THE UNIVERSITY OF BUEA

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

CEF 440: INTERNET PROGRAMMING AND MOBILE PROGRAMMING

Group 22

REQUIREMENTS ANALYSIS

Task 3

Name	Matricule	Option
1. Enow Myke-Austine Eta	FE21A183	Software
2. Mokfembam Fabrice Kongnyuy	FE21A240	Software
3. Ndangoh Boris Bobga	FE19A072	Network
4. Ndong Henry Ndang	FE21A248	Software
5. Niba Verine Kajock	FE21A267	Network
6. Takem Jim-Rawlings E.	FE21A309	Software

COURSE INSTRUCTOR: Dr. NKEMENI VALERY

Contents

1. INTRODUCTION

The foundation for any successful mobile application is a comprehensive understanding of its requirements. In the previous report, we meticulously gathered requirements for our disaster management system through a multi-pronged approach. We delved into existing documentation, conducted citizen interviews and brainstorming sessions, and analyzed stakeholder needs. This rich tapestry of information has yielded a well-defined set of functional and non-functional requirements that will guide the system's development.

However, requirements alone are just the first step. To translate these needs into a tangible and effective mobile application, we must embark on the critical stage of requirements analysis. This report delves into the heart of **requirements analysis**, meticulously dissecting the gathered requirements to:

- **Identify relationships and dependencies:** We'll explore how different requirements interconnect and influence each other.
- **Prioritize functionalities:** Not all requirements hold equal weight. Here, we'll establish a clear hierarchy, ensuring the most critical features are addressed first.
- **Identify potential conflicts or inconsistencies:** A keen eye will be cast upon the requirements to uncover any potential contradictions or areas requiring further clarification.
- **Feasibility assessment:** We'll evaluate the technical and resource feasibility of implementing each requirement, ensuring the final product aligns with practical constraints.
- **Refine and document requirements:** The analysis will lead to the refinement of some requirements and the creation of a clear, well-structured document for the development team.

Through rigorous requirements analysis, we transform the raw materials of requirements into a blueprint for a robust and user-centric disaster management system. This refined blueprint will serve as a guiding light for the development team, ensuring the final application effectively addresses the needs of stakeholders and empowers communities to navigate the challenges of disasters with greater preparedness, resilience, and ultimately, safety.

2. TOOLS AND METHODS FOR REQUIREMENTS ANALYSIS

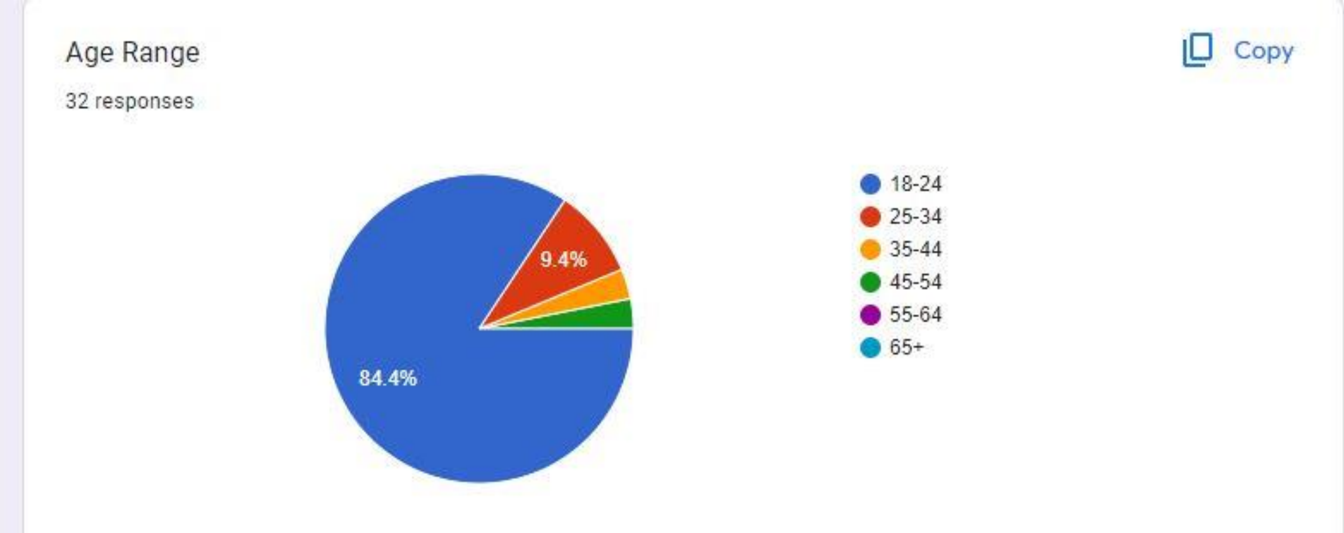
I. GOOGLE FORMS

While Google Forms wasn't directly involved in the requirements analysis stage, it played a crucial role in the initial phase of our project. We leveraged Google Forms to create a web survey for data collection. This survey served as a valuable tool for gathering requirements from a broad range of stakeholders, including citizens and disaster response personnel. This platform offers a robust suite of built-in analytic tool that proved instrumental in analyzing the survey data. These tool allowed us to:

- **Identify Trends:** By visualizing responses in charts and graphs, we could easily identify recurring themes and patterns in user feedback. This helped us pinpoint functionalities most desired by the user base.

