🏗️ Project: Situation-Adaptive Safe Routing AI Navigation (Draft)

> ⚠️ Status: Provisional (仮定)
> 本ドキュメントに記載されている仕様、技術選定、およびスコープは初期アイデア段階のものです。
> 実装の進行やハッカソンのルール確認状況に応じて、柔軟に変更・ピボットすることを前提としています。

1. 企画概要 (Overview)

- **プロダクト名（仮）:** Situation-Adaptive Safe Routing AI Navigation
- **コンセプト:** 「いつもの移動は“安心寄り”に、非常時は“生存寄り”に切り替わるナビ」
- **コアバリュー:**
  - 既存ナビの「最短・最速」ではなく、「リスク回避」を最適化する。
  - ブラックボックスな指示ではなく、Gemini 3 が「なぜその道を避けたか」を思考・説明する。

2. ハッカソン戦略 (Strategy)

Target: Google Cloud Japan AI Hackathon Vol.4

- **狙うテーマ:** Agentic AI（自律的に考え行動するAI）
- **必須要件:**
  - **AI:** Gemini 3 (Experimental / Flash) を活用し、高度な推論（Thinking Process）を取り入れる。
  - **Cloud:** Google Cloud (Firebase Genkit / Cloud Functions) をバックエンドに採用する。
- **提出戦略 (重要):**
  - **形態:** iOSネイティブアプリ (SwiftUI)

- **デプロイURL対策:** アプリ自体のURLはないため、**「動作デモ動画(YouTube)」** または **「GitHubリポジトリ」** を提出URLとして扱う。（※要FAQ再確認）

3. システムアーキテクチャ (Architecture)

構成: Client-Server モデル（"ADK" をバックエンドに集約）

Frontend (iOS)

- **Tech:** Swift, SwiftUI, MapKit (or Google Maps SDK)
- **役割:**
  - UI/UXの提供（日常モード/非常時モードの劇的な変化）。
  - 位置情報とユーザーの移動モードをバックエンドへ送信。
  - AIからの回答（ルート座標 + 説明テキスト）を描画。

Backend (Agentic AI Layer)

- **Tech:** Firebase Genkit (Node.js/TypeScript) **← ここで Google ADK を活用**
- **役割:** マルチエージェントオーケストレーション
  1. **Input Agent:** クライアントからの要求と、外部データ（天気・擬似的な事故データ）を統合。
  2. **Risk Evaluator (Gemini 3):**
     - *Thinking Process:* 「ユーザーは非常時モードを選択している。現在地周辺は大雨だ。通常のリスク重み付けを変更し、低地を徹底的に避ける思考を行う」
  3. **Route Selector:** リスク評価に基づき、最適な経路を選定（※MVPでは擬似ロジック可）。
  4. **Narrator Agent:** ユーザーへの「納得感のある説明」を生成。

4. MVP機能スコープ (Scope for Hackathon)

「狭く、深く、派手に」 実装する。

- **対象エリア:** 特定の1都市（例：渋谷区）に限定。データもこのエリア分のみMockで用意。
- **機能:**

    - **モード切替:** 「日常（青）」⇔「非常時（赤）」のトグルスイッチ。
    - **AI説明:** モード変更時、即座にGemini 3が状況を判断し、テキストで理由を語る。
    - **地図描画:** スタート→ゴールを結ぶラインを描画（危険エリアを避けていることが視覚的にわかること）。
- **データ:** リアルタイムAPIの完全連携は必須としない。ハッカソン用に「事故データ」「浸水エリア」をJSON等でMockとして持たせることを許容する。

5. 開発ロードマップ (Next Actions)

1. **環境構築 (Backend):**
    - PCにて Firebase Genkit のプロジェクトを初期化 (TypeScript)。
    - Gemini 3 のAPIキーを取得し、Genkitから "Hello World" できるか確認。
2. **環境構築 (Frontend):**
    - Xcodeで新規プロジェクト作成。
    - マップ表示と「モード切替スイッチ」の基本UI実装。
3. **疎通確認:**
    - iOSからBackendのエンドポイントを叩き、Geminiの返答がアプリに表示されるかテスト。
4. **作り込み:**
    - 「思考するプロンプト」の調整。
    - デモ動画映えするUIアニメーションの実装。

📝 メモ (To-Do)

☐ ハッカソンのFAQを再確認し、iOSアプリ提出時の「デプロイURL」の扱いを確定させる。

☐ Gemini 3 の "Thinking" 機能を有効にするためのパラメータ設定を調べる。

☐ デモ動画の構成案（絵コンテ）を練る。