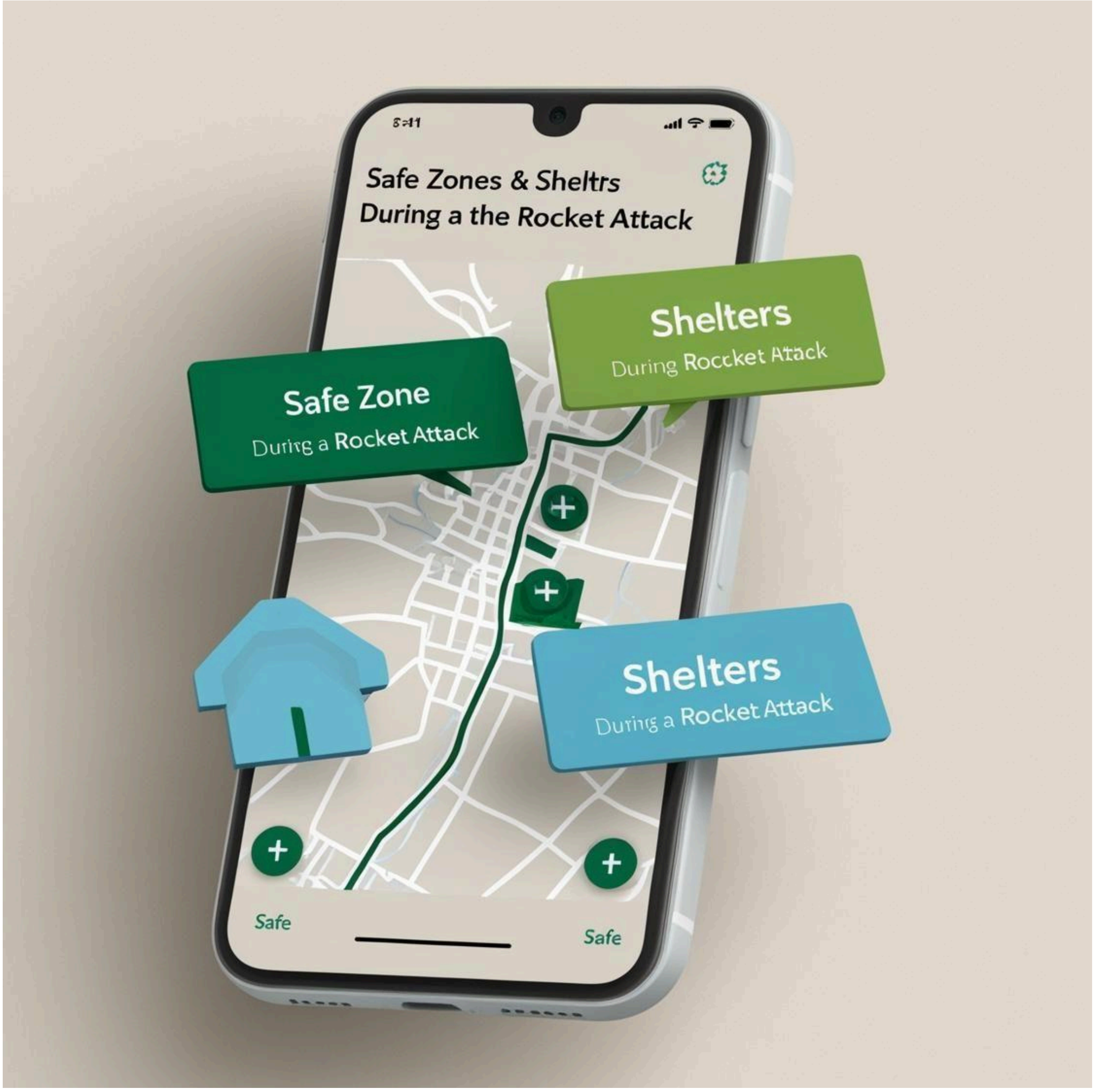
Working Title: "SafeSpot - Emergency Shelter Finder App"
Author: Guy Oved,Itay Raviv,Sharon Pais
Date: 11/12/2024



1. Introduction

1.1 Project Overview

In regions where emergency situations such as missile attacks or severe weather events occur, quick access to protected shelters is critical. "SafeSpot" is a mobile application designed to help users locate the nearest safe shelter (e.g., public bomb shelters, designated secure rooms, protected staircases) in real-time. Using the device's GPS, the app will identify and display shelters on an interactive map, provide step-by-step navigation, and offer essential information such as shelter type, capacity, and maintenance status.

