ASHESI

ASHESI UNIVERSITY COLLEGE

DISASTER MANAGEMENT SYSTEM: PROVIDING AN EFFECTIVE COMMUNICATION MECHANISM & ACCESSIBLE DATA ANALYSIS IN GHANA

APPLIED PROJECT

B.Sc. Computer Science

JOSEPH ODARKWEI MILLS

2018

Disaster Management System: Providing An Effective Communication Mechanism & Accessible Data Analysis in Ghana

Joseph Odarkwei Mills

B.Sc. Computer Science

Supervisor: **Dr. Ayawoa Dagbovie**

Project type: **Undergraduate Applied Project**

Submitted to the Computer Science Department, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science Degree in computer science.

Declaration

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

_____.

Candidate's name Candidate's signature Date

I hereby declare that the preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied projects laid down by Ashesi University College.

_____.

Supervisor's name Supervisor's signature Date

Acknowledgement

My sincere gratitude goes to the Almighty God, my family and my friends for the various ways I have received motivation, help and guidance during this project. To my supervisor, Dr. Ayawoa Dagbovie, I say thank you very much for your guidance and supervision: we made it together.

Abstract

Disaster is an unfortunate, yet common phenomenon in our world today. This phenomenon stakes undesirable claims on lives, infrastructure and services alike. Those affected both directly and indirectly must overcome challenges, not only during but also, after the occurrence of disasters. Imperatively, it is always prefered to be better equipped to prevent the disaster and avoid it altogether. Yet that is impossible as not all disasters are man-made. Disaster management then must be up to the task to reduce consequences to the barest minimum. This has not been the case in Ghana.

There is a great need for efficient communication channels and accessible data that will greatly help to preserve lives and property in the country. This project focuses directly on meeting this need. This paper outlines the various technologies and processes employed in developing a disaster management system that has great potential in solving the said problem. The application is an Android application with a simple and intuitive design. Challenges encountered in the development process are also discussed. In addition, suggestions on future work on this application are also given.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Background

Disaster is an unfortunate, yet common phenomenon in our world today. This phenomenon stakes undesirable claims on lives, infrastructure and services alike. Those affected both directly and indirectly must overcome challenges, not only during but also, after the occurrence of disasters. Imperatively, it is always prefered to be better equipped to prevent the disaster and avoid it altogether. Yet that is impossible as not all disasters are man-made. Disaster management then must be up to the task to reduce consequences to the barest minimum. In the simplest of terms, it is the "process of planning and taking actions to minimize the social and physical impact of disasters and reduce the community's vulnerability to the consequences of disasters" (Li et al., 2017). Disaster management relies heavily on readily accessible data. The ability to adequately analyze data gives the country an edge in preventing and managing disasters. Unfortunately, Ghana lacks this all-essential ability. Huge limitations in accessible data, coupled with ineffective communication channels have left the country well weakened in her efforts to effectively prevent or manage disasters.

This shortcoming has resulted in the unfortunate situation where humanitarian organizations respond more to media pressure than to exact humanitarian needs (Amin & Goldstein, 2008). The reliance on media has arisen because of the lack of accurate data. These circumstances are evident and more common in developing parts of the world. No wonder "in Sub-Saharan Africa, disaster damages are double traditional estimates, averaging $14 billion per year" (ModernGhana, 2016). More specifically, Sub-Saharan Africa, particularly Ghana, faces a lack of accuracy when it comes to reporting on death toll, rescue operation statistics, disaster damages etc.

1.2 Related Work & Literature

Ghanaian attempts to curtail these menaces have been on the low thus far. One of such good inputs is the application SnooCODE RED, developed by TinyDavid Limited. It solves the problem of a bad addressing system in Ghana (Nartey, 2017). The application, which has been adopted by the Ghana National Ambulance Service, generates a unique 6-digit alphanumeric code that gives the location of users with the capacity to work offline. It is helpful, but contributes only a small piece of the puzzle: it is not enough to effectively manage disasters in the country. There is the need for solutions that will be particularly helpful in amassing useful data during and after disasters.

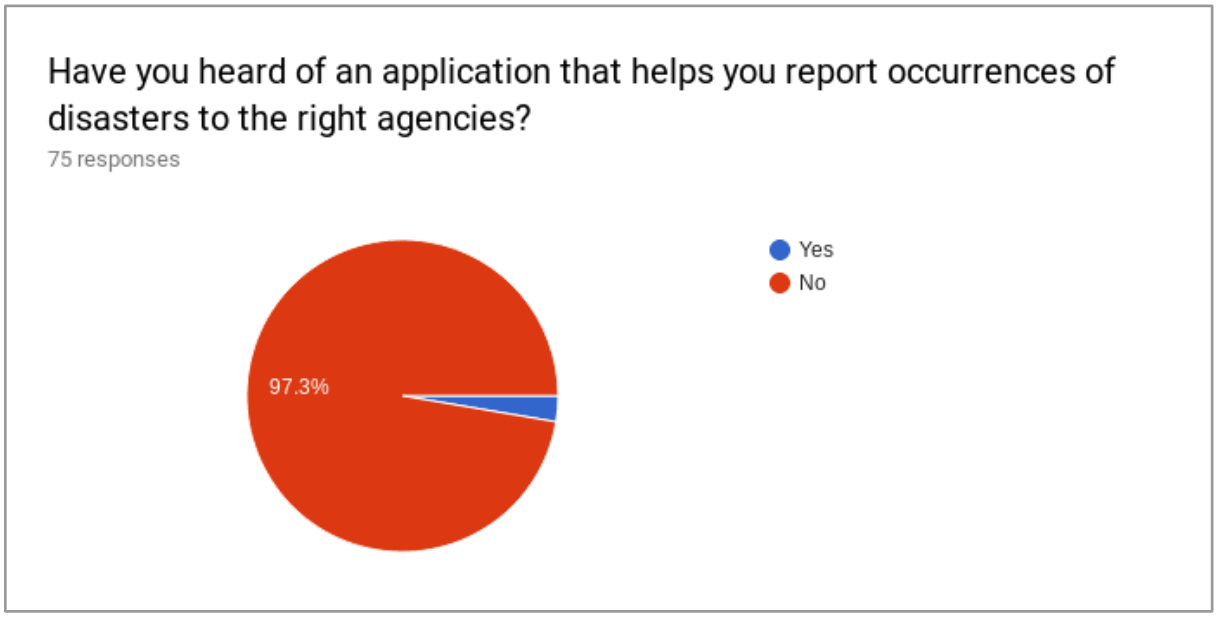


Fig. 1.0: The absence of a disaster management application in Ghana

Outside the borders of Ghana, other inputs include an application called Disaster Monitor. It gives “near real-time alerts about natural disaster[s] with a potential humanitarian impact” (Dominoc925, 2018). The application presents an interactive map marked with color-coded icons that represent disaster alerts at the corresponding locations. It provides a simple tabbed interface for viewing alerts either on the map or on a list. With the help of additional

side navigation, users can view alerts by disaster category. Disaster types include volcanoes, earthquakes, floods, cyclones and tsunamis. As the name suggests, users have to monitor alerts manually without push notifications. This means that, if there is a disaster at your location, you can only find out by manually opening the application, refreshing the feed and searching through the alerts for your location.