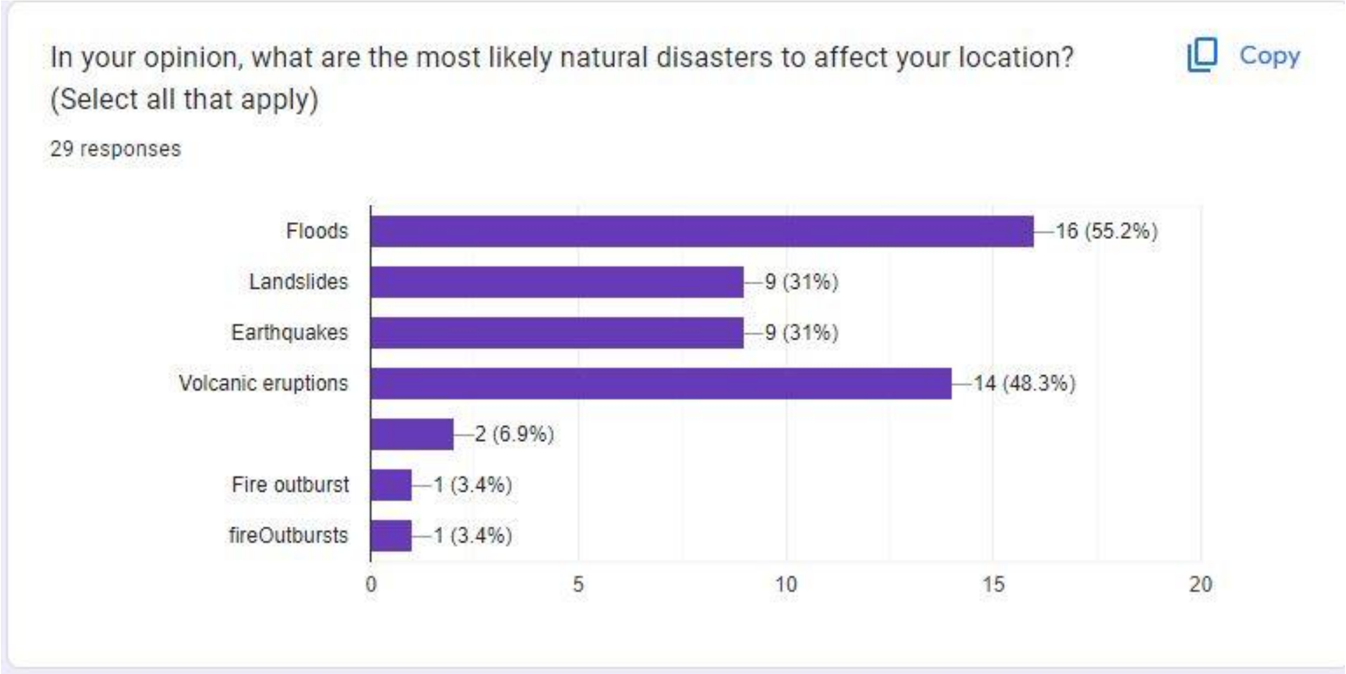
- **Gauge User Sentiment:** Analyzing the answer choices and open-ended responses provided insights into user anxieties and concerns. This information was critical for understanding user needs and ensuring the disaster management system effectively addresses their pain points.
- **Prioritize Requirements:** By analyzing the frequency of specific feature requests and the strength of user sentiment surrounding them, we could prioritize functionalities based on their perceived importance to the target audience. This data-driven approach ensured the development effort focused on features with the most significant impact.

