Enhancing Disaster Preparedness: Development of an Android Mobile Application for Real-time Disaster Reporting, Notification, and Danger Zone Visualization

Jun Han

A thesis submitted to
Auckland University of Technology
in partial fulfilment of the requirements of the degree of
Master of Computer and Information Sciences (MCIS)

2024

School of Engineering, Computer and Mathematical Sciences
Faculty of Design and Creative Technology

Table of Contents

List of Figures

List of Tables

List of Acronyms

APP	Application
AI	Artificial Intelligence
APIs	Application Programming Interfaces
DEAPP	Disasters Emergency Events Application
DSR	Design Science Research
DSRM	Design Science Research Methodology
GIS	Geographic Information System
GPS	The Global Positioning System
IDEs	Integrated Development Environments
IT	Information Technologies
IS	Information Systems
NEMA	National Emergency Management Agency
ORM	Object-Relational Mapping
SDK	Software Development Kit
SDLC	Software Development Life Cycle
UI	User Interface
UML	Unified Model Language

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor used artificial intelligence tools or generative artificial intelligence tools (unless it is clearly stated, and referenced, along with the purpose of use), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Jun Han

3/10/2024

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my primary supervisor, Dr. Krassie Petrova, for her invaluable guidance, support, and encouragement throughout this journey. I am especially thankful for the opportunity to research disaster management and mobile applications, a field I am passionate about. Her expertise, patience, and thoughtful feedback helped me navigate both academic challenges and personal difficulties. Her understanding and flexibility during my demanding work commitments, along with her unwavering support after the loss of my completed prototype due to hardware failure. Without her support, I could not have completed this thesis.

Secondly, I would like to thank my secondary supervisor, Dr. Sarita Pais, for her guidance throughout this study.

I am also grateful for the support from the AUT CMS Postgraduate Programmes, especially for Amber Osgood to help me in coordinating with the school.

I must express my heartfelt gratitude to my family and my best friends, Sumeet Badhan and Min Weng, for their unconditional support, prayers, and encouragement. Without them, I could not have completed my studies.

Last but not least, I am thankful for my colleagues at work for their support and encouragement.

Glory to God.

Ethical Approval

This research has obtained the Ethical approval from the Auckland University of Technology Ethics Committee (AUTEC) on 20th May 2024, the application number is 24/128. The Ethics approval letter can be found in Appendix section A.

Abstract

In the face of increasing disasters, effective disaster preparedness and response are crucial. Nowadays, smartphones are widely used by everyone. In the case of a disaster, it's possible to use mobile technology to allow general public to report incidents and alert them about the events and the location of emerging disaster danger zones.

Currently, disaster and emergency mobile applications can be categorized into two types: Early warning systems and Location-based alert systems. Early warning systems provide alerts to mobile users if a disaster occurs, however, these applications often lack detailed information about the disaster events and direct users to visit a website for more information. On the other hand, location-based alert systems allow users to receive alerts and information by setting up their location in the application. However, if users fail to set a location, they will not receive any alert about the disaster. All current mobile applications only passively receive alerts and cannot enable users to directly report disaster events.

This research focuses on enhancing disaster preparedness through the development of an Android mobile application prototype designed for real-time disaster reporting, notification, and danger zone visualization. The application prototype allows users to report disasters as they occur, receive timely notifications about ongoing threats, and visualize affected areas on a mobile phone map. This helps the public to avoid entering into hazardous areas and allows emergency services to efficiently dispatch equipment and materials.

Furthermore, this application prototype is evaluated through disaster simulations among the general public. The feedback is discussed in terms of both potential benefits and limitations. Suggestions for future development work are also provided.

1. Introduction

As of 2024, there are approximately 7.1 billion smartphone users worldwide, with over 86% of the population owning one or more smartphones (GilPress, 2024; Turner, 2024). Compared to the traditional functionality of phones which was primarily limited to marking calls and sending messages, smartphones have undergone a complete transformation (Brewer et al., 2016). Nowadays, people use smartphones for a multitude of tasks, such as checking email, connecting with family, friends, watching movies, reading books, taking photos and videos, shopping online, using GPS for navigation, monitoring sleep, blood sugar and heart rate, online banking, checking weather forecasts, making appointments, attending online meeting and online classes, and more (Turner, 2024). Smartphones and their applications are not only driving technological innovation but also changing people's lifestyles, work styles and industry structures. (Amez & Baert, 2020; Busch et al., 2021; Nawaz, 2024). In particular, with the widespread adoption of smartphones, new opportunities are brought to disaster management due to the widespread use of mobile phones and technological advancements (Nawaz, 2024).

The main focus of this research is to leverage smartphones and their applications to enable the general public to report disaster and emergency events, as well as to access the detailed information about any danger around an area or any calamity that needs immediate attention.

This research develops an Android mobile application prototype, which facilitates real-time reporting of disaster and emergency events by the general public. Additionally, the application provides timely notifications and allows users to visualize danger zones. Section 1.1 introduces the background and rationale of this research. Section 1.2 outlines the research aims and research question. Following this, section 1.3 provides research approach and the structure of this research thesis.

1.1. Research Rationale

In the recent years, the frequency and severity of both natural and man-made disasters on global scale have shown a troubling upward trend, presenting extraordinary challenges to the society (Chaudhary & Piracha, 2021; de Asis et al., 2019). These events, like floods, droughts, hurricanes, earthquakes and war, not only cause significant casualties and economic losses but also impose severe damage on natural environments and infrastructure, with extensive consequences for sustainable development. (Crunch, May 2021; Solokha et al., 2023)

While doing my research and preparing my literature review last year, New Zealand experienced a devastating event when Cyclone Gabrielle hitting on 13 February 2023. This cyclone affected over half of the country's population, causing widespread damage to homes, businesses, and infrastructure, particularly in Auckland (iDMC, 2024). Looking back three years ago, when COVID-19 pandemic hit, New Zealand and the entire world suffered major losses because of the implementation of lockdowns. The losses that incurred during this period, besides economic ones, included extensive psychological and physiological effects on individuals(National Library of Medicine, 2023).

Currently, in New Zealand, the public relies on radio, TV news, Civil Defence websites and social media for emergency information. The ministry of Civil Defence and the relevant emergency management groups will publish the latest information through these channels to let public know what is happing and what we should do, and they also provide emergency warnings to mobile phones through SMS (New Zealand Government, 2021).

While these methods are helpful, they may not be highly effective, as people need to access specific devices such as radio or TV or browse the relevant websites to get the information for disaster preparedness. Nowadays, as we discussed earlier with the usage of smartphones by majority of the population, the focus has to be shifted to the applicability of such devices for disasters management. Depending on their purpose, such mobile applications can be divided into two categories (Tan et al., 2017):

- **Disaster / Emergency Alerts**

Pamudji et al. (2020) created an integrated early warning system with a mobile application that provides alerts and knowledge of earthquakes and tsunamis. The mobile application links to Indonesia's National climatological, metrological and geographical agency websites to obtain the information. (Pamudji et al., 2020). Similarly, Anta et al. (2021) built a mobile application for flood disasters to alert and monitor the flood information in Jakarta.

However, these early warning system applications provide alerts but often lack detailed information about disaster events. If the user wants to get more information, the application will direct them to visit the website.

- **Geographic information system (GIS)**

Nowak et al. (2020) state that using geographical location could be benefit when develop a mobile application as it will help improve efficiency in field work and enhance location-based insights. When a disaster or emergency happens, the user could use the developed mobile application to share or pin their location to report the event or get alert information.

In New Zealand, a similar mobile application is called the Red Cross hazard app. The users could use the application to request hazards alert information by adding their location (New Zealand Red Cross, 2022) However, if a user does not pin their location, they will not be able to view the disaster area on their phone. Additionally, these mobile applications do not provide visual information, such as displaying the danger zone on the map on their personal mobile device.

1.2. Research Aims and Question

This research aims to develop a mobile application prototype that enables the public to report various emerging disasters, including but not limited to natural disaster, chemical spills, car crashes, and personal threats, among others. It also aims to provide timely notifications, access disaster event details, and visualisation of danger zones directly on user's mobile maps. The research question of this study can be formulated as:

How to use mobile technology to enhance real-time disaster reporting, notification, and danger zone visualization to support disaster preparedness?

By leveraging mobile technology, this research seeks to enhance real-time disaster reporting, notification, and danger zone visualization, ultimately enhancing public disaster preparedness. Additionally, this research explores key features necessary for an effective emergency reporting system and identifies critical information for public awareness. This initiative not only helps in preventing public entry into hazardous areas but also expedites the dispatch of emergency supplies and equipment by public emergency services to affected areas.

1.3. Research Approach and Thesis Structure

This research follows and is guided by the Design Science Research Methodology (DSRM), which informs the structuring of the thesis into eight chapters.

Chapter 1 introduces the background and motivation for conducting this research, including research objectives and question.

Chapter 2 is a literature review that examines the different technologies used in the disaster and emergency management. The keys features that might be required when developing a mobile application are also discussed. This chapter then compares the mobile applications currently available on the market.

Chapter 3 introduces the design science research methodology along with rationale of selecting this approach for this study. In this chapter, a proposed research model containing different phases to guide the research is discussed in detail.

Chapter 4 focuses on the design solution stage, discussing the detail of prototype application, including its functionalities, UML diagrams and database diagram.

Chapter 5 covers the Implementation stage. Introducing the design architecture for the prototype application and technologies used in its development. The key functionalities are discussed with code snippets.

Chapter 6 evaluates the prototype application using a questionnaire with fifteen participants selected randomly.

Chapter 7 presents the results from the evaluation.

Chapter 8 discusses the changes and improvements made based on the participants' feedback from the evaluation.

Chapter 9 discusses the results in relation to the research question. The limitations and future work are also discussed in this chapter.

2. Literature Review

The literature review is an important chapter in this research. Based on Knopf (2006) opinion, the literature review not only demonstrates the researcher's understanding of existing research on the selected topic, but also provides a foundation for the new research (Knopf, 2006). Lim et al. (2022) states that the literature review also helps identify gaps between the current research and the proposed studies. It could offer a justification for the new research. Furthermore, the literature review can be used to identify and build a framework for the research (Paul & Criado, 2020). Therefore, this research, it reviews various disasters, and emergency management systems, applications and technologies that used in the field, aiming to find the most suitable solution for the disaster management system which could be used for general public.

The literature review in this thesis is structured into the following sections:

Section 2.1 presents statistics on the impact of disaster worldwide and introduces the classification of disasters and emergencies.

Section 2.2 provides an overview of disaster and emergency management, highlighting the importance of Information Technologies (IT) used in this field.

Section 2.3 gives an overview of mobile technologies used in disaster management.

Section 2.4 reviews the key characteristics of mobile applications for disasters and emergencies.

Section 2.5 describes the visualization of mobile applications in disaster and emergency management.

Section 2.6 reviews the existing disaster and emergency mobile applications available in the market.

Section 2.7 examines the current disaster and emergency management in New Zealand. The review aims to identify the strengths and weaknesses of the proposed application for New Zealand users.

Section 2.8 highlights the research question development, identifying gaps in existing disaster-related mobile applications, particularly the lack of integration for real-time reporting, notifications, and danger zone visualization, and proposes exploring how mobile technology can bridge these gaps to enhance disaster preparedness.

2.1. The impacts of Disasters

According to Torani et al. (2019),disasters can be natural or man-made disasters. Their occurrences can range from minor inconveniences to large-scale tragedies and have the potential to cause significant disruption to a community or even a country (Monte et al., 2021). Natural disasters include, for example, storms, floods, earthquakes, tsunamis, and wildfires. Man-made disasters include chemical spills, power outages, oil spills, explosions, and others (Carter, 2008). Disasters can have many impacts, including physical destruction, economic loss, injury and death, psychological distress, and environmental degradation (Caldera & Wirasinghe, 2022).

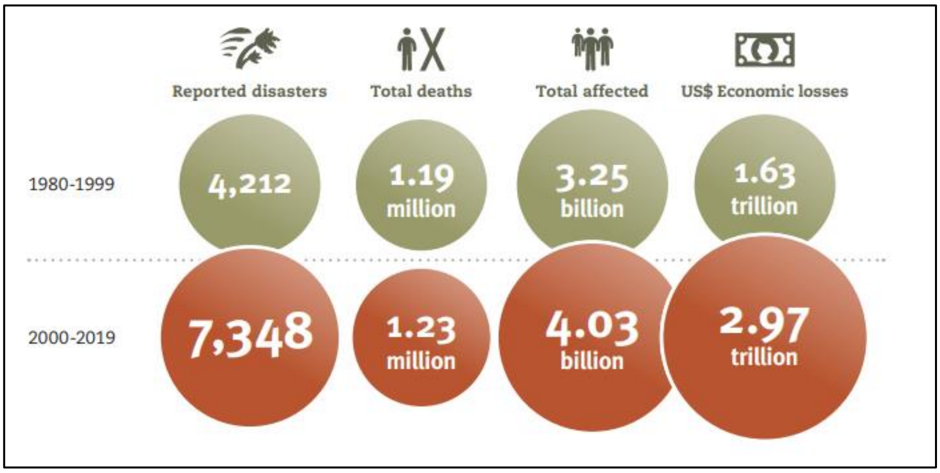


Figure 2.1: Disaster impacts: Comparing 1980 – 1999 with 2000-2019 (UNDRR, 2020)

On 7th July 2021, the United Nations Office released a statistics report summarizing data from 1980 to 2019. According to the report, Figure 2.1 illustrated that the number of reported disasters rose substantially from 4,212 between 1980 and 1999 to 7,318 between 2000 and 2019, representing a 74.5% increase. The number of individuals impacted by disasters grew significantly by 24% from 3.25 billion in the earlier period to 4.03 billion in the more recent one. Furthermore, economic losses increased by 82.7%, rising from $1.63 trillion in 1980 -1999 to $2.97 trillion in 2000 – 2019. These trends, as shown in Figure 2.1, highlight a significant rise in disaster frequency, impact and economic losses over this period. Although, the data shows only a small increase in the number of deaths, the broader impacts of disasters, for example, displacement and extended economic recovery are substantial (Ritchie & Rosado, 2022).

2.2. Importance of IT in Disasters and Emergency Management

When disaster and emergency occur, it is important to be aware of the incident immediately and ensure that general public receives information about the event without any delay. The importance of Information Technology in disaster and emergency management cannot be overstated. Because disaster and emergency management focus on reducing the impact of these events on people and community. This requires timely responses, wise decision-marking, and effective communication (Asadzadeh et al., 2020) . In the past, responding to disaster and emergency situations relied on manual processes. Frontline officers had to gather all the event information and notify the public through broadcasting and television (Chen et al., 2011) . Nowadays, Information Technology has totally changed the way to manage the disaster and emergency. By using advanced digital technologies, it provides more efficient, effective and timely responses to a wide range of disaster events (Lettieri et al., 2009). Asadzadeh et al. (2020) also pointed out that the core object of disaster and emergency management is to minimize the impact on individuals, community and business. Achieving this requires prompt responses, informed decision-marking, and effective communication. Information Technologies enable the quickly gathering, analysing, and disseminating information. An access of such information by government or private authorities is beneficial to provide accurate assessment of disaster situation that further assists to make an informed and coordinated response of action (Dhakal et al., 2022).

Information Technology has an advantage that it can help to provide situational awareness during disaster and emergency situations. (Djalante & Lassa, 2019; Munir et al., 2022). It can also assist in providing the real-time update when connected to remote monitoring systems, which is a great way to assess changing conditions in the field. Such updates when received by the authorities is the crucial aspect of informed decision making by the authorities. For example, remote sensors can be used by emergency services to collect relevant data from disaster affected regions. Such data can be used in the present or future to create evacuation plans for public (Meechang et al., 2020). Advancing Information Technology has also made it possible to provide transfer of information about such events with speed and accuracy through reliable networks. As such, it has significantly reduced the delays that could happen because of manual processing of data (Mizutori, 2019; Mojtahedi et al., 2021).

Information technology has become a vital tool in responding to emergencies and disasters. Not only it has been successful in enhancing situational awareness to provide a better understanding to affected communities, but it has also been able to process such information quickly and effectively in real time frame. Furthermore, IT has brought different groups and organisations to work together to achieve common goal. For example, IT systems are used by different authorities to access available information quickly. This makes it easier for the authorities to identify the need of the situation and allocate resources where they are most effective, that would lead to faster and efficient responses, in disaster and emergency management. (Mukhopadhyay & Bhattacherjee, 2015; Nobre et al., 2019). Also important to notice is that IT as such has been able to connect various authorities through integrated platform. Such platform enables to share updates about the situation, communicate with one another in effective manner and coordinate the use of resources. This has resulted in faster decision making and a reduction in response time, that is critical during emergency response (Robertson et al., 2019; Sakurai & Murayama, 2019).

Now the technology has further moved towards usage of Artificial Intelligence that have gone a step further to provide recommendations in such situations. It is because of IT that it has become possible to process large amounts of data quickly, helping decision-makers make better choices more rapidly during disaster and emergencies situation (Stute et al., 2020).

2.3. Technologies Used in Disaster and Emergency

Today, technologies provide various methods to respond disasters. The main technologies used in disaster and emergencies today include big data analytics, drones, artificial intelligence, blockchain technology, satellites technology, the internet of Things (IoT), and smartphone technology as reviewed below:

2.3.1. Big Data Analytics Technology

With advanced methods and algorithms, big data analytics examine huge amounts of data sets to identify important trends and patterns that can help people to make better decisions (Yu et al., 2018). These huge amounts of data comes from many different sources (Sarker, Wu, et al., 2020). In 2020, when COVID-19 pandemic was affecting people all over the world, big data analytics was very helpful. For example, it analysed data from medical records, social media, mobile phones, and other sources to see how the virus was spreading in real time, and using this information, government and public health agencies were able to quickly identify high-risk areas and take appropriate action (Alsunaidi et al., 2021; Bragazzi et al., 2020).

Big data analytics not only helps manage COVID-19 but is also used to predict natural disasters through the study of past weather patterns. By analysing social media and sensor data, it can track disasters in real time, so that resources to be allocated effectively during relief efforts (Jaber et al., 2022; Sarker, Peng, et al., 2020).

2.3.2. Drones

In modern times, drones are a product of new technology advancement that flies without a pilot on board. It can be remotely controlled by a person from the ground (Chamayou, 2015). Drones have found their wide usage in agriculture and could be seen as a significant potential achievement during emergency disaster response (Ahirwar et al., 2019; Daud et al., 2022). During search and rescue operations, drones can be used as helpful tool that can fly over dangerous areas, where rescue teams might struggle to reach.(Van Tilburg, 2017). They use thermal imaging to identify signs of life, and can also be used for delivering supplies such as food and medication (Zwęgliński, 2020).

2.3.3. Artificial Intelligence (AI)

In Emergency situations, quick and effective responses are essential. AI is now commonly used in disaster management (Abid et al., 2021). AI is able to assist governments, relief organizations, and communities in better preparing for, responding to, and recovering from disasters through data analysis, predictive models, and automated systems (Sun et al., 2020).

According to Development Asia (2021 March), AI has the capability to quickly analyse large amount of satellite images in a short period of time. For example, AI was used to detect landslides during the 2018 Kokkaido earthquake in Japan. While engineers would take about five days to identify damaged areas, AI could do it in just five minutes with 93% accuracy. Similarly, AI used in 2011 Tohoku earthquake in Japan was able to identify damaged houses with 94% accuracy in finding houses that were later washed away by the tsunami.

2.3.4. Blockchain Technology

Blockchain technology is a way to securely and transparently record transactions. It works like a digital ledger where each transaction is grouped into a “block” (Pour, 2021). These blocks are linked together to form a “chain” and are stored on many different computers (Hunt et al., 2022). This system provide a high level of security and trust, because once data is recorded, it cannot be changed (Pour & Gheorghe, 2021). In disaster management, especially during large-scale events involving victims from multiple countries, how to accurately identify the deceased is crucial. Blockchain technology is used in this area to securely store postmortem and pre-disaster data of missing persons and then to perform the comparison, which could reduce data breaches and errors (Alsalamah & Nuzzolese, 2020).

2.3.5. Satellites Technology

Satellite technology refers to use of artificial satellites for various purposes, such as scientific research, communication, remote sensing and navigation (Kopacz et al., 2020). Artificial satellites are devices launched into Earth's orbit that can operate in space for extended periods and transmit data back to Earth via ground-based receiving stations. Satellite technology has been widely used in disaster management. For example, satellite imagery is used to monitor floods, earthquakes or landslides (Pachika, 2022). Satellite technology, through the Global Positioning System (GPS), enables in locating stranded or trapped individuals and supports the delivery of emergency supplies such as medications and food to affected areas (Chetty, 2023). When ground-based communication infrastructure is damaged, satellite communication ensures the smooth execution of rescue and relief operations (Long et al., 2021). Satellites continuously monitor the Earth's surface and atmosphere, providing data that helps understand patterns and triggers of natural disasters, which can reduce future disaster impacts (Zhao et al., 2021).

2.3.6. Internet of Things (IoT) Technology

IoT enables to collect data and share data automatically (Bhandari, 2022). It works by using sensors and software to communicate with each other (Sharma et al., 2021), which helps gather real-time information and automate tasks (Li & Da Xu, 2020). IoT uses early warning systems to monitor things such as increase in water levels, gas leaks or potential flood/ earthquake alerts before they reach critical level (Esposito et al., 2022). IoT devices are also helpful in managing any emergency supplies such as medical equipment to ensure resources are available and allocated to the areas that need them (Krichen et al., 2023).

2.3.7. Smartphone Technology

The advent of smartphone technology has shown significant changes in how people interact with their environment (Krichen et al., 2023). Smartphones are not just devices that are used for communication, anymore. They are advanced technological devices that use Global Positioning System (GPS) navigation, has sensors, high-tech cameras, Bluetooth and data processing capacities in a real-time manner. Such features make smartphones essential tools during disaster management (Pal et al., 2020). These functionalities along with live data transmission and sharing, video calling, and location tracking can play important role during emergencies. This is because smartphones can provide quick and effective communication, data support, that is crucial in managing crisis (Cicek & Kantarci, 2023; Latvakoski et al., 2022).

The government authorities can access information around the world when they have real-time data easily accessible (Budimir et al., 2021). This immediate access allows the agencies to coordinate resources and rescue efforts promptly (Pine, 2017). Smartphones, as such, acts as a great tool to send out emergency alerts and notifications. Such information spreads critical information to public quickly. It also notifies any threats nearby so that the public can take appropriate precautions (Escolano et al., 2023). A smartphone with high resolution camera and sensors, and special applications is a powerful to collect vital information and share it across (Wang et al., 2020). People who manage to rescue themselves in a disaster event can use this data, photos and videos to send to emergency teams within a certain time framework (Astarita et al., 2020). Such data is helpful in providing a clear picture of the situation to authorities for effective coordination (Aten et al., 2011; Maryam et al., 2016). With the advent of social media applications such as Facebook, Twitter, smartphones have furthermore provided a channel of communication to raise public awareness especially in the event of disaster or emergencies (Aboualola et al., 2023). Sharing of such information is valuable during emergencies, and is also a great tool to strengthen communities (Bachmann et al., 2015).

With a high population using smartphones, it has become easy for individuals to access and contribute towards any information related to disaster. As discussed earlier, we are using technologies such as smartphones and mobile for this research as they are user-friendly, accessible, adaptable and widespread that makes them a useful tool for communication enhancement.