1.3 Proposed Solution

In that light, this capstone project aims to provide an effective communication mechanism and an avenue for accessible data analysis through an Android application called Disaster Aides. It will help facilitate the reporting of more accurate information in times of crisis. The administrators perform the task of monitoring disaster occurrences and reports, disseminating alerts that users than access even with the application closed. The user application has four modes of operation: standby, alert, disaster and recovery modes. This application follows the model developed by Bekhor, Cohen, Doytsher, Kanza, & Sagiv (2015) to help with crisis management during earthquakes.

In standby mode, the application collects and stores geosocial data about users including places visited, friends communicated with etc. In this mode, there is no disaster at hand. Next, the alert mode refers to the mode where users are alerted about impending crises; the application makes use of the data available to central agencies to make such predictions and warn users accordingly. The application starts offering guidance from this mode. The next mode is the disaster mode. Here, a disaster has struck. The user can communicate with speed through this application to the appropriate emergency agency. The user sends his/her geographical location along with a captioned image or video to show the current state of the crisis. This is to facilitate better response time from the relief agencies. Once a crisis is reported, everyone near the person will also be alerted and offered guidance. The last mode is the

recovery mode. In the aftermath of a disaster, this application will facilitate effective accounting for affected persons and distribution of aid based on needs and not media pressure.

1.4 Motivation

The application is strictly in line with the National Disaster Management Organizations' (NADMO) objective: "to strengthen disaster prevention and response mechanisms" (NADMO, 2017). This application will fill in the gaps of current efforts by giving personalized alerts to people and their social connections, using geosocial data to offer better relief and recovery services. These services will be offered for disasters in general and not just one kind of disaster. Most importantly, there is an expressed need for an application of this nature. Fig 1.4 shows this desire from my preliminary feasibility study.

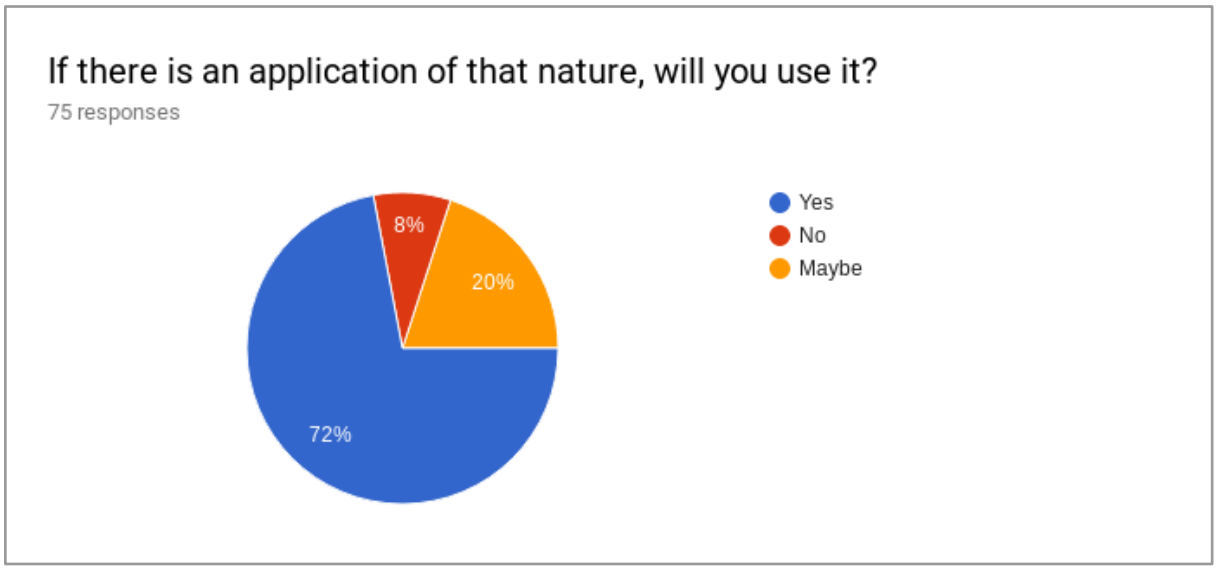


Fig 1.1: Pie chart showing the desire for a disaster management application

Chapter 2: Requirements

This chapter provides a detailed description of Disaster Aide. Its features and functionalities will be expounded on as well. This document is targeted at stakeholders on all levels, including developers. The requirements were gathered through questionnaires, interviews and related applications and literature.

2.1 Requirement Gathering

Table 2.1: Requirement gathering

Resource	Information to be gathered	Methodology
Similar applications	a. Functionalities b. Design patterns	Using and testing the applications
Literature	a. Existing implementation methods. b. Current limitations and deficiencies. c. Preferred features and functionalities. d. Design patterns for disaster management applications.	Reading
Citizens	a. Usability requirements b. Functional requirements	a. Interviews b. Questionnaires
NADMO	a. Policies and standards for disaster management b. Existing technology	Interview
Android API	a. Implementation methods b. Packages, classes and other Android components that would be needed	Reading the API reference
Firebase & FirebaseUI API	a. Information on tools and efficient ways of storing data b. Information on efficient ways of offline storage and synchronization	Reading the API reference

2.2 Software Requirements Specification

2.2.1 Scope

Disaster Aide is an Android application that allows citizens to report occurrences of various disasters with a captioned image or video along with the location (GPS and word-described). The user specifies the type of disaster while reporting so that response can be streamlined. The specified types include floods, earthquakes, motor accidents, meteorological, fire and epidemic. An option to select other disaster types is given when reporting. The administrator portals of the agencies will give them the opportunity to send alerts and disseminate helpful information concerning disasters. Data analysis will be possible because of the central point of accessible data.

2.2.2 System Description

The Android application will implement an alert system using push notifications. In-app alerts are presented graphically on a map interface accompanied by a section for up-to-date news and rated information. Users will also be able to view their reports and their respective statuses (i.e. sent, pending, unsuccessful). A catalogue of contacts for the various stakeholders (police, fire service, NADMO etc) will also be provided so that users can know who to contact.