1.2 Objectives

- **Rapid Access to Information:** Enable users to quickly find and reach the closest protected space during an emergency.
- **Enhanced Preparedness:** Improve community resilience by providing up-to-date shelter data readily available on a mobile device.
- **User-Friendly Experience:** Present information through an intuitive user interface, allowing even non-technical individuals to operate the app efficiently under stress.
- **Scalability and Flexibility:** Support multiple languages, offline usage, and integration with various data sources and navigation APIs.

1.3 Target Audience

The primary users are residents of areas prone to emergencies who need immediate access to reliable shelter information. Secondary users could include visitors or travelers who might not be familiar with local safety infrastructure.

2. Project Scope and Features

2.1 Core Features

Real-Time Location Tracking:

- Leverages the phone's GPS to determine the user's position.
- Continuously updates user location for accurate navigation and distance calculations.

Interactive Map Display:

- Presents a map (via Google Maps or OpenStreetMap) with markers indicating nearby shelters.
- Allows zooming, panning, and filtering shelters by type or availability.

Navigation and Directions:

- Provides turn-by-turn instructions to the selected shelter.
- Offers estimated travel time by foot or vehicle.
- Integrates with external map services for advanced navigation routes.

Shelter Information:

- Displays shelter details: type (public, private, stairwell, secure room), capacity (if available), and any maintenance or availability notes.
- Enables users to view distance, estimated arrival time, and other context-specific data.

Offline Mode:

- Allows pre-download of map tiles and shelter data for connectivity-limited scenarios.
- Ensures minimal functionality (location display and previously downloaded shelter markers) even without an active internet connection.

Push Notifications for Emergencies (Future Feature):

- Integrates with a notification system to alert users about urgent events or changes in shelter statuses.
- Offers updates during crises, guiding users to safer locations if conditions change.

2.2 Additional Features (Considerations)

- **User Feedback & Ratings:** Option for users to report issues with shelters or suggest updates to data.
- **Bookmarking & Favorites:** Allows users to save frequently accessed shelters for quick reference.
- **Accessibility Features:** Large icons, voice assistance, and high-contrast modes for visually impaired users.

3. Design Considerations

3.1 User Interface (UI) Concepts

The UI will focus on clarity, simplicity, and rapid interpretation of information. Key design principles:

- **Minimalist Layout:** Emphasize the map and the user's current location, with contextual overlays rather than cluttered UI elements.
- **Clear Iconography:** Use recognizable icons (e.g., shelter symbol) to differentiate shelter types.
- **Consistent Color Scheme:** Employ colors associated with safety and emergency readiness (greens, blues for calmness; red/orange highlights for critical alerts).
- **Multi-Lingual Layout:** Ensure that the UI seamlessly supports right-to-left (Hebrew, Arabic) and left-to-right (English) scripts. The interface should adjust text alignment, icons, and layout direction dynamically.

3.2 User Experience (UX) Flows

- **Onboarding:** Brief explanation of app functionalities, asking for location permissions, and selecting preferred language.
- **Main Screen (Map View):** Immediately shows the user's position and nearby shelters as markers. A bottom sheet or floating panel can list shelters by proximity.
- **Shelter Details Screen:** Appears after tapping a marker. Displays details, a "Navigate" button, and relevant data.
- **Settings Screen:** Allows changing language, downloading offline maps, managing notifications, and accessing help or tutorial materials.

3.3 Accessibility and Responsiveness

- **Scalable Text Sizes:** Users can increase or decrease font sizes as needed.
- **Color Contrast:** Sufficient contrast between text and background to ensure readability under different conditions.
- **Voice Assistance Compatibility:** The app should be compatible with native screen readers.

