**Functional Requirements**

**1. Disaster Alert and Detection System**

* Provide **real-time disaster alerts** to users based on official sources and social media data analysis.
* Continuously monitor and analyze social media feeds for disaster-related information.
* Display risk zones on an interactive map using layered data (e.g., risk level, shelters).
* Send push notifications for immediate disaster warnings.

**2. Live Map Module**

* Show danger zones, shelters, hospitals, gathering points, and safe zones on a live map.
* Offer **AI-powered optimal route calculations** considering current disaster risks.
* Provide **offline map support** to ensure usability without internet.
* Update map dynamically as new data arrives or user location changes.
* Detect user location and display it on the map.

**3. AI Help Bot**

* Provide **voice and text-based assistance** to users.
* Support **psychological first aid** and basic counseling via conversational AI.
* Provide guidance, information retrieval, and step-by-step instructions during disasters.
* Enable offline usage with cached knowledge base for critical info access.

**4. Community Support Platform**

* Allow users to request and offer help (matching system).
* Facilitate missing persons reporting and tracking.
* Support donation campaigns and crowdsourced aid collection.
* Enable messaging and communication among community members.
* Implement moderation features to manage user content and interactions.

**5. User Membership and Security System**

* Support **anonymous mode** for privacy-focused users.
* Allow creation of **child profiles** with restricted access.
* Support both **ad-supported (freemium) and ad-free subscription models**.
* Provide secure user authentication and data protection mechanisms.

**6. Disaster Knowledge Library**

* Offer educational resources such as disaster guides, simulations, and tests.
* Enable offline access to library content.
* Provide interactive content (quizzes, simulations) for preparedness.

**7. Crisis Mode Features**

* Enable **offline mode** with essential app functionalities.
* Support **energy-saving mode** to extend device battery life during crises.
* Provide **emergency signaling** features (e.g., SOS signals).

**8. AI Behavioral Analysis and Suggestions**

* Track user interactions to provide personalized disaster preparedness tips.
* Offer automated suggestions based on behavioral data and location.

**9. Admin and Moderation Panel**

* Manage users, help requests, and community content.
* Moderate reported content and interactions.
* Oversee AI data training and feedback loops.
* Monitor system health and usage analytics.

**Non-Functional Requirements**

**1. Performance**

* The app should load disaster alerts and map data within 3 seconds under normal network conditions.
* Real-time updates must have minimal latency (less than 5 seconds delay).
* Offline functionalities should be responsive and usable within 2 seconds response time.

**2. Scalability**

* Backend infrastructure must scale to handle thousands of concurrent users during major disasters.
* Modular architecture to add new features without impacting existing functionality.

**3. Availability and Reliability**

* Ensure 99.9% uptime for critical services like alert notifications and map updates.
* Implement fallback mechanisms if primary data sources fail (e.g., cached data).

**4. Usability and Accessibility**

* Provide intuitive UI/UX for all user groups, including children and elderly users.
* Support multiple languages (at least 3 major languages initially).
* Include voice navigation and support for accessibility standards (e.g., screen readers).

**5. Security**

* Protect user data with end-to-end encryption, especially location and personal info.
* Comply with international privacy regulations (GDPR, KVKK).
* Provide secure login and session management with multi-factor authentication options.
* Support anonymous usage without storing personally identifiable information unless opted in.

**6. Maintainability**

* Codebase should be modular and documented to facilitate future development.
* Support easy update deployment without downtime.
* Log and monitor errors centrally for rapid troubleshooting.

**7. Compatibility**

* Support Android and iOS platforms via Flutter or React Native.
* Ensure compatibility with a wide range of device models, including low-end smartphones.

**8. Offline Support**

* Enable offline access to critical app features like maps, help bot, and knowledge base.
* Automatically synchronize offline data when connectivity is restored.

**9. Localization**

* Support multilingual content and UI localization.
* Date, time, and measurement units should adapt to user’s region.

**10. Legal and Compliance**

* Ensure all AI responses include disclaimers for emergency advice.
* Ensure donation and financial transactions comply with relevant legal frameworks.
* Store user data according to legal retention policies.