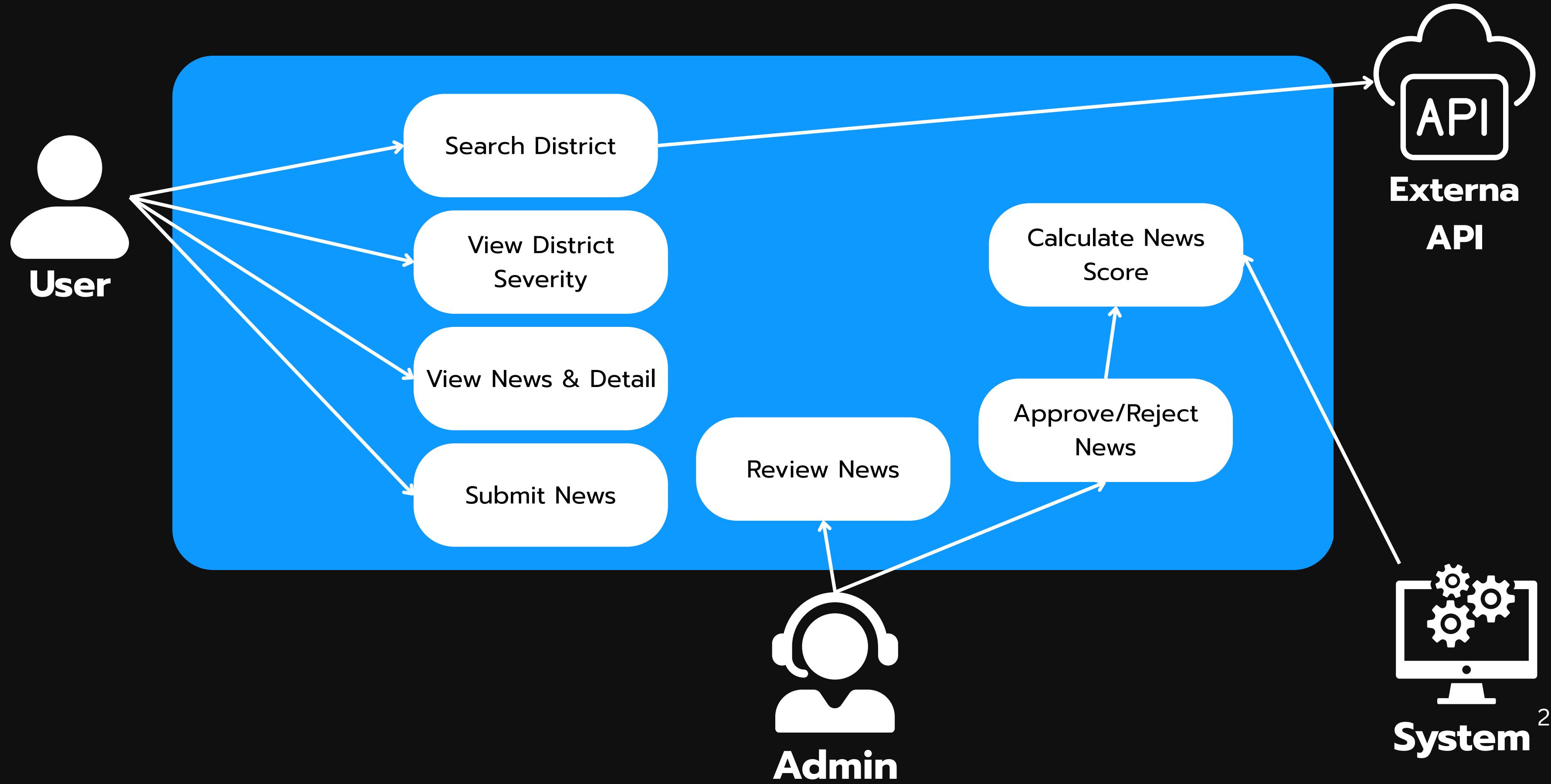


Post News (User Submission)