4. Technical Overview

4.1 Architecture

- **Frontend:**
  - **Technology:** React Native or Flutter for cross-platform mobile development (iOS, Android).
  - **UI Libraries:** Material Design Components (Flutter) or React Native Paper.
  - **State Management:** Redux (for React Native) or Provider/Bloc (for Flutter).
- **Backend & Data Management:**
  - **Platform:** Firebase or a custom Node.js backend hosted on a cloud platform.
  - **Data Storage:** Firestore (NoSQL) for real-time updates. Shelters' static data updated periodically.
  - **Authentication & Security:** May not be critical for initial MVP, but consider future user registration or verification.
- **Mapping and Navigation API:**
  - **Google Maps API or OpenStreetMap:** For map tiles, markers, and geocoding services.
  - **Navigation:** Google Directions API or OpenRouteService for routing and ETA calculations.
- **Offline Capability:**
  - **Local Storage:** Pre-downloaded JSON data and map tiles stored on the device. SQLite or local file storage for caching.
  - **Fallback Logic:** The app attempts to load from cache when the network is unavailable.

4.2 Data Sources

- **Shelter Data:**
  - Obtained from authoritative sources (e.g., a government agency or defense command providing publicly available shelter data).
  - Data includes GPS coordinates, shelter type, last maintenance date, and possibly capacity.
  - Regular updates might be scheduled via a server that fetches and stores new data in Firestore.

4.3 Security and Privacy

- **Location Privacy:** App requests user consent to access location data and uses it solely for navigation and proximity calculations.
- **Data Integrity:** Ensure data from external sources is validated and sanitized.
- **Communication Security:** Use HTTPS for all network requests, secure API keys, and follow best practices for data handling.

5. Planned Work and Timeline

5.1 Development Phases

1. **Requirements & Design (2-3 weeks)**
   - Finalize user stories, wireframes, and UI/UX design.
   - Determine precise APIs and data formats.
2. **Proof of Concept (2 weeks)**
   - Implement a basic map view with user location.
   - Display mock shelter markers.
   - Integrate language switching.
3. **Core Feature Implementation (4-6 weeks)**
   - Add real shelter data loading from a backend source.
   - Implement navigation features (routing and ETA).
   - Introduce offline mode (basic caching).
4. **Testing & Refinement (2-3 weeks)**
   - Conduct unit and integration tests.
   - Run user experience tests with initial users.
   - Improve performance and address feedback.
5. **Release Preparation (1-2 weeks)**
   - Polish UI, add final translations.
   - Prepare App Store/Play Store listings.
   - Final security and privacy checks.

Total Estimated Timeline: Approximately 12-16 weeks from initial concept to first public release (MVP).

5.2 Deliverables

- **Design Documents:** High-fidelity mockups, finalized user stories.
- **Working Prototype:** A functional MVP demonstrating main features.
- **Testing Reports:** Documentation of test cases, test results, and fixes.
- **Deployment Instructions:** Guidelines for installing and running the app, along with a maintenance plan.

6. Evaluation and Success Metrics

6.1 User Acceptance

- **Adoption Rate:** Number of downloads and active users over time.
- **User Feedback:** Positive ratings, comments, and suggestions for improvements.
- **Average Time-to-Shelter:** Reduction in the time it takes users to find a shelter location during drills or emergency tests.

6.2 Performance Indicators

- **App Responsiveness:** Map loading time, shelter data retrieval speed.
- **Accuracy:** Reliability of location services and correctness of shelter data.
- **Offline Reliability:** Ability to function properly without connectivity in pre-downloaded areas.

7. Risks and Mitigations

- **Data Accuracy Risk:** If the shelter data is outdated or incorrect, users may lose trust.
  Mitigation: Regular updates and verifications, user reports of incorrect data.
- **API Changes or Limits:** Third-party map APIs may change or enforce limits.
  Mitigation: Consider alternative map providers (OpenStreetMap) and caching strategies.
- **Language and Accessibility Challenges:** Ensuring full accessibility and correct text alignment in multiple languages can be complex.
  Mitigation: Rigorous testing with native speakers and accessibility tools.

8. Conclusion

"SafeSpot" aims to enhance personal safety and community resilience by providing quick, reliable shelter-finding capabilities during emergencies. By combining intuitive UI, accurate location data, and responsive design, this application aspires to become an essential tool for individuals living in or visiting high-risk areas.