REPUBLIC OF CAMEROON
PEACE-WORK-FATHERLAND
UNIVERSITY OF BUEA
BUEA, SOUTH-WEST REGION
P.O BOX 63.

RÉPUBLIQUE DU CAMEROUN
PAIX-TRAVAIL-PATRIE
UNIVERSITÉ DE BUEA
BUEA, RÉGION DU SUD-OUEST
B.P. 63.

FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING
SOFTWARE ENGINEERING
CEF440: Internet and Mobile Programming

SYSTEM REQUIREMENTS FOR A DISASTER MANAGEMENT MOBILE APPLICATION

Presented by:

GROUP 2

NAMES	MATRICULE
QUINUEL TABOT NDIP-AGBOR	FE21A300
SIRRI THERESIA ANYE	FE21A306
NGONCHI RAMATOU YOLAND	FE21A157
CHE BLAISE NJI	FE21A260
LIMA CHARLES	FE21A225

Course Facilitator:
Dr. NKEMENI Valery, PhD

Date Issued:
21st April 2024

Table of Content

1. Abstract

This document outlines the system requirements for a mobile disaster management application. Traditional disaster management methods often lack efficiency due to manual processes. This project proposes a mobile application to address these challenges by creating a comprehensive platform for disaster preparedness, response, and recovery.

This mobile application aims to leverage the power of mobile technology to improve communication, information sharing, and overall effectiveness in disaster management situations.

2. Introduction

In today's world, natural disasters and emergencies are a constant threat. To effectively manage these situations, we need robust disaster management systems. Traditional methods, often reliant on manual processes and scattered communication channels, struggle to deliver timely and coordinated responses. This project proposes a solution: a Mobile-Based Disaster Management System.

Expected Outcomes

- A multifunctional mobile application supporting all stages of disaster management: preparedness, response, recovery, and mitigation. The application will have user-friendly interfaces catering to both individuals and emergency responders.
- A real-time alerting and notification system delivering critical information during emergencies. Alerts will be customized based on user location and preferences.
- The ability for users to report incidents and request assistance directly through the application. This will facilitate efficient resource allocation and timely response.
- Integration with geospatial data and mapping services, providing users with real-time information about disaster zones, evacuation routes, shelter locations, and other relevant spatial data.
- Features promoting community engagement and collaboration, such as forums, chat rooms, and social media integration. These tools will enable information sharing, peer support, and collective action during disaster response and recovery.

3. System Requirements For A Disaster Management Mobile Application

System requirements define the characteristics and functionalities a disaster management mobile application must possess to meet its objectives. These requirements can be categorized into different types:

- **Functional Requirements:** These define the specific actions the application should be able to perform. In our case, this includes features like real-time alerts, incident reporting, and access to resources.

- **Non-Functional Requirements:** These address overall qualities of the application, like performance, security, and usability. Examples include real-time delivery of alerts and ensuring the application is accessible to diverse users.

- **Business Requirements:** These focus on how the application aligns with the organization's goals. This could involve facilitating information sharing among stakeholders or coordinating response efforts.

- **Technical Requirements:** These specify the technical aspects needed to develop the application, such as integrating with mapping services or implementing multi-language support.

- **Stakeholder and User Requirements:** These capture the specific needs of the application's intended users. Examples include customized alerts based on location and forums for community engagement.

3.1. Functional Requirements

1. **Real-time Alerts and Notifications:**

    - System should deliver critical information like warnings about impending disasters, evacuation orders, and safety instructions.
    - Allow for user preferences to customize alerts based on location severity (e.g., receive alerts only for major disasters or within a specific radius).
    - Multiple notification channels can be used (push notifications, text messages, in-app alerts).

2. **Incident Reporting and Assistance:**

    - Users can report incidents (e.g., flooding, fire damage, injuries) directly through the app.
    - Reporting should include options to capture details like location (using GPS), type of incident, photos/videos (if possible), and any additional relevant information.
    - The system should allow users to request assistance from specific emergency services (police, fire department, ambulance) based on the reported incident.

3. **Geospatial Data and Mapping Services:**

    - Integrate with real-time geospatial data services to display disaster-affected areas on interactive maps.
    - Maps should show key information like evacuation routes, shelter locations, flood zones, and critical infrastructure status.
    - Users' current location should be displayed on the map for better situational awareness.

4. **Community Engagement and Collaboration:**

- Integrate features like forums, chat rooms, and social media feeds to facilitate information sharing and communication among users and stakeholders involved in disaster response.
- Users can share updates, request help, offer assistance, and connect with others in their community during emergencies.
- Allow for anonymous posting if desired to encourage broader participation.