Below are some of the images to show the analysis



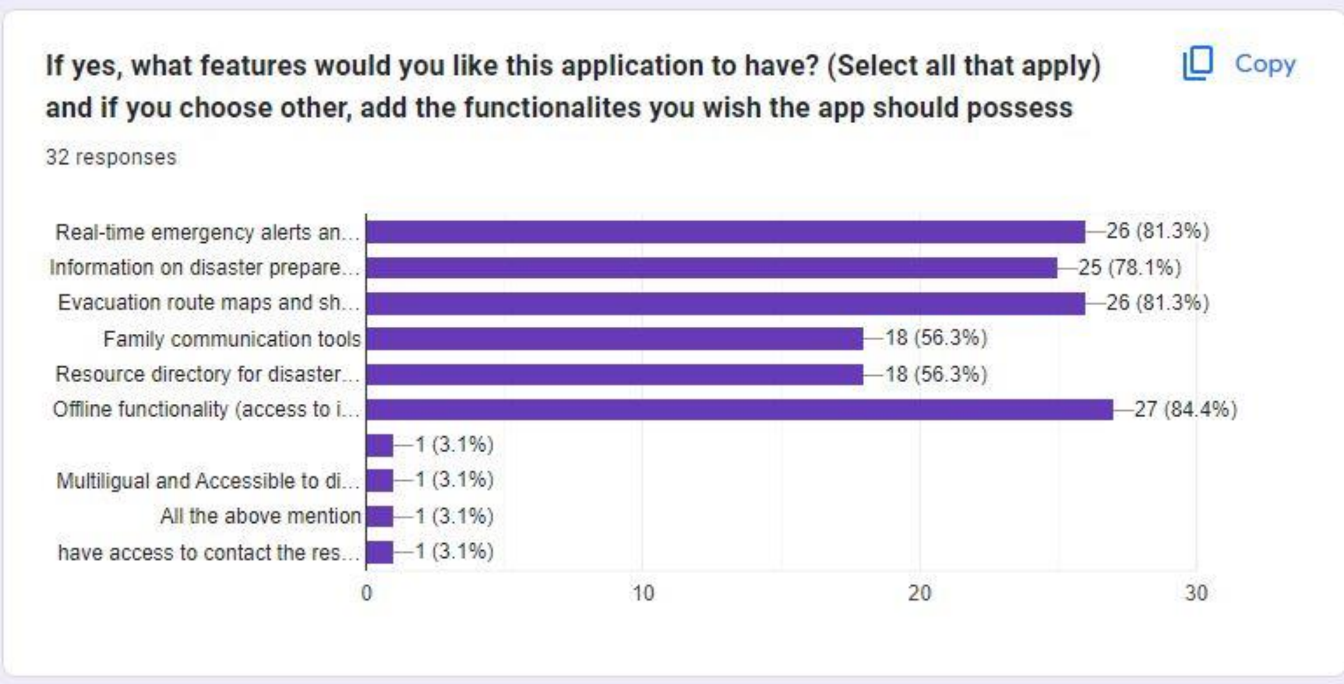
Age Range

32 responses



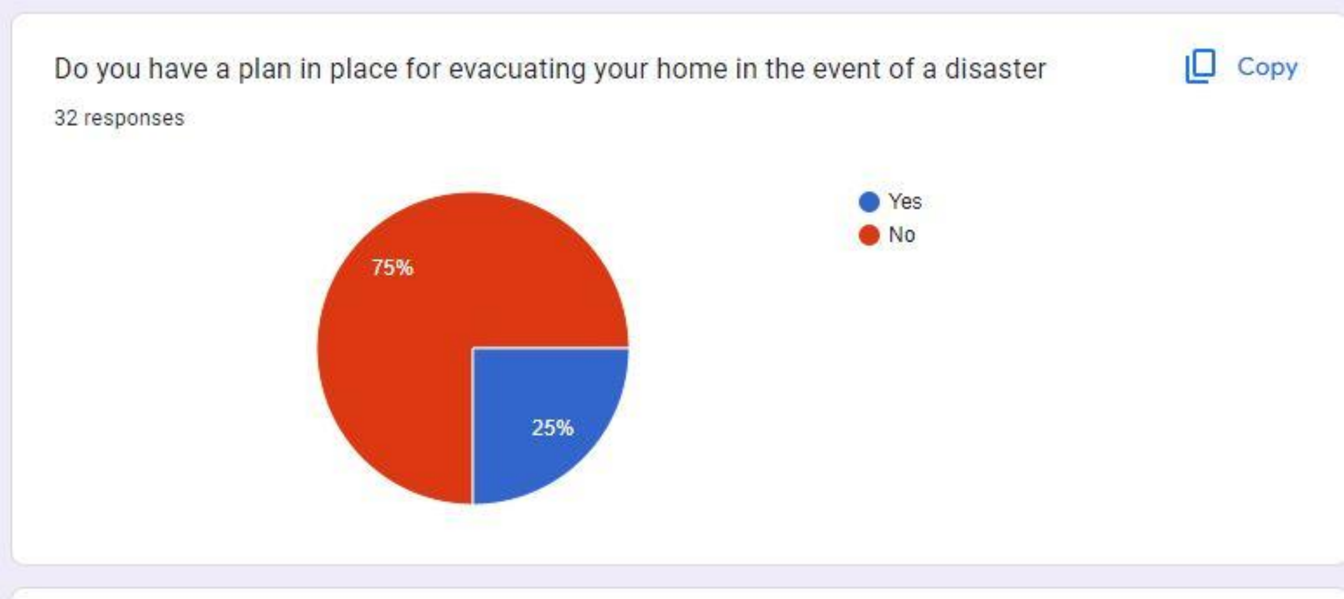
In your opinion, what are the most likely natural disasters to affect your location? (Select all that apply)

29 responses



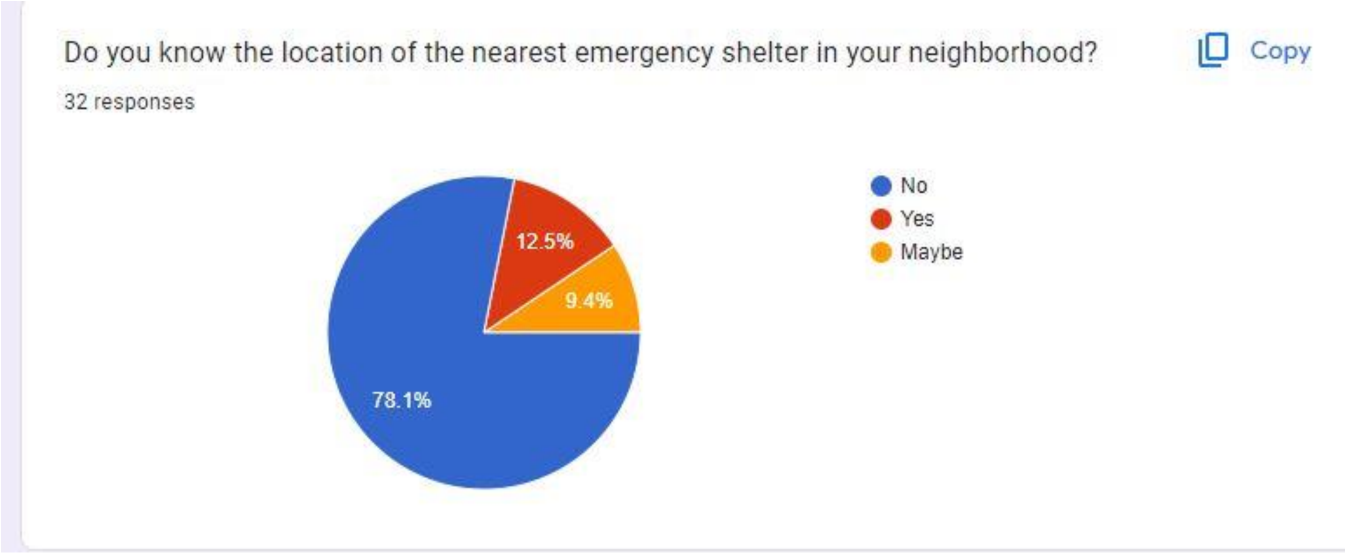
If yes, what features would you like this application to have? (Select all that apply) and if you choose other, add the functionalites you wish the app should possess

32 responses



Do you have a plan in place for evacuating your home in the event of a disaster

32 responses



Do you know the location of the nearest emergency shelter in your neighborhood?

