

**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**

---

## EEF 440: Internet Program and Mobile Programming

**TASK 2**

**Group 18**

## Table of content

II. Requirement gathering for our disaster management mobile application .....	1
Approach to requirement gathering .....	1
1. Stakeholders and their roles in the Disaster management mobile application.....	3
2. Functional and Non-Functional requirements .....	4
3. Constraints of our disaster management mobile application.....	7

## **II. Requirement gathering for our disaster management mobile application**

Requirements gathering for a disaster management mobile app involves systematically collecting, documenting, and analyzing the needs and expectations of stakeholders, including users, government agencies, emergency responders, and community organizations.

### **Approach to requirement gathering**

#### **1. Identify Stakeholders:**

- Determine all stakeholders involved in disaster management, including end-users, government agencies, NGOs, emergency responders, and community organizations.

#### **2. Conduct Stakeholder Analysis:**

- Understand the roles, responsibilities, and objectives of each stakeholder group.
- Determine their level of influence, interest, and contribution to the project.

#### **3. Define Project Scope:**

- Clearly outline the goals, objectives, and constraints of the project.
- Define the geographical scope, types of disasters to be addressed, and target user demographics.

#### **4. Conduct User Research:**

- Gather insights from potential users through surveys, interviews, focus groups, and observation.
- Understand their needs, preferences, pain points, and expectations regarding disaster management.

#### **5. Document Functional Requirements:**

- Document specific features and functionalities required in the mobile app based on user needs and stakeholder requirements.
- Prioritize requirements based on their importance and feasibility.

**6. Document Non-Functional Requirements:**

- Identify and document non-functional requirements such as performance, security, scalability, usability, and compliance.
- Ensure that non-functional requirements align with stakeholder expectations and industry standards.

**7. Perform Risk Assessment:**

- Identify potential risks and challenges associated with the development and implementation of the mobile app.
- Assess the impact and likelihood of each risk and develop mitigation strategies.

**8. Validate Requirements:**

- Review and validate requirements with stakeholders to ensure accuracy, completeness, and alignment with project objectives.
- Address any discrepancies or conflicting requirements through negotiation and consensus-building.

**9. Document Requirements Specifications:**

- Create detailed requirements documentation, including functional specifications, use cases, user stories, wireframes, and prototypes.
- Ensure that requirements are well-documented, unambiguous, and traceable throughout the development process.

**10. Iterate and Refine:**

- Continuously gather feedback from stakeholders and end-users throughout the development lifecycle.
- Iterate on requirements based on feedback, changing priorities, and evolving project needs.

**11. Obtain Stakeholder Approval:**

- Obtain formal approval from key stakeholders on the final set of requirements before proceeding to the design and development phase.

By following these steps, we can ensure that the requirements gathering process for our disaster management mobile app is thorough, collaborative, and well-aligned with the needs of stakeholders and end-users.

## **1. Stakeholders and their roles in the Disaster management mobile application**

In a disaster management mobile application, various stakeholders play critical roles in ensuring its effectiveness, adoption, and success. Here are the key stakeholders associated with their roles in the application:

### **1. Emergency Responders:**

- **Constitution:** Emergency responders include firefighters, paramedics, police officers, and other first responders who are directly involved in disaster response and rescue operations.
- **Involvement:** They use the mobile application to receive real-time alerts, access critical information (such as incident locations, hazards, and resource availability), communicate with other responders, and coordinate response efforts.

### **2. Government Agencies:**

- **Constitution:** Government agencies at local, regional, and national levels are responsible for coordinating disaster management and providing support during emergencies.
- **Involvement:** They may use the application to disseminate official alerts and warnings, share situational awareness updates, manage resources (such as shelters and evacuation routes), and coordinate response efforts with other agencies and stakeholders.

### **3. Non-Governmental Organizations (NGOs):**

- **Role:** NGOs play a vital role in providing humanitarian aid, relief services, and support to affected communities during disasters.
- **Involvement:** NGOs may use the application to coordinate volunteer activities, request assistance or donations, disseminate information about relief efforts, and collaborate with government agencies and other organizations involved in disaster response and recovery.

### **4. Citizens and Residents:**

- **Constitution:** Citizens and residents living in disaster-prone areas are directly impacted by emergencies and play a critical role in preparedness, response, and recovery efforts.
- **Involvement:** They use the application to receive alerts and warnings, access information on evacuation routes and shelters, report emergencies or hazards, request assistance, and communicate with responders and other community members.

## 5. **Media Organizations:**

- **Constitution:** Media organizations, including news outlets and journalists, play a crucial role in disseminating accurate information, raising awareness, and shaping public perceptions during disasters.
- **Involvement:** They may use the application to access official updates and alerts, report on disaster events and response activities, and share information with the public to facilitate informed decision-making and community resilience.

## 6. **Technology Providers and Developers:**

- **Role:** Technology providers and developers are responsible for designing, developing, and maintaining the disaster management mobile application.
- **Involvement:** They collaborate with other stakeholders to understand requirements, develop and test the application, provide technical support and updates, and ensure the application's usability, reliability, and security.