5. **Preparedness and Mitigation Resources:**

    - Provide a centralized repository of educational resources, guides, and checklists to help users prepare for different types of disasters.
    - Content should cover topics like creating emergency plans, assembling disaster kits, and practicing safety measures.
    - Resources can be downloadable for offline access.

6. **Emergency Contacts and Services:**

    - Offer a comprehensive directory of emergency contact information for various services (police, fire department, hospitals, poison control).
    - Allow users to easily call or message emergency services directly from the app.
    - Information should be categorized and searchable for quick access during emergencies.

7. **Offline Capability and Data Sync:**

    - The app should function to some extent even without an internet connection.
    - Users should be able to access critical information like alerts, maps (pre-downloaded versions), and emergency resources offline.
    - The system should queue incident reports and other user-generated data when offline, automatically syncing it with the server when an internet connection is re-established.

8. **Multi-Language Support:**

    - The app interface and content should be available in multiple languages to cater to a diverse user base.
    - Allow users to easily switch between languages based on their preference.
    - Localization should consider cultural nuances to ensure clear and effective communication.

3.2. Non-Functional Requirements

1. **Privacy and Security Measures:**

- **Data Encryption:** Implement strong encryption algorithms to protect user data during transmission and storage. This includes sensitive information like location, incident details, and potentially even personal information used for registration (if applicable).

- **Authentication and Authorization:** Consider user authentication mechanisms (logins, passwords) to restrict access to sensitive functionalities or user data (optional depending on app design).

- **Data Minimization:** Collect only the data essential for app functionality. Avoid unnecessary data collection to minimize privacy concerns.

- **Transparent Privacy Policy:** Clearly outline the app's data collection practices, how user data is used, and user rights regarding their data.

2. **Real-time Delivery of Alerts and Notifications:**

- **Low Latency Communication:** Utilize efficient communication protocols and server infrastructure to minimize delays in delivering critical alerts and notifications. This is crucial for ensuring timely warnings during emergencies.

- **Push Notification Optimization:** Optimize push notification delivery to minimize reliance on a stable internet connection. Explore alternative methods like SMS or local network broadcasts for critical alerts in case of internet outages.
- **Notification Retries:** Implement mechanisms to retry sending notifications if there are initial delivery failures due to network issues.

3. **Accuracy and Reliability of Geospatial Data:**

- **Data Source Validation:** Establish partnerships with reliable sources for geospatial data, such as government agencies or reputable mapping services.

- **Data Refresh Mechanisms:** Implement processes to regularly update geospatial data, especially during emergencies when information about disaster zones and evacuation routes can change rapidly.

- **Data Accuracy Checks:** Integrate data validation techniques to identify and correct any potential inaccuracies in the geospatial data before it's displayed in the app.

4. **Accessibility of the Application:**

- **WCAG Compliance:** Strive for conformance with the Web Content Accessibility Guidelines (WCAG) to ensure the app is accessible to users with disabilities. This includes features like:

    - **Screen Reader Compatibility:** The app interface should be compatible with screen reader software for visually impaired users.
    - **Increased Color Contrast:** Implement sufficient color contrast between text and background elements for better readability by users with visual impairments.
    - **Keyboard Navigation:** Allow users to navigate the app interface using a keyboard instead of a touchscreen, catering to users with motor impairments.

3.3 Business Requirements

1. **Streamlined Coordination Between Emergency Responders:**

    - The app should facilitate seamless communication and information sharing between different emergency response agencies (police, fire department, ambulance, etc.).
    - Real-time incident reporting and updates from the app can help responders coordinate deployment efforts, avoid duplication, and optimize resource allocation.
    - Secure messaging features within the app can enable responders to collaborate and share critical information during emergencies.

2. **Information Sharing Among Users and Stakeholders**

    - The app should create a platform for information exchange between citizens, community organizations, and government agencies.
    - Users can report incidents, share updates on local conditions, and offer or request assistance through the app's communication features.
    - This two-way flow of information can improve situational awareness for both authorities and the public, leading to faster and more effective response efforts.
3. **Centralized Repository of Emergency Contact Information:**

    - Having a readily accessible list of emergency contacts within the app can be crucial during crises.
    - This can include phone numbers, website links, and even location information for essential services like hospitals, shelters, and disaster relief agencies.
    - A centralized repository ensures users have quick access to the right resources in an emergency situation, saving valuable time.

Additional Business Considerations:

- **Reduced Response Times and Improved Efficiency:** Faster information sharing and coordinated response through the app can lead to quicker emergency response times, potentially saving lives and minimizing property

damage.