2.4. Key Mobile Application Features of Disaster and Emergency Management

The previous sections showed how smartphones are effective tools and can be used during disaster management. Smartphones use mobile application, and such applications rely on the hardware and operating systems of the devices to perform functions effectively. According to Paul et al. (2021) research, mobile applications are very important in handling disaster and emergency because they allow quick communication, sharing locations, and spreading information. The mobile applications offer real-time updates and send disaster alerts to assist with sharing of location.

A lot of research has summarized such mobile applications and their features that could be included for effective management of disasters. These features are divided into 4 phases: preparation, response, recovery and mitigation, that are discussed below in detail. (MoldStud, 2024; Syukron et al., 2024).

1. **Preparation phase:**

According to Fernando et al. (2019), disaster preparation would mean a plan and preparation for potential disasters that can occur in the near future. It includes activities that focus on developing emergency plans, educating people, and training response teams. The preparation phase would include mobile application that focus on preparation aspect of early warning alerts, forecasting of disaster and preparing appropriate tips or tutorials. (Syukron et al., 2024). An early warning system can allow user to find a secure place before a disaster, so they can protect themselves (MoldStud, 2024). Disaster forecasting could also mean support weather forecasting that can monitor wind movements, predicting unnatural forces at play (Srivastava et al., 2020). Syukron et al. (2024) identified that information tips and training should be a part of mobile application to help people understand and prepare for disasters.

2. **Response phase:**

During response phase, mobile applications would help with real time communication and provide guidance during a disaster. They can send emergency alerts and allow GPS location sharing (Bachmann et al., 2015; Toledo et al., 2017). Unlike early warning alerts, real-time alerts are sent during a disaster, not before it happens (Li & Ulaganathan, 2017). Location sharing provides users geography information about the disaster's location or affected area, often displayed on an interactive map for real-time tracking (Sharma et al., 2020).

3. **Recovery phase:**

In the recovery phase, it is to help individuals and communities to return to normal after a disaster. The features may be included like connecting users with relief resources, update on recovery efforts, donation platforms, access to mental health services (Furutani & Minami, 2021; Tan et al., 2017). For example, disaster updates allow user to view disaster statistics (Syukron et al., 2024).

4. **Mitigation phase:**

The mitigation phase focuses on reducing the long-term impact of future disasters. They may offer risk assessments based on location, tips on strengthening homes or buildings, and tools for monitoring potential hazards (Suzianti et al., 2020). This could, for instance, also recommend actions to be taken after the calamity, revision as well as enhancing of disaster preparedness plans, and enhancement of community-wide measures that can be taken to understand any future events (Wahyudi, 2022).

2.5. Visualization in Disaster and Emergency Management

Visualisation is equally important during disaster and emergency management. It is an easy way of turning complex date into visuals. Different maps, charts and diagrams can be made to simplify the way data can be understood (Dusse et al., 2016). This helps in making quick decision making while looking at disaster details of the affected area, where people are located and resources are required. (Kubíček & Staněk, 2006).

With visualizations, organizations can better understand the situation, plan effective responses, and coordinate aid (Nurbekova et al., 2020). For example, real-time maps can show where help is needed most. Additionally, visualizations help identify areas that might be at risk in the future, improving preparedness and safety plans (Neene & Kabemba, 2017).

2.6. Existing Disaster / Emergency Mobile Applications in the Market

Both the Apple Store and Google Play Store currently offer a wide range of disaster and emergency mobile applications. In this section, these applications are categorized based on their functionality before, during, and after a disaster, with some examples of existing applications for each phase.

- **Before a Disaster /Emergency**

Mobile applications like **Weather Underground** (see Figure 2.2) provide near real-time updates on 18 types of natural hazards (Weather Underground, 2024).

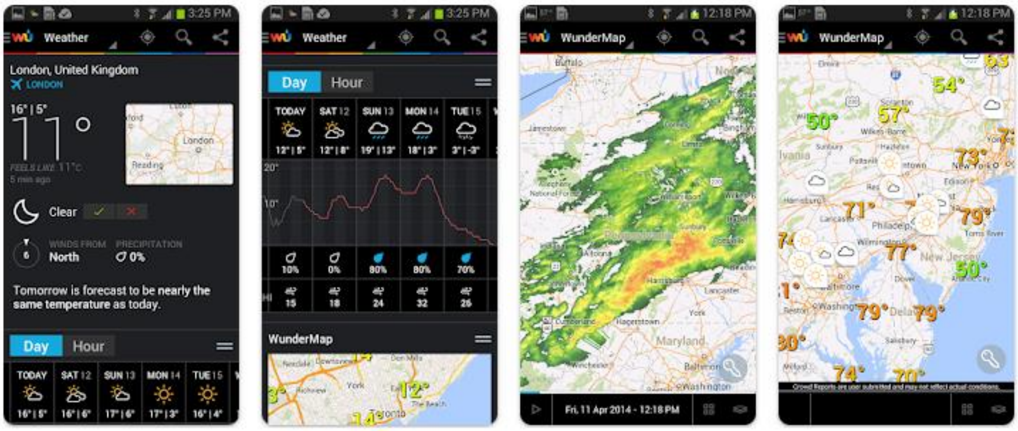


Figure 2.2: Weather underground mobile application (Weather Underground, 2024)

Similarly, **Disaster Alert** (see Figure 2.3), offers important hazard alerts and safety information to people worldwide. However, these applications do not let users to report any natural or man-made disaster events.

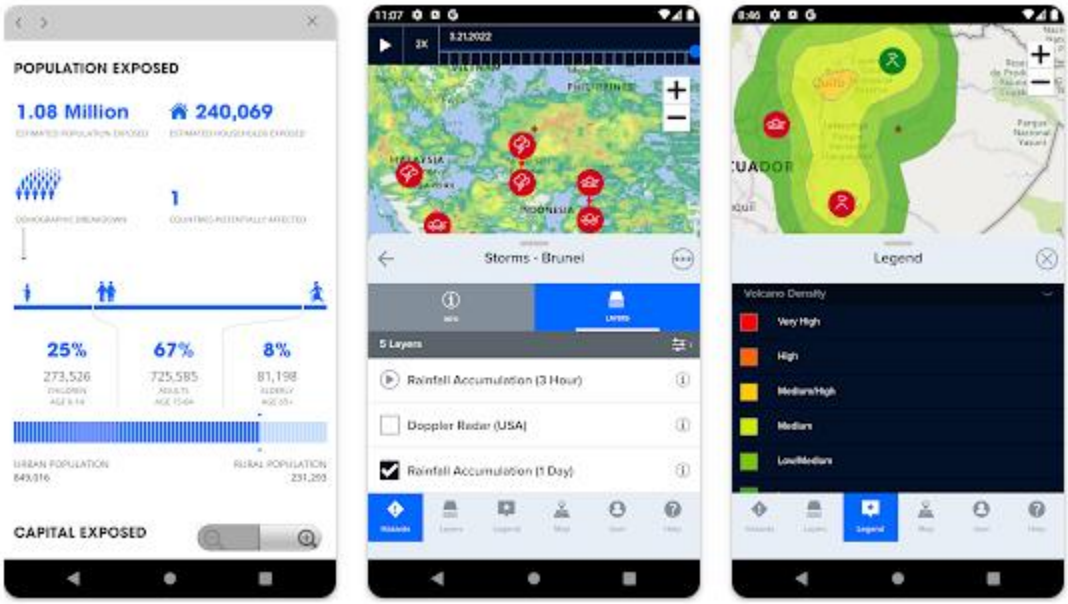


Figure 2.3: Disaster Alerts mobile application (Pacific Disaster Center, 2024)

- **During a Disaster /Emergency**

During a disaster or emergency, users can monitor ongoing events through mobile application like the Natural Disaster Monitor (see Figure 2.4). This app provides real-time updates on various natural disasters worldwide, including earthquakes, volcanoes, tsunamis, and floods. It also sends global disaster alerts, helping users stay informed about current threats and their impact (dominoc925, 2024). However, this type of applications primarily focuses on natural disasters and does not have features for user to report a disaster.

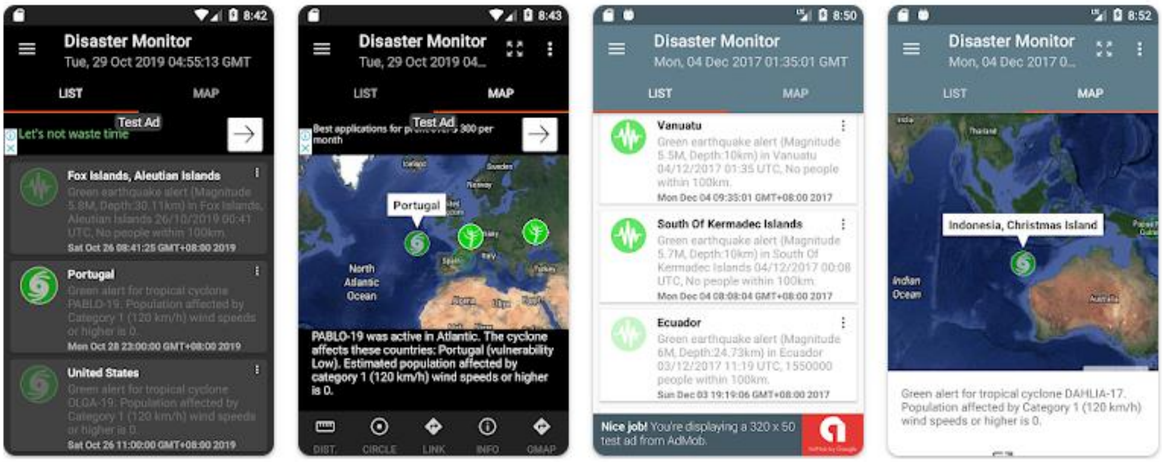


Figure 2.4: Natural Disaster Monitor mobile application (dominoc925, 2024)

- **After a Disaster /Emergency**

After a disaster or emergency, social media application like Facebook can be used to notify family and friends that you are safe. These platforms allow users to post updates and check in, helping to keep ones informed. However, it's important to be cautious as these platforms may also be a source of misinformation.

2.7. New Zealand Disaster and Emergency Management

New Zealand, located in the South Pacific Ocean, is at high risk for natural disasters because of its location and the many tectonic plates around it. As a result, the country frequently experiences earthquakes, volcanic eruptions, tsunamis, landslides, and severe floods (ehinz, n.d). Because of the potential for significant hazard impacts over the past decades, New Zealand has put in place an advanced system for disaster and emergency management (New Zealand Civil Defence, 2023). The New Zealand Disaster and Emergency Management system has been developed to reduce the social, environmental and economic losses from disaster events. It outlines each agency's responsibilities during a disaster event, coordinated through the National Emergency Management Agency (NEMA). The system is based on four key components: Risk Reduction, Emergency Readiness, Emergency Response and Recovery. In an emergency, NEMA coordinates and supports the affected agencies and local authorities to ensure the response is effective, efficient and integrated. Many organizations across central and local government are represented in the NEMA's emergency management system, including first responders, the military, local search and rescue teams, public health and welfare services, Environmental Protection Authorities and the media (National Emergency Management Agency, 2024). NEMA works closely with local authorities to develop local emergency plans which outline how they will manage an emergency in their region. These plans include detailed evacuation procedures, listing essential services that need to be provided in an emergency and identifying the roles and responsibilities of key personnel (DPMC, 2023).

Before a disaster, New Zealand operates several national early warning systems which alert the public to potential emergencies. The NEMA monitors seismic activity, floods and cyclonic conditions and is activated when an emergency is detected. NEWS triggers communication via radio, television, SMS, social media and other channels to alert people in affected areas of an upcoming event and guides appropriate protective actions. NEMA also operates the New Zealand Civil Defence Emergency Network (CDN), which provides emergency warnings to mobile phones and supports civil defence messages via the Internet (New Zealand Civil Defence, 2023).

Mobile applications, such as GeoNet (See Figure 2.5) for earthquake alerts and the MetSevice application for tracking updates on extreme weather events, are currently used at this stage (MetService, 2024).

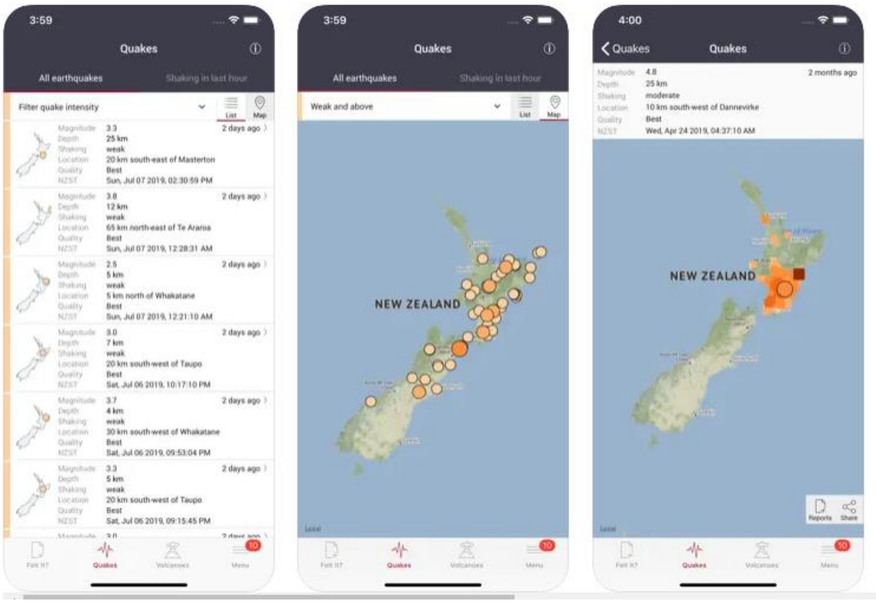


Figure 2.5: GeoNet Quake Mobile Application (GNS SCience, 2024)

Red Cross Hazards is another application used in New Zealand. It provides an emergency response kit and plan to help the public how to prepare for disasters. This application also has a feature that allows users to focus on specific location and share updates with friends and family about the situation (Nelson City Council, 2023).

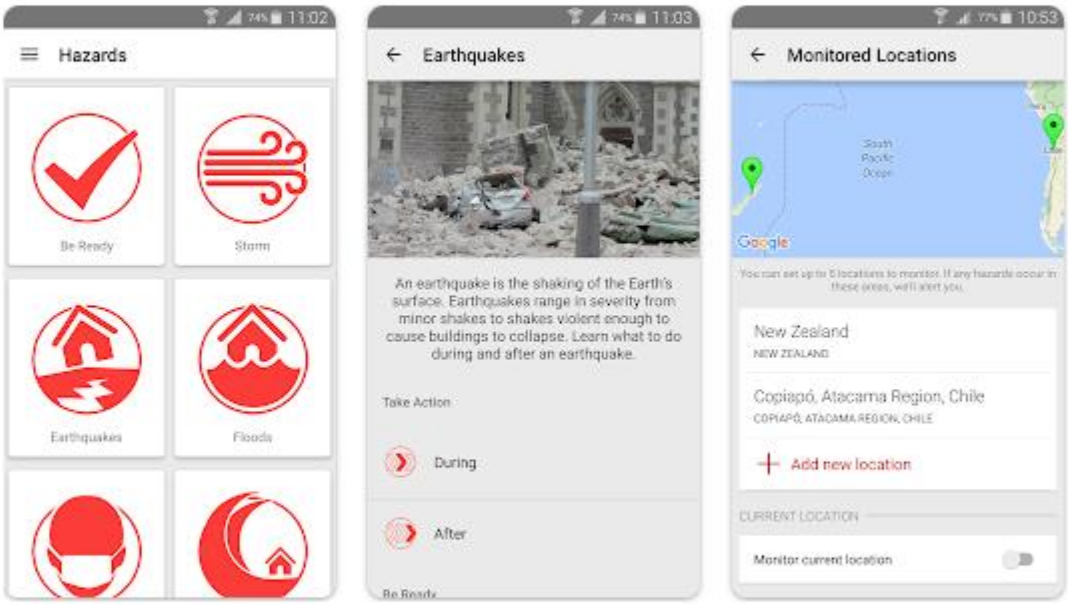


Figure 2.6: Hazards – Red Cross Mobile application (New Zealand Red Cross, 2022)

During and after a disaster in New Zealand, the public still relies on the NEMA website, radio, local civil deference emergency management social media or online news sources, such as Stuff and Radio NZ, to stay informed (New Zealand Red Cross, n.d.).

The purpose of this research is to create a platform for public to actively participate and contribute information, while still receiving updates from authorities. The review of the applications indicated that no current application is available in the market that allows users to report disasters and contribute to the platform, in real time framework, including visualisations of danger zones, from convenient devices such as mobile phones. This gap highlights the need to develop a new mobile application that can address this idea of information sharing during disasters.

2.8. Research Question Development

From the literature reviewed, it is clear that mobile technology plays a significant role in disaster management, offering essential features for real-time updates, emergency alerts, location sharing, and disaster response. However, despite the advancements in disaster-related mobile applications, a critical gap remains. Current applications primarily focus on alerting users and sharing general information during or after disasters, but few applications allow users to actively report disasters or visualize danger zones in real time. The ability for individuals to contribute information, while also receiving updates from authorities, has not been fully explored.

Additionally, the existing mobile applications in the market (e.g., Weather Underground, GeoNet, Red Cross Hazards) tend to focus on specific disaster phases like preparedness or recovery, but there is a lack of integrated systems that combine reporting, notification, and real-time visualization in one platform. This leaves a gap in enhancing disaster preparedness by empowering both authorities and the general public to collaborate more effectively in real time.

Thus, the research question, **"How to use mobile technology to enhance real-time disaster reporting, notification, and danger zone visualization to support disaster preparedness?"**, emerges from the literature gap. The aim is to explore how a mobile application can bridge these gaps by offering a platform for both public participation and official updates, integrating reporting and visualization of danger zones, and ultimately improving disaster preparedness.

This question addresses the need for a comprehensive system that not only informs users but also allows them to contribute valuable information in real time, thereby enhancing disaster management efforts and fostering better preparedness in disaster-prone areas.

2.9. Chapter Summary

This chapter reviewed the various technologies commonly used in the disaster and emergency management. It also discussed mobile applications used before, during and after a disaster both worldwide and in New Zealand. The literature finding review indicated that mobile applications that can contribute for reporting and visualisation of affected areas during a disaster, will help users to better assess the situation and act accordingly.

3. Research Methodology

The purpose of this research is to explore ways to enhance real-time disaster reporting, notification, and danger zone visualization to support disaster preparedness. To address these challenges effectively and develop a practical solution, a mobile application prototype has to be created, and its functionality demonstrated. This prototype aims to demonstrate the significance and benefits for the community and assist in improving disaster preparedness. The methodology of research involves developing mobile application and assess its performance. To define objectives, create prototypes and validate its effectiveness, Design Science is an optimal approach. This research introduces new advantages and innovative solutions while tackling existing targets. Therefore, Design Science Research (DSR) is the choice of methodology to generate this prototype. The approach here is to identify research motivations, define objectives, develop the system, demonstrate its utility and evaluate its impact. Following sections provide detailed steps in Design Science.

Before the application of DSR methodology, a literature review has to be conducted to introduce background and state of knowledge in disaster management field. This review also aims to understand context of existing solutions and advancements in technology relevant to application development. Insights gained from previous research will be a guide to the assessment of prototype's potential benefits.

DSR methodology was employed once the literature review was conducted. Information systems would also be approached using independent and customised methods as this system is not constrained by specific methodologies. Section 3.1 outlines DSR methodology along with prior research perspectives. Section 3.2 will explain the research framework and its phases.

3.1. Design Science Research Methodology (DSRM)

The origins of the DSRM can be tracked back to the field of Information Systems (IS), though it was not invented by a single individual. The theoretical foundation of DSRM from contributions by multiple scholars (Venable et al., 2017). However, Alan R. Hevner is widely recognised as one of the key figures who formally introduced design science into information systems research. In 2004, Hevner, along with other co-authors, published a paper titled "*Design Science in Information Systems Research*", which outlined the core principles and research process of design science and explained how the creation and evaluation of artifacts can generate theoretical knowledge (Rothenberger & Kuechler, 2012).

Guideline	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Figure 3.1: DSR Guidelines (Hevner et al., 2004)

Hevner et al. (2004) presented 7 guidelines through their research work by offering a framework to conduct research in the field of information systems (See Figure 3.1). These seven guidelines highlight the significance of developing artifacts and evaluating them to address real-world challenges. They argued that a meticulous designing and a thorough testing is required for artifacts to be effective. It ensures that artifacts not only solve problems at hand but also contribute to new knowledge leading to better innovations.

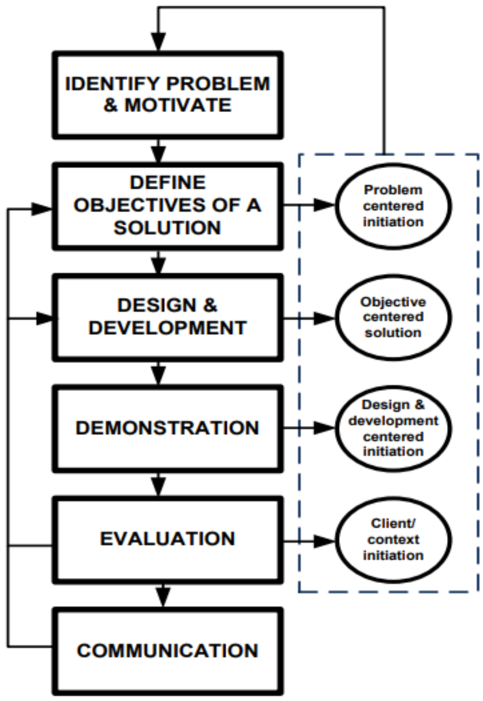


Figure 3.2: DSRM process model (Hevner et al., 2004)

These seven guidelines provided a process model to conduct design research in a well-structured manner (Hevner et al., 2004). As can be seen in Figure 3.2, the model has several key stages: **Problem identification** that involves recognising and defining the issues clearly and addressing the research. Then is the **Object Definition,** when the identified problems require specific goals and objectives. In Design and Development stage, the solution or artifact is created that meets those objectives. The next stage is **Demonstration**, where solution is applied that illustrates how well it solves the problem. The **Evaluation** stage assesses the solution’s performance and its impact to ensure it meets research goals. The final stage is the **Communication** stage, where the results and findings are shared with research community and relevant stakeholders. (Dresch et al., 2015; Hevner et al., 2004; Offermann et al., 2009; Wieringa, 2014).

Based on the study of DSRM above, DSR aims to solve issues by developing a new artifact. To address any real-world problem, researchers must address two knowledge challenges. First, they must recognize the problem and evaluate if the suggested solution successfully addresses it. Second, they should to predict the results of implementing the design in the given context (Wieringa, 2014). The problem in our case is that there is currently no single system or app, there integrates all the essential features for disaster management. Such system or app is predicted to combine all necessary functionalities to provide public with vital information and improve disaster preparedness.

Peffers et al. (2007) describe that DSR as involving steps like building theories, developing systems, experimenting, and observing results. However, the design process does not always have to follow these exact steps. For example, Winter and vom Brocke (2021) focus on creating solutions to specific problems by carefully building and testing them. This research follows a similar path with four main stages: proposing a solution, developing the system, running simulations, and evaluating the outcomes. By suggesting a solution first, we can define the types of system needed, which is then developed and evaluated through simulations.