## **2. Functional and Non-Functional requirements**

Both functional and non-functional requirements are essential for defining the behavior, performance, and quality characteristics of a system or application. They help ensure that the final product meets the needs of its users and stakeholders while also meeting performance, security, and other quality standards.

### **Functional Requirements:**

Functional requirements describe the specific behaviors and functionalities that a system or software application must perform to meet the needs of its users. ie Functional requirements specify what the system or application should do

#### 1. **User Registration and Authentication:**

- Users should be able to create accounts and log in securely.
- Authentication methods may include email/password, social media login, or biometric authentication.

#### 2. **Emergency Alerts and Notifications:**

- The app should deliver real-time alerts and notifications for various types of disasters based on the user's location.
- Users should be able to customize their alert preferences.

### **3. Geolocation and Mapping:**

- The app should provide mapping functionality to display disaster-prone areas, evacuation routes, shelters, and resources.
- Users should be able to view their current location on the map.

### **4. Emergency Resources and Information:**

- The app should offer access to emergency contacts, procedures, safety tips, and resources.
- Information should be categorized and easily searchable.

### **5. Reporting and Communication:**

- Users should be able to report emergencies and request assistance through the app.
- The app should facilitate two-way communication between users and emergency responders.

### **6. Emergency Preparedness Tools:**

- The app should provide checklists, guides, and educational resources for emergency preparedness and planning.
- Users should be able to create and manage personalized emergency kits.

### **7. Community Support and Coordination:**

- The app should enable users to connect with neighbors, friends, and family members during emergencies.
- Community forums or chat groups should be available for sharing information and providing support.

## **Non-Functional Requirements:**

Non-functional requirements describe the qualities or characteristics that a system or software application must possess but do not directly relate to its functionality. It describes how the system or application should perform

### **1. Performance:**

- The app should have fast response times, even during peak usage periods.
- It should be able to handle a large number of concurrent users.

### **2. Reliability:**

- The app should be highly reliable, with minimal downtime.
- It should have mechanisms in place to recover from failures quickly.

### 3. **Security:**

- The app should adhere to strict security standards to protect user data and privacy.
- It should employ encryption for data transmission and storage.

### 4. **Scalability:**

- The app should be scalable to accommodate increasing user numbers and data volumes.
- It should be able to handle sudden spikes in usage during emergencies.

### 5. **Accessibility:**

- The app should be accessible to users with disabilities, following accessibility guidelines.
- It should support features such as screen readers and voice commands.

### 6. **Usability:**

- The app should have an intuitive user interface and user-friendly navigation.
- It should be easy to understand and use, even for users under stress during emergencies.

### 7. **Interoperability:**

- The app should be able to integrate with external systems such as emergency response systems, weather APIs, and government databases.
- It should adhere to industry standards for data exchange and communication.

### 8. **Compliance:**

- The app should comply with relevant laws and regulations, such as data protection regulations (e.g., GDPR, CCPA).
- It should have clear terms of service and privacy policies.



### **3. Constraints of our disaster management mobile application**

Constraints in a disaster management mobile application refer to limitations, restrictions, or factors that may hinder or affect the development, deployment, or operation of the application.

By identifying and addressing these constraints early in the development process, we aim to mitigate risks, prioritize requirements, and optimize the design and functionality of the disaster management mobile application to better serve the needs of users and communities during emergencies.

1. **Resource Limitations:** Limited budget, time, and personnel can constrain the development and deployment of the application. Adequate resources are necessary for designing robust features, conducting thorough testing, and ensuring timely updates and maintenance.
2. **Technological Constraints:** Compatibility issues, hardware limitations, and network connectivity constraints may affect the functionality and performance of the application. The app should be designed to work across different devices, operating systems, and network conditions, considering factors such as screen size, processing power, and bandwidth.
3. **Regulatory and Compliance Constraints:** Compliance with regulatory requirements, data protection laws, and industry standards is essential for ensuring the security and privacy of user data. The application must adhere to legal and ethical guidelines related to data collection, storage, sharing, and consent.
4. **Geographical Constraints:** The application may need to support users in various geographical locations with different disaster risks, infrastructure, and emergency response systems.
5. **Data Constraints:** Limited access to real-time data, incomplete data sets, and data quality issues can impact the accuracy and reliability of the application. Integration with data sources, data validation mechanisms, and data synchronization strategies are necessary to ensure that the application delivers timely and accurate information to users.
6. **User Constraints:** Users may face constraints such as language barriers, literacy levels, disabilities, or limited access to technology. The application should be accessible, intuitive, and inclusive, with features like multilingual support, assistive technologies, and user-friendly interfaces to accommodate diverse user needs.
7. **Infrastructure Constraints:** Limited infrastructure, power outages, and communication disruptions can affect the availability and reliability of the application during emergencies. The application should have offline capabilities, caching mechanisms, and fallback options to ensure continuous access to critical information and functionality.
8. **Interoperability Constraints:** Integration with existing systems, platforms, and emergency response protocols may be challenging due to interoperability constraints.

Standardization of data formats, APIs, and communication protocols can facilitate seamless integration and interoperability with other systems and stakeholders.