2.2.3 Assumptions and Limitations

2.2.3.1 Assumptions

It is reasonably assumed (by observation) that the most widely used operating system for mobile phones used in Ghana is Android. Furthermore, the minimum Android version for Android users in Ghana is assumed to be KitKat (4.4).

2.2.3.2 Limitations

This project is limited by Androids restriction on peer-to-peer connections without data networks. This limits the option to share files totally offline, like the AirDrop functionality on iPhones. The application will only be available for mobile phones running the Android

operating system. The unavailability of learned models for image processing pf disaster report for validation also leaves a shortcoming to this project.

2.2.4 Functional Requirements

2.2.4.1 Use Cases

This section outlines the use cases for the Disaster Aide.

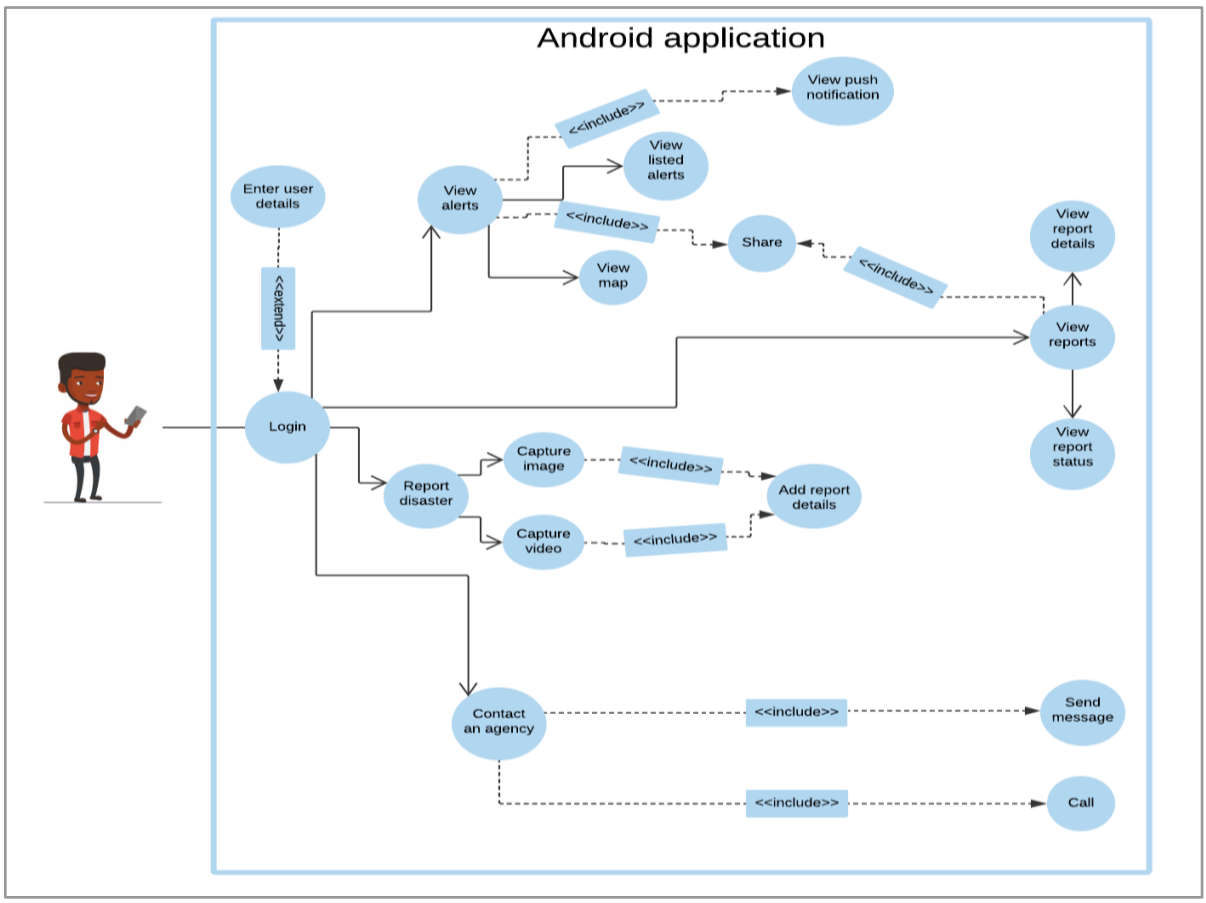


Fig. 2.0: Use case diagram

2.2.4.1.1 Login

Table 2.2.4.1.1a: Use case for login

Use Case Name	Login
Actor	Citizen
Trigger	The citizen taps on the application's icon from the phone's menu
Precondition	a. The citizen must be on the home screen or applications menu. b. It must be the first time of using of the application or user must be logged out.
Procedure	a. The user taps on the application's icon for the login screen to show. b. The user enters his/her phone number and click on the login button. c. A text message is sent to the user's phone containing a verification code. d. Authentication is done automatically. e. Landing screen is shown containing trending news and updates.

2.2.4.1.2 Report Disaster

Table 2.2.4.1.1b: Use case for reporting disasters

Use Case Name	Report disaster
Actor	Citizen
Trigger	The user taps on the "Capture" button in the bottom navigation menu.
Precondition	The application must be opened and logged in. The capture button is accessible no matter what action the user is carrying out so that reporting can be done with speed.
Procedure	a. Two floating buttons appear when the capture button is selected: one for image capture and the other for video capture. b. The user selects one option and captures either an image or a video. c. After confirming captured media, the media is previewed on the Details screen where the user adds a caption, disaster category and word-description of location. d. Disaster categories are presented as toggle buttons with descriptive images. A section is provided for a disaster category not presented. e. User taps on submit button to report the disaster. f. A toast notification confirms the success of the submission. g. The disaster report is added to the reports section.

2.2.4.1.3 View Report

Table 2.2.4.1.1c: Use case for viewing disaster reports

Use Case Name	View report
Actor	Citizen
Trigger	The user taps on the Reports button in the bottom navigation menu
Precondition	The user can be on any screen except when capturing a disaster report or adding details to the media.
Procedure	a. A list of reports are displayed in a list view when the report button is selected. b. A report is represented by a ‘card’. Each ‘card’ contains a preview of the media, caption, location, category, date, location and status of the report. c. A report can be viewed in full by tapping on the prefered card. d. A screen with the full report details shows next. e. The user may see the response action taken by the appropriate agency.