- **Enhanced Public Safety and Awareness:** Providing real-time alerts and preparedness resources can empower citizens to make informed decisions and take necessary precautions during disasters.

- **Improved Communication and Collaboration:** The app can foster better communication and collaboration between authorities, emergency responders, and the public, leading to a more unified response effort.

- **Data Collection and Analysis:** User-generated data from the app (incident reports, location information) can be valuable for post-disaster analysis and future preparedness efforts.

- **Scalability and Sustainability:** The app's design should consider scalability to accommodate a large number of users during widespread emergencies. It should also have a sustainable funding model to ensure ongoing maintenance and updates.

3.4 Technical Requirements

- **Integration with Geospatial Data and Mapping Services:**
    - The app needs APIs (Application Programming Interfaces) to connect with real-time geospatial data providers. Popular options include Google Maps Platform, HERE Maps, or Open Street Map (OSM).
    - These services offer geospatial data like road networks, landmarks, and boundaries, enabling the creation of interactive maps within the app.

- **Implementation of Offline Capability and Data Sync:**

- The app should leverage offline storage mechanisms like SQLite or local databases to cache critical information (pre-downloaded maps, emergency contacts, basic functionalities).
- Data synchronization protocols need to be implemented to ensure seamless transfer of user data (incident reports, location updates) between the app and the server when an internet connection becomes available.

- **Implementation of Multi-language Support:**

    - The app's user interface (UI) and content must be adaptable to display in multiple languages. This can be achieved through techniques like resource files containing translated text strings.
    - The system should allow users to choose their preferred language from within the app settings.
    - Localization tools might be necessary to ensure proper cultural adaptations in translated content.

Tools

1) UI Design (Figma) : Besides the fact that it is really good and gets the job done, it is very easy to use, free and all team members have some minimum amount of experience with it.

2) Mobile Application UI (React Native) : RN has a low barrier of entry so it is quite easy to pick up by someone with some Javascript or React knowledge no matter how basic. Also, it is cross platform and hence, we won't have to build a separate application for IOS devices. Furthermore, it has a very large community and ecosystem so it is very easy to seek and find help when you need it.

3) Styling UI (SaSS and NativeWind): SaSS is a flavour of CSS while NativeWind is in simple terms, custom tailwind css but for RN. We chose these two so that we can focus less on how our system looks and more on what it actually accomplishes. It helps us to iterate much faster with the look and feel of the application.

4) **Database (MySQL / MongoDB):** The entire team has been working with MSQL right from the start of our program at FET so we are fairly good with it. Also, MYSQl is vertically scalable and structured so suits our needs perfectly in this case. MongoDB is a noSQL database track which is fairly easy to use and understand within a short period of time.

5) Backend Language/Framework (Typescript/NestJs): NestJs is a NodeJs framework for building performant and scalable backend applications(APIS). The fact that it uses typescript also means it provides a high level of type safety for us thus catching a lot of errors at compile time instead of runtime. This makes it very suitable for our use case. More important is the fact that part of the team has some experience working with it so it won't require the entire team to learn an entire framework hence making us move faster.

6) Deployment Backend (Azure Container Apps): We intend to host the final backend as an Azure Container application. This means we'll have to build the app as a docker container in the end before deployment. The reason for this choice is that one of our team members already has a functioning account on Azure and so it makes sense to use it instead of looking for something brand new. Also, it just works and is not too difficult to pick up by other team members. Mobile App: Unfortunately, we will only be able to deploy on Android at this point via APKs as the google play store 10 will require a $25 fee for deployment and even worse than that, the Apple App store requires a $99 charge and doesn't even support APKs. Depending on the direction the other team members chose to go, we might consider these other options even after this course.

7) Documentation: This application will be documented with the help of Swagger documentation with the direction of the OpenApi specification. This is going to be very vital for the developers of the UI as they'll need guidance when consuming the API services that power the application.

3.5 Stakeholders and User Requirements

- **Stakeholders:** These are individuals or organizations that have a vested interest in the success of the application but may not directly use it themselves. They are impacted by the application's effectiveness and functionality.
- **Emergency Responders (Police, Fire Department, Ambulance):**
    - **Needs:** Real-time information on incidents, secure communication channels for coordination, efficient resource allocation based on user reports.
    - **Benefits:** Improved response times, better situational awareness, streamlined collaboration.

- **Government Agencies:**
    - **Needs:** Dissemination of critical information to the public, data collection for post-disaster analysis, centralized platform for resource management.
    - **Benefits:** Enhanced public safety and awareness, improved coordination between emergency services, informed decision-making for future preparedness efforts.