While many researchers test their systems in real-world situations, design science research can be done in different ways. Instead of fully deploying the system, it can focus on evaluating the knowledge gained from the system's design (Venable et al., 2016) . The goal of design science research is to match the research objectives, such as evaluating how well the system works (Méndez Fernández & Wieringa, 2013). Our study explores the advantages of using real-time disaster reporting, notifications, and danger zone visualization. The design science approach aims to assess how these features improve disaster preparedness and response. Since DSR often creates new knowledge for the field (Kuechler & Vaishnavi, 2012), it's an ideal method for this research.

3.2. Research Framework

Our research framework draws from both a literature review and the DSRM. It includes six different distinct phases, illustrated in Figure 3.3, with each phase described in detail below.

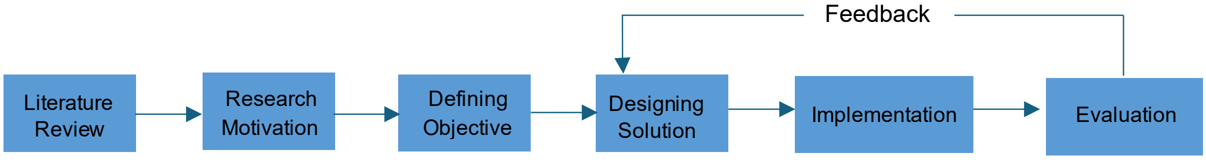


Figure 3.3: Research Framework *(source: Author)*

3.2.1. Phase 1: Literature Review

The literature review conducted in this research highlights the current limitations in disaster management systems. While many existing systems or applications focus on early warnings, weather forecasting, and public education about natural disasters, they fall short in several areas. Typically, notifications are sent by the government emergency agencies only after a disaster has occurred, and the public receives these alerts via phone messages. For real-time updates, they still rely on radio, TV, or civil defence websites.

Our review identified a significant gap: recent applications do not offer features for users to report disaster events or view danger zones directly on their mobile devices. We also explored the use of modern technologies in disaster management, noting that smartphones and mobile applications are increasingly being utilized in this field. This evidence supports the potential of using mobile technology to fill the existing gaps.

In this research, we propose developing a mobile application designed to address these shortcomings. This app would allow users to report disasters, receive real-time notifications, view danger zones on a map, and access updated news about the events.

Most of the literature was found through searches of the Auckland University of Technology library and Google Scholar. A small portion of the sources came from websites, such as those related to New Zealand Disaster and Emergency Management.

To ensure that all sources and technologies are relevant and up-to-date, the search primarily included literature published within the last 5 years, with information on mobile applications being used from the past two years.

3.2.2. Phase 2: Research Motivation

Design science research can solve the problems present in the current system or build a solution that is more effective and efficient than previous ones. In chapter one – Introduction section, we have identified that in existing emerging disaster management systems, the public mostly relies passively on notifications issued by the government emergency agencies. If they need to learn more about disaster information and stay informed, they still have to go through the Civil Defence website, radio, or television.

However, disasters can happen at any time, and aside from natural disasters, there may also be man-made disasters, such as car accidents. If an incident is not discovered or reported in time, it can delay emergency response efforts. Therefore, the motivation of this research is to develop an application that allows the public to actively report a disaster using widely available mobile phone apps, enabling them to access disaster information and danger zone at any time, rather than relying on calling 111 to report the disaster, if someone faces a language barrier and cannot speak fluent English, it could lead to confusion. Additionally, it prevents relying solely on websites, television or radio for information. We also explore the potential features and benefits that our research solution may provide. At least, the proposed features and benefits should facilitate real-time disaster reporting, receiving notifications, and visualizing danger zone, while also enhancing disaster preparedness.

3.2.3. Phase 3: Defining Objective

In this phase, we have clearly identified that our goal is to develop a mobile application. To achieve this goal, we need to plan and analyse all the required features which will be discussed in Chapter 4. Additionally, the development technologies such as programming languages and platforms will be covered in Chapter 5.

3.2.4. Phase 4: Designing Solution

After defining the research objective, we begin to build up the prototype mobile application in this phase. The development process is guided by Software Development Life Cycle (SDLC), which comprises three major stages: design, development, and testing. In this phrase, we focus on designing solution as discussed in Chapter 4.

The proposed system architecture is presented during this stage. Unified Modelling Language (UML) diagrams are used to visualize and document the structure and behaviour of the system. The UML diagrams include: use case diagrams to illustrate how use cases interact between actors and system, where actors represent the admin who manages the system and the general public as users of the system in this research; a class diagram to demonstrate the structures, methods, and attributes of the system. Additionally, database diagrams, which are introduced in this phase to show how data is stored in the database. The functional requirements and non-requirements are also identified in this phase.

3.2.5. Phase 5: Implementation

The implementation phase involves converting the design from the previous phase into a working prototype solution, as detailed in Chapter 5. The implementation phase includes both frontend and backend development. Frontend development focuses on creating the mobile application prototype that allows users to report disaster events, receive disaster event notifications, view danger zones and access updated disaster news. The backend development includes administrator to verify events, send events notifications to user mobile application, and manage events.

The system development architecture, technologies and Integrated Development Environments (IDEs) used are discussed, including those for frontend, backend and database development. Additionally, this phase addresses the testing process, tools and prototype code storage. The design of frontend and backend UIs, key functionalities, and relevant code snippets are also discussed.

3.2.6. Phase 6: Evaluation

The final stage assesses the effectiveness and usefulness of the mobile application prototype developed. This evaluation will be conducted according to a structured framework with specific actions, which will be detailed in Chapter 6.

The process of evaluation has several key steps: random selection of 15 participants, simulating a mock emergency event that uses the prototype, and evaluating application of the prototype through collected feedback from a prepared questionnaire. Such approach ensures thorough understanding of prototype's application, its performance and the impact in real world scenarios.

After evaluation, we strive to make change based on the feedback received. This may include revisiting designing, implementation and evaluation phases as shown in Fig 3.3. However, if significant redesigning is required, it would be beyond the scope of this research.

This research obtained its approval from the Auckland University of Technology Ethics Committee (AUTEC), the Ethical Approval can be found from Appendix A.

3.3. Chapter Summary

This chapter explains Design Science Research (DSR) methodology and the reason to choose this method. It explains DSR used in each phase of the research, emphasising how this approach contributed to the development of mobile application prototype. The prototype, further, addresses the challenges of disaster management, specifying the lack of features in current mobile applications, that allows users to report disaster.

4. Designing Solution

In the previous section, the research methodology and framework were introduced to determine why the prototype is needed and what it should do. This section discusses how to begin designing solution for the mobile application prototype. It is necessary to prepare the design system specification to identify both functional and non-functional requirements, create UML diagrams, design the database, However, before starting the actual development, it's essential to choose the right Software Development Lifecycle (SDLC), as it helps how to build the prototype effectively.

4.1. Mobile Application Prototype Development Process

The SDLC is defined as a process that helps developers and project managers create software in an organized way (Scroggins, 2014). It breaks the process into smaller stages to make sure the final product meets customer expectations and is delivered on time and within budget (Sharma, 2017). According to the Figure 4.1, these stages are commonly divided into six stages:

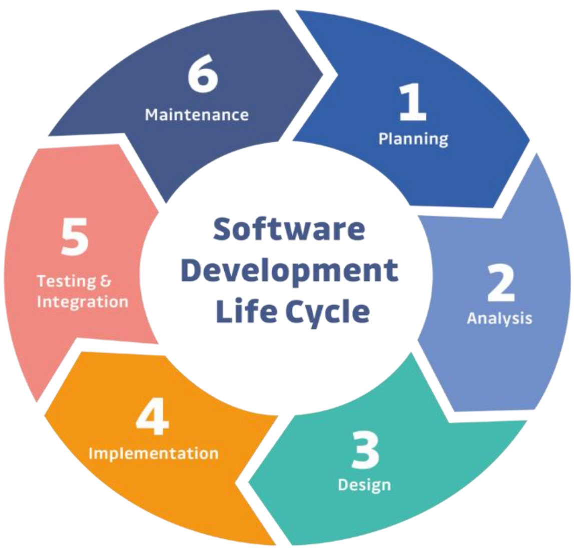


Figure 4.1: Software Development Lifecycle(Medium, 2023)

Planning stage: This stage outlines how to build the software and includes elements such as creating a timeline for the project, assigning tasks to different group of people, figuring out how much money will be needed to build the software , and also thinking about how to handle potential risks that could happen during the project (Kute & Thorat, 2014). The planning stage is like making a detailed map before going on a trip, so everyone knows where to go and what to do along the way.

Analysis stage: This stage is to figure out exactly what the software should do. The analysis stage helps development teams understand the problem, for example, what features the system should have and what problems it will solve. It requires to communication with various stakeholders to gather information, and then the team documents everything the software needs to do, which are the requirements. Additionally, this stage, it involves studying different feasibilities, for example, checking whether the teams has enough resources or expertise to build the system, and if not whether these can be acquired without difficulties (Ragunath et al., 2010). The analysis stage is like planning to build a house, before starting, one should understand what the house needs, how big it will be, and what materials are required.

Design stage: This stage involves figuring out exactly how the software will work and how it will be built. It includes creating various models and diagrams, such as UML diagrams, to visually represent how the system functions. These visual tools help in planning and understanding the software's structure and interactions (Langer & Langer, 2008). Examples of diagrams commonly used include:

- Use Case diagrams illustrate how users interact with the system and what functionalities the system provides (Scroggins, 2014).
- Class diagrams show the structure of the system by detailing the classes, attributes, and methods (Scroggins, 2014).

Additional, database design is a crucial part of this stage. It involves creating diagrams show how data will be organized, stored, and accessed in the database (Olorunshola & Ogwueleka, 2022). The design stage is like drawing blueprints before building a house, ensure that every aspect of the software is planned out before actual development begins.

Implementation stage: in this stage, programmers start writing the code to create the software based on the design plans. It's like starting to build the house using the blueprints. Although testing is a separate stage, some testing happens during implementation to catch any issues early and fix them (Raj et al., 2014).

Testing stage: After the software has been built, it is important to check if it works as it's supposed to. This stage like a final inspection before a house is ready for people to live in. Types of testing unit testing, which involves testing individual parts or "units" of the software, and automated testing, which uses special tools or scripts to automatically test the software (Jindal, 2016).

Maintenance stage: This stage involves working with deployment. Once the testing phase is completed and the software is approved, it's released for use, whether for internal use or for clients. Th maintenance stage, which occurs post-deployment, supports fixing bugs, making necessary improvements, and updating the software to adapt to new requirements if required (Pukdesree, 2017).

As discussed above, the SDLC is helpful process that breaks down software development into smaller stages, that ensures the final products meet expectations. However, when it comes to how these stages are approached, there are two common methods: Waterfall and Agile (Palmquist et al., 2013). Waterfall is a traditional, step-by-step approach. Each stage is completed before moving to the next stage, making it easier to manage but less flexible if changes are needed later. On the other hand, Agile is a more flexible method where teams work in short cycles, making adjustments based on ongoing feedback (Senarath, 2021).

While Agile offers more adaptability, for the purpose of this research, Waterfall proves to be the appropriate approach due to the clear structure and specific requirements we need to fulfil, as previously discussed in earlier sections. There will be no need for any extra features or significant changes or disruptions later in the development process.

This allows researcher to concentrate fully on completing each stage thoroughly before moving to the next.

4.2. Mobile Application Requirements Specification

This section outlines the requirements and the UML diagrams for the mobile application prototype. To figure out these requirements, it is first essential to understand how the system should work. The proposed system architecture includes the necessary activities, as illustrated in Figure 4.2.

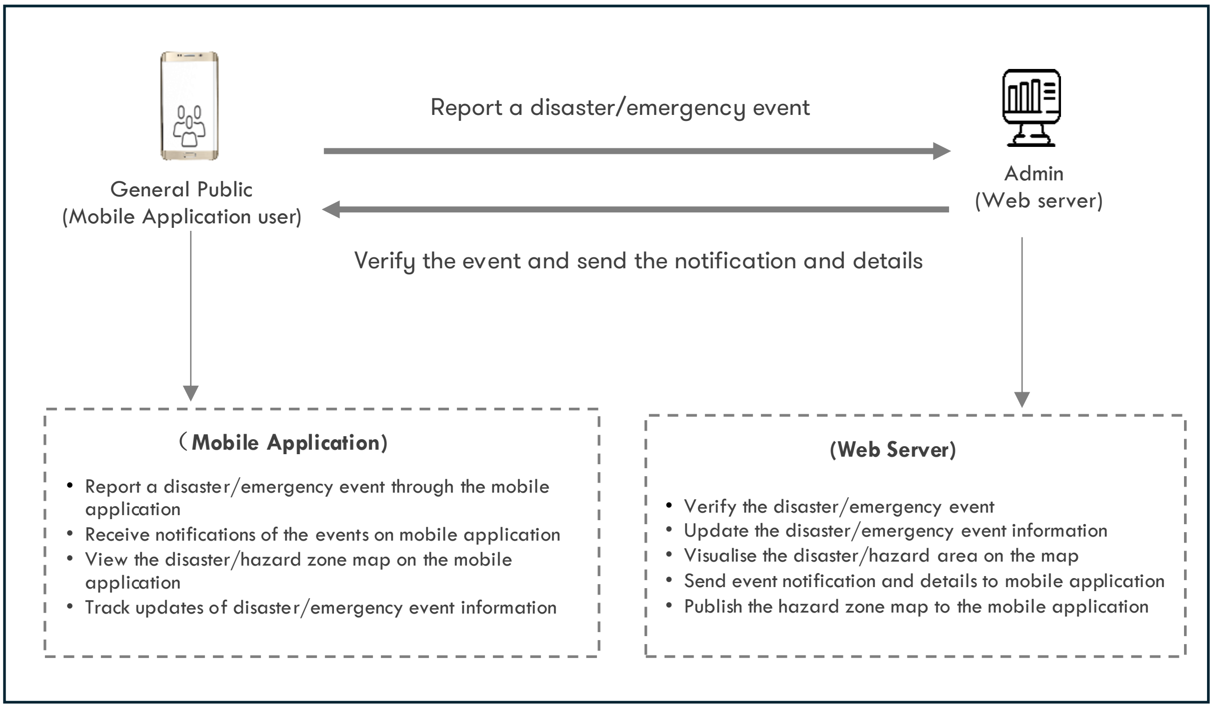


Figure 4.2: Proposed Disaster Emergency Management System Architecture *(Source: Author)*

To report a disaster or emergency event, the general public will use the mobile application to submit the incident. The admin on the web server will then verify the event to prevent fake reports or update event details. After verification, the admin will send a notification to alert all users of the application. Two systems need to be developed in this prototype: the mobile application and the web server.

The activities for mobile application include allowing users to report disasters when they occur, receiving event notifications, viewing the disaster hazard zone, and tracking updates of disaster information. The web server system should enable the admin to verify the disaster events, update event details, visualise the disaster danger zone on a phone map, and send notifications, along with the danger zone map and details, to be published to mobile application.

4.2.1. User Characteristics

In this prototype design, two user roles are included, as listed in Table 4:1 below:

User	Description
users (Mobile Application)	• The **users** are defined as individuals from the general public who interact with the designed mobile phone application. • The users can use the application' features. • Since this prototype is designed for research purposes, certain age groups, such as users under a specific age, are not considered in this research.
admin (Web Server)	• The admin is an administrator that manage disaster management system on the web server.

Table 4.1: User Characteristics (source: Author)

4.2.2. Functional Requirements

According to Tiun et al. (2020), functional requirements provide a clear outline of what the system should do, it is the specific actions, tasks, or behaviours that a system must perform. These requirements define the features or operations that the system needs to carry out in order to meet its objectives and fulfil the needs of its users (Guevara-Vega et al., 2019). In this prototype, all functional requirements are introduced and described below:

A) Functional Requirements of Mobile application

User management:

- The mobile application user should be able to register an account.

- To register, the user should enter a username and password. (Since this is a prototype for demonstration purposes, the system does not require any constraints, validation or format on user registration).
- Once registered, the user should be able to log in to the application.
- After logging in, the user should have access to three features: `NEW EVENT`, `Current EVENTS`, `NEWS`.

Reporting Events System:

- The user should be able to report a disaster / emergency event by clicking on the "NEW EVENT" button.
- To expedite the reporting process and allow users to report a disaster quickly, the design intended to simplify the report process:
    - After clicking on the `NEW EVENT` button, a map should be able to pop up, and by default, it should be able to automatically captures the user's current location with a pin marker on the map. If the user's current location is not disaster site, they shall be able to browse the map and manually pin the correct location.
    - After pinning the correct location, the user should be able to fill out the report form. The form should contain the event name to specific the type of event, for example fire or earthquake; the data and time when the event occurred; the level of the event, such as moderate or critical; the location (in case the user did not pin the correct location on the map, they can still enter the location manually, for example, `Auckland City`); information describing the disaster event; the name of the person reporting the event, and an estimated of how large the affected area is, such as 1 km or 2 km.
    - The user should be able to provide the event details in the report form, if time is limited, only two fields in the form are mandatory to complete: the

event name and the date and time of event occurrence. Other fields are optional.

View Current Disasters Events:

- If the user wants to view current disaster events, they shall be able to click on the `CURRENT EVENTS` button to access a list of all ongoing disaster events and view the details of each event.
- If the user wishes to view the danger zone areas of a particular event, they shall be able to click on the event information, which will pops up a map displaying the danger zone visualisation.

Emergency Management News:

- If the user wants to check emergency management news from the New Zeland National Emergency Management Agency, they shall be able to click on the `NEWS` button within the application. The news should be embedded directly in the application, so users do not need to open a browser to view the latest updates.

B) Functional Requirements of Web Server

User management:

- Considering security concerns, the web server should not allow any user to self-register. An initial admin account should be created automatically when setting up the database. Once the admin is logged into the web server, they should be able to create additional users and assign roles on those users. For example, roles can be predefined by the admin, such as, “Administrator”, or “Standard User`, each with different levels of permissions to access specific functionalities. As this is a prototype for demonstration purposes, there is no need to create other users beyond the admin in the version.

Disaster Events Management:

- The admin should be able to manage disaster events submitted by mobile application users. Managing events includes viewing event details, updating event information, and deleting disaster events.
  - To view the details of an event, the admin should click on the `View` button, which will open a screen displaying all the information about the event.
  - To View the danger zones of an event, the admin should be able to click on the `View Hazard Areas` button to see the affect areas on the map.
  - To update event details, the admin should be able to edit all disaster information, including the event name, event date and time, severity level, affected area, the exact position of the disaster by longitude and latitude, event address, description, reporter name and contact information.
  - The admin shall be able to delete a disaster event.
  - Considering this prototype focuses on mobile application users actively reporting disasters, and the admin verifying these events, therefore, the function to add new events on the web server is not required in this version. This feature can be included in the full version for the next stage of research.
- When mobile application users report disaster events, the default status of these events on the web server is set to `Pending`. The admin shall be able to verify the event. Once an event is verified, a notification should be automatically sent, the events details will be published in the mobile application. Mobile application users should then receive the notification and be able to view the full events details.

4.2.3. Non-Functional Requirements

Non-Functional Requirements explains how well system a system should work. It focuses on the overall quality and user experience rather than specific functions (Chung et al., 2012).

A) **Performance**: According to Ameller et al. (2012), performance is to measure how efficiently the system operates. In this prototype, it is crucial that both mobile application and web server perform optimally, because it helps to ensure that users are able to report disaster events or view disasters details in real-time. A high-speed internet is required for this prototype, as it supports the smooth operation of both platforms. Because it is not just about one user to report an event during emergencies, there could be many users reporting events at the same time. Therefore, the system should be able to handle multiple users simultaneously without delays, so all disaster reports are processed and displayed efficiently. Additionally, the web server must be optimized to send notifications instantly to users as soon as there are updates about disaster events.

B) **Reliability:** The second non-functional requirement to consider is reliability**.** This prototype should be available at all the times to ensure that mobile application users can access information and report incidents whenever needed.

C) **Security:** Security is a major concern for this prototype, as quick and reliable decision-making is crucial during a disaster. If security is compromised, through cyberattacks such as data manipulation or denial of service, it could delay or mislead critical responses. For example, if hackers change the disaster location, it could disrupt rescue efforts and worsen the impact of the disaster.

D) **Usability:** Another non-functional requirement to consider is usability, which refers to how easy the system is to use (Chung & do Prado Leite, 2009). This prototype must be designed for ease of use, as users need to report incidents and access to critical information immediately during a disaster. Therefore, a user-friendly system enables users to report disasters and find event information quickly, thereby reducing panic and confusion, and facilitating effective responses coordination.

4.2.4. UML Diagrams

Unified Modelling Language (UML) is a way to draw diagrams to explain how a system or software works. It helps people understand, design, and plan how a system function before it is built. Compared to written system descriptions, which can be interpreted in different ways by different people, diagrams provide a more precise and clear understanding. This ensures that everyone interprets the system in the same way (Yu & Mylopoulos, 1994). This section will use activity diagram and use case diagrams to demonstrate how the prototype works.

A) Activity Diagram

An activity diagram is similar to a flowchart that shows the steps or actions involved in a process. It helps visualize how a system works by breaking in down into smaller steps and showing how those steps are connected (Tilley & Rosenblatt, 2017).

In this prototype, users are required to report disaster events, and the admin verifies and publishes these events. Figure 4.3 illustrated the activity diagram of the reporting process. The diagram starts with mobile application users (who download and install the mobile application on their smartphones) to reporting disaster events by pinning the location, filling out the report form, and submitting the event to web server when a disaster occurs. Once the event is submitted, and the web server sets the event status to `pending`. The admin then verifies the event, for example, they will check its authenticity and update the collected event details. They will also send notifications and publish all the details to alert the application users, keeping them informed and prepared. The verification process is outside the scope of this prototype, it can be explored in future research.