2.2.4.1.4 View Alert

Table 2.2.4.1.1d: Use case for viewing alerts

Use Case Name	View alert
Actor	Citizen
Trigger	The user may select the trending or map sections or a push notification may show an alert.
Precondition	The user may either have the application opened or closed.
Procedure	a. When the user is outside the application, a push notification appears for an emergency alert. 1. Tapping on the notification brings the application to the foreground. 2. The user can now view more details about the alert and take any necessary action. b. When the application is in use, a push notification also alerts the user. In addition to that, the trending section of the application is updated with the details. 1. Tapping on the notification brings to the Trending section. 2. The user can now view more details about the alert and take any necessary action.

2.2.4.1.5 Contact Agency

Table 2.2.4.1.1e: Use case for contacting an agency

Use Case Name	Contact agency
Actor	Citizen
Trigger	The user taps on the phone or message icon on a listed agency.
Precondition	The user must have selected Contacts section of the application on the .
Procedure	a. From the list of agencies provided, the user may either tap on the phone icon to call or the message icon to send a message. b. If the phone icon is selected, the default phone application on the user's phone comes to the foreground with the agency's phone number already dialed. The user must now tap on the call button to place the call. c. If the message icon is selected, the default SMS application on the user's phone comes to the foreground with the agency's phone number already dialed. The user must now type the message and send.

2.2.5 Non-functional Requirements

2.2.5.1 Performance

The system should have a good performance in terms of speed and data usage. Because of how crucial this system is, users should be able to report disasters with the least amount of data. The application should be responsive as well.

2.2.5.2 Reliability

This system should, with high probability, accept validated input and process all work correctly and completely. It should give correct outputs and responses as expected.

2.2.5.3 Security

User information must be kept secure. Personal data is of particular interest since it is very sensitive. If the app is used for any purpose, it should be guaranteed that it is the user who initiated the action.

2.2.5.4 System availability

Owing to the crucial nature of such an application, the system will be available everyday of the week. There will no exempted days or times of the day as disasters can occur at anytime.

2.2.5.5 Error handling

The system will check inputs from user so that no bad inputs are accepted. For example, when reporting a disaster, the user must enter correct data corresponding to the field being used. Fields for text (alphabetic) only should not accept numbers.

2.2.5.6 Conventions and Standards

The application follow the Material Design standards proposed by Google (Google Inc., 2014) under the Apache 2 Licence.

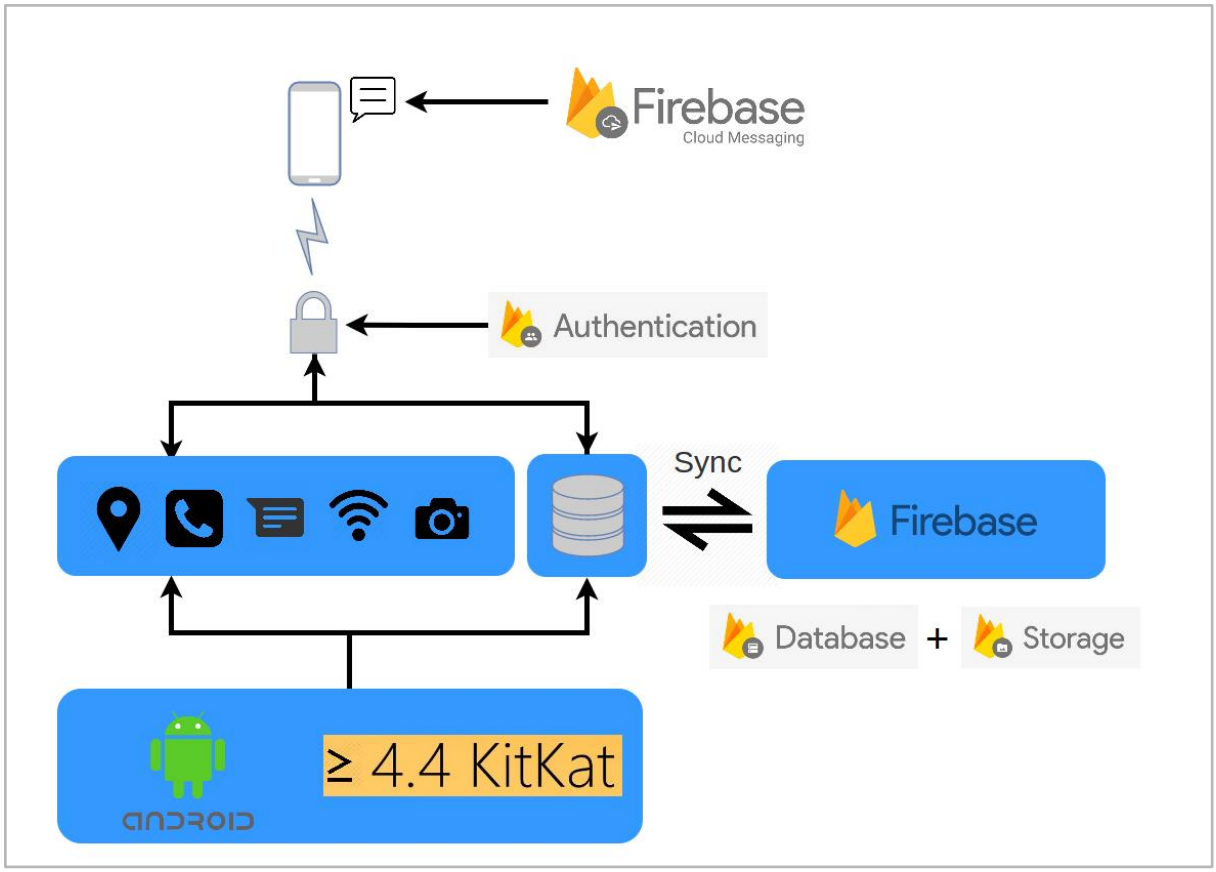
Chapter 3: Architecture and Design

This chapter gives a high level description of the disaster management system's architecture. The architecture pattern followed in the development of the application is also expounded in detail.

3.1 System Architecture

There are two main points of usage for the application: an end user (through a mobile phone) and an administrator from an agency (through a web portal). All users must be authenticated before they can use the application. This is to ensure that the security requirement mentioned in the previous chapter is met. Firebase authentication handles the authentication process. An authenticated end user then has access to the next layer, which refers to the components of the phone that the application uses (camera, GPS, phone, etc).