- **Non-profit Organizations and NGOs:**
    - **Needs:** Platform for sharing resources and connecting with affected communities, ability to coordinate volunteer efforts.
    - **Benefits:** Increased visibility and outreach, streamlined resource distribution, efficient deployment of volunteers.

Users (General Public): These are the individuals who will directly interact with the mobile application to access its features and functionalities.

- **Needs:** Timely and location-specific alerts, ability to report incidents and request assistance, access to emergency resources and evacuation information.
    - **Benefits:** Improved preparedness, increased sense of security, ability to connect with others during emergencies, access to critical resources during disasters.

- **Additional Considerations:**
    - **Accessibility:** Ensure the application caters to users with disabilities (visual impairments, hearing impairments, etc.) through features like text-to-speech functionality or closed captions.
    - **Multilingual Support:** The application should be available in multiple languages to reach a diverse user base.

4. Requirement Gathering

Developing a successful disaster management mobile application hinges on understanding the needs of its users and stakeholders. This document outlines a comprehensive approach to requirement gathering for this crucial technology.

Requirement Gathering Techniques:

We employed various techniques to gather requirements from stakeholders and users:

- **Stakeholder Workshops:** Facilitate discussions to identify critical needs, functionalities, and success metrics.
- **User Interviews and Surveys:** Gather user feedback on their expectations from the application and potential pain points during disaster scenarios.
- **Focus Groups:** Conduct moderated discussions with diverse user groups to explore needs, preferences, and accessibility considerations.
- **Scenario Mapping:** Simulate potential disaster situations and brainstorm functionalities that would be helpful in those scenarios.
- **Competitive Analysis:** Review existing disaster management applications to identify best practices and potential areas for improvement.

Documenting Requirements:

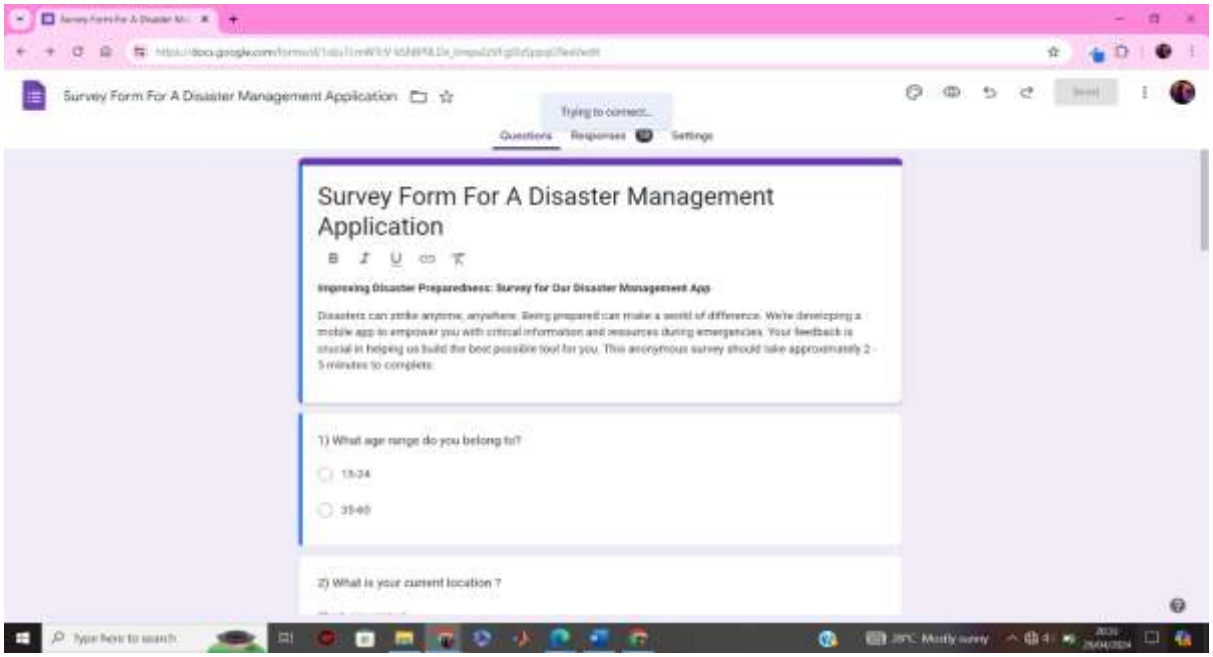
Capture all gathered requirements in a clear and organized manner. This involved:

- **Use Cases:** Describe specific user interactions with the application to achieve particular goals.
- **User Stories:** Briefly describe functionalities from the user's perspective (e.g., "As a resident, I want to receive real-time alerts about approaching storms").
- **Functional Requirements:** Define the specific actions the application should be able to perform (e.g., display interactive maps, allow users to report incidents).
- **Non-Functional Requirements:** Specify performance, usability, security, and accessibility considerations.