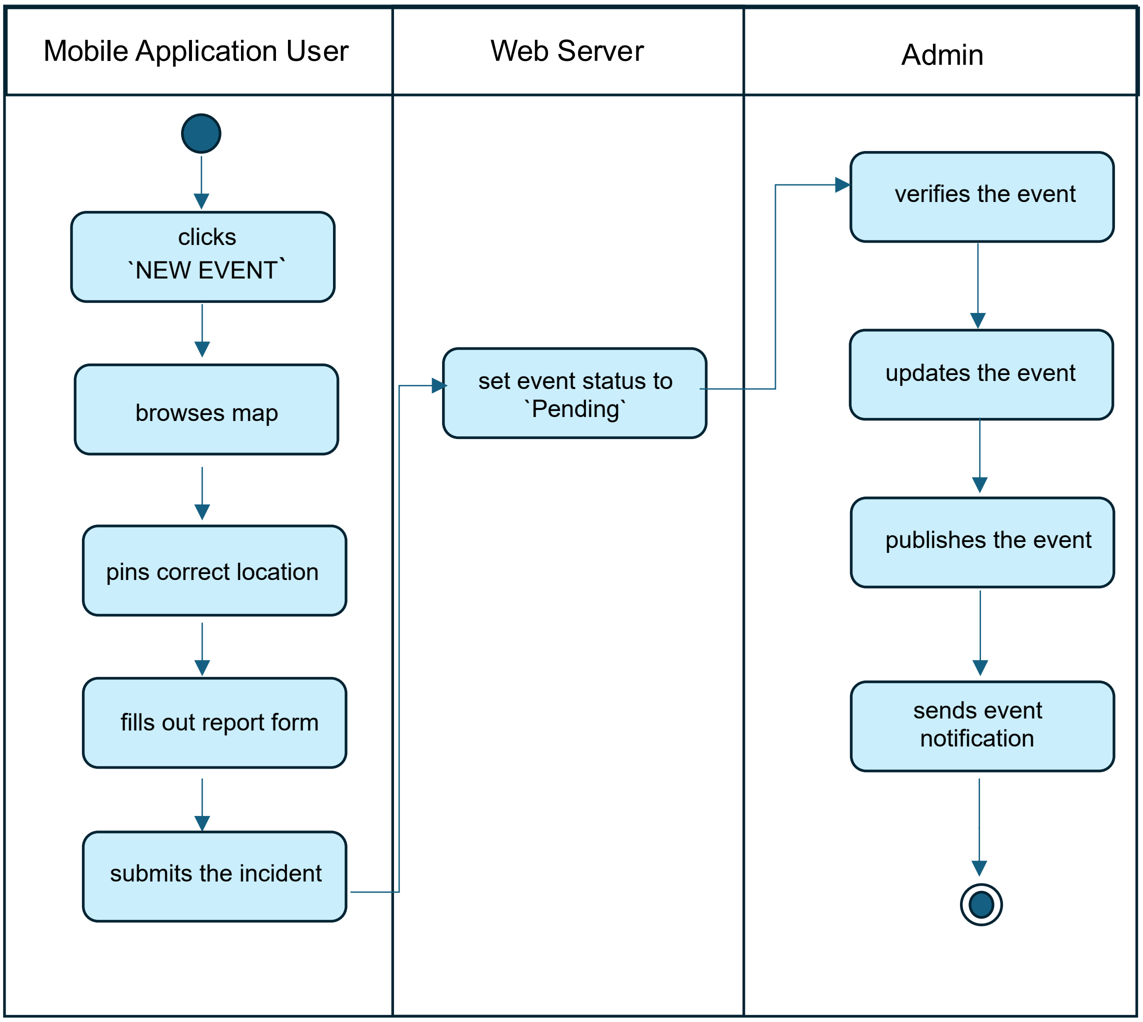


Figure 4.3: Activity diagram - Disaster Event Reporting Process *(Source: Author)*

B) Use Case Diagrams

In software design and development, a use case diagram shows who can use the system and what they can do with it. It highlights various roles, such as users or staff, known as actors, and illustrates how they interact with the system. (Tilley & Rosenblatt, 2017). This prototype design involves two actors: mobile application users and the web server admin. Figure 4:4 shows a high-level use case diagram that provides an overview of how these two actors interact with the prototype, and what functionalities the system must fulfil.

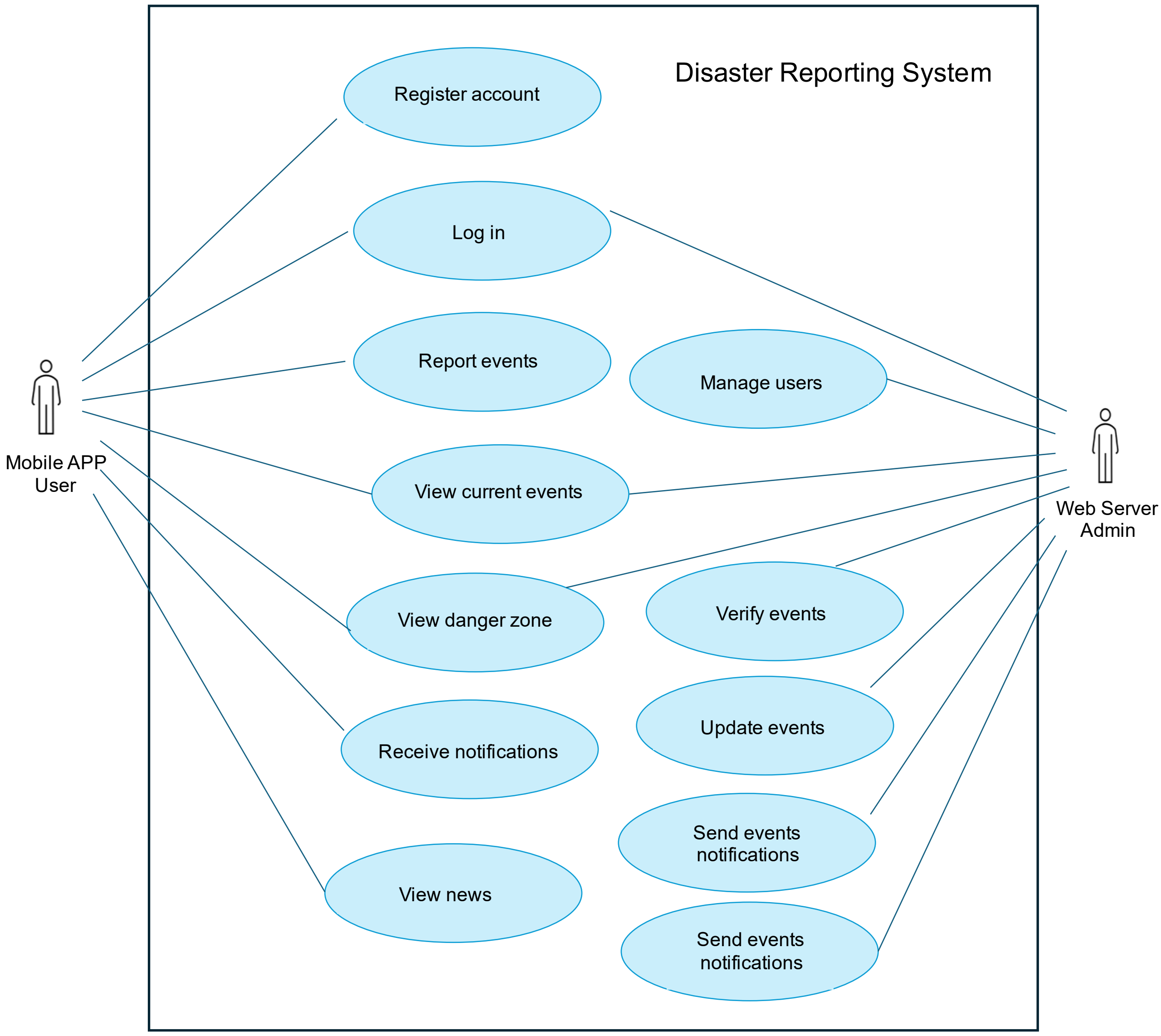


Figure 4.4: Use Case Diagram - Disaster Reporting System *(Source: Author)*

4.2.5. Database Diagram

Figure 4.5 illustrates the database structure of the prototype application for events and users. The user and event data are required to store in an online database to ensure it is accessible at all times. The `event_info` table contains all fields necessary to identify an event. For example, `event_name` specifies the name of the event, while `lat_value` and `lon_value` represent the latitude and longitude of the event. The data types and sizes are also specified in the database structure; for instance, the `event_name` data type is `varchar` with a size is 32.

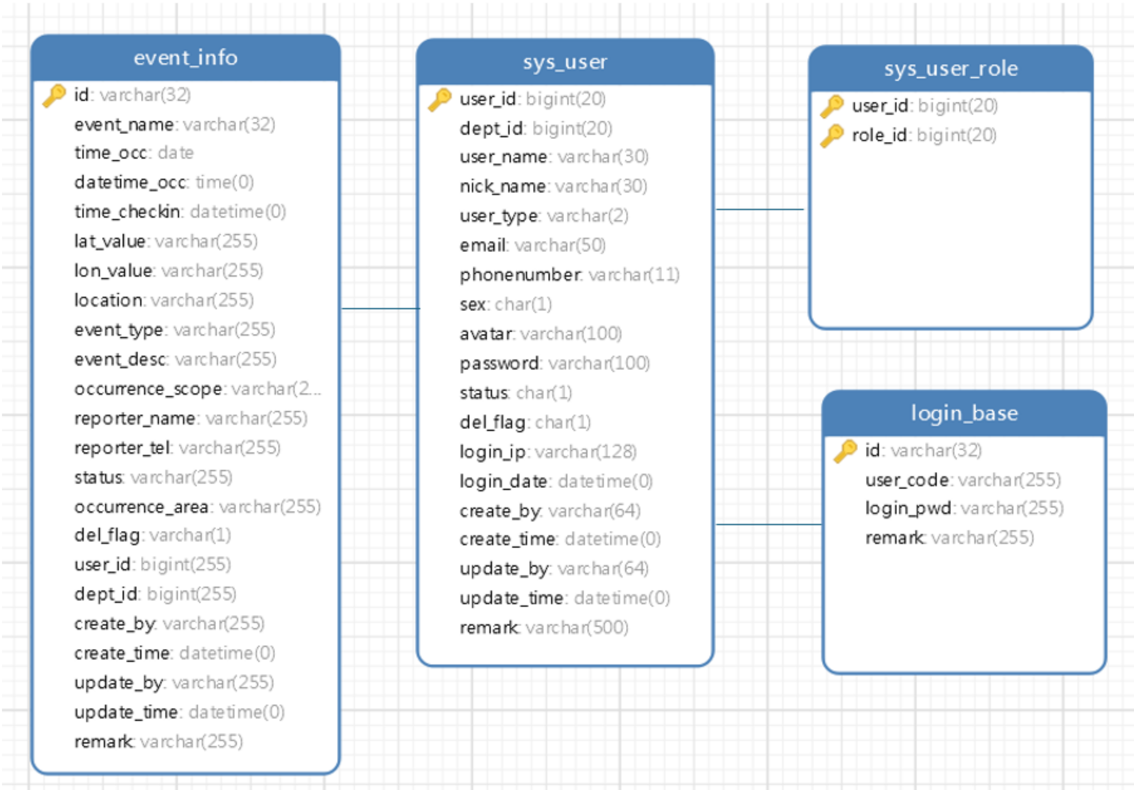


Figure 4.5: Database Diagram of Prototype Application (Source: Author)

4.2.6. Chapter Summary

This chapter outlines the functional and non-functional requirements used in developing the prototype application. It also covers the use case diagram, activity diagram and database diagram.

5. Implementation

Currently, the mobile phone market is dominated by two major operating systems: Android and iOS (GlobalStats, 2024). According to Zohud and Zein (2021) and Nurbekova et al. (2020), in mobile application development, there are several programming languages are commonly used based on the platform and the specific requirements of the application.

iOS development:

iOS is developed by Apple Inc. it's only used for Apple's mobile devices such as iPhone and iPad (Posey, 2024). Key languages for iOS platform include:

Swift: Swift is the main programming language which is used for native iOS development such as iPhone or iPad. It is a replacement of Objective-C. Swift is safe and optimized for performance (Baresi et al., 2024).

Objective-C: Objective-C is still used in existing iOS projects and libraries. It remains an essential skill for maintaining older iOS applications (Baresi et al., 2024).

Android Development:

Android is an open-source platform developed by Google, operates on the Linux Kernel (Dzezhyts, 2013). Three-quarters of all smartphones worldwide run Android (Apps, 2024), including devices from Samsung, Google (Pixel Phones), Amazon Fire, Sony and Xiaomi.(Android, n.d.) Common languages used for Android development include :

Java: Java is the primary language used to develop an Android application. It provides an extensive library and community support (Cheon, 2019).

Kotlin: Kotlin is supported by Google. It offers many features and seamless interoperability with Java, making development faster and more concise (Ardito et al., 2020).

Cross-Platform Development

Instead of creating separate apps for iOS and Android, another approach called cross-platform development offers developer to create a single codebase and deploy it across both operating stems: iOS and Android. (Charkaoui & Adraoui, 2014). Examples of cross-platform development approaches include:

React Native: Created by Facebook, React Native leverages JavaScript along with React framework to develop applications that are compatible with both iOS and Android systems. (Danielsson, 2016).

Flutter: Created by Google, Flutter uses Dart programming language to develop mobile applications for iOS and Android platforms. Flutter provides a rich set of UI components and tools, allowing developers to quickly build modern, visual appealing user interfaces. (Tashildar et al., 2020).

Xamarin (C#): Uses C# to develop apps for both iOS and Android. It provides access to native APIs and libraries.

Web-based: This approach utilizes web technologies like HTML, CSS, and JavaScript to develop what are known as Hybrid applications. These applications are packaged within a native container, allowing them to be installed and run on mobile devices, similar to native applications from an app store. Basically, hybrid applications combine web technologies with the functionalities of native applications (Amatya & Kurti, 2014).

Considering the mobile development application designed in this research, after users report an emergency event, it is necessary to verify the authenticity of this event. To accomplish this, it is necessary not only implement a verification system, but also to manage event publication, notifications, visualization the danger zone and overall event management. Therefore, in addition to completing the mobile development application, an event management system also needs to be developed. Given the researcher's proficiency in Java programming, the Spring Boot + Vue.js framework was chosen for this research. Spring Boot, developed in Java, aligns well with Android development for prototyping the mobile application, which primarily uses Java as one of its main programming languages. This choice leverages the extensive availability of Java-based development tools in the public domain and fits well with the researcher's skills. For the frontend of the events management system, Vue.js was selected due to its effectiveness in creating dynamic user interfaces.

This chapter focuses on the implementation stage of the application. Section 5.1 discusses the system architecture and technologies used in the development process and introduces the Integrated Development Environments (IDEs) used, along with their respective versions. Section 5.2 provide screenshots of both the mobile application prototype and the event management system frontend, accompanied by corresponding code snippets and explanations. Section 5.3 demonstrates key functionalities through provided code snippets and descriptions. Section 5.4 covers the prototype testing.

5.1. System Development Architecture and Technologies

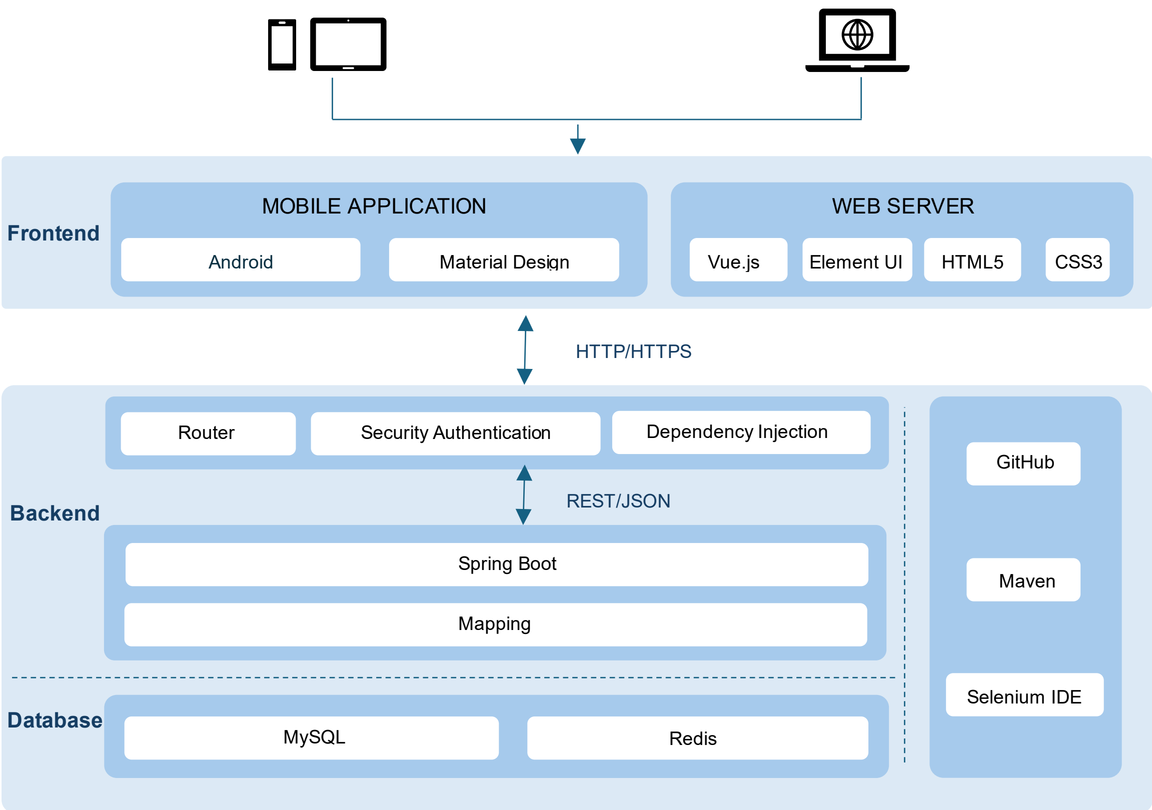


Figure 5.1: Proposed System Prototype Development Architecture (Source: Author)

Figure 5.1 illustrates the proposed architecture for the system prototype development. It consists of three main parts: Frontend, Backend, and Database. This architecture was used for building both the frontend and backend, as describes as follows:

5.1.1. Frontend Development

The frontend development of the prototype is divided into two parts: **MOBILE APPLICATION** and **WEB SERVER** as shown in Figure 5.1. The `**MOBILE APPLICATION**` component manages the mobile application for reporting, receiving and visualizing disaster events and danger zones. The `**WEB SERVER**` component handles the events management system, including the verification of the disaster event, publishing the event, and managing events. Various technologies were used to build the prototype's frontend: Android, Material Design, Vue.js, Element UI, HTML5, and CSS3 were utilized to develop and style both mobile application and the web server event management system, as detailed below:

A) MOBILE APPLICATION

Android (Java): Android is the system that runs mobile apps on Android devices like phones and tablets, providing the necessary tools for app development. In this research prototype system development, Android was used to create a prototype version of the app specifically for Android devices. Java was the programming language used to write the app's code, and Android's SDK (Software Development Kit) provided APIs (Application Programming Interfaces) that help build the app's appearance, manage how users interact with it, and connect it to other services (Steele & To, 2010). The Android framework organizes how the application looks and works, manages how different parts of the app communicate, and handles background tasks like loading data or making network requests. For instance, in the mobile application prototype development, Activities and Fragments control the app's screens and interactions, Intents help different parts of the application talk to each other, and `AsyncTasks` handle tasks running in the background (Steele & To, 2010).

Material Design: Material Design is a guide from Google that helps make apps look good and work the same on all devices, like phones and tablets. It provides rules for designing things like buttons and colors, and how these elements should behave when user interacts with them (IxDF, 2024). For example, in this development, some material components used such as `AppBarLayout' and `Toolbar` that help organise the top section of the application with easy-to-use buttons and titles. The `DrawerLayout` used to create the sliding menu, making it easy to navigate through different section of the application. The "List View" displays items in a clean and readable way. By following Material Design, the application looks good and works smoothly across various devices, ensuring that it is both visually appealing and user-friendly.

B) WEB SERVER

HTML5, CSS3, Vue.js and Element UI: When developing a website, HTML5 is used to define the webpage's structure and content. For example, the `header` tag is to define the webpage header. CSS3 is responsible for handling the style and layout, they work together creating the visual presentation (Macaulay, 2017). Vue.js is a framework used to create dynamic and interactive websites. Element UI is a collection of pre-designed components like buttons and forms that work with Vue.js that speeding up the design process and ensuring everything looks polished (Kumpulainen, 2021; Mobaraki et al., 2007; Sarrion). In the web server frontend development, the HTML5, CSS3 is to make the webpage structure and styling within Vue.js and Element UI components to create the frontend interface. For example, as shown in the Figure 5.2, the `el-form-item` is an Element UI component used to wrap form items, such as input fields. The inline style sets the width of the input fields to 63% of the container's width. These frontend technologies work together to provide effective layout and visual effects for the prototype.

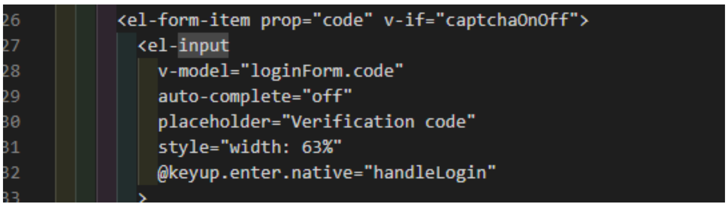


Figure 5.2: Element UI Form Input (Source: Author)

5.1.2. Backend Development

Spring Boot is a framework built on top of Spring that simplifies Java development by reducing the amount of configuration required. It integrates smoothly with databases, web frameworks, and security features (Walls, 2015). In this prototype, the backend development utilizes the Spring Boot framework, as shown in Figure 5.3, when a user reports a disaster event, routing is used to direct each HTTP request to the appropriate controller method based on the URL and HTTP method. For mapping, the prototype uses Object-Relational Mapping (ORM) to link database records to Java objects, which simplifies data handling. Dependency injection in Sprint Boot manages relationships between classes automatically. For example, injecting a disaster reporting service into the controller to handle disaster-related tasks. In terms of security, Spring Security is implemented to ensure that only authorized users can report or view disaster information. The system will verify users and admin who must login with username and password. In the backend of this prototype, specific permissions are assigned, such as granting admin the ability to manage disaster events.

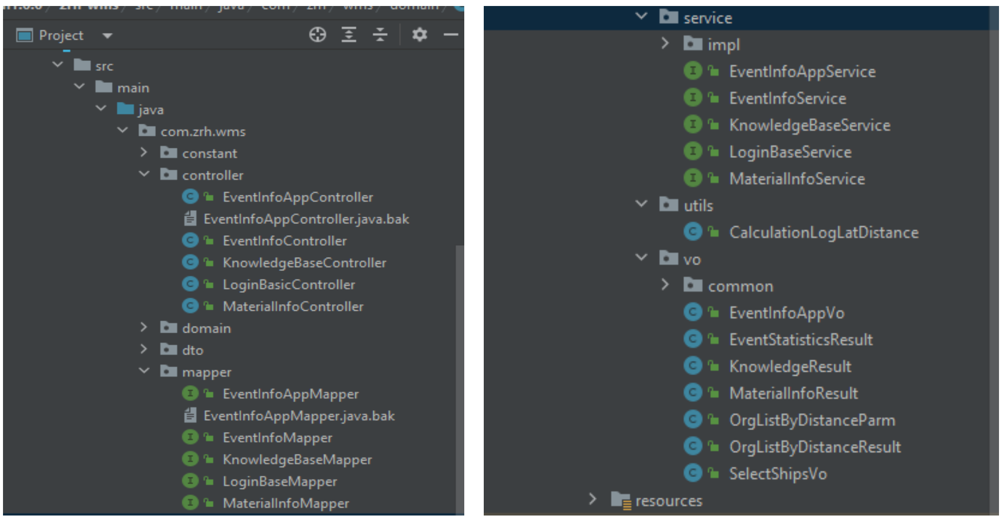


Figure 5.3: Backend Structure (Source: Author)

5.1.3. Databases

The database within this architecture is used to store various types of data, such as user information, details of disaster events. There are two databases are used:

MySQL is a popular relational database management system supporting SQL query language. It is managed with Spring Data JPA in the Spring Boot backend for data access layer management, implementing CRUD operations which are create, retrieve, update and delete (Bell, 2012; Zhang et al., 2021).

Redis is an open-source, in-memory data structure store supporting various data structures such as strings, hashes, lists, and more, including publish/subscribe patterns. Integrated into the Spring Boot backend for caching frequently accessed data or session information, reducing load on MySQL database and enhancing system concurrency and responsiveness (Li, 2022). The caching mechanism is important because the prototype requires real-time performance, especially when users report or view disaster events.