32 responses

II. REQUIREMENTS EVALUATION BY BRAINSTORMING

After gathering all our requirements, we evaluated all them by brainstorming using the SMART analysis to ensure the requirements feasibility and alignment with project objectives. Evaluation was also done to see whether or not to consider these requirements during our project development.

<u>SMART Analysis</u>

1. Specific (S):
   a. Ensure that the requirement is clear, concise, and well-defined.
   b. Ask: Is the requirement specific enough? Can it be further refined?
2. Measurable (M):
   a. Make sure the requirement can be quantified or measured.
   b. Ask: Can we define success criteria or metrics related to this requirement?
3. Attainable (A):
   a. Evaluate whether the requirement is feasible within the project's constraints (time, budget, resources).
   b. Consider technical limitations and available expertise.
   c. Ask: Is it realistic to achieve this requirement given our current capabilities?
4. Relevant (R):
   a. Assess whether the requirement directly contributes to project objectives.
   b. Avoid including nice-to-have features that don't align with the system's purpose.
   c. Ask: Does this requirement address a critical need or add significant value?
5. Time-Bound (T):
   a. Set a timeframe for achieving the requirement.
   b. Consider deadlines, project milestones, and overall project duration.
   c. Ask: Can we complete this requirement within the desired timeframe?

Now, in order to analyze these requirements, we need to know what we have gathered so far. Below is a list of all the requirements we got from our requirements gathering process:

1. **User Feedback and Support**: Users wanted a feature that enables them to provide feedback regarding the disaster management system, and, they expressed the need for support channels to address any concerns or worries they may have during system's usage.
2. **Offline Functionality**: Users emphasized the necessity for the disaster management system to be accessible even in scenarios where internet connectivity is unavailable. Offline functionality could include offline access to maps, emergency contact information, and saved user data.
3. **Multi-language Support**: Recognizing the diverse linguistic preferences within the community, users requested support for multiple languages within the disaster management system. This requirement ensures that information and instructions provided by the system are comprehensible to all users, regardless of their language proficiency.
4. **Report Disaster**: Users expressed the need for a feature that allows them to report the occurrence of a disaster promptly. This requirement emphasizes the importance of enabling users to contribute to the collective awareness of disaster situations, facilitating timely response and coordination efforts by relevant authorities.
5. **Evacuation Routes**: Users emphasized the importance of having access to clear and up-to-date evacuation routes in the event of a disaster. This requirement underscores the critical role of the system in providing life-saving information to users, enabling them to navigate safely to designated evacuation points and minimize potential risks during evacuation procedures.
6. **Request Help**: Users desired a feature that enables them to request assistance or support from relevant authorities or community responders during emergencies. This requirement emphasizes the importance of communication and coordination between users in distress and the appropriate response teams, enhancing overall emergency response efficiency and effectiveness.
7. **Real-time Alerts and Notifications**: Users expressed the need for timely alerts and notifications regarding potential or ongoing disaster situations. This requirement emphasizes the importance of real-time communication and situational awareness, enabling users to stay informed and take appropriate actions to ensure their safety and well-being.
8. **Receive Alerts Offline (SMS)**: Recognizing the limitations of internet connectivity during emergencies, users specifically requested the ability to receive alerts and notifications via SMS. This requirement ensures that critical information reaches them promptly regardless of their online status.
9. **Usability**: Users highlighted the importance of a user-friendly interface and intuitive design in the disaster management system. This requirement emphasizes the need for a system that is easy to navigate, understand, and operate, especially during high-stress situations. Ensuring usability enhances user adoption and effectiveness of the system in supporting disaster preparedness and response efforts.
10. **Resource Management:**
    - Effective resource management involves pre-positioning essential supplies and equipment, ensuring readiness to respond promptly to disaster events.
    - Proper resource allocation facilitates the efficient distribution of aid and support to affected areas during the recovery phase, helping communities rebuild and restore normalcy.
    - Strategic resource management contributes to long-term resilience by allocating resources towards mitigation efforts, such as infrastructure improvements and risk reduction measures.
11. **Communication:**
    - Clear communication channels enable rapid dissemination of information and instructions during emergency response operations, enhancing coordination and situational awareness.
    - Continuous communication facilitates ongoing updates and support services for affected communities during the recovery phase, fostering resilience and community engagement.

- Effective communication campaigns raise awareness about disaster risks and preparedness measures, empowering individuals and communities to take proactive steps to reduce vulnerabilities.

12. **Geospatial Data and Mapping:**
    Geospatial data and mapping technologies provide critical situational awareness for emergency responders, helping them identify impacted areas, assess damage, and prioritize response efforts.
13. **Incident Reporting:**
    Incident reporting mechanisms enable individuals to report emergencies and provide essential information to responders, facilitating rapid assessment and deployment of resources during the response phase.
14. **Damage Assessment:**
    Damage assessment tools and techniques support the evaluation of infrastructure damage and inform decision-making processes for resource allocation and emergency response efforts.
15. **Disaster Detection:**
    Early detection of potential disasters allows for timely intervention and mitigation measures to reduce the impact and severity of the event, ultimately enhancing community resilience and safety.