These components exist in the Android environment represented by the last layer. The user must fulfil a minimum requirement of having Android 4.4 running. The local data of the app is kept in constant synchronisation to Firebase database and storage. Firebase offers effective synchronisation techniques that make it perfect for a real-time application like this. This architecture ensures that all necessary functionalities for the citizen can be achieved. The web portal offers an administrator a means of accessing this real time data. Once again Firebase synchronises data to keep the accessible data fresh all the time. The high level system architecture of the proposed system is shown in Fig. 3.1 below.



3.2 Architecture Pattern

This application follows the model-view-presenter (MVP) architecture for application development. In MVP, the presenter is responsible for getting user actions from the view in order to update the model accordingly and vice versa. The presenter 'listens' for the most up-to-date data from the model, determines whether the view should be updated and binds the fresh data to the view when necessary (Zukanov, 2015). The presenter determines whether a view should be updated using event listeners. Android and Firebase provide a variety of event listeners to make this possible.

Unlike the popular model-view-controller (MVC) approach, the model does not interact directly with the view. The view implements no logic as its "sole purpose becomes rendering of the data that was bound to [it] by the presenter and capturing user input." (Zukanov, 2015). Figure 3.2 presents a diagrammatic representation of the MVP architecture (right) in contrast with the MVC architecture (left).

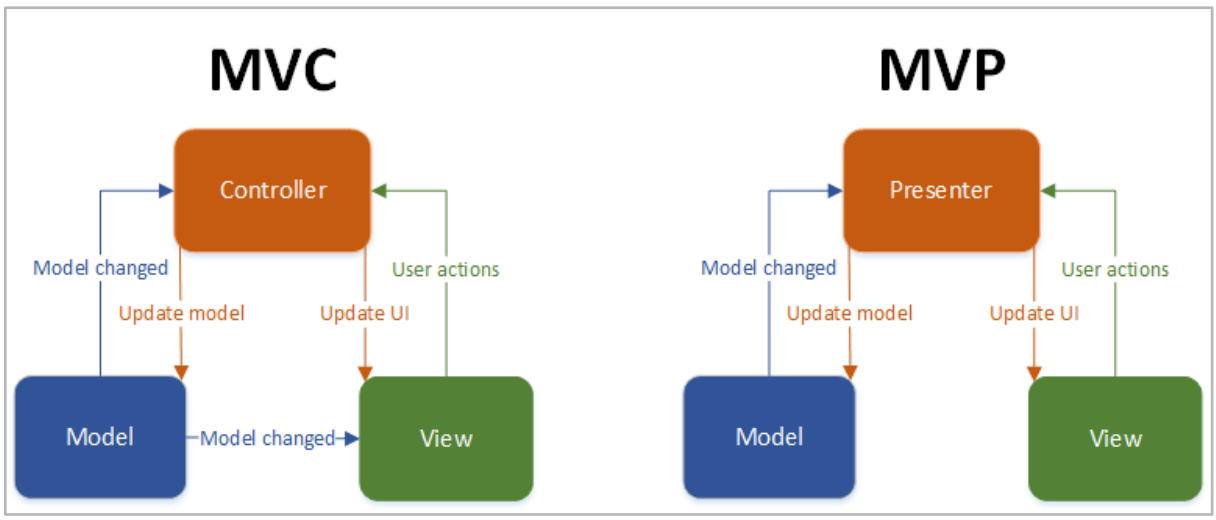


Fig. 3.1: MVP vs MVC architecture patterns (Zukanov, 2015)

Chapter 4: Design and Implementation

This chapter outlines the design of the user interfaces of the application. Following that is a list of tools and technologies used in the implementation of the disaster management system.

4.1 Process Model

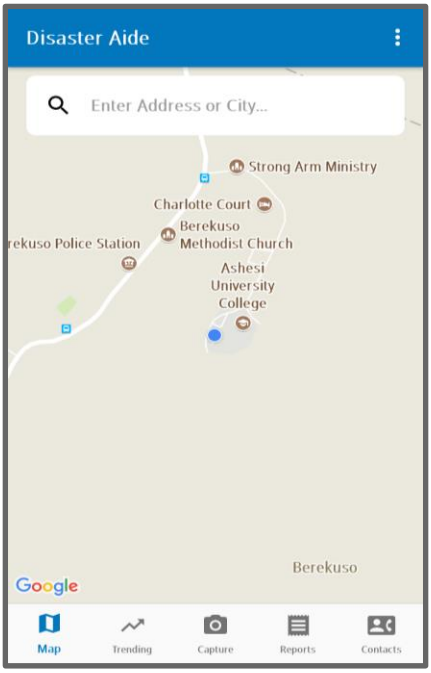
The application was developed incrementally using Extreme Programming (XP), a kind of agile software development. XP allows for incremental planning, small releases, simple designs, test-first development, refactoring and continuous integration during software development (Summerville, 2011). Development was carried out incrementally using ‘stories’ that are further broken into development tasks. Each story represents a module of a functionality of the application and specifies how to progress from the start to the completion of an action.

Another model used in the development process is the software-reuse model. In software development, it is common to find ready-to-use implementations that already satisfy the requirements specification for the project. This application made use of web services provided by Firebase and other Google APIs (eg. Google Maps API).

4.2 User Interface Design

The user interface of the application was developed with the crucial nature of the application in mind. Users should be able to capture a disaster no matter what they were doing. Hence a bottom navigation is used for the menus so that the capture button can be accessed no matter what interface is in the foreground of the application. The colors used also depict a need for peace and tranquility - the very things that disasters steal from us. The interfaces in for each functionality are intuitive.

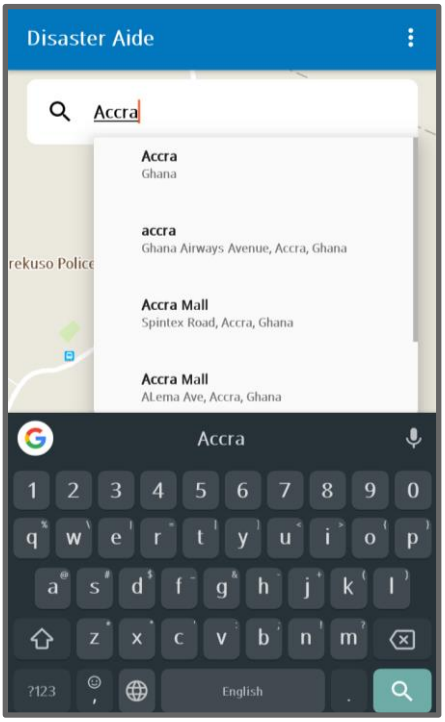
4.3 User Interfaces



4.3.1 Map

This interface allows the user to interact with the map. The current location of the device is represented by the blue dot.

