

REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

**MINISTER OF HIGHER
EDUCATION**

UNIVERSITY OF BUEA



REPUBLIQUE DU CAMEROUN

PAIX-TRAVAIL-PATRIE

**MINISTERE DE L'ENSEIGNEMENT
SUPERIEURE**

UNIVERSITE DE BUEA

Disaster Management Mobile Application

TASK 3 : Requirement Analysis

Group 18

Table of content

Introduction.....	3
Requirement Classification.....	4
I. Functional Requirements	4
II. Non-Functional Requirements	6
External Interface Requirements	7
Requirements Prioritization	9
➤ High Priority Functional Requirements	9
➤ Medium Priority Functional Requirements.....	9
➤ High Priority Non-Functional Requirements	9
➤ Medium Priority Non-Functional Requirements.....	10
Constraints	10
Conclusion	10

Introduction

In the face of a disaster, timely information and clear action plans are critical for survival and recovery. A well-designed disaster management mobile application can empower individuals and communities to prepare for, respond to, and recover from emergencies. However, building such an application requires a strong foundation by understanding what users truly need. This is where Requirement Analysis comes in.

Requirement Analysis is the initial phase of development, where we meticulously analyze the documented needs, objectives, and constraints of stakeholders to define the functional and non-functional requirements of our system. Through this process, we bridge the gap between user needs and the technical capabilities of the application.

Requirement analysis serves as the foundation for the design, development and deployment of a solution that meets the expectations and goals of stakeholders with respect to managing disasters with the use of a mobile application.

Requirement Classification

In classifying our requirements as functional or non-functional for our disaster management mobile application, we based our self on the distinction between what the system does (functional) and how it does it (non-functional).

I. Functional Requirements

Functional requirements define the specific behaviors, features, and capabilities that the system must provide to support the desired functionality of the application. These requirements typically answer the question: "What does the system do?".

In our disaster management mobile application, the functional requirement include:

1. Alerting and Notification System:

- Real-time alerts and notifications about disasters, emergencies, and critical events based on user location and preferences.
- Customizable alert settings for different types of disasters and severity levels.

2. Geolocation and Mapping Integration:

- Integration with mapping services to identify disaster-prone areas, evacuation routes, and nearby resources.
- Geolocation features to track users' locations during emergencies and provide relevant information.

3. Resource Management Tools:

- Tools for managing and coordinating emergency resources, including personnel, equipment, supplies, and facilities.
- Features for updating resource availability, requesting assistance, and tracking resource deployment in real-time.

4. Communication and Collaboration Features:

- Communication channels for stakeholders to share updates, exchange information, and coordinate response efforts.
- Group chats, messaging, and video conferencing capabilities to facilitate collaboration among emergency responders, government agencies, NGOs, and community members.

5. Community Engagement Functionalities:

- Educational resources, preparedness tips, and volunteer opportunities to engage and empower the community.
- Incident reporting mechanisms for users to report incidents, request assistance, and contribute to community-led initiatives for disaster response and recovery.

6. Data Collection and Analysis Capabilities:

- Data collection tools for capturing information related to disaster events, response activities, and community needs.
- Analytics dashboards and reporting features for analyzing and visualizing data to monitor trends, assess impacts, and make informed decisions.

7. Personal Preparedness Tools:

- Checklists, guides, and resources for individuals and families to prepare for disasters and emergencies.
- Personalized preparedness plans and reminders for completing preparedness tasks and staying informed.

8. Emergency Contacts and Resources Directory:

- Comprehensive directory of emergency contacts, resources, and service providers, including government agencies, healthcare facilities, and community organizations.
- Search and filtering capabilities to quickly locate relevant contacts and resources based on location and category.

9. Incident Reporting and Tracking:

- Features for reporting incidents, documenting observations, and tracking response activities in real-time.
- Incident management tools for assigning tasks, updating status, and communicating progress among response teams.

10. Push Notifications and Emergency Alerts:

- Push notification system for delivering timely alerts and updates to users' mobile devices during emergencies.

- Integration with emergency alert systems, such as Wireless Emergency Alerts (WEA), to broadcast critical information to affected populations

II. Non-Functional Requirements

Non-functional requirements specify the qualities or constraints on how the system performs or behaves, rather than what it does. These requirements typically answer the question: "How does the system perform?"

In our disaster management mobile application, the non-functional requirements include:

1. **Performance:**

- Response time for delivering alerts and notifications.
- Scalability to handle increasing user loads during emergencies.

2. **Security:**

- Data encryption to protect sensitive user information.
- Secure authentication mechanisms to control access to the application.

3. **Usability:**

- Accessibility features to ensure usability for users with disabilities.
- Intuitive design and clear instructions for ease of use.

4. **Reliability:**

- Uptime and availability of the application during emergencies.
- Fault tolerance and disaster recovery mechanisms.