Now we perform the SMART analysis on all the requirements to evaluate whether or not they are feasible, attainable and non-redundant as shown below:

1. **User Feedback and Support:**
    - **Specific:** Enable users to provide feedback and access support channels.
    - **Measurable:** Track the number of user feedback submissions and support requests.
    - **Achievable:** Feasible to implement feedback forms and support channels.
    - **Relevant:** Relevant for improving system usability and addressing user concerns.
    - **Time-Bound:** Implement within the next development cycle.
2. **Offline Functionality:**
    - **Specific:** Provide offline access to maps, emergency contacts, and saved data.
    - **Measurable:** Test offline functionality in various scenarios.
    - **Achievable:** Feasible with proper design and data synchronization.
    - **Relevant:** Critical for users during emergencies.
    - **Time-Bound:** Ensure offline features are available in the next release.
3. **Multi-language Support:**
    - **Specific:** Support multiple languages for user interfaces and alerts.
    - **Measurable:** Count the number of supported languages.
    - **Achievable:** Possible through localization efforts.
    - **Relevant:** Addresses diverse user needs.
    - **Time-Bound:** Implement language support by the end of the quarter.
4. **Report Disaster:**
    - **Specific:** Enable users to report disaster incidents.
    - **Measurable:** Track the number of reported incidents.
    - **Achievable:** Feasible with incident reporting forms.
    - **Relevant:** Vital for situational awareness and response coordination.
    - **Time-Bound:** Available in the next system update.
5. **Evacuation Routes:**
    - **Specific:** Provide up-to-date evacuation routes.
    - **Measurable:** Validate accuracy of route information.
    - **Achievable:** Possible through geospatial data integration.
    - **Relevant:** Critical for user safety during disasters.
    - **Time-Bound:** Ensure routes are available before the next disaster season.

6. **Request Help:**
    - **Specific:** Allow users to request assistance.
    - **Measurable:** Track help requests.
    - **Achievable:** Implement a help button or chat feature.
    - **Relevant:** Essential for user safety and resource allocation.
    - **Time-Bound:** Available in the next system update.
7. **Real-time Alerts and Notifications:**
    - **Specific:** Provide timely alerts during disasters.
    - **Measurable:** Measure alert delivery time.
    - **Achievable:** Feasible with real-time communication channels.
    - **Relevant:** Critical for situational awareness.
    - **Time-Bound:** Ensure real-time alerts are operational within a month.
8. **Receive Alerts Offline (SMS):**
    - **Specific:** Enable SMS alerts.
    - **Measurable:** Confirm successful SMS delivery.
    - **Achievable:** Possible through SMS gateways.
    - **Relevant:** Addresses connectivity limitations.
    - **Time-Bound:** Implement SMS alerts within two weeks.
9. **Usability:**
    - **Specific:** Ensure an intuitive user interface.
    - **Measurable:** Conduct usability testing and gather feedback.
    - **Achievable:** Achievable through user-centered design.
    - **Relevant:** Enhances user adoption and effectiveness.
    - **Time-Bound:** Continuously improve usability throughout development.
10. **Resource Management:**
    - **Specific:** Efficiently manage resources for each phase.
    - **Measurable:** Monitor resource allocation and utilization.
    - **Achievable:** Requires resource planning and coordination.
    - **Relevant:** Critical for disaster response effectiveness.
    - **Time-Bound:** Implement resource management features by the end of the year.
11. **Communication:**
    - **Specific:** Establish communication channels for each phase.
    - **Measurable:** Evaluate communication effectiveness.
    - **Achievable:** Requires protocols and tools.
    - **Relevant:** Essential for coordination and information sharing.
    - **Time-Bound:** Ensure communication channels are operational before the next disaster event.

12. **Geospatial Data and Mapping:**

- **Specific:** Implement geospatial data integration and mapping technologies.
- **Measurable:** Evaluate accuracy and coverage of mapped data.
- **Achievable:** Feasible through GIS tools and APIs.
- **Relevant:** Critical for situational awareness during disaster response.
- **Time-Bound:** Ensure geospatial features are operational within six months.

13. **Incident Reporting:**

- **Specific:** Provide a mechanism for users to report emergencies.
- **Measurable:** Track the number of incident reports submitted.
- **Achievable:** Implement an incident reporting form or button.
- **Relevant:** Facilitates rapid assessment and resource deployment.
- **Time-Bound:** Available in the next system update.

14. **Damage Assessment:**

- **Specific:** Develop tools for assessing infrastructure damage.
- **Measurable:** Evaluate accuracy and efficiency of damage assessment.
- **Achievable:** Requires collaboration with experts and field data collection.
- **Relevant:** Guides resource allocation and recovery efforts.
- **Time-Bound:** Operational within three months.

15. **Disaster Detection:**

- **Specific:** Implement early detection mechanisms (e.g., sensors, data analytics).
- **Measurable:** Monitor detection accuracy and response time.
- **Achievable:** Requires data integration and predictive models.
- **Relevant:** Mitigates impact by allowing timely intervention.
- **Time-Bound:** Deploy detection systems within a year.