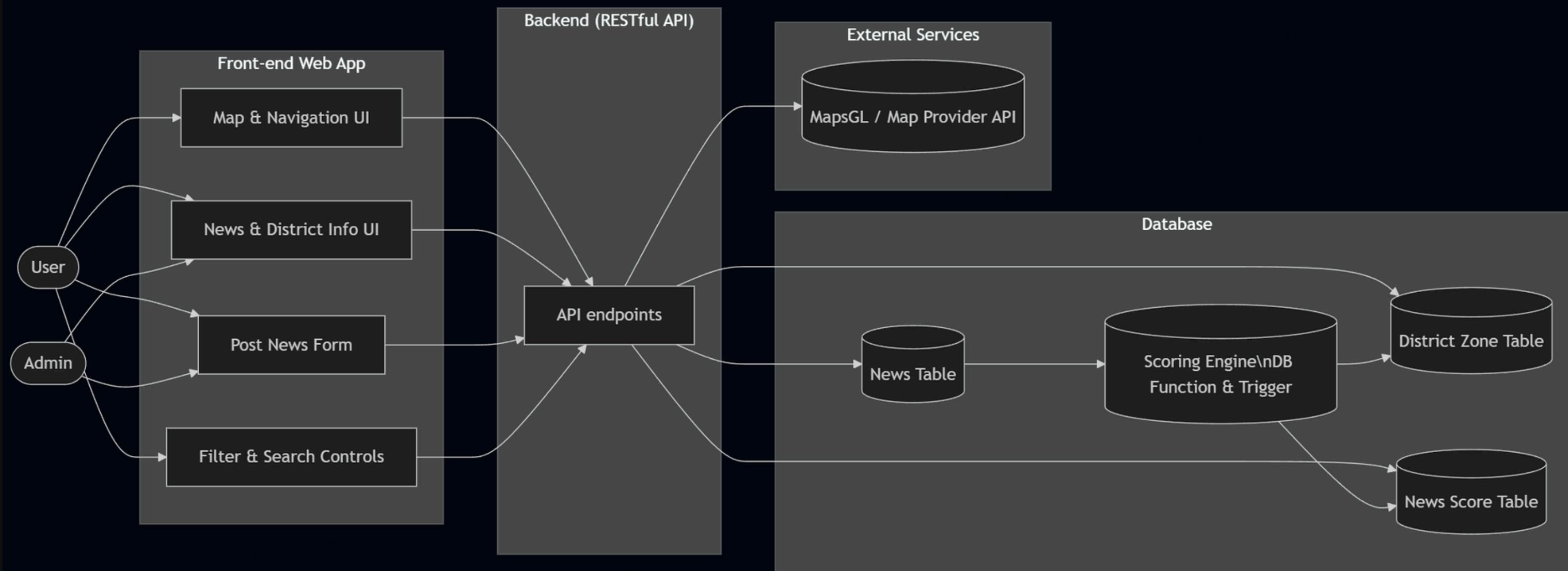


Authentication (Google OAuth)

Project Details - Use Case Diagram



Project Details - Component Diagram



Project Details - Architecture Design

Frontend + Backend Database



Deploy On

▲Vercel

 supabase

Members & Roles



66070503434
Parunchai Kaewkhampa
Scrum Master & Front-end Developer



66070503436
Punnawat Mungkalarungsi
UX/UI Designer & Back-end Developer



66070503445
Peerawish Channukool
Front-end Developer



66070503446
Pakkawat Kaolim
Project Owner & Back-end Developer



66070503451
Veerathach Rattanarojt
Back-end Developer



66070503462
Kamil Pattanasakulloy
Back-end Developer



68540460045
Raphael Canu
Front-end Developer



68540460044
Gabin Dalibard
Front-end Developer

demo

Problem & Solution

Problem 1 – History Feature Could Not Be Completed in Time

- The original History Mode (1M, 6M, 1Y, 5Y) required complex filtering, scoring, and UI integration.
- Development schedule and workload did not allow full implementation within the sprint timeline.

Solution

- Reduced the History feature to a minimal version aligned with MVP requirements.
- Prioritized essential functionality and postponed advanced history browsing to a future release.
- Ensured core user flows remained stable and usable without the full history scope.

Problem 2 – Data Consistency Issues

- Early in development, frontend components and backend responses used different field names and formats.
- This caused mismatched data, rendering errors, and inconsistent mapping of news, zones, and scores.

Solution

- Standardized the data structure across both ends with unified response formats.
- Updated REST endpoints and frontend interfaces to follow the same schema.

Conclusion

The **Dangerless** project successfully delivered a complete MVP that includes an interactive safety map, real-time red-zone overlays, safety news markers, district severity scoring, and a validated user reporting system. Through 5 structured sprints, the team built a fully functional platform with authentication, map integration, and backend API connectivity.

Key achievements include:

- A reliable, user-friendly system that helps travelers assess neighborhood safety in real time.
- A complete end-to-end pipeline: data input → verification → scoring → visualization.
- A responsive UI that functions across mobile and desktop.
- Strong team collaboration using Agile methodology, improving skills in planning, communication, frontend–backend integration, and problem-solving.

Lessons learned:

- Prioritization is essential – focusing on core features produced a stable MVP.
- Early alignment through SRS and ER diagrams greatly reduced development errors.
- Map-based products require careful performance optimization and UI attention.

Future Work

Offline Mode

- Allow users to view last-seen maps and incidents without internet – useful for travelers with unstable connectivity.

Advanced History Mode

- Reintroduce and enhance historical news and incident timelines (1M–5Y), allowing pattern analysis and trend visualization.

Push Notifications & Live Alerts

- Notify users when entering a dangerous area or when new incidents occur nearby.

Multi-language Support

- Expand accessibility to international travelers.



DANGERLESS

An application for providing travelers with real-time neighborhood safety ratings and local safety news alerts.