Fig. 4.0: Map interface



A search bar is provided at the top for users to search locations. Auto completion is added using the Google Places API.

Fig. 4.1: Search with autocomplete feature on map interface

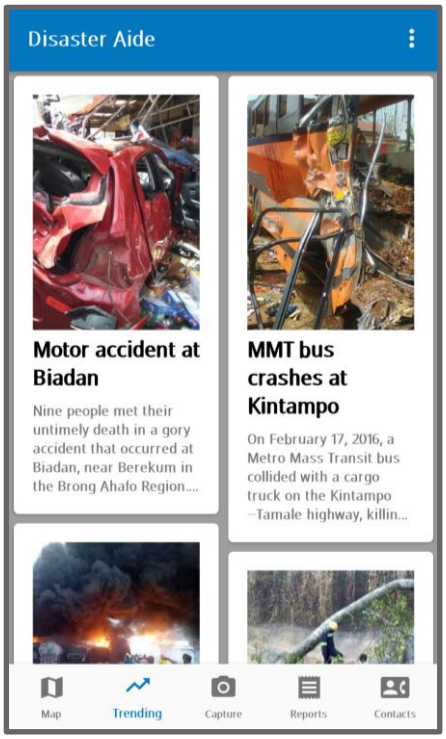


Fig. 4.2: Trending interface

4.3.2 Trending

The user is kept up to date with trending news and updates on disasters and relief efforts. This is the landing page of the application.

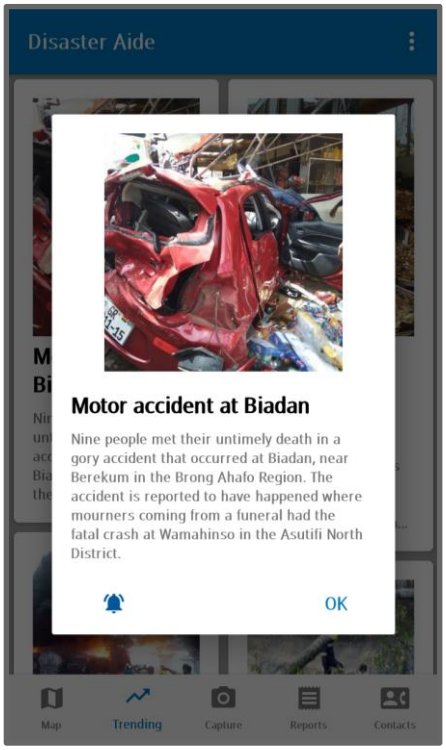


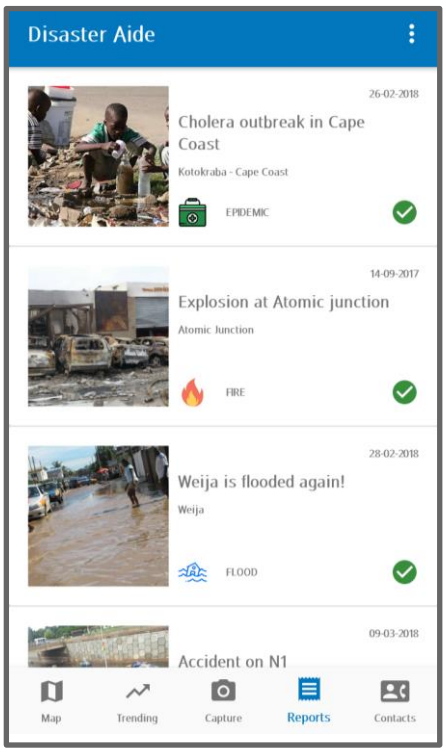
Fig. 4.3: Trending dialog interface

This dialog displays more information about the selected trending topic. Users may subscribe/turn on notifications for the selected topic.

4.3.3 Contacts

This interface allows users to call or text the listed agencies.

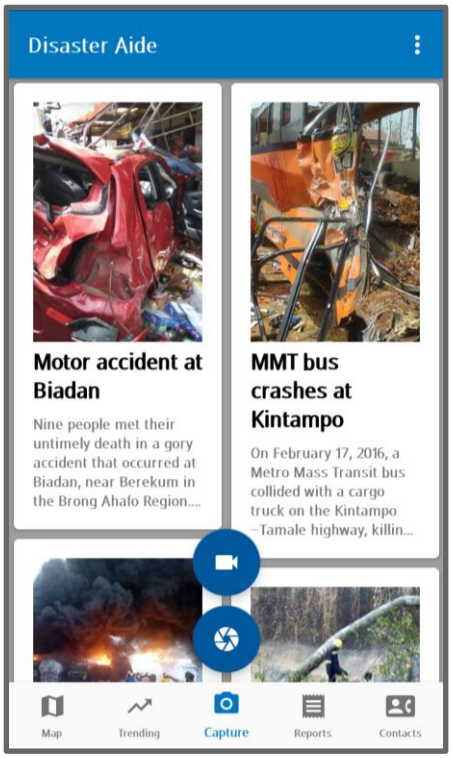
Fig. 4.4: Contacts interface



4.3.4 Reports

Reports made by the user are listed in this section of the application.

Fig. 4.5: Reports interface

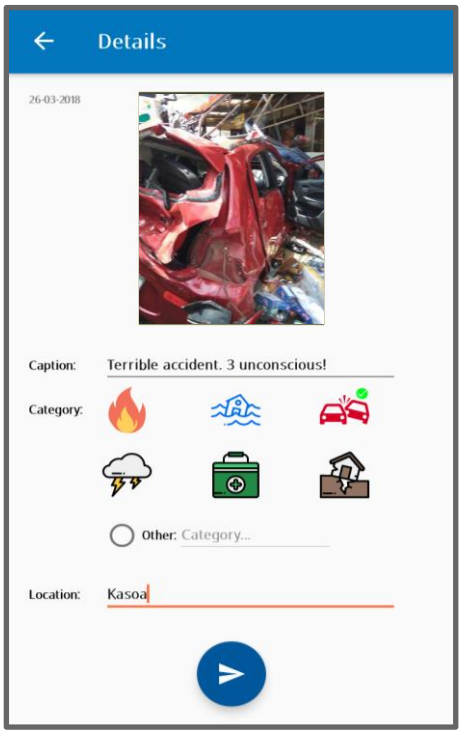


4.3.5 Capture

Users capture disasters by tapping on the capture button on the bottom navigation view. A bottom navigation view is used so that menu items (especially the capture option) can be accessed no matter which screen is currently in the foreground.