5. **Interoperability:**

- Integration with external systems, APIs, and communication protocols.
- Compatibility with interoperability standards used in disaster management.

6. **Localization:**

- Adapting the application to different languages, cultures, and regions, enhancing its usability and accessibility for users from diverse backgrounds.
- Localization of content, resources, and user interfaces for diverse communities.

7. **Accessibility:**

- Accessibility features to ensure usability for users with disabilities.
- Compliance with accessibility standards, such as WCAG, for inclusive design.

8. **Scalability:**

- Ability to scale resources and infrastructure to accommodate increasing user demand during emergencies.
- Scalable architecture for handling spikes in traffic and data volume.

9. **Offline Functionality:**

- Offline access to essential features and information in areas with limited or no internet connectivity.
- Data synchronization capabilities to update information once connectivity is restored.

10. **Cross-Platform Compatibility:**

- This refers to the ability of the application to run seamlessly on different platforms, such as iOS and Android

External Interface Requirements

External interface requirements for our disaster management mobile application encompass interactions between the application and external systems, devices, or users. These interfaces facilitate communication, data exchange, and integration with external entities to enhance the functionality and effectiveness of our application.

Here are some external interface requirements:

User Interfaces:

- **Mobile Devices:**
 - The application should have user interfaces optimized for various mobile devices, including smartphones and tablets, to ensure compatibility and usability across different screen sizes and resolutions.
 - User interfaces should be intuitive, responsive, and accessible, with clear navigation and user-friendly design elements.

Hardware Interfaces:

- **Location Services:**
 - The application should utilize hardware interfaces, such as GPS or Wi-Fi, to access the device's location information and provide geolocation features, including real-time tracking and mapping functionalities.
- **Sensors:**
 - Integration with device sensors, such as accelerometers and gyroscopes, may be used to detect motion, orientation, and environmental conditions relevant to disaster management activities.

Software Interfaces:

- **Mapping Services:**
 - Integration with mapping services, such as Google Maps, to display geospatial data, including disaster-prone areas, evacuation routes, and nearby resources.
- **Weather APIs:**
 - Interfacing with weather APIs to access real-time weather data, forecasts, and alerts, enabling users to monitor weather conditions and anticipate potential disasters or emergencies.

Communication Interfaces:

- **Push Notification Services:**
 - Integration with push notification services, such as Firebase Cloud Messaging (FCM) or Apple Push Notification Service (APNs), to deliver real-time alerts and notifications to users' mobile devices.
- **Messaging Platforms:**
 - Integration with messaging platforms, such as SMS or email gateways, to reach users who may not have the application installed or access to mobile data during emergencies.

Emergency Services Interfaces:

- **Emergency Response Systems:**
 - Interfacing with emergency response systems, such as 119 to enable users to report incidents, request assistance, and initiate emergency response protocols directly from the application.

External Hardware Devices:

- **Wearable Devices:**
 - Compatibility with wearable devices, such as smartwatches, to extend functionality and provide additional monitoring and alerting capabilities for users engaged in disaster management activities.

Requirements Prioritization

Prioritizing functional and non-functional requirements for our disaster management mobile application involves assessing their importance, urgency, and impact on the application's effectiveness in supporting disaster preparedness, response, and recovery efforts.

➤ High Priority Functional Requirements

1. Alerting and Notification System
2. Geolocation and Mapping Integration
3. Resource Management
4. Communication and Collaboration
5. Incident Reporting and Tracking
6. Personal Preparedness Tools
7. Emergency Contacts and Resources Directory

➤ Medium Priority Functional Requirements

1. Community Engagement
2. Data Collection and Analysis
3. Emergency Contacts and Resources Directory

➤ High Priority Non-Functional Requirements

1. Performance
2. Security
3. Usability
4. Reliability
5. Scalability

6. Localization

➤ **Medium Priority Non-Functional Requirements**

1. Interoperability

Constraints

Constraints in the development of our disaster management mobile application may arise from various factors, limiting the implementation of some requirements. Here are some common constraints that may affect the implementation of our functional and non-functional requirements:

1. **Time Constraints:**

- Limited development time may result in prioritizing essential features over less critical ones, leading to the postponement or omission of certain functionalities.

2. **Resource Limitations:**

- Constraints on human resources, budget, or technology infrastructure may impact the development team's ability to implement all desired features within the project scope.

3. **Scope Creep:**

- Expansion of project scope beyond initial expectations may lead to the inclusion of additional features or functionalities, diverting resources and attention from previously prioritized requirements.

4. **External Dependencies:**

- Dependencies on external systems, APIs, or third-party services may introduce risks of delays or disruptions in implementation, particularly if those dependencies experience issues or changes.

Conclusion

Requirement analysis provides valuable insights and guidance for the subsequent phases of the software development lifecycle. It lays the foundation for a successful design, development, and deployment of our disaster management mobile application that effectively meets the needs of stakeholders.