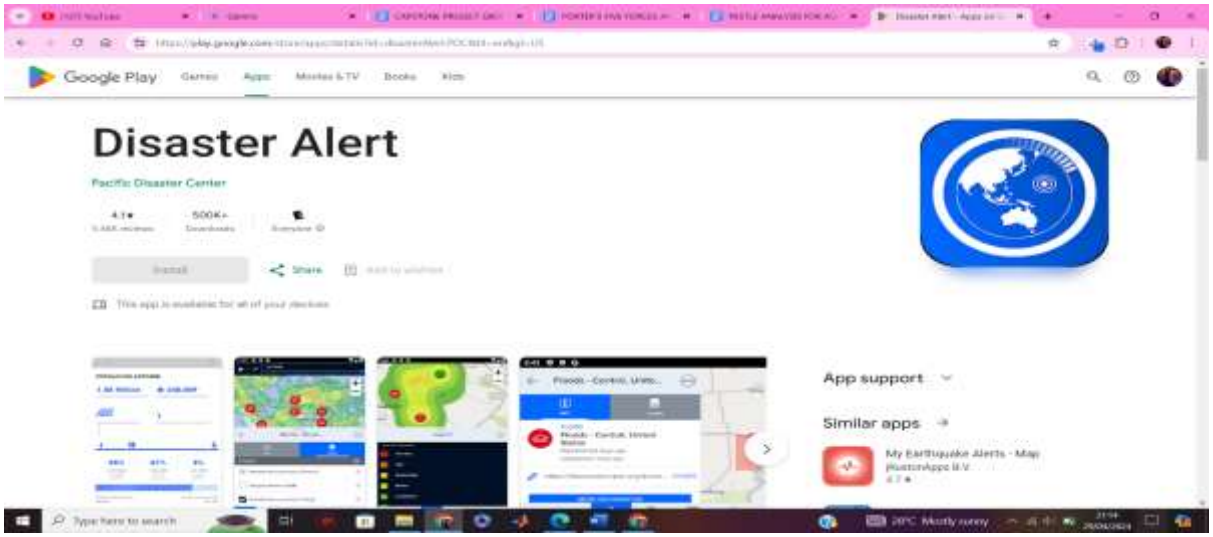
Prioritization:

Not all requirements have equal importance. So we employed strategies like survey forms, interviews, and research .

Sample Survey Form:



Sample Research Maps:


Google Play
Games
Apps
Movies & TV
Books
Kids
Disaster Alert
Pacific Disaster Center
4.1★
500K+
Share
App support
Similar apps
My Earthquake Alerts - Map
Type here to search

5. Conclusion

Effective disaster management hinges on timely communication, coordinated response, and empowered communities. Traditional methods often struggle to meet these demands. This project proposes a Mobile-Based Disaster Management System to address these challenges.

The proposed system leverages the power of mobile technology to create a comprehensive platform for all stages of disaster management: preparedness, response, and recovery. The mobile application will equip users with real-time alerts, incident reporting capabilities, access to emergency resources, and communication channels with authorities and other stakeholders during crises.

This project outlines various requirements, categorized by function, target audience, technical considerations, and business goals. By meticulously addressing these requirements, we can develop a robust and user-friendly mobile application that empowers individuals, communities, and emergency responders. This application will ultimately contribute to a more prepared, informed, and connected society during disasters, fostering a more resilient future in the face of natural hazards and emergencies.

6. References

1) Enhanced 911 http://library.ema.gov.au/LIBERO/WebOpac.cls

2) SMS Alerts http://www.egov4dev.org/mgovapplic.htm

3) SMS emergency broadcasting http://www.egov4dev.org/mgovapplic.htm

4) Mobile devices to government authorities www.jsce-int.org/Report/report/flood_euro.pdf

5)The value of GSM in a natural disaster situation. http://www.gsmworld.com/news/press_2000/press_releases_43.shtml

6) New emergency response system helps fight disasters (EGERIS) http://www.innovations-report.com/html/reports/communication_media/report-29094.html

7)Disaster Emergency Management Information System (AFAYBIS) http://www.isprs.org/istanbul2004/comm4/papers/339.pdf