5.1.4. Additional Technologies

As shown in Figure 5.1 above, in addition to the frontend and backend technologies, several other technologies were used in the prototype development process:

HTTP/HTTPS are the fundamental protocols for transferring data over the web (Hong et al., 2023). When a user interacts with the web system or mobile application, HTTP requests are sent to the server to retrieve or send data. HTTPS uses encryption (via SSL/TLS) to protect data transmitted between the client (web browser or mobile application) and the server (Paracha et al., 2020). REST is a design approach for creating networked applications. It uses standard HTTP methods (GET, POST, PUT and DELETE) carry out various operations. (Martin-Lopez et al., 2022). RESTful APIs are frequently used to facilitate interactions between the frontend and backend (Ozdemir, 2020). JSON is a compact and easy-to-read format for data exchange. It is designed to be simple for humans to understand and straightforward for computers to access. JSON is often within RESTful APIs to structure and transmit data between clients and servers (Ahmad et al., 2021). In this development, The Vue.js communicated with the Spring Boot backend using RESTful APIs. HTTP/HTTPS are used as the underlying protocol for these API calls. The data exchanged is in JSON format when a user retrieves a list of disaster events from the server or reports a new disaster to the server.

GitHub is cloud-based service that allows developers to store, manage, and track changes to their code repositories (Hata et al., 2022). The researcher encountered a hardware failure that resulted in the loss of all prototype code without a backup during development process. Since then, the GitHub has been used to securely stored the code and related assets in the cloud, as well as to track and manage changes.

Maven is a build automation tool used primarily for Java project to manage the project dependencies (Downey & Downey, 2021). Maven is used in the prototype development to manage the various libraries and frameworks required for the prototype.

Selenium IDE is an automation testing tool used to verifying functions and features of web applications (Krishna & Gopinath, 2021). In this development, selenium IDE was installed on Chrome web browser to test the disaster verification function.

5.1.5. Integrated Development Environments Used in Development

The Integrated Development Environments (IDEs) used in this development are:

IntelliJ IDEA: This IDE is used for backend Java development. The version used in this development is IntelliJ IDEA 2023.1.1 (Ultimate Edition), with a license granted by AUT.

Visual Studio Code: This IDE is used for web server frontend development. Visual Studio Code is open source does not require a licence. The version used in this development is Visual Studio Code 1.93.0.

Android Studio: This IDE is used for mobile application development. Android Studio is also open source. The version used in this development is Android Studio Flamingo | 2022.2.1 Patch 1.

5.2. Interface Design and Code Implementation

This section introduces the main interface designs for mobile application and web server along with code snippets and explanations.

5.2.1. Main Mobile Application Interfaces Designs

This mobile application is an Android app built using Android Studio. The main application interface designs are shown in different figures. The left screenshot displays the actual application as seen on the phone, while the right screenshot shows the corresponding code snippets and design from Android studio project.

A) APP Name and Icon Display on Phone

Figure 5.4 below illustrates the mobile application's icon on the phone. This application is named \`DEAPP\` - Disaster/Emergency Events Application. The code snippet demonstrates the configuration settings of the Android application within the \`AndroidMainfest.xml\` file. It configures the essential settings for the application, including the app name (see line 22) and icon (see line 21).

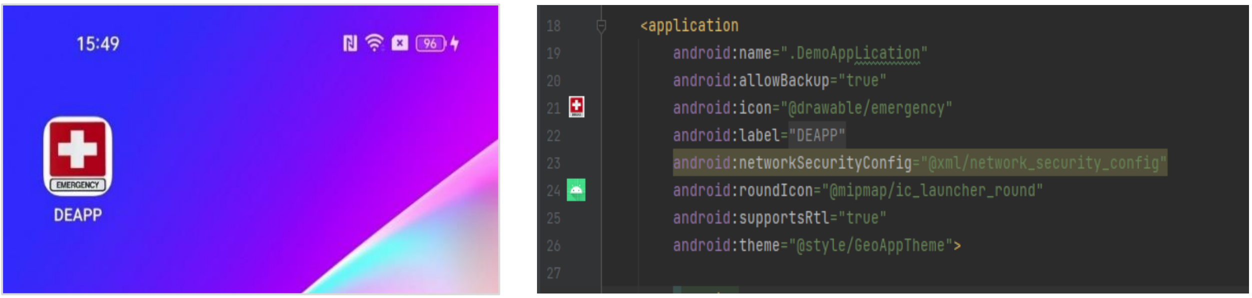


Figure 5.4: APP Icon on Home Screen (Source: Author)

B) Login / Registration Interface

To use the application, users need to click on the `DEAPP` icon on their phone. The login interface provides two functions: Register and Login. As mentioned in the chapter 4, this application prototype is more concrete in terms of its functionalities, so the constraints are not required in this design for the `Register` function. To login or register, users only need to provide their `username` and `password`, as show in Figure 5.5. The Android `LinearLayout` arranges the login and registration buttons horizontally. This layout adjusts to fit the screen, so that the buttons could share space evenly. The added margins and paddings help make the interface visually appealing and easy to tap.

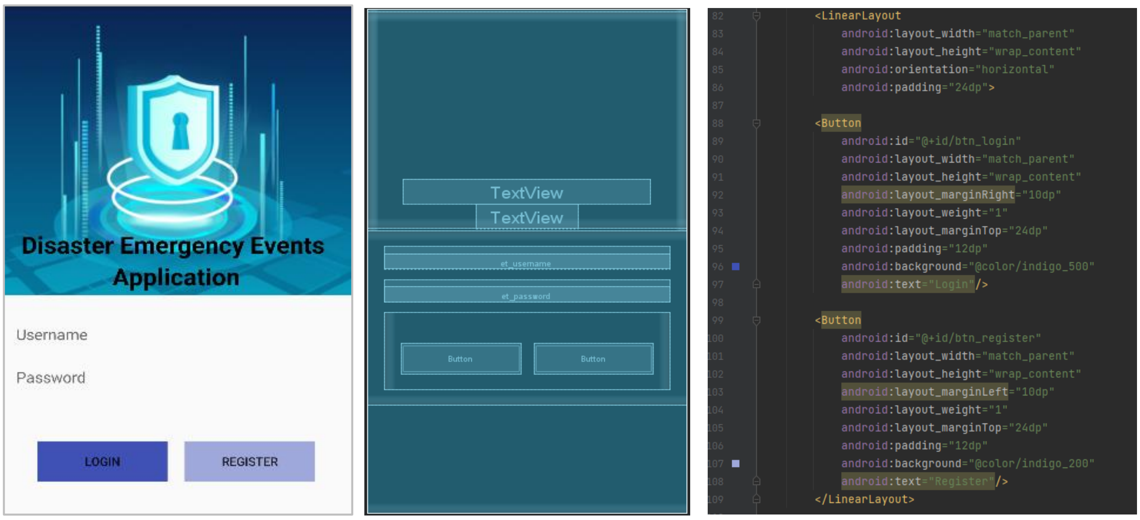


Figure 5.5: Login and Registration Screen (Source: Author)

C) Dashboard Interface

After logging, users can access the dashboard (see Figure 5.6). The application allows users to view three buttons: `NEW EVENT`, `CURRENT EVENTS` and `NEWS`, as discussed in chapter 4. At the top of the screen, there is a toolbar containing the `menu` and `logout` options, which use the `AppBarLayout`. The `AppBarLayout` was discussed in the technologies section, where it was mentioned that it utilises `Material UI`. The three buttons are placed within a `TableLayout`, which is centered in the layout. The single `TableRow` contains: the `New Event` button with a map icon, the `Current Events` button with a list icon, and the `News` button with a news icon. All buttons have a transparent background and black text.

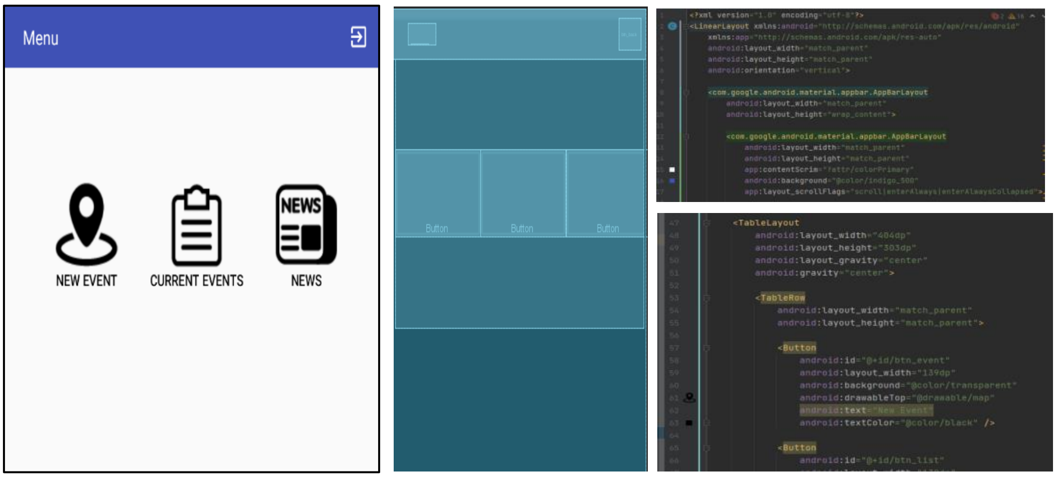


Figure 5.6: Dashboard Interface *(Source: Author)*

D) Disaster Event Reporting Interface

To report a disaster event, users should click on the `NEW EVENT` button, which will pop up a map, as shown in Figure 5.7. If the disaster location is correct, the user can slide to the side to access the report form. The form contains necessary attributes to describe the event, with mandatory entries for `Event name`, `Occur Date` and `Occur time`.

To facilitate quick reporting, the occurrence date and time will default to the user's current date and time. If they wish to change it, they can select the correct data and time; otherwise, the user can click the `CONFIRM` button to submit the event. To edit the date and time, the user simply clicks on the calendar and selects the exact time when the disaster occurs. No additional input is required for the date and time. If the user has enough time, they can complete the optional fields to provide more detailed information about the event.

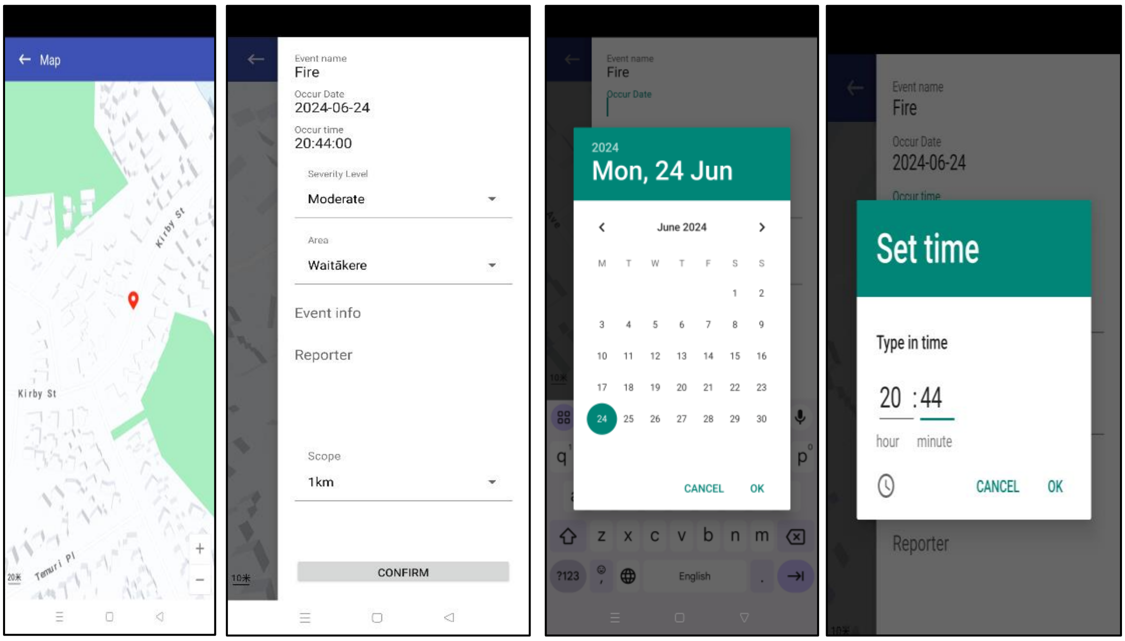
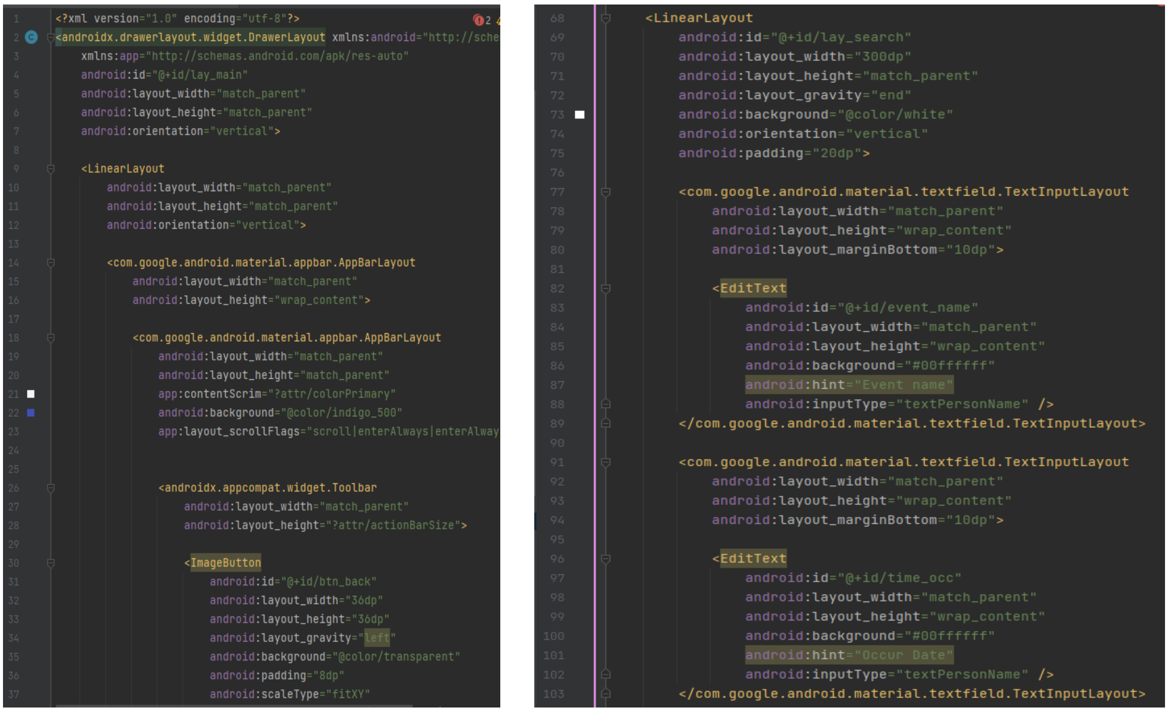


Figure 5.7: Report Disaster Events Interface   (Source: Author)



```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://sche
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/lay_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <com.google.android.material.appbar.AppBarLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:contentScrim="?attr/colorPrimary"
                android:background="@color/indigo_500"
                app:layout_scrollFlags="scroll|enterAlways|enterAlway


                <androidx.appcompat.widget.Toolbar
                    android:layout_width="match_parent"
                    android:layout_height="?attr/actionBarSize">

                    <ImageButton
                        android:id="@+id/btn_back"
                        android:layout_width="36dp"
                        android:layout_height="36dp"
                        android:layout_gravity="left"
                        android:background="@color/transparent"
                        android:padding="8dp"
                        android:scaleType="fitXY"
```

```xml
    <LinearLayout
        android:id="@+id/lay_search"
        android:layout_width="300dp"
        android:layout_height="match_parent"
        android:layout_gravity="end"
        android:background="@color/white"
        android:orientation="vertical"
        android:padding="20dp">

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp">

            <EditText
                android:id="@+id/event_name"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="#00ffffff"
                android:hint="Event name"
                android:inputType="textPersonName" />
        </com.google.android.material.textfield.TextInputLayout>

        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="10dp">

            <EditText
                android:id="@+id/time_occ"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="#00ffffff"
                android:hint="Occur Date"
                android:inputType="textPersonName" />
        </com.google.android.material.textfield.TextInputLayout>
```

Figure 5.8: Layout for Disaster Event Reporting Interface   (Source: Author)

Figure 5.8 is partial code snippets that illustrate how the report disaster events interface is designed. The root element is a `DrawerLayout`, which enables a sliding navigation drawer that fills the screen. Within it, a `LinearLayout` serves as the main container for

UI components. A `MapView` from the Baidu Maps API is incorporated to display a map for location selection. On the right side, the `TextInputLayouts` and `EditTexts` for capturing event details, including event name, occurrence date and time, severity level, area, and description. Additionally, there are `Spinner` for selecting options, such as danger zone area (e.g. 1km, 2 km), rather than requiring manual input. Optional inputs, and a confirm button are also included.

E) View Disaster Events

To view current disaster events, the user should click on the `CURRENT EVENTS` button. Which will display all events in a list. As shown in the Figure 5.9, the code snippets demonstrate the layout used to present disaster events in a user-friendly interface with a navigational toolbar. The `AppBarLayout` includes a toolbar featuring a back button (`ImageButton`) and a title (`TextView`). The `RelativeLayout` contains a `ListView` for displaying the events, with a custom list item selector.

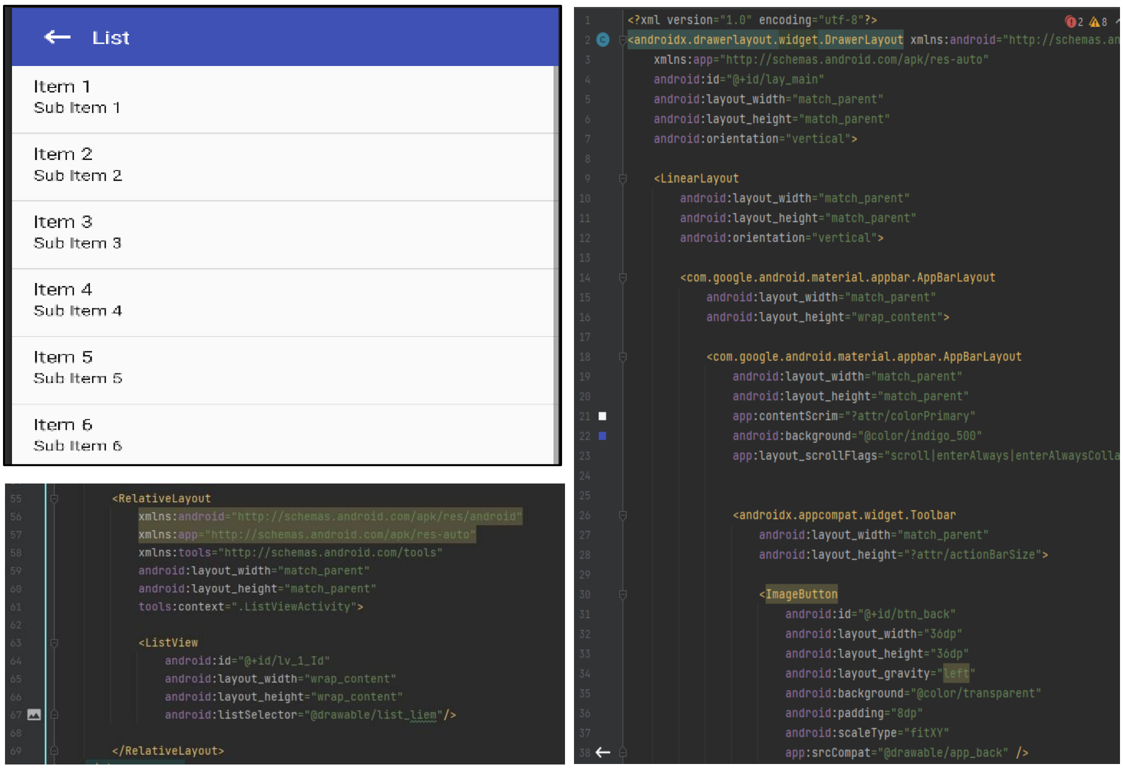


Figure 5.9: View Disaster Events Interface (Source: Author)

5.2.2. Web Server Interface Design

In this prototype, the web server is used for administrators to verify and manage the disaster events, it is not intended for use by the general public. Therefore, in this section, only present the main web page interface of managing disaster events, as show in Figure 5.10.

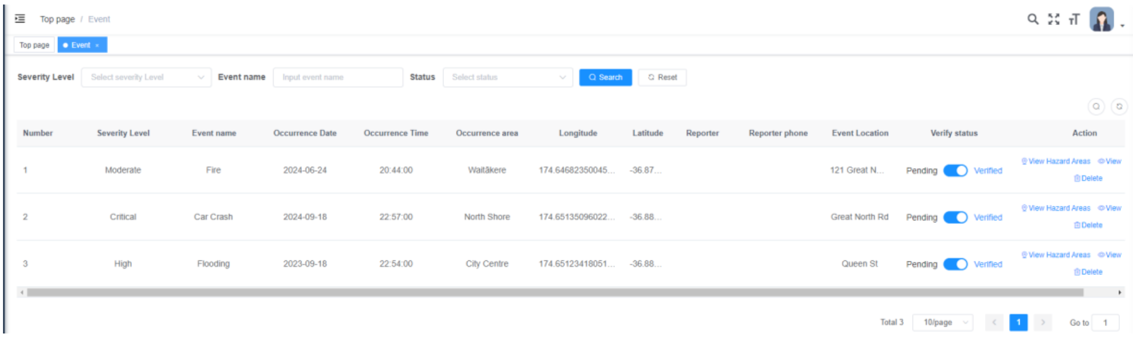
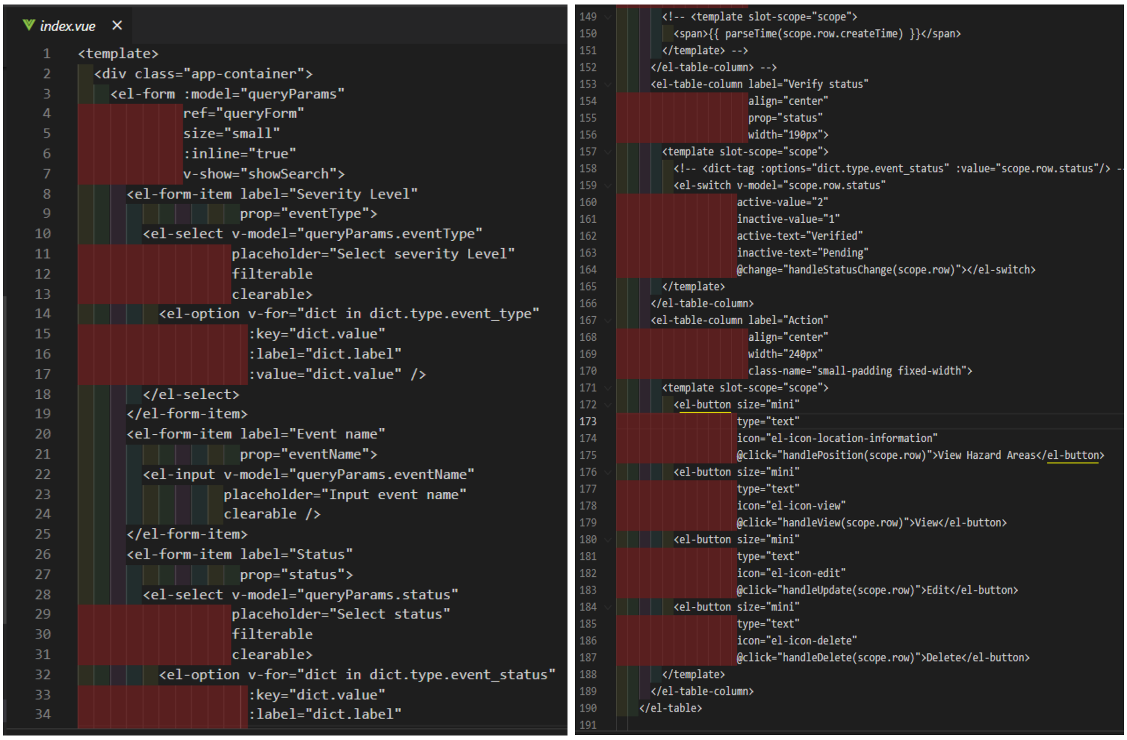


Figure 5.10: Web Server- Event Management Interface (Source: Author)



```
index.vue
<template>
  <div class="app-container">
    <el-form :model="queryParams"
             ref="queryForm"
             size="small"
             :inline="true"
             v-show="showSearch">
      <el-form-item label="Severity Level"
                    prop="eventType">
        <el-select v-model="queryParams.eventType"
                   placeholder="Select severity Level"
                   filterable
                   clearable>
          <el-option v-for="dict in dict.type.event_type"
                     :key="dict.value"
                     :label="dict.label"
                     :value="dict.value" />
        </el-select>
      </el-form-item>
      <el-form-item label="Event name"
                    prop="eventName">
        <el-input v-model="queryParams.eventName"
                  placeholder="Input event name"
                  clearable />
      </el-form-item>
      <el-form-item label="Status"
                    prop="status">
        <el-select v-model="queryParams.status"
                   placeholder="Select status"
                   filterable
                   clearable>
          <el-option v-for="dict in dict.type.event_status"
                     :key="dict.value"
                     :label="dict.label"
```

```
      <!-- <template slot-scope="scope">
        <span>{{ parseTime(scope.row.createTime) }}</span>
      </template> -->
    </el-table-column> -->
    <el-table-column label="Verify status"
                     align="center"
                     prop="status"
                     width="190px">
      <template slot-scope="scope">
        <!-- <dict-tag :options="dict.type.event_status" :value="scope.row.status"/> -
        <el-switch v-model="scope.row.status"
                   active-value="2"
                   inactive-value="1"
                   active-text="Verified"
                   inactive-text="Pending"
                   @change="handleStatusChange(scope.row)"></el-switch>
      </template>
    </el-table-column>
    <el-table-column label="Action"
                     align="center"
                     width="240px"
                     class-name="small-padding fixed-width">
      <template slot-scope="scope">
        <el-button size="mini"
                   type="text"
                   icon="el-icon-location-information"
                   @click="handlePosition(scope.row)">View Hazard Areas</el-button>
        <el-button size="mini"
                   type="text"
                   icon="el-icon-view"
                   @click="handleView(scope.row)">View</el-button>
        <el-button size="mini"
                   type="text"
                   icon="el-icon-edit"
                   @click="handleUpdate(scope.row)">Edit</el-button>
        <el-button size="mini"
                   type="text"
                   icon="el-icon-delete"
                   @click="handleDelete(scope.row)">Delete</el-button>
      </template>
    </el-table-column>
  </el-table>
```

Figure 5.11: Event Management Interface Code Snippet (Source: Author)

The web server uses Vue.js along with the Element UI framework. Figure 5.11 illustrates a partial code snippet of this interface, showcasing components such as `el-form`, `el-select`, and `el-input`, which are used to build a search form and table for managing disaster events. The form is used to filter or search disaster based on parameters like severity level, event name and status (Pending or Verified). The table displays a list of disaster events with columns for information such as occurrence date/time, reporter, and event location. It also includes the actions like viewing, editing, deleting events, and verifying events using the `el-switch` component, which allows the admin to toggle the verification status of an event between `Pending` and `Verified`.

5.3. Key Functionalities Demonstrations

This section discusses the key functionalities with code snippets. These functions involve the mobile application prototype for user login, disaster reporting, view disaster events, danger zone visualization, and accessing the latest disaster news from New Zealand Civil defence website. For the web server, it will demonstrate the functions of sending event notifications and verifying events. Since the source code is quite extensive, the full version is not displayed in the code snippets; instead, specific portions are illustrated to explain each function.

5.3.1. Login Function

The login function allows mobile application users to access the application securely. As shown in Figure 5.12, when the login button is clicked, it captures the username and password, creates a `LoginBase` object, and sends a login request using the `HTTP.Request` method. If the response indicates a successful login with an HTTP status code of 200, it saves the username and password in `SharedPreferenccces` and starts the `MenuActivity` to display the dashboard. If the login fails, it shows a toast message with the error.

```java
// Login page activity
public class LoginActivity extends AppCompatActivity {

    // UI components
    private EditText et_username;
    private EditText et_password;
    private Button btn_login;
    private Button btn_register;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        // Initialize UI components
        et_username = findViewById(R.id.et_username);
        et_password = findViewById(R.id.et_password);
        btn_login = findViewById(R.id.btn_login);
        btn_register = findViewById(R.id.btn_register);

        final SharedPreferences sharedPreferences = getSharedPreferences("wm

        // Login functionality
        btn_login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                LoginBase loginBase = new LoginBase();
                loginBase.userCode = String.valueOf(et_username.getText());
                loginBase.loginPwd = String.valueOf(et_password.getText());
```

```java
        // Registration functionality
        btn_register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                LoginBase loginBase = new LoginBase();
                loginBase.userCode = String.valueOf(et_username.getText()
                loginBase.loginPwd = String.valueOf(et_password.getText()

                HttpResponse<LoginBase> response = Http.Request("register
                if (response.Code.equals("200")) {
                    startActivity(new Intent(LoginActivity.this, MenuActi
                } else {
                    Toast.makeText(LoginActivity.this, response.Msg, Toas
                }
                et_password.setText("");
            }
        });
    }
}
```

Figure 5.12: Login Function Code Snippet (Source: Author)

5.3.2. Report a Disaster Event Function

To report a disaster event, the mobile application user must first click on the map to pin the location, and then fill out the report form. Figure 5.13 illustrates how to handle map clicks in the application, allowing user to click on the map to place a marker. When a user clicks, the application updates the displayed latitude and longitude based on the click position and manages any potential errors that may arise during the process. The method `onMapClick` is invoked when the user clicks on the map, with the `LatLng point` parameter representing the geographical coordinates of the clicked location.

```java
@Override
public void onMapClick(LatLng point) {
    markerOption.icon(BitmapDescriptorFactory.fromResource(R.mipmap.marker))
    centerLatLng = point;
    addCenterMarker(centerLatLng);
    try {

        lat_value.setText(String.valueOf(centerLatLng.latitude));
        lon_value.setText(String.valueOf(centerLatLng.longitude));
    } catch (Exception e) {
        throw new RuntimeException(e);
    }

}
```

Figure 5.13: Pin Location Code Snippet (Source: Author)

Figure 5.14 shows how the application handles user input for reporting an event, including validation, submission, and feedback to the user. When the button is clicked, it triggers a series of validation checks and actions related to reporting a disaster event. First, it checks if the date and time are valid; if so, the process continues. Then it checks if any of the required fields are invalid using the `isStringValid()` method. If any are invalid, a toast message will pop up. If all validations pass, an `EventInfo` object is created and a request is made to a server to submit the event data.

```
this.btn_refresh.setOnClickListener((v) → {
        if(isDateStringValid(time_occ.getText().toString().trim())){
            if(time_occ.getText().length()==10){
                if(isStringValid(event_name.getText().toString())
                        || isStringValid(lat_value.getText().toString())
                        || isStringValid(lon_value.getText().toString())
                )
                {
                    Toast.makeText( context: MapActivity.this,  text: "Date format error", Toast.LENGTH_SHORT).show();
                }else{
                    EventInfo eventInfo = new EventInfo();
                    eventInfo.eventName = String.valueOf(event_name.getText());
                    eventInfo.timeOcc = String.valueOf(time_occ.getText());
                    eventInfo.dateTimeOcc = String.valueOf(datetime_occ.getText());
                    eventInfo.eventType = String.valueOf(((KeyValuePair) event_type.getSelectedItem()).Key);
                    eventInfo.latValue = String.valueOf(lat_value.getText());
                    eventInfo.lonValue = String.valueOf(lon_value.getText());
                    eventInfo.location = String.valueOf(location.getText());
                    eventInfo.eventDesc = String.valueOf(event_desc.getText());
                    eventInfo.reporterName = String.valueOf(reporter_name.getText());
                    eventInfo.reporterTel = String.valueOf(reporter_tel.getText());
                    eventInfo.occurrenceArea = String.valueOf(((KeyValuePair) occurrence_area.getSelectedItem()).Key);
                    eventInfo.occurrenceScope = String.valueOf(((KeyValuePair) occurrence_scope.getSelectedItem()).Key);
                    HttpResponse<EventInfo> response = Http.Request1( uri: "insert", eventInfo, EventInfo.class);
                    if (response.Code.equals("200")) {
                        Toast.makeText( context: MapActivity.this,  text: "Upload event successfully", Toast.LENGTH_SHORT).show();
                    } else {
                        Toast.makeText( context: MapActivity.this, response.Msg, Toast.LENGTH_SHORT).show();
                    }
```

Figure 5.14: Report Disaster Code Snippet *(Source: Author)*

5.3.3. View Disaster Events Function

To view disaster events, mobile application users need to click on the `CURRENT EVENTS` button on the dashboard. This action opens the events list screen, which displays all current events along with disaster information. Figure 5.15 illustrates a list of event data displayed using a `ListView` and use a custom adapter, `MyAdapter`, to populate the `ListView` with event information fetched from the web server.

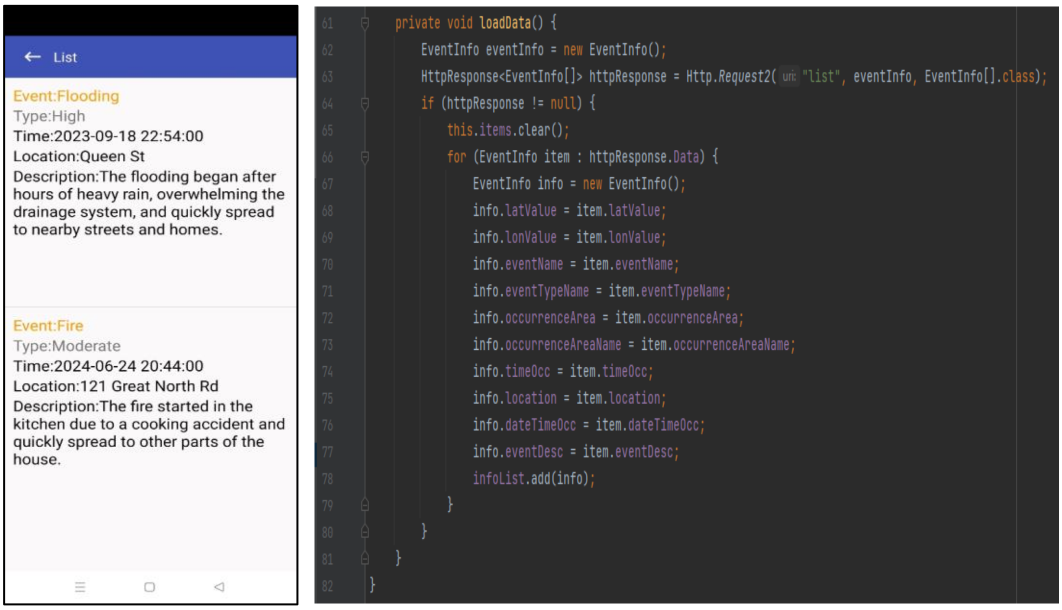


Figure 5.15: View Disaster Events (Source: Author)

5.3.4. Danger Zone Visualisation Function

When mobile application users view disaster incident information, if they also want to check the danger zone, they can click on a specific event. The system will then open and display a visualization map of the danger zone, with the affected area surrounded by a circle to indicate the extent of the disaster. The pin markers on the map represent the user's current location. As shown in Figure 5.16, the code snippet illustrates the `loadData()` method, which is responsible for loading event data from the backend, interpreting the response, and visualizing the each event's on a Baidu Map.

The `loadData()` method retrieves relevant disaster information and parses the JSON response to extract key details, such as event coordinates and the radius that defines the danger zone. This radius value corresponds to the **`occurrence scope`** shown in Figure 5.15, line 106, which determines the size of the circle representing the affected area. The circle visually represents the geographical areas associated with each event, it helps users can quickly assess the severity of the disaster and the extent of the danger zone. So that users will make informed decisions about their safety based on the size of the circle and the proximity to their current location.

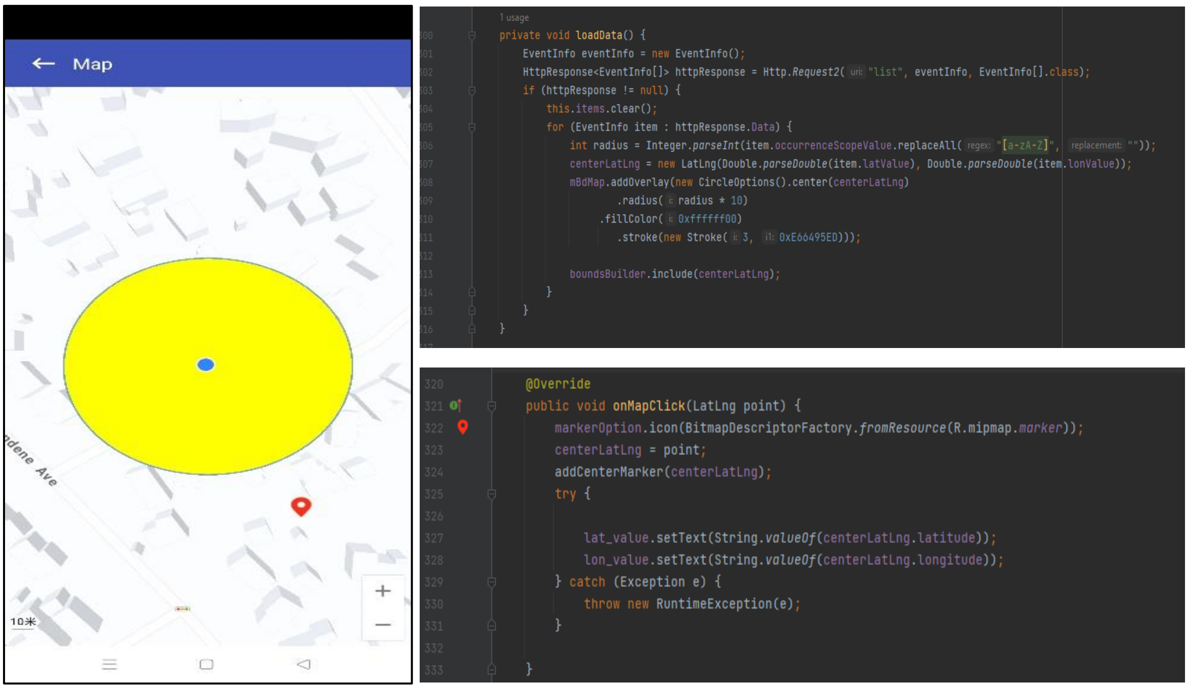


Figure 5.16: Danger zone Visualisation Function(Source: Author)

The rational for selecting Baidu Maps needed to explain. Initially, Google Maps was intended to display the affected disaster areas. However, during development, it was realized that the test device (a Redmi Phone) did not support Google Maps, making it impossible to display the disaster zones as originally planned. After careful consideration, the researcher decided to use Baidu Maps instead. The main focus of this study is to assess whether users can view the disaster zones through the application and understand the scope of the affected areas, thereby helping them in responding more effectively to disasters. The critical aspect here is not the choice between Google Maps or Baidu Maps, but how well users can access and interpret the disaster information.

Moreover, economic factors influenced this decision. Since Baidu Maps is the default mapping service on the test device, integrating it became a more cost-effective and practical solution. The objective of the study is to facilitate users' ability to retrieve disaster area information, rather than to depend on a specific mapping service. Therefore, the choice of map provider does not significantly affect the primary research goal.

In future developments, integrating multiple map services, such as Google Maps, Baidu Maps, or other local providers, would be a logical progression to ensure broader compatibility and flexibility across various devices.

5.3.5. View New Zealand Civil Defence News Function

To view New Zealand Civil Defence news, the user simply clicks the `NEWS` button on the dashboard. The system then seamlessly embeds the civil defence website directly within the application, making it easy for users to access the news without needing to open an external browser. This integration enhances the user experiences by keeping the entire process within the application, making it more convenient and efficient for users to stay informed and up to date on important news. As shown in Figure 5.17, the New Zealand Civil Defence news page is displayed within the `WebView` component embedded in the application, using the `wv.loaderUrl()` method to load the webpage into the `WebView`.

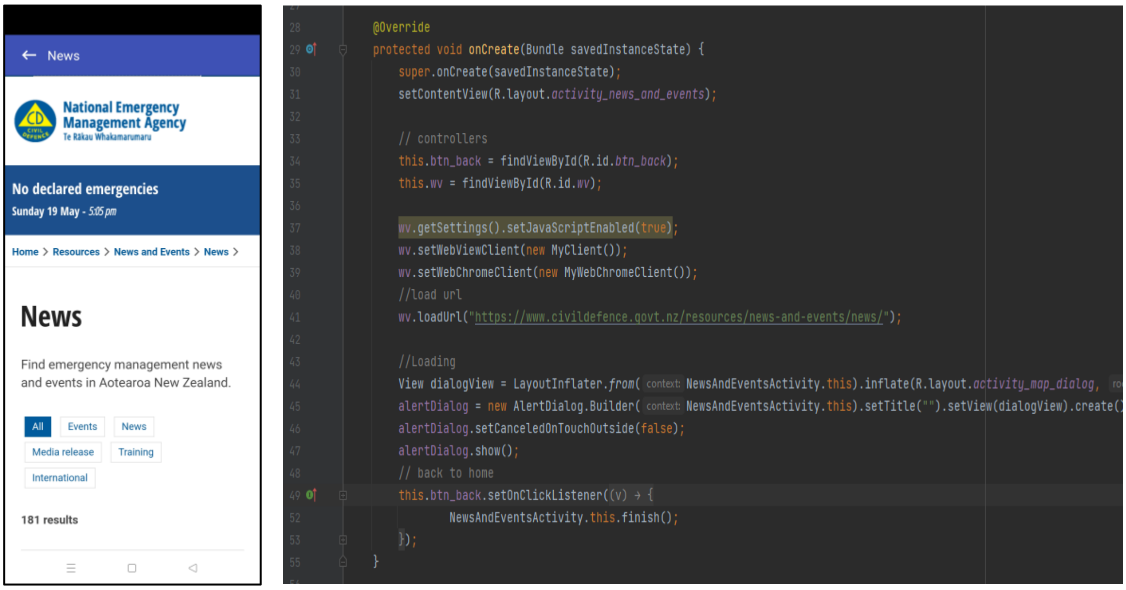


Figure 5.17: View New Zealand Civil Defence News (Source: Author)

5.3.6. Verify Event Function

After users report a disaster incident, the admin will verify it through the web server. There are two verification statuses: `**Pending**`, which is the default status assigned to any submitted event from the application, and `**Verified** `, which is assigned once the admin has reviewed and confirmed the incident. Figure 5.18 illustrates that the `handleStatus` method manages the status change of an event. It prompts the admin for confirmation before making any changes. If `row.status` is `2`, it is set to `Verified`, otherwise, it is set to `Pending`.

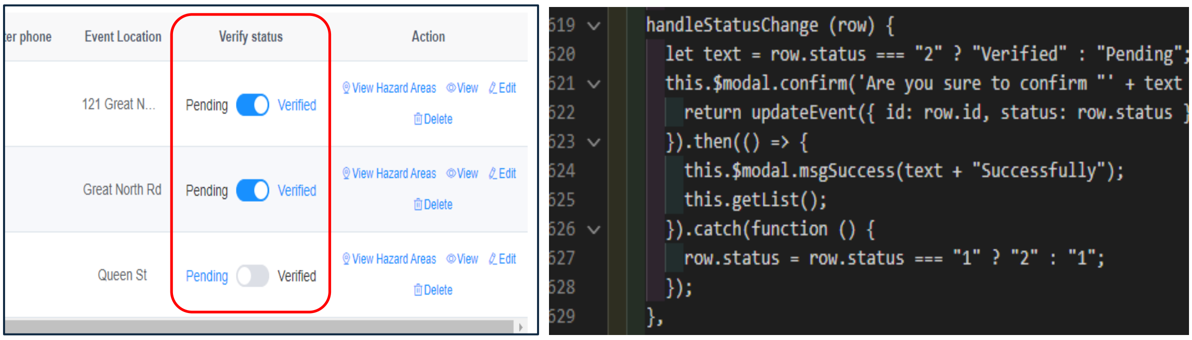


Figure 5.18: Verify Disaster Event (Source: Author)

5.3.7. Send Event Notification Function

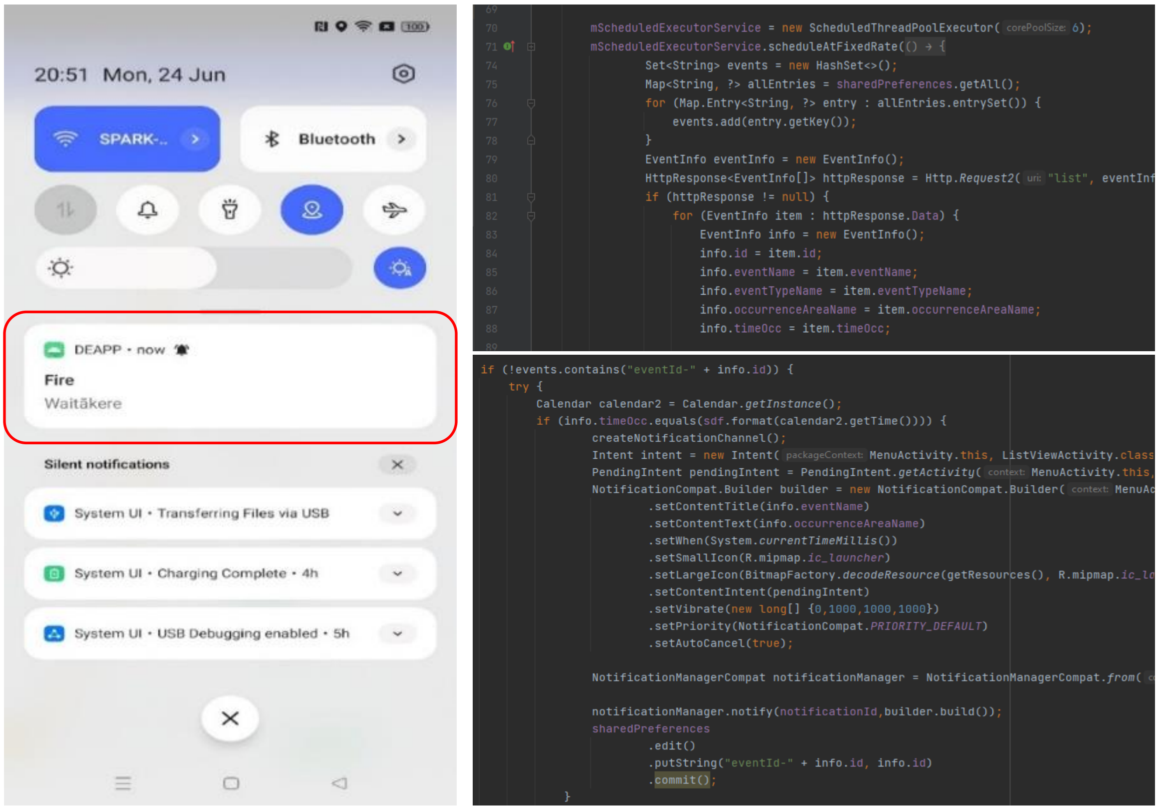


Figure 5.19: Send Event Notification (Source: Author)

After the admin verifies the disaster event, it is immediately published to the application, accompanied by a notification sent to the user's mobile application. Figure 5.19 is designed to provide real-time disaster notifications to users by regularly checking for new events and sending notifications when they occur. It integrates user preferences and notifications into the application. When users receive a notification, it will display on the lock screen, allowing them to open and view it.

