**High Level Design (HLD) - Street Smart**

**1. Introduction**

Street Smart is a mobile application designed to help users navigate unfamiliar or potentially unsafe areas by providing optimized routes based on real-time safety data. The app calculates the safest route using multiple parameters, including street lighting, business density, crime statistics, and user reports.

**2. System Architecture**

**2.1 Overview**  
Street Smart follows a client-server architecture, where the mobile application communicates with a backend server that processes and provides route recommendations. The backend integrates multiple data sources to assess the safety of different routes.

**2.2 Components**

* **Mobile Application** (Android)
  1. User interface for route selection and preferences.
  2. Real-time map display and navigation.
  3. Safety alerts and notifications.
* **Backend Server**
  1. RESTfulAPI – Manages communication between the mobile application and backend services.
  2. Safety Score Calculation Engine – Uses the map layer data (lighting, hazards, crime, business density, user reports) to assign safety scores to possible paths.  
     Continuously updates route safety scores based on live data: new reports, road closures, changing conditions.
  3. Data Integration & Map Layer Management–

1. Data collection and storage:

* Lighting Layer   
  Data is collected from public sources and updated daily through official reports and real-time user reports.
* Crime Data Layer   
   Official crime statistics from government APIs   
  Historical data for each area.
* Business Density Layer   
  Uses external sources such as Google Places API to update open business availability hourly.
  + - User Reports Layer   
      Users can report hazards, and the system filters duplicate reports, evaluates report credibility, and removes spam.

2. Route Calculation:

* + - Each route segment is scored based on lighting, crime data, hazards, and user reports.
    - The Safety Score Calculation Engine queries relevant layers for the defined start and destination points.

3. User Display:

* + - Users can manually control the display of different layers (enable crime layer, disable business layer).
    - Dangerous areas are color-coded, making it easy to identify risk zones.
    - Dynamic updates- layers refresh automatically when opening the map.

4. Real-Time Navigation Adjustments:

* + - If a sudden hazard is reported (e.g., a crime incident or road closure), the server updates the relevant data layer and recalculates safety scores.
    - The user receives an alert and can choose a safer alternative route.
  1. Authentication and user management.
  2. Machine Learning-Based Safety Enhancements-

Predictive Safety Score Model – Uses Random Forest to predict safe routes based on historical and real-time data.

Dynamic Risk Adjustment – Continuously updates route safety scores based on live environmental and user feedback.

* **Database**
  1. User profiles and preferences.
  2. Historical route safety data.
  3. Crime statistics and environmental conditions.
  4. User-generated reports and feedback.
  5. Map Layers Data- Lighting info, crime stats, business listings, hazards, user reports.
* **External Data Sources**
  1. Government crime data APIs.
  2. Business activity and density data.
  3. Public infrastructure data (lighting, road conditions).

**3. Functional Requirements**

* **User Registration & Authentication**
  1. Sign-up via phone number or email.
* **Route Calculation & Navigation**
  1. Route optimization based on safety parameters.
  2. Multiple route suggestions with estimated safety levels.
  3. Real-time rerouting based on updated conditions.
* **User Safety Reports**
  1. Users can report issues like poor lighting, suspicious activity, or crime incidents.
* **Real-Time Updates & Alerts**
  1. Notifications for new safety incidents along a route.
  2. Alerts for route deviations or entering high-risk areas.
* **Map Layer Display (Mobile App)**

1. Visual overlay for each layer (lighting conditions, hazards, etc.) on an interactive map.
2. Ability to toggle layers on/off (e.g., see current hazards or crime hotspots).

**4. Technology Stack**

|  |  |
| --- | --- |
| **Component** | **Technology** |
| Mobile Application | Flutter (Dart) |
| Backend Server | Node.js Express |
| Database | PostgreSQL |
| Machine Learning | Python |
| Maps & Navigation | Google Maps API / OpenStreetMap |