Selecting 'capture' reveals two buttons for video and image capture. Selecting any will open up the camera app so that the right media can be captured.

Fig. 4.6: Capture interface



4.3.6 Report Details

Here the user has captured a disaster. Details have to be added to the report before being sent. Details to be added include: caption, disaster category and location in words. Disaster categories are presented using images on toggle buttons. An 'other' section is provided for cases when disaster category is not represented.

Fig. 4.7: Report details interface

4.4 Tools and Technologies

4.4.1 Android Studio

Android studio is an integrated development environment for Android application development. Development was done solely in Android Studio.

4.4.2 Firebase

Firebase is a product of Google that offers services in web hosting, realtime database, storage, authentication, cloud messaging, advertisements and many more. This project specifically uses the realtime database, storage, cloud messaging and authentication features.

4.4.3 Google Maps API

The Google Maps API allows applications to make use of Google Maps.

4.4.4 Google Places API

Google Places API provides access to the geolocation data of cities, addresses, and even organizations. With this, the application gives autocomplete suggestions while the user searches for places on the map.

4.4.5 GitHub

GitHub is a versioning system that enables developers to keep track of versions in incremental development. Changes and additions made were frequently saved on GitHub.

Chapter 5: Testing and Results

This chapter talks about the various testing techniques that were employed to test the application. Tests carried out include component testing, system testing and user testing.

5.1 Component Testing

The application was developed using the XP model of agile development (as explained in Section 4.1). Hence, testing of components was done frequently between builds. No automated testing tools were employed throughout development. Each build added small incremental improvements to the application.

5.1.1 Notifications Unit

Notifications were implemented using the Firebase Cloud Messaging service for Android. The notifications test lab on the Firebase console was used to test notifications to single devices, to all users with the application and by topic. The trending section of the application allows users to subscribe or turn on notifications for certain disasters or alerts. Notifications on these topics were delivered successfully to only those subscribed to the topics. General notifications were also delivered to all users successfully.

5.1.2 Trending Unit

The trending section offers users updates and alerts on trending topics, be it about disasters or not. Making use of Firebase Database and Storage, alerts added or modified on the console reflected in the application in real time. This met the performance requirement of keeping information fresh with fast synchronization across devices.

5.1.3 Map Unit

In this section of the application, the map interface shows the device's location via a blue circle. Device locations were accurately pinned on the map interface. With this unit comes

the need for permission to access the device's location and GPS to be turned on. Using the latest standards in requesting for permissions, the application rightfully requested for location permission the first time the map section was opened. Upon rejecting, the application continuously prompted for the permission to be granted as it forms a crucial part of the application and was indispensable.

Another component of this unit is the search functionality. Locations were successfully searched and navigated to upon confirmation. The Google Places API worked correctly and navigated to the correct places searched. This search bar also came with autocomplete suggestions to cap a good user experience.

5.1.4 Reports Unit

This component test dealt with the viewing of reports by each user. First of all, users had to see only reports they had made. This was successfully implemented. Then also, the next test was on synchronization. The reports users made were updated on the server in real time. The status of reports were seen by a status icon on the bottom-right of every report. The coloring was intuitive to show users the status of their report: green for a successful submission and yellow for a pending submission.

5.1.5 Contacts Unit

This section of the application provided contact details of the various agencies in Ghana. Users were able to select the agency to contact and either call or send a message to that agency. The default messaging or phone application of the user's phone was used.

5.1.6 Offline Storage

Offline capabilities were enabled in the application to enhance the user experience. Data was synchronized as soon as internet connection was restored. Writes to the database could be

done offline with corresponding interface updates. The automatic synchronization of data made application usage interactive even in the absence of internet connection.

5.1.7 Login & Logout

Login and logout functionalities also passed their tests. Login sessions were also effectively implemented so that users were not logged out of the application when they were rightly logged in. Users were able to login to the application using their phone numbers; this is simpler than the usual username-password or email-password as it covered all users despite their background. A text message with a login code was sent successfully to the user's phone and the application logs in automatically upon receiving the text message. The application provided a seamless login process. Logging out of the system was done by selecting the logout option in the menu.

5.2 System Testing

System testing was done after all units were merged together successfully. However, this does not mean that system testing was done at the end of development. All sections were successfully accessed from each other. Offline storage capabilities worked as expected. Users were able to log in and out of the application successfully. Data synchronization worked well. Again, due to XP, the entire application was tested as a whole incrementally.

5.2.1 System Requirements

The following are the minimum system requirements for the application:

- ➔ Android version: Android KitKat (4.4)
- ➔ Memory: 100MB
- ➔ Storage: 20MB

5.3 User Testing

The application was tested by students from Ashesi University College. Since the usage of the application does not require any particular background, testers were randomly selected. Twenty testers were allowed to test the application. For some users, the application was installed on an Infinix Hot 4 having 1 Android 7.installed. These were allowed to test the application on the phone provided. The rest of the users had the application installed on their Android phones for testing. All users were able to test the application successfully.

In general, the application had a 4-star rating (out of 5), reflecting its usefulness, intuitive design, and resourceful nature. The feedback from this session can be divided into three major divisions: report validation, usefulness and the alert mechanism. These general topics were the main talking points from the testers.

About five testers expressed concern about the need to validate disaster reports. Currently the application does not have any image processing module to validate each report. Though this was not in the scope of this project, it is indeed a necessary feature of such an application and must be attended to in future work on this application. This feature would require the training of machine learning models for validation of each disaster category. The work around in this implementation is that, when a user sends a report, users nearby are notified for them to validate the report.

Next, the way users got alerts was of utmost importance to testers. For an application that should alert you in the event of a disaster, the way in which the alert comes is very crucial. Notifications were received in real time as tested using the Notifications Lab of Firebase. Users were notified whether the application was opened or not. Trending news subscriptions also determined whether users got notifications concerning the particular topic or not.

After testing the application, all users admitted that the application indeed met the need it was intended to. They all reiterated the need for a better way of reporting occurrences of

disasters in the country. Testers said that the application is an application that they would use if deployed nationwide. This proves the initial response from requirements gathering as illustrated in Fig. 1.1.

Chapter 6: Conclusion and Recommendations

In this paper, the procedures for developing the disaster management application has been properly outlined. Requirements gathered from prospective users of the application and other stakeholders guided the development process to ensure that the application is going to be used. The application was implemented with an intuitive design, and tested on all levels. The application received an overall 4-star rating.

6.1 Shortcomings

This implementation did not meet all requirements and so needs further improvements. One shortcomings of this implementation is that it allows for only image capture. Also, although the application works offline, reports can only be received by the agencies when the users have a working internet connection. The limitations in the Android operating system make this very difficult to implement. Technologies like WiFi Direct cannot solve this problem because they require connection of devices to be done manually (as implemented by sharing applications like Xender). Another limitation is that, the application is not available for iOS.

6.2 Future Work

In the future, the application should support video capturing. For this functionality, the native camera has to be controlled so that there is a maximum number of seconds the video can last. This will ensure that data consumption is kept low. The application should also be available for other operating systems like Windows and iOS. Furthermore, there should be a validation on every disaster report sent. Image segmentation modules can be developed for each disaster category using machine learning. This will help to automatically detect false reports so that the application is not misused. Finally, a way of sending reports (text and media) totally offline without internet connection would be a great addition.

Also, a web portal should be developed to offer a centralised point for data analysis. This system will be used by disaster management agencies to monitor and manage disasters in real time. This will make the operations of these agencies much more efficient. In addition a first aid guide can be added so that users have a guide to follow when a disaster strikes and the responding agency has not arrived yet.

Overall, the disaster management system has been a successful project. It has helped to sharpen my software engineering and Android development skills. This project has challenged me to apply knowledge and skills acquired from most of my courses to solve a pressing need in my country. This application has great potential in helping to better manage disasters, save property and preserve lives in Ghana.

Bibliography

Amin, S., & Goldstein, M. (Eds.). (2008). *Data Against Disasters*. Washington, DC: The World Bank. Retrieved from https://elibrary.worldbank.org/doi/abs/10.1596/978-0-8213-7452-8

Bekhor, S., Cohen, S., Doytsher, Y., Kanza, Y., & Sagiv, Y. (2015). A Personalized GeoSocial App for Surviving an Earthquake. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on the Use of GIS in Emergency Management* (p. 21:1–21:6). New York, NY, USA: ACM. https://doi.org/10.1145/2835596.2835616

Dominoc925. (2018). Disaster Monitor (Version 5.2.17) [Mobile application software]. Retrieved from http://dominoc925.blogspot.com/

Google Inc. (2014). Introduction - Material Design. Retrieved March 6, 2018, from https://material.io/guidelines/material-design/introduction.html#

Li, T., Xie, N., Zeng, C., Zhou, W., Zheng, L., Jiang, Y., … Iyengar, S. S. (2017). Data-Driven Techniques in Disaster Information Management. *ACM Comput. Surv.*, *50*(1), 1:1–1:45. https://doi.org/10.1145/3017678

ModernGhana. (2016). Natural Disasters Cost $520bn in Losses Every Year. Retrieved September 15, 2017, from https://www.modernghana.com/news/736426/natural-disasters-cost-520bn-in-losses-every-year.html

NADMO. (2017). Functions & Objectives - National Disaster Management Organisation (NADMO). Retrieved September 15, 2017, from http://www.nadmo.gov.gh/index.php/about-us/functions-objectives

Nartey, R. (2017, February 15). Addressing App unveiled in Ghana. *Today Newspaper*. Retrieved from http://www.todaygh.com/addressing-app-unveiled-ghana/

Summerville, I. (2011). *Software Engineering* (9th ed.). USA: Pearson Education Inc.

Zukanov, V. (2015, July 12). MVP and MVC in Android - part 1. Retrieved March 25, 2018, from https://www.techyourchance.com/mvp-mvc-android-1/

Appendix

A: Questionnaire

This questionnaire is to help gather information needed to develop a disaster management application for citizens to report occurrences of disasters and effectively communicate with disaster management agencies in Ghana. The application when developed will also provide accessible data for these agencies that will inform decisions that will help better manage/avoid disasters in the country. Your involvement in this project is purely voluntary and your responses are truly confidential. Thank you.
*Required

1. Gender *
   Mark only one oval.
   - Male
   - Female
   - Other:

2. Age range *
   Mark only one oval.
   - Below 15 yrs
   - 15 - 20 yrs
   - 21 - 30 yrs
   - 31 - 50 yrs
   - Over 50 yrs

3. Please select an estimate of your monthly data consumption plan? *
   Mark only one oval.
   - Less than 500MB
   - 500MB - 1GB
   - 1GB - 2GB
   - Over 2GB
   - I use WiFi most often.
   - I don't know.

4. Have you ever personally been a victim or eyewitness of any kind of disaster? *
   Mark only one oval.
   - Yes

No Skip to question 9.
Prefer not to say Skip to question 9.

Personal Experience or Eyewitness

5. Did you contact any person/organization for help during the disaster? *
Mark only one oval.
Yes
No

6. If "yes", who did you reach out to?
Tick all that apply. Skip if you answered "no" to the previous question.
Parent/Guardian
Sibling
Partner
NADMO
Fire Service
Police
Ambulance Service
Other:

7. Did you post any message/media about the disaster on any social media platform? *
Mark only one oval.
Yes
No
Prefer not to say

8. If "yes" which social media platform did you post to?
Tick all that apply. Skip if you answered "no" to the previous question.
Tick all that apply.
WhatsApp
Facebook
Twitter
Instagram
Other:

Disaster Reporting

9. In reporting a disaster to the appropriate agencies, what information do you think is relevant to send? *
Tick all that apply.
Location
Image or video
Caption or brief description
Type of disaster
Other:

10. Rank the following people/agencies that can be contacted during a disaster in the order of their importance. *

1 - most important | 5 - least important
Mark only one oval per row.

	1	2	3	4	5
Partner					
Disaster Management Agency (eg. NADMO, Fire service etc)					
Parent(s)/Guardian(s)					
Sibling					
Media house (eg. radio station)					

11. If you have tried contacting any agency in the event of a disaster, kindly share your experience.

Disaster management software

12. Have you heard of an application that helps you report occurrences of disasters to the right agencies? *
Mark only one oval.
Yes
No

13. If there is an application of that nature, will you use it? *
Mark only one oval.

Yes	After the last question in this section, skip to question 15.
No	After the last question in this section, skip to question 16.
Maybe	After the last question in this section, skip to question 15.

14. If "no", what will you do instead?
Tick all that apply. Skip if you answered "yes" to the previous question.
Call 191 or any emergency line
Post videos or images on social media
Call family or friends
Other:

Functionality

15. What functionalities will you like such an application to have? *
Tick all that apply.
Attach image/video
Send location
Share disaster reports on social media
View trending news about disasters
See weather updates
Map interface with disaster alerts
Contacts of all necessary agencies
Automatically alert friends and family when in disaster zone
Other:

Comments

16. Please share your ideas or any comments you have that have not been covered in this questionnaire.