5.4. Prototype Testing

Although each function has been compiled and tested during the development process, it is important to conduct final testing before handing the mobile prototype over to participants for evaluation. The testing methods used included unit testing with Selenium IDE to verify the disaster event function and black box testing to evaluate the mobile application's overall functionality. For example, the black box testing involved reporting a disaster event by pinning a mock event location, entering the occurrence date and time, submitting the mock event, and reviewing the events from the web server. Figure 5.20 illustrates the use of Selenium IDE to test the event verification function. All designed functions have been successfully tested. The next step is to invite participants to conduct the evaluation.

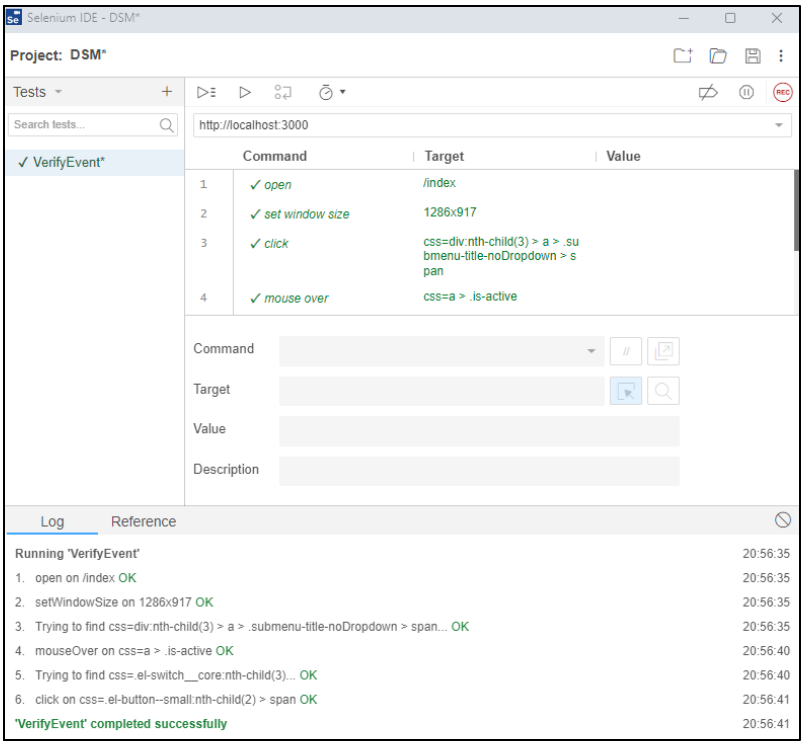


Figure 5.20: Selenium IDE – Event Verification Testing (Source: Author)

5.5. Chapter Summary

This chapter outlines the implementation process of the prototype interface. It also discusses the selection of appropriate front-end and back-end languages, along with other programming techniques. Additionally, the chapter showcases the technical development of the prototype through coding examples and screenshots.

6. Evaluation

This chapter explains the evaluation process for the mobile application prototype, following the chosen methodology. The evaluation accesses the artifact of design science research and is based on a survey. It includes the design of questionnaire, participants selection, the evaluation process and the results.

6.1. Questionnaire

According to Hajesmaeel-Gohari et al. (2022), questionnaires commonly evaluate user satisfaction, usability, and acceptance of mobile applications. Wich and Kramer (2015) also emphasised that the importance of evaluating usability in evaluation questionnaires for mobile application.

In this research, it is considered crucial to quickly report disaster and emergency incidents. Therefore, this evaluation not only evaluates the feasibility of the functionalities but also assesses the ability to report incidents rapidly which is related to usability evaluation. Based on this consideration, this questionnaire includes the following types of questions:

- **User satisfaction and experience**: It evaluates usability.
- **Features usage and Needs**: It identifies any features or functionalities users find particularly useful.
- **Issues and suggestions**: It gathers suggestions or feedback for improvements.
- **Future intentions and Recommendations**: It determines if users want to use this application in the future or recommend to friends or family.

The questionnaire contains 13 questions (See Appendix B), consisting of both qualitative and quantitative items. To match functionality of the prototype and make it easier to analyse the findings, the 13 questions are divided into 5 sections. The questions and their rationale are discussed below:

Section 1: Disaster Event Reporting Function Evaluation

There are two questions in this section:

Question 1. How easy was it to use disaster event reporting feature within the mobile application?

This question evaluates the usability of the disaster event reporting feature. It provides answers choice ranging from 'extremely difficult' to 'extremely easy,' that allows users to express their experience with the feature. The goal is to assess whether the mobile application is intuitive and user-friendly, ensuring that it meets user expectations for ease of use. A positive response indicates that users find the application satisfying and efficient in performing essential tasks like reporting disaster events, which is critical for the application's overall success.

Question 2. How useful do you find the mobile application in engaging users to report disaster events and danger zones?

This question aims to evaluate the application's effectiveness in simplifying the process of reporting disaster events. Ideally, users should interact less with manual typing and more with selecting or clicking on provided fields. This approach allows users to quickly report events by ensuring the necessary information is readily available and easy to input. The question assesses whether the provided fields facilitate efficient reporting of events.

Section 2: Emergency Notifications Evaluation

There is one question in this section:

Question 3. Were you able to receive emergency notification through the application that a disaster event has occurred? Please elaborate.

To receive timely notification is very important, this question is to check if the user is able to receive emergency notification through the application.

Section 3: Danger Zone Visualization Evaluation

To view the danger zone on mobile phone map is very important function in this application, this section contains two questions:

Question 4. Were you able to easily identify the boundaries of the danger zone on the map using this application?

This question is to assess the effectiveness of the application's mapping feature in clearly delineating the boundaries of danger zone. It evaluates whether users can easily understand and visualize the extent of potential danger areas based on the application's map interface.

Question 5. How useful was the visual representation of the danger zone in helping you understand the potential impact of the disaster?

This question seeks to evaluate the utility of visual representation of the danger zones provided by the application. It is to determine if the visual presentation enhances user' understanding of the geographical extent of the danger zone depicted on the map. It focuses on the user's personal experience with the application's visual representation of danger zone.

Section 4: Public Safety Impact Evaluation

There is one question in this section:

Question 6. Do you believe that having access to this type of application could help the public avoid entering dangerous areas during the disasters? Please provide your comments.

This question is designed to gather user opinions on whether they think the application, if widely accessible, could effectively contribute to public safety by preventing people from entering hazardous areas during disasters. It focuses on whether the application increases public safety awareness.

Section 5: Overall Feedback

This section is designed to gather feedback from users about their overall satisfaction and perception of the application' performance and functionalities and suggestion for future improvements. There are 7 questions in this section: question 7 and 8 is designed to gather feedback on the current application, while question 9 to 13 focus on seeking suggestions for future improvements.

Question 7: How satisfied are you with the overall performance of the application in reporting disaster events, receiving emergency notifications, and visualizing danger zones?

This question is intended to assess the user's overall satisfaction with the application's functionalities on the disaster events reporting, timely emergency notifications, and effectively visualizing danger zones on the map.

Question 8: With regards to the application that you are evaluating, do you believe that (Select all that apply):

- ☐ The application effectively provides real-time emergency notifications.
- ☐ The application delivers accurate and up-to-date information about danger zones.
- ☐ I find the maps and danger zone visualization features user-friendly.
- ☐ The application offers valuable guidance for responding to emergencies or navigating danger zones.
- ☐ It effectively notifies about nearby emergency services or resources.
- ☐ The application's speed and responsiveness meet my expectations.
- ☐ The application provides detailed information on potential dangers in specific areas.
- ☐ Additional Comments (if your thoughts are not listed above):___________________

This question enables users to provide detailed feedback on specific aspects of the application's performance and functions. Users could select multiple options if they believe that accurately reflect the application's strengths and areas after evaluation.

Question 9: If you have used any mobile applications designed for emergency notification and danger zone visualization in the past, please share your thoughts on your experience with those applications and compare them to the one you are currently evaluating.

This is open-ended question designed to determine if users have previous experience with similar applications. There are two main purposes to design this question: firstly, to identify if such mobile applications have been used in the public which we were not know to the researcher during the research stage, secondly, to identify features that could potentially be used for future improvements.

Question 10: If you were to use a mobile application to receive alerts about disaster danger zones, what features would you find most valuable?

- ☐ Real-time danger zone mapping.
- ☐ Two-way communication to report incidents.
- ☐ Navigation and routing to safe areas.
- ☐ Ability to share alerts with others.
- ☐ Notification with sound and visual cues.
- ☐ Other (please specify): _________________

In addition to visualizing danger zones during a disaster, which other features are considered most important. This question aims to find out the specific features that user prioritizes for a mobile application focused on alerting about disaster danger zones. This will help prioritize development efforts according to the user expectations and needs.

Question 11: What types of information are most important for you to receive in an alert about a disaster danger zone?

- ☐ Evacuation instructions.

- Emergency contact information.
- Estimated time of arrival for the danger zone.
- Description of the disaster event.
- Location and proximity of the danger zone.
- Other (Please specify). _______________

This question seeks to understand the types of information users consider essential when receiving alerts about disaster danger zones. By providing options such as evacuation instructions, emergency contact information, and specifics about the disaster event, it helps customize the content and format of alerts to better meet user needs during critical situations, Additionally, this is information could be also beneficial for authoritative groups, such as New Zealand Civil Defence.

Question 12: Based on your experience with the current application, what specific improvements or changes do you think would enhance its disaster event reporting functionality?

This question is an open-ended question that encourages users to provide practical feedback on how to improve the application's disaster event reporting capabilities. It allows users to suggest any specific enhancements or changes they believe it would make the reporting process more effective, efficient, or user-friendly.

Question 13: What additional features or functions, beyond those in the current application, would you like to see implemented to make it more effective for disaster reporting and visualization?

This is final question that asks users to suggest any new features or functionalities they feel are missing in the current application but would enhance its effectiveness in disaster reporting and visualization in future. It helps uncover innovative ideas and user expectations for future development, guiding the application's evolution to better support users during crisis situations.

6.2. Participants Selection

Disasters can happen anytime, anywhere, and anyone may face them. Therefore, this evaluation targets the general public, inviting people who can use a smartphone and is familiar with mobile application to participate. Recruitment began with advertising on a church noticeboard, chosen for its membership representing various ethnic backgrounds, genders, and age groups, making it representative of the general public. Interested individuals expressed their interest by sending an email to the researcher, who then sent them a consent form, Candidates willing to participate signed the consent form.

Fifteen participants were randomly selected to evaluate the mobile application prototype in this research. All documents used in the evaluation including the questionnaire, advertisement information, consent form, and information sheets have been approved by Auckland University of Technology Ethics Committee (AUTEC), under approval number 24/128.

6.3. Evaluation Process

The evaluation required participants to use the application to simulate a mock disaster event and a sever was the set up to verify the event and publish it. The devices were used in the evaluation are as follows:

Specification	Mobile Phone	Server
Band and Model	OPPO A94 5G (Model: CPH2211)	DELL Laptop Vostro 3420
Purpose to use	Evaluation of the application	Verification and publishing event to mobile device
Operating System	ColorOS 13.0, Android 13OS	Window 11 Pro
Processor	Dimensity 800U Octa-core	12^{th} Gen Intel® Core (TM i7-1255U 1.70GHz
Random Access Memory (RAM)	8 GB	16 GB
Network	Wireless fidelity (WIFI)	WIFI

Table 6.1: Required Evaluation Devices

The evaluation process follows the framework shown in Figure 6.1, which consists of four main stages. The actions for each stage of the evaluation are explained as follows:

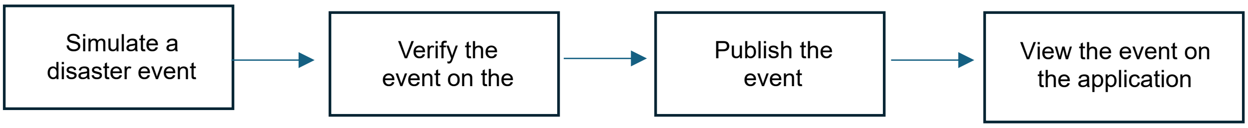


Figure 6.1: Evaluation Workflow Framework

Stage 1: Simulate a disaster event

1. The researcher provides a smartphone to the participant, which the phone has pre-installed with the mobile application prototype.
2. The participant opens the mobile application to simulate reporting a mock disaster event. They follow the steps below:
   a) Click on the application icon to open the application
   b) Click on the "Event" icon.
   c) Pin the location on the popup map, then click on the "confirm" button to report the event.
   d) Alternatively, the participant may input additional event details, such as event name, occurrence date and time, event type, event description, and reporter name, though this is optional.
3. The participant closes the mobile application and awaits the notification.

Stage 2: Verify the event on the sever

4. The researcher utilizes a laptop to open the web server application, verifying the event, updating all details.

Stage 3: Publish the event

5. The researcher sends the event notification and publishes the event details from the web server to the mobile application prototype.

Stage 4: View the event on the mobile application

6. The participant opens the phone to check if the event notification is displayed on the notification panel.

7. The participant accesses the mobile application to view the event details by selecting the “Event lists” icon.
8. The participant clicks on the selected event to open and view the danger zone visualization to verify its visibility.
9. The participant provides feedback on the given questionnaire.
10. The researcher awaits the participant to complete all the questions on the questionnaire.
11. The researcher collects the completed questionnaire from the participant.

6.4. Evaluation results

Kaur et al. (2018) stated that the descriptive statistics method could be used to analyse data that contain variables or measure the frequencies. Forman and Damschroder (2007) pointed out that qualitative content analysis is used to analyse the contents of textual data, it could categorize textual data in order to understand its content.

In this research, the data will be analysed using a mixed methods approach: descriptive statistics and Qualitative content analysis. Descriptive statistics is used to calculate the response frequencies, summarise the data and find out the patterns to explain the data. Qualitative content analysis is used for open-ended questions, try to capture the content and participants’ comments and get a deeper understanding of the data relationship and meanings. This mixed methods approach will help to assess the effectiveness of the mobile application.

Participants feedback in this evaluation are labelled as P1, P2, and so on, no personal or sensitive details are disclosed. Evaluation results have been recorded in the tabular format, with each table corresponding to a specific question. Below are the results table for these 13 questions. All 15 participants provided answers to each question.

Question 1**.** How easy was it to use the disaster event reporting feature within the mobile application?

ANSWER CHOICES	RESPONSES
Extremely Difficult	0
Difficult	0
Neutral	0
Easy	9
Extremely Easy	6

Table 6.2: Answers to Question 1

Question 2. How useful do you find the mobile application in engaging users to report disaster events and danger zones?

ANSWER CHOICES	RESPONSES
Not Effective	0
Slightly Effective	0
Moderately Effective	3
Effective	3
Very Effective	9

Table 6.3: Answers to Question 2

Question 3. Were you able to receive emergency notification through the application that a disaster event has occurred?

ANSWER CHOICES	RESPONSES
Yes	15
No	0

Table 6.4: Answers to Question 3

Question 4. Were you able to easily identify the boundaries of the danger zone on the map using the application?

ANSWER CHOICES	RESPONSES
Yes	15
No	0
Other	0

Table 6.5: Answers to Question 4

Question 5. How useful was the visual representation of the danger zone in helping you understand the potential impact of the disaster?

ANSWER CHOICES	RESPONSES
Not useful	0
Slightly useful	0
Moderately useful	4
Very useful	5
Extremely useful	6

Table 6.6: Answers to Question 5

Question 6. Do you believe that having access to this type of application could help the public avoid entering dangerous areas during the disasters? Please provide your comments.

Participants	Answers
P1	Yes, of course.
P2	Yes, is very helpful. Publicity also very important the more people use this app, the more effective to avoid unnecessary harm to the public.
P3	Yes, application is easy to use and understand.
P4	Yes, it would be very helpful to common people as well.
P5	Yes, the app would help people avoid those dangerous areas.
P6	Very much so useful and informative real time alert system.
P7	Yes
P8	I guess it is very helpful, and helping to avoid any possible traffic.
P9	Yes, absolutely
P10	• Yes, there are a lot of disasters that keep happening and having this app will surely help a lot of people. • It provides the information that's significant for the situation, and it's very easy to use, so I'm positive that everyone can use it.
P11	I believe it is more effective than the current alert notification we get from the civil defence.
P12	Yes, I do
P13	Yes, it will surely aware the people for the respective incident, but if won't show the disaster, for example, there is a fire in some area, but the person doesn't know where it is happening, he/she can't be aware. So an image of this incident is a must.
P14	Yes, if this type of app is being prioritized perfectly, it will help prevent people entering zone and help people inside to evacuate.
P15	With the area outline area, it would be easy and obvious to avoid.

Table 6.7: Answers to Question 6

Question 7. How satisfied are you with overall performance of the application in reporting disaster events, receiving emergency notifications and visualizing danger zones?

ANSWER CHOICES	RESPONSES
Very Dissatisfied	0
Dissatisfied	0
Neutral	1
Satisfied	5
Very Satisfied	9

Table 6.8: Answers to Question 7

Question 8. With regards to the application that you are evaluating, do you believe that (Select all that apply).

ANSWER CHOICES	RESPONSES
The application effectively provides real-time emergency notifications.	**15**
The application delivers accurate and up-to-date information about danger zones.	**15**
I find the maps and danger zone visualization features user-friendly.	**13**
The application offers valuable guidance for responding to emergencies or navigating danger zones.	**11**
It effectively notifies about nearby emergency services or resources.	**12**
The application's speed and responsiveness meet my expectations.	**13**
The application provides detailed information on potential dangers in specific areas.	**13**
Additional Comments (if your thoughts are not listed above): P2 - To avoid misuse on this app, the verified system needs to be very accurate. P4 - I can't wait for this application to be started. P9 - Can I limit notification based on personalized areas of importance? P15 - • Danger zone circle can obstruct map info in large areas. • Notification rely on phone data, which can be unreliable during critical disasters. Can notifications consider SMS in critical situation? • People may panic in emergencies, so it’s good that the UI is fairly simple and user friendly.	**4**

Table 6.9: Answers to Question 8

Question 9. If you have used any mobile applications designed for emergency notification and danger zone visualization in the past, please share your thoughts on your experience with those applications and compare them to the one you are currently evaluating**.**

Participants	Answers
P1	No
P2	N/A
P3	No
P4	I have never used this type of application
P5	No
P6	No
P7	No
P8	No
P9	N/A
P10	I have not used any apps that is similar to this one, but I've received alerts from the emergency team/sos before. It does show the emergency situation but most of the times, it's not time sensitive and it does not provide detailed information about the situation, so it is most likely overlooked.
P11	I have not used any
P12	No, I haven't
P13	No
P14	No
P15	No

Table 6.10: Answers to Question 9

Question 10. If you were to use a mobile application to receive alerts about disaster danger zone, what features would you find most valuable? (Select all that apply)

ANSWER CHOICES	RESPONSES
Real-time danger zone mapping.	**14**
Two-way communication to report incidents.	**8**
Navigation and routing to safe areas.	**14**
Ability to share alerts with others.	**9**
Notification with sound and visual cues.	**12**
Other (please specify): P15 - If help or emergency services are on the way.	**1**

Table 6.11: Answers to Question 10

Question 11. What types of information are most important for you to receive in an alert about a disaster danger zone? (Select all that apply)

ANSWER CHOICES	RESPONSES
Evacuation instructions.	**14**
Emergency contact information.	**12**
Estimated time of arrival for the danger zone.	**12**
Description of the disaster event.	**11**
Location and proximity of the danger zone.	**14**
Other (Please specify): P13- visuals and helping tracker, like it shows how far the medics and police and other thing. P15 - Tips on what to do to escape danger zone safely. What to do in case of fire etc.	**2**

Table 6.12: Answers to Question 11

Question 12. Based on your experience with the current application, what specific improvements or changes do you think would enhance its disaster event reporting functionality?

Participants	Answers
P1	This application is most effectively used online access through internet
P2	So far is a very good app, very useful in NZ
P3	No issues so far
P4	This is good
P5	UI elements on the event's could be more clear and easier to access.
P6	No it's good
P7	No
P8	No
P9	• Time >Include AM/PM. (It is difficult for kiwis, especially for elder people to understand the 24 - hours format. For example 22:00 can be confusing to determine whether it is nighttime or daytime). • Danger zone indicator >lower opacity to view map details, • Should have different colors indicate level of criticalness (red for critical, blue for low, etc.) • Notification to phone only for areas, I specify needing to know about so I don't get flooded with notifications for disaster that do not affect us (especially for low lower danger) • Make it easier to tell that user needs to first select area on map, and then swipe to make report.
P10	Have a live view in maps would be a good improvement for the app, as someone who's really bad at directions, seeing the specific location through live view would be helpful
P11	The danger zone radius should update automatically.
P12	Update event (refresh the map)
P13	It should show the evacuation process, firstly visual of the disaster, and the helping tracker where the helpers are like ambulance and police and also show of safer zone.
P14	I think that there should be another way to input the area of disaster for someone not used to using map.
P15	For danger area circle make it transparent so the ahead details are still clear.
	The ability to put danger location by typing down address instead of only physically pinning.

Table 6.13: Answers to Question 12

Question 13. What additional features or functions, beyond those in the current application, would you like to see implemented to make it more effective for disaster reporting and visualization?

Participants	Answers
P1	N/A
P2	At this point, I just want to use the app for a while, so I can give more clear and accurate feedback on this application. But so far very satisfied with it.
P3	It's a very interesting concepts with more usage, I am sure changes can be made at different stages.
P4	It should available to general public with mobile phone.
P5	For users to be able to see past "danger events" of an area.
P6	No it's very good
P7	No
P8	No
P9	• Forced notification in critical events based on GPS location. • Filter notification based on personal preferences (location, severity) • Extra languages - perhaps consider dropdown selection for event names to make translation easier • Perhaps add audio events for blind people and drivers.
P10	• First aid kit near me and hospitals near me • For disaster, it's gonna be helpful if people can see where they can get aid kits, especially during emergency situations. Cause even if the ambulance comes, there can delays and attending to situation immediately is the most idle things.
P11	No
P12	Notification sound more seriously.
P13	• It should have a photo option, or video. • It should contain the number of people in the danger zone. • It should have a default of incident popped up automatically (like earthquake, fire etc)
P14	I think that the details should be popped up after pining the map so user can immediately type in the details.
P15	Evacuation zones or safe area highlighted

Table 6.14: Answers to Question 13

7. Evaluation Results Analysis

Based on the evaluation feedback, this section discusses the evaluation results, in order to improve public disaster preparedness through an application that supports disaster event awareness and danger zone visualisation. The feedback helps to demonstrate the application prototype 's effectiveness and its important in disaster and emergency reporting. It shows how well the app meets the needs of users and highlights its value in improving public disaster preparedness by providing essential features and critical information.

The discussion analyses the feedback provided by 15 participants to 13 questions, categorizing it based on the mobile application functionalities and effectiveness. Additionally, it includes a discussion on the strengths and weaknesses identified.

7.1. Disaster Event Reporting Function - Question 1, 2 and 12

Based on the feedback from 15 participants, it is evident that this application is user-friendly for reporting disaster events. In response to Question 1, nine participants found it is “easy” to use, while six participants found it “extremely easy”. Additionally, in response to Question 2, nine participants found it is “very effective”, three founts it “effective” for reporting the disaster events. and three participants found it “moderately effective”.

In Question 12, the participant 5 mentioned that the UI elements could be clearer, and participant 9 suggested that the application should inform user to first pin the danger zone location, and then swipe the menu bar on the right side of phone to report the event.

The current design of this application focuses more on implementation of functionalities, rather than providing a user manual or instructions. To report an event, users must first pin the location on the map. They can then enter any disaster information if the situation permits or simply click the submit button. However, if the users are unaware of these steps, they may find it difficult for report incidents. Therefore, a future improvement could include adding instructions within the application to clarify the reporting process.

Participant 9 from Question 12 also commented on the necessity to include AM/PM format for reporting the time of disaster events. Currently, users can either select the time using the provided clock on the application or manual enter it in 24-hour format. Participant 12 mentioned that the 24-hour format might be challenging for Kiwis, especially the elderly. Further investigation is needed to determine their familiarity with AM/PM format. Therefore, a future improvement could involve adding an AM/PM format option for time input.

7.2. Emergency Notification Function – Question 3, 8, 12 and 13

According to the feedback from Question 3 and Question 8, all participants received timely emergency notifications on the mobile phone. Some useful suggestion provided by participants include:

Currently, the notification sounds use the phone’s default notification sound, and the application does not provide a strong alert audio. According to the results from Table 6.13, Participant 12 suggested that the notification sound should be more serious to effectively alert users.

Based on the results from Table 6.9 and 6.13, Participant 9 suggested that the notifications could be customizable based on user preferences or phone GPS location and event severity. For example, Users in Auckland may not need notifications for events in Christchurch, or they may not need to notifications for lower danger events. Currently, users receive emergency notifications if they have installed the application on their phone.

These two comments are very useful, that could guide future improvements.

7.3. Danger Zone Visualization Function – Question 4, 5, 6, 10 and 12

The danger zone visualization function is fulfilled in this research. Based on the feedback from Table 6.5, all participants found it easy to identify the boundaries of the danger zone on the map within the application. From Table 6.6 six participants thought it is “extremely useful”, five participants found it “very useful” while four participants found that it “moderately useful” in helping users to understand the potential impact of the disaster. Table 6. 7 results indicates that all participants believed that this application could help the public avoid entering dangerous areas during disasters.

A suggestion from Participant 15 in Table 6.13 proposed making the danger zone area, currently circled, more transparent to enhance visibility of area details. The current design highlights the danger zone with a yellow fill background and blue outline. This could be improved by adjusting the opacity to allow clearer viewing of details underneath the colored circle.

7.4. Overall Satisfaction - Question 7, 8 and 9

As the application is designed to report disaster event, receive timely notifications, and provide events details and visualization of danger zones, all 15 participants confirmed that this application is effectively provide these functionalities. Nine participants were very satisfied with this application, five participants expressed satisfaction, while one participant was neutral.

It is important to note that according to the results form Table 6.10, none of 15 participants has used a similar mobile application before. This confirms the findings of this research in literature review section that currently, there is no similar mobile application available in the app stores. There is an urgent need to fill this gap.

Participant 10 compared this application with the NZ Civil Defence Emergency Alert system which sends alert via SMS to all phones users. They commented that these alerts do not provide detailed information about the situation, making them easily overlooked.

7.5. Important Features - Question 10 and 11

Based on the feedback from Tables 6.11 and 6.12, such a mobile application should include the main features that allow users to report disaster events, receive notification alerts, and accessing incident details. Additionally, real-time danger zone mapping, navigation, and routing to safe areas are crucial features that should also be included into a mobile application designed to alert about danger zones as 14 participants selected these features.

7.6. Suggestion and Improvements - Question 6, 8, 12 and 13

In the Table 6.9 results, Participant 2 suggested that this application should be more accurate in verifying incidents. For example, if someone report a fake incident without proper verification, it could cause significant issues.

According to the feedback from Table 6.8, 6.13 and 6.14, Participant 13 suggested that the application should provide a live view with images and videos on the map allowing users to monitor events in real time. Participant 10 and 13 also recommended that within the danger zone area, the application could display the evacuation pross, including the directions to emergency services such as ambulance and police, as well as nearby safe zone.

Participant 9 indicated that this application could benefit from additional languages options or audio function for visually impaired users.

Participant 5 suggested that the application should include a feature to view past dangerous events in a specific area.

8. Improvements Based on User Feedback

After collecting feedback from participants, the research framework mentioned in Chapter 3 is implemented which mean it might become necessary to loop back the design, work on implementation and evaluation phases to refine application. This could further change the application's design. While all effort would be made to incorporate those changes to enhance functionality, any further development is beyond the scope of this research within limited time. However, one such feedback did result in an immediate implementation, based on the user's suggestion.

Currently, the disaster affected area is circled with a yellow background that is not transparent. One participant provided feedback that the danger zone should have lower opacity to make it easier to view the map underneath the affected area. Based on this feedback, the researcher has made changes to the opacity value, making it more transparent to allow visibility of what lies beneath the highlighted area. This enhancement of transparency in the danger zone visuals improves readability. Figure 8.1 is a comparison screenshot that indicates the state before and after the opacity adjustment, clearly showing what is underneath.

Figure 8.1: Comparison of Disaster-Affected Area Visibility Before and After Opacity Adjustment

9. Discussion and Conclusion

This section discusses the benefits, implications, limitations of the research. This identification will help the research to move in the right direction for application improvement.

9.1. How Findings Answer the Research Question

As stated in Chapter 1, the research question addressed in this study is:

How to use mobile technology to enhance real-time disaster reporting, notification, and danger zone visualization to support disaster preparedness?

The findings of this research indicate that the proposed mobile application prototype could bring real benefit for the community and enhance the disaster preparedness by unitizing its functionalities. In addition, this application could reduce the workload of the 111-call centre of New Zealand and improve the efficiency of reporting a disaster event.

People and communities equipped with such mobile applications would be better prepared for disasters, thereby minimizing the damage caused by providing timely information and resources. Utilizing the application could enhance community resilience and strengthen their ability to withstand disasters effectively.

9.2. Limitations

Although, the 15 participants provided positive feedback that the application is very easy to use and facilitates timely reporting and notifications of disasters and visualizing danger zone. There are still some limitations that need to be addressed and improved:

- Providing clear instructions to guide users on how report disasters.
- Simplifying the input process, offering more options such as AM/PM time formats. and pre-entering common disaster names for quicker and more accurate reporting.

9.3. Future work

Next step, this application should be evaluated by authorised group such as the New Zealand Civil Defence, or the Polic department or hospital. It should also allow users to check disaster events statistics based on various criteria, such as location or year. For example, users could search for the frequency of disasters in a specific area or track the number of disasters that occurred over the past decade.

In fact, this feature has already been created in the web server prototype, as shown in Figure 9.1. The events statistics allow admin to search based on the year, area, and severity level. The search results are displayed in two pie charts showing data by area and severity level, and a line chart showing the number of disasters per year. From the participants' feedback, it can be confirmed that this feature is very important. However, since the current research is focused on how users can report disaster events and view events details, including danger zone visualisation to enhance disaster preparedness, this feature was not included in this version of the prototype. Therefore, in future work, this feature can be expanded, allowing not only admin but also all mobile application users to search for and view past disaster statistics.

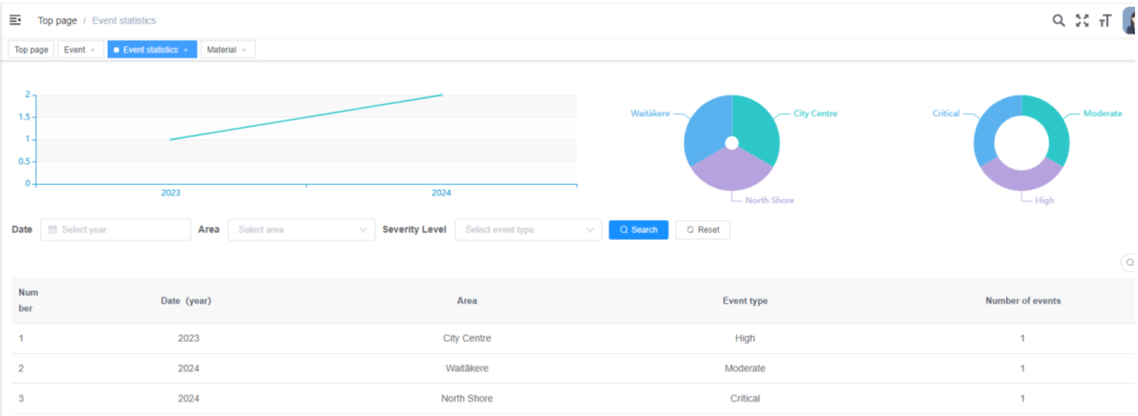


Figure 9.1: Disaster Events Statistics (Source: Author)

The application is designed for Android OS, but it should also be developed for iOS users who use iPhones or other supported operating systems. In the further development, it should consider various maps integrations, as mentioned in the implementation chapter, not only using Baidu Maps, but also Google Maps, or other mapping services.

From this research, it is evident that there is a practical need for this mobile application to be used in the community. It has good prospects. At the same time, there are some areas that need improvement.

Currently, drones are widely used for various purpose. Common uses include delivery, agriculture, search and rescue operations, military application and more. (Adnan & Khamis, 2022; Cheng et al., 2024; Shah et al., 2024). With the advancement of drone technology, they can also be utilized in this mobile application. For future research, it is proposed to explore the use of drones to verify disaster events. Drones could facilitate easier access to critical situations for verification purposes. Additionally, leveraging wireless communication technology could enable live streaming capabilities during verification processes. This chapter outlines Design Science Research (DSR) methodology and explains the reasons why it was selected. It also provides an in-depth explanation of DSR steps involved and followed throughout the research process, emphasising how this approach has supported the development of mobile application prototype that addresses the given situation in hand about disaster management challenges.

9.4. Conclusion

Facing increasingly frequent disasters, there is currently no more practical and effective method for the public to actively report disaster events or receive real-time disaster information. This research, through a review of relevant literature and an investigation into the use of existing disaster alert systems, combined with the widespread use of smartphones, has developed a mobile application. Through the development of this mobile application, the public has the opportunity to quickly report information in the event of a disaster and stay informed about disaster information and affected areas in a timely manner. This enables better disaster preparedness when disasters strike.

This research follows the DSR methodology, which involves stages such as problem identification, proposing solutions, development, and evaluation. The application development utilizes the Spring Boot + Vue.js framework. Fifteen participants were surveyed to evaluate the developed mobile application prototype. The evaluation covered feasibility, usability, and performance aspects of the application, including considerations for future enhancements.

According to the evaluation results, this mobile application helps the public report and understand disaster information promptly and prepare better for disasters. Additional features and suggestions can be further refined in future research to enhance the mobile application. The thesis also included the improvement based on the survey feedback.

References

Abid, S. K., Sulaiman, N., Chan, S. W., Nazir, U., Abid, M., Han, H., Ariza-Montes, A., & Vega-Muñoz, A. (2021). Toward an integrated disaster management approach: how artificial intelligence can boost disaster management. *Sustainability*, *13*(22), 12560.

Aboualola, M., Abualsaud, K., Khattab, T., Zorba, N., & Hassanein, H. S. (2023). Edge technologies for disaster management: A survey of social media and artificial intelligence integration. *IEEE Access*.

Adnan, W. H., & Khamis, M. F. (2022). Drone use in military and civilian application: Risk to national security. *Journal of Media and Information Warfare (JMIW)*, *15*(1), 60-70.

Ahirwar, S., Swarnkar, R., Bhukya, S., & Namwade, G. (2019). Application of drone in agriculture. *International Journal of Current Microbiology and Applied Sciences*, *8*(01), 2500-2505.

Ahmad, I., Suwarni, E., Borman, R. I., Rossi, F., & Jusman, Y. (2021). Implementation of RESTful API Web Services Architecture in Takeaway Application Development. 2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS).

Alsalamah, S., & Nuzzolese, E. (2020). Promising blockchain technology applications and use case designs for the identification of multinational victims of mass disasters. *Frontiers in Blockchain*, *3*, 34.

Alsunaidi, S. J., Almuhaideb, A. M., Ibrahim, N. M., Shaikh, F. S., Alqudaihi, K. S., Alhaidari, F. A., Khan, I. U., Aslam, N., & Alshahrani, M. S. (2021). Applications of big data analytics to control COVID-19 pandemic. *Sensors*, *21*(7), 2282.

Amatya, S., & Kurti, A. (2014). Cross-platform mobile development: challenges and opportunities. *ICT Innovations 2013: ICT Innovations and Education*, 219-229.

Ameller, D., Ayala, C., Cabot, J., & Franch, X. (2012). How do software architects consider non-functional requirements: An exploratory study. In *2012 20th IEEE international requirements engineering conference (RE)* (pp. 41-50). IEEE.

Amez, S., & Baert, S. (2020). Smartphone use and academic performance: A literature review. *International Journal of Educational Research*, *103*, 101618.

Android. (n.d.). *Android is for everyone*. https://www.android.com/everyone/enabling-opportunity/#creating-choices.

Anta, V. L. P., Liestyo, I. A., & Warnars, H. L. H. S. (2021). Mobile Application for flood disaster in Jakarta. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS).

Apps, B. o. (2024). *Android Statistics (2024)*. https://www.businessofapps.com/data/android-statistics/.

Ardito, L., Coppola, R., Malnati, G., & Torchiano, M. (2020). Effectiveness of Kotlin vs. Java in android app development tasks. *Information and Software Technology*, *127*, 106374.

Asadzadeh, A., Pakkhoo, S., Saeidabad, M. M., Khezri, H., & Ferdousi, R. (2020). Information technology in emergency management of COVID-19 outbreak. *Informatics in medicine unlocked*, *21*, 100475.

Astarita, V., Giofrè, V. P., Guido, G., Stefano, G., & Vitale, A. (2020). Mobile computing for disaster emergency management: Empirical requirements analysis for a cooperative crowdsourced system for emergency management operation. *Smart Cities*, *3*(1), 31-47.

Aten, J. D., Leavell, K., Gonzalez, R., Luke, T., Defee, J., & Harrison, K. (2011). Everyday technologies for extraordinary circumstances: Possibilities for enhancing disaster communication. *Psychological Trauma: Theory, Research, Practice, and Policy*, *3*(1), 16.

Bachmann, D. J., Jamison, N. K., Martin, A., Delgado, J., & Kman, N. E. (2015). Emergency preparedness and disaster response: there's an app for that. *Prehospital and disaster medicine*, *30*(5), 486-490.

Baresi, L., Di Penta, M., Quattrocchi, G., & Tamburri, D. A. (2024). How have iOS Development Technologies Changed over Time? A Study in Open-Source. Proceedings of the IEEE/ACM 11th International Conference on Mobile Software Engineering and Systems.

Bell, C. A. (2012). *Expert MySQL: Charles Bell (Second edition.)*. APress.

Bhandari, V. (2022). Use of technology in disaster management. *Unity Journal*, *3*(01), 292-304.

Bragazzi, N. L., Dai, H., Damiani, G., Behzadifar, M., Martini, M., & Wu, J. (2020). How big data and artificial intelligence can help better manage the COVID-19 pandemic. *International journal of environmental research and public health*, *17*(9), 3176.

Brewer, R., Garcia, R. C., Schwaba, T., Gergle, D., & Piper, A. M. (2016). Exploring traditional phones as an e-mail interface for older adults. *ACM Transactions on Accessible Computing (TACCESS)*, *8*(2), 1-20.

Budimir, M., Bee, E., & Paul, J. (2021). Using mobile phone technologies for Disaster Risk Management: reflections from SHEAR.

Busch, P. A., Hausvik, G. I., Ropstad, O. K., & Pettersen, D. (2021). Smartphone usage among older adults. *Computers in Human Behavior*, *121*, 106783.

Caldera, H. J., & Wirasinghe, S. (2022). A universal severity classification for natural disasters. *Natural hazards*, *111*(2), 1533-1573.

Carter, W. N. (2008). Disaster management: A disaster manager's handbook.

Chamayou, G. (2015). *Drone theory*. Penguin UK.

Charkaoui, S., & Adraoui, Z. (2014). Cross-platform mobile development approaches. 2014 Third IEEE International Colloquium in Information Science and Technology (CIST).

Chaudhary, M. T., & Piracha, A. (2021). Natural Disasters—Origins, Impacts, Management. *Encyclopedia*, *1*(4), 1101-1131. https://www.mdpi.com/2673-8392/1/4/84.

Chen, A. Y., Peña-Mora, F., & Ouyang, Y. (2011). A collaborative GIS framework to support equipment distribution for civil engineering disaster response operations. *Automation in construction*, *20*(5), 637-648.

Cheng, C., Adulyasak, Y., & Rousseau, L.-M. (2024). Robust drone delivery with weather information. *Manufacturing & Service Operations Management*.

Cheon, Y. (2019). Multiplatform application development for android and java. 2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA).

Chetty, P. (2023). *Satellite technology and its applications*. Notion Press.

Chung, L., & do Prado Leite, J. C. S. (2009). On non-functional requirements in software engineering. *Conceptual modeling: Foundations and applications: Essays in honor of john mylopoulos*, 363-379.

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). *Non-functional requirements in software engineering* (Vol. 5). Springer Science & Business Media.

Cicek, D., & Kantarci, B. (2023). Use of mobile crowdsensing in disaster management: A systematic review, challenges, and open issues. *Sensors*, *23*(3), 1699.

Crunch, C. (May 2021). Disaster Year in Review 2020 Global Trends and Perspectives. https://cred.be/sites/default/files/CredCrunch62.pdf.

Danielsson, W. (2016). React Native application development. *Linköpings universitet, Swedia*, *10*(4), 10.

Daud, S. M. S. M., Yusof, M. Y. P. M., Heo, C. C., Khoo, L. S., Singh, M. K. C., Mahmood, M. S., & Nawawi, H. (2022). Applications of drone in disaster management: A scoping review. *Science & Justice*, *62*(1), 30-42.

de Asis, L. A., Romero, B. N., de Asis-Estigoy, K. M. A., & Estigoy Jr, A. A. (2019). Community-based Disaster Preparedness and Challenges in Catubig, Northern Samar: Basis for Communication in Disasters Plan.

Development Asia. (2021 March). *how ai can boost disaster response and recovery*. https://development.asia/insight/how-ai-can-boost-disaster-response-and-recovery.

Dhakal, S., Zhang, L., & Lv, X. (2022). Ontology-based semantic modelling to support knowledge-based document classification on disaster-resilient construction practices. *International Journal of Construction Management*, *22*(11), 2059-2078.

Djalante, R., & Lassa, S. (2019). Governing complexities and its implication on the Sendai Framework for Disaster Risk Reduction priority 2 on governance. *Progress in Disaster Science*, *2*, 100010.

dominoc925. (2024). *Natural Disaster Monitor,*. https://play.google.com/store/apps/details?id=com.dom925.disastermon&hl=en.

Downey, T., & Downey, T. (2021). Spring Framework. *Guide to Web Development with Java: Understanding Website Creation*, 121-170.

DPMC. (2023). *National Emergency Management Agency*. Retrieved from https://www.dpmc.govt.nz/departmental-agency/nema.

Dresch, A., Lacerda, D. P., Antunes Jr, J. A. V., Dresch, A., Lacerda, D. P., & Antunes, J. A. V. (2015). *Design science research*. Springer.

Dusse, F., Júnior, P. S., Alves, A. T., Novais, R., Vieira, V., & Mendonça, M. (2016). Information visualization for emergency management: A systematic mapping study. *Expert Systems with Applications*, *45*, 424-437.

Dzezhyts, Y. (2013). Android application development.

ehinz. (n.d). *About natural hazards*.

Escolano, V. J. C., Caballero, A. R., Albina, E. M., Hernandez, A. A., & Juanatas, R. A. (2023). Acceptance of Mobile Application on Disaster Preparedness: Towards Decision Intelligence in Disaster Management. 2023 8th International Conference on Business and Industrial Research (ICBIR).

Esposito, M., Palma, L., Belli, A., Sabbatini, L., & Pierleoni, P. (2022). Recent advances in internet of things solutions for early warning systems: A review. *Sensors*, *22*(6), 2124.

Fernando, M. C. G., Solomo, M. V. S., & Lagman, A. C. (2019). iHanda: A mobile application for disaster preparedness. *International Journal of Simulation Systems, Science & Technology*, *20*.

Forman, J., & Damschroder, L. (2007). Qualitative content analysis. In *Empirical methods for bioethics: A primer* (pp. 39-62). Emerald Group Publishing Limited.

Furutani, T., & Minami, M. (2021). Drones for disaster risk reduction and crisis response. *Emerging Technologies for Disaster Resilience: Practical Cases and Theories*, 51-62.

GilPress. (2024). *How Many People Own Smartphones? (2024-2029)*. https://whatsthebigdata.com/smartphone-stats/.

GlobalStats, S. (2024). *Mobile Operating System Market Share Worldwide - June 2024*. https://gs.statcounter.com/os-market-share/mobile/worldwide.

GNS SCience. (2024). *GeoNet Quake*. https://apps.apple.com/nz/app/geonet-quake/id533054360?platform=iphone.

Guevara-Vega, C. P., Guzmán-Chamorro, E. D., Guevara-Vega, V. A., Andrade, A. V. B., & Quiña-Mera, J. A. (2019). Functional requirement management automation and the impact on software projects: case study in Ecuador. Information Technology and Systems: Proceedings of ICITS 2019.

Hajesmaeel-Gohari, S., Khordastan, F., Fatehi, F., Samzadeh, H., & Bahaadinbeigy, K. (2022). The most used questionnaires for evaluating satisfaction, usability.

acceptance, and quality outcomes of mobile health. *BMC Medical Informatics and Decision Making*, *22*(1), 22.

Hata, H., Novielli, N., Baltes, S., Kula, R. G., & Treude, C. (2022). GitHub Discussions: An exploratory study of early adoption. *Empirical Software Engineering*, *27*, 1-32.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.

Hong, S., Kang, J., & Kwon, S. (2023). Performance Comparison of HTTP, HTTPS, and MQTT for IoT Applications. *The International Journal of Advanced Smart Convergence*, *12*(1), 9-17.

Hunt, K., Narayanan, A., & Zhuang, J. (2022). Blockchain in humanitarian operations management: A review of research and practice. *Socio-Economic Planning Sciences*, *80*, 101175.

iDMC. (2024). *New Zealand - Disaster resilience and tailored responses mitigate impact of Cyclone Gabrielle*. https://www.internal-displacement.org/spotlights/New-Zealand-Disaster-resilience-and-tailored-responses-mitigate-impact-of-Cyclone-Gabrielle/.

IxDF. (2024). *What is Material Design?* https://www.interaction-design.org/literature/topics/material-design.

Jaber, M. M., Ali, M. H., Abd, S. K., Jassim, M. M., Alkhayyat, A., Aziz, H. W., & Alkhuwaylidee, A. R. (2