Based on the SMART ANALYSIS, here's a list of requirements we will include in our project development:

1. **User Feedback and Support**: Essential for improving user satisfaction and system effectiveness.
2. **User Registration and Authentication**: The application should allow users to register and create an account.
3. **Offline Functionality**: Crucial for ensuring accessibility in areas with limited connectivity.
4. **Multi-language Support**: Enhances inclusivity and usability for diverse user demographics.
5. **Evacuation Routes**: Critical for user safety and effective disaster response.
6. **Request Help**: Improves communication and coordination between users and response teams.
7. **Real-time Alerts and Notifications**: Vital for keeping users informed and safe during emergencies.
8. Receive Alerts Offline (SMS): Addresses communication challenges during emergencies.
9. **Resource Management**: Essential for efficient allocation of resources across all disaster phases.
10. **Communication**: Clear communication channels are crucial for effective disaster response, recovery, and mitigation.
11. **Geospatial Data and Mapping**: Enhances situational awareness and decision-making for responders.
12. **Incident Reporting**: Users should be able to report incidents and request assistance directly through the mobile application.
13. **Usability**: Ensures user adoption and effectiveness of the system, especially in high-stress situations
14. **Performance:** Ensuring the app operates efficiently, with fast loading times and responsiveness is crucial for providing timely information during emergencies.
15. **Security:** Protecting user data, ensuring secure communication channels, and guarding against potential cyber threats are essential for maintaining trust and integrity.
16. **Reliability:** Users must trust the system to provide accurate and timely information consistently, especially during emergencies.

17. **Scalability:** The system should be capable of handling increased demand during emergencies without significant degradation in performance.
18. **Maintainability**: The application should be designed for ease of maintenance and updates to address bugs, add features, or adapt to changing requirements.
19. **Accessibility:** The application should be accessible to users with disabilities, complying with accessibility standards to ensure equal access to information and functionality.
20. **Interoperability**: The application should be able to seamlessly integrate with other systems and platforms used by emergency responders and agencies.

3. CATEGORISING OUR REQUIREMENTS

I. FUNCTIONAL REQUIREMENTS

Functional requirements of a system, in simple terms, are the specific tasks and features that the system needs to perform to meet the needs of its users. They define what the system is supposed to do and how it should behave under certain conditions.

The functional requirements of our disaster management system are listed below:

i. **User Registration and Authentication**: The application should allow users to register and create an account. It should support secure authentication mechanisms to protect user's personal data like email, password, name etc.

ii. **Real-Time Alerts and Notifications**: The application should incorporate a system to send real-time alerts and notifications to users about impending disasters, evacuation orders, and other relevant information. These alerts should be customizable based on users' geographic locations and preferences.

iii. **Incident Reporting**: Users should be able to report incidents and request assistance directly through the mobile application. The system should support different types of incident reports, including text, photos, and videos.

iv. **Emergency Resource Access**: The application should provide access to emergency resources such as first aid guides, emergency contact numbers, and safety tips.

v. **Communication with Authorities**: The application should facilitate communication between users and authorities. This could include features like direct messaging, emergency calls, and request tracking.

vi. **Geospatial Data Integration**: The application should integrate with geospatial data and mapping services to provide users with interactive maps displaying real-time information about disaster-affected areas, evacuation routes, shelter locations, and other relevant spatial data.

vii. **Community Engagement Features**: The application should include features to promote community engagement and collaboration, such as forums, chat rooms, and social media integration. These tools should facilitate information sharing, peer support, and collective action among users and stakeholders involved in disaster response and recovery efforts. The application to should also permit users with resources to become emergency volunteers.

viii. **Data Privacy and Security**: The application should ensure the privacy and security of user data. This includes complying with relevant data protection regulations and implementing appropriate security measures.

ix. **Evacuation Routes**: Users emphasized the importance of having access to clear and up-to-date evacuation routes in the event of a disaster. This requirement underscores the critical role of the system in providing life-saving information to users, enabling them to navigate safely to designated evacuation points and minimize potential risks during evacuation procedures.

x. **Request Help**: Users desired a feature that enables them to request assistance or support from relevant authorities or community responders during emergencies. This requirement emphasizes the importance of communication and coordination between users in distress and the appropriate response teams, enhancing overall emergency response efficiency and effectiveness.

xi. **Offline Functionality**: Considering that disasters can disrupt internet connectivity, the application should provide essential features in offline mode, such as viewing downloaded emergency guides or maps and community chat.

xii. **Multilingual Support**: To ensure the application is usable by a diverse range of users, it should support multiple languages.

xiii. **User Feedback and Support**: The application should provide mechanisms for users to give feedback and get support. This could include a help center, FAQ section, and customer support contact.

II. NON-FUNCTIONAL REQUIREMENTS

To ensure the effectiveness of our disaster management system, several non-functional requirements are crucial. These non-functional requirements address how the system should perform or behave rather than what specific features it offers.

i. **Usability**: The mobile application should be intuitive and easy to use, especially during high-stress situations such as emergencies. The system should be easy to use without prior experience or training, for people with varying technical skills and should support multiple languages.

ii. **Performance**: The system should be able to handle a large number of concurrent users during emergencies without significant degradation in performance. Response times for critical functionalities such as alert delivery and incident reporting should be kept minimal, even under heavy load conditions.

iii. **Security**: The system should adhere to industry best practices for security to protect users' personal information and sensitive data. Secure communication protocols should be used to transmit data between the mobile application and backend servers. Access controls and authentication mechanisms should prevent unauthorized access to system resources.

iv. **Reliability**: The system should be highly reliable and available at all times, especially during disasters and emergencies. Measures should be in place to ensure continuous operation, including redundancy, failover mechanisms, and disaster recovery plans.

v. **Scalability**: this is the ease with which users are added to the system. The system should be able to handle a large influx of users concurrently during disasters without a decrease in performance.

vi. **Maintainability**: The system should be easy to update and maintain over time to adapt to changing technologies and disaster risks.

vii. **Accessibility**: The system should be accessible for people with disabilities, including features for visually impaired and hearing-impaired users.

viii. **Interoperability**: The system should be interoperable with existing disaster management systems and external services. Integration interfaces should adhere to industry standards to facilitate seamless data exchange with other systems and agencies.

4. PRIORITISING REQUIREMENTS

Prioritizing the requirements is crucial to ensure the most critical needs are addressed first and determining which requirements should be implemented first and which can be addressed later. By analyzing the urgency, impact, and complexity of the identified requirements, the requirements were prioritized as follows:

i. **High Priority:**

1. Real-time Alerting and Notification System: Critical for user safety and response effectiveness during emergencies.
2. Ability to Report Incidents and Request Assistance: Empowers users and facilitates coordination with responders, crucial for effective disaster response.
3. Integration with Geospatial Data and Mapping Services: Provides essential spatial information for decision-making, enhancing situational awareness and resource allocation.
4. Communication with Authorities: enables users and authorities to communicate with each other, this will ease disaster response and assistance. This is very crucial for effective response and recovery.
5. Offline Functionality: Ensures accessibility in areas with limited connectivity, and is very crucial as during emergencies actions needs to be carried out without the necessity of internet connection.

ii. **Medium Priority:**

1. Multifunctional Mobile Application: Supports all stages of disaster management, improving user experience and system usability.
2. Community Engagement and Collaboration Features: Fosters community resilience and information sharing among users and stakeholders.
3. User Feedback and Support: Enhances user satisfaction and system effectiveness through user input and support mechanisms.

iii. **Low Priority:**

1. Multi-language Support: Enhances inclusivity and usability but may not be critical during emergency situations.
2. Receive Alerts Offline (SMS): Addresses communication challenges during emergencies but may overlap with other alert/notification features.

5. VALIDATION OF REQUIRMENTS

Having meticulously gathered requirements from diverse stakeholders through various methods, we now enter the crucial stage of requirements validation. This process serves to verify that the captured requirements accurately reflect the stakeholders' needs and expectations for the disaster management system. Through rigorous validation, we ensure we are building the **correct system**, as opposed to simply building a system correctly. Successful validation paves the way for accurate documentation and serves as a solid foundation for the development phase.

Requirements Validation Checks

Requirements validation employs various checks to ensure a comprehensive and well-defined set of specifications. Here's a breakdown of the key checks performed:

- **Completeness Check:** This verifies if all the necessary requirements are documented and no crucial functionalities are missing.
- **Consistency Check:** This ensures that the requirements don't contradict each other and there are no inconsistencies between different parts of the specification.
- **Validity Check:** This confirms that the requirements are realistic, achievable with the available technology and resources, and aligned with the overall project goals.
- **Realism Check:** This assesses if the requirements are practical and can be implemented within the given constraints.
- **Ambiguity Check:** This eliminates any vagueness or unclear language in the requirements, ensuring they are well-defined and easy to interpret.

Requirements Validation Techniques

We employ a multi-pronged approach to requirements validation, encompassing the following techniques:

1. **Reviewing the Requirements:** This involves a thorough review of the requirements by the team alongside key stakeholders. This collaborative effort aims to identify any inconsistencies (consistency check), ambiguities (ambiguity check), or missing requirements (completeness check) within the documented needs.
2. **Stakeholder Feedback:** The requirements are presented to end-users and other relevant stakeholders for their feedback. This ensures the system aligns with their needs and expectations (validity check). Stakeholder input is crucial for moving forward with a clear understanding of the project's goals.
3. **Use Cases:** Use cases are developed based on the requirements. This technique helps us understand how the system will be used and highlights potential issues or missing requirements (completeness check). By analyzing use cases, we can ensure the system caters to various user roles and scenarios (realism check).
4. **Test Case Generation:** We ensure the requirements documented in the Software Requirements Specification (SRS) are testable. This involves conducting tests aimed at revealing errors within each requirement (validity check). By creating testable requirements, we can verify their functionality and identify potential problems early on.
5. **Walkthrough:** This method focuses on facilitated discussions between stakeholders as they review the system's requirements. The walkthrough aims to identify potential gaps in the planned system (completeness check). Through discussion and collaboration, stakeholders can assess the feasibility of requirements (realism check) and reach agreements on the final specifications (consistency check).

By employing these techniques and addressing the different validation checks, we can ensure a comprehensive and accurate set of requirements that forms the bedrock for a successful disaster management system.

6. CONCLUSION

Through meticulous analysis and validation, we've transformed raw requirements into a refined blueprint: the Software Requirements Specification (SRS) document (below). This SRS serves as the foundation for the development phase, ensuring the final disaster management system effectively addresses stakeholder needs and empowers communities.

The analysis employed various techniques (user feedback, brainstorming, SMART analysis) to capture comprehensive requirements. Rigorous validation (stakeholder feedback, test case generation) guaranteed we're building the right system, addressing inconsistencies and laying a solid foundation for development.

This requirements analysis and the resulting SRS pave the way for a disaster management system that is more than just technology. It's a beacon of hope and resilience, empowering communities and equipping users with the knowledge and tools to navigate disasters and emerge stronger. As we embark on the development phase, we do so with a clear vision and a commitment to building a system that truly makes a difference.

7. SOFTWARE REQUIREMENT SPECIFICATION

1. Introduction

This document outlines the Software Requirements Specification (SRS) for a mobile disaster management application. The application aims to empower communities by providing them with critical information, communication channels, and resources to prepare for, respond to, and recover from disasters.

Through a comprehensive requirements gathering process that included user surveys, stakeholder interviews, and brainstorming sessions, we have identified a well-defined set of functional and non-functional requirements. This SRS translates these requirements into a clear and concise blueprint for the development team.

2. System Description

The disaster management mobile application will be a user-friendly platform available for download on various smartphone operating systems. The application will prioritize critical features for emergency situations while offering functionalities to support all phases of disaster management (preparedness, response, and recovery).

3. Requirements

3.1 Functional Requirements

- **User Registration and Authentication:** The application should allow users to register and create an account for secure access.
- **Real-Time Alerts and Notifications:** The application should deliver real-time alerts and notifications to users regarding impending disasters, evacuation orders, and other relevant information. These alerts should be customizable based on user location and preferences.

- **Incident Reporting:** Users should be able to report incidents and request assistance directly through the application. The system should support various reporting methods, including text, photos, and videos.
- **Emergency Resource Access:** The application should provide access to crucial emergency resources, such as first-aid guides, emergency contact information, and safety tips.
- **Communication with Authorities:** The application should facilitate two-way communication between users and authorities. This could include features like direct messaging, emergency calls, and request tracking.
- **Geospatial Data Integration:** The application should integrate with geospatial data and mapping services to display real-time information about disaster-affected areas. This includes evacuation routes, shelter locations, and other relevant spatial data.
- **Community Engagement Features:** The application should promote community engagement and collaboration through forums, chat rooms, and social media integration. These tools should enable information sharing, peer support, and collective action during disaster response and recovery efforts. Users with resources should be able to register as volunteers.
- **Data Privacy and Security:** The application should ensure the privacy and security of user data by complying with relevant data protection regulations and implementing robust security measures.
- **Evacuation Routes:** Users should have access to clear and up-to-date evacuation routes during emergencies. This information is critical for navigating safely to designated evacuation points and minimizing risks.
- **Request Help:** Users should be able to request assistance or support from relevant authorities or responders during emergencies. This feature enhances communication and coordination, improving overall emergency response efficiency.
- **Offline Functionality:** The application should provide essential features in offline mode, considering that disasters can disrupt internet connectivity. This could include access to downloaded emergency guides, maps, and limited community chat functionalities.
- **Multilingual Support:** To ensure inclusivity for a diverse user base, the application should offer support for multiple languages.

3.2 Non-Functional Requirements

- **Usability:** The application should be intuitive and user-friendly, especially during emergencies. It should be easy to navigate without prior experience or training and cater to users with varying technical skills. Multilingual support is crucial for this requirement.
- **Performance:** The system should be able to handle a significant number of concurrent users during emergencies without compromising performance. Response times for critical functionalities like alert delivery and incident reporting should be minimal even under heavy load.
- **Security:** The application should adhere to industry best practices for security to safeguard user privacy and sensitive data. Secure communication protocols should be used for data transmission, and access controls and authentication mechanisms should prevent unauthorized access.
- **Reliability:** The system should be highly reliable and available at all times, particularly during disasters and emergencies. Measures should be taken to ensure continuous operation, including redundancy, failover mechanisms, and disaster recovery plans.

- **Scalability:** The system should scale efficiently to accommodate a growing user base. It should handle a large influx of users during disasters without impacting performance.
- **Maintainability:** The application should be designed for ease of maintenance and updates over time to adapt to evolving technologies and disaster risks.
- **Accessibility:** The application should be accessible to users with disabilities, incorporating features for visually impaired and hearing-impaired users.
- **Interoperability:** The system should seamlessly integrate with existing disaster management systems and external services. Integration interfaces should adhere to industry standards to facilitate data exchange with other systems and agencies.

4. External Interface Requirements

The Mobile Disaster Management System interacts with various external entities through well-defined interfaces. These interfaces ensure seamless data exchange and functionality with other systems and services.

4.1 User Interfaces (UI)

The application will have separate user interfaces for different user groups:

- **Individual User Interface:** This interface caters to the general public and provides functionalities like:
  - Viewing real-time alerts and notifications.
  - Reporting incidents and requesting assistance.
  - Accessing emergency resources and preparedness information.
  - Engaging in community forums and discussions.
  - Viewing evacuation routes and shelter locations.
- **Emergency Responder Interface (Administrator Interface):** This interface caters to authorized personnel like emergency responders, disaster management officials, and volunteers. It provides functionalities like:
  - Managing user accounts and access permissions.
  - Issuing real-time alerts and notifications.
  - Monitoring incident reports and resource requests.
  - Coordinating response efforts and resource allocation.
  - Accessing real-time geospatial data and analytics.

4.2 Hardware Interfaces

The application should be accessible from a wide range of standard mobile devices with internet connectivity. This includes smartphones and tablets with various operating systems like Android, iOS, etc. The application should adapt to different screen sizes and resolutions to ensure optimal user experience across various devices.

4.3 Software Interfaces

The application will integrate with several external software systems and services to provide comprehensive disaster management functionalities. Here are some key software interfaces:

- **Geospatial Data and Mapping Services:** Integration with services like Google Maps or OpenStreetMap allows displaying real-time information about disaster zones, evacuation routes, and shelter locations on interactive maps.
- **Alert Notification Systems:** The application might interface with existing emergency alert notification systems to receive and disseminate critical information like weather warnings and evacuation orders.
- **Data Management Systems:** Integration with government databases or disaster management agency systems could facilitate the exchange of information on resource availability, volunteer registration, and damage assessments.
- **Social Media Platforms:** Integration with social media platforms can enhance community engagement by allowing users to share information and updates during disasters. However, this integration should be implemented with strict data privacy considerations.

5. Conclusion

The Mobile Disaster Management System, outlined in this SRS document, aims to be a valuable tool for all stakeholders involved in disaster preparedness, response, and recovery efforts. By defining clear functional and non-functional requirements, along with external interface requirements, this document provides a roadmap for developing a comprehensive and effective mobile application. The system will empower users with critical information, facilitate communication, and support coordinated response during disasters, ultimately contributing to a safer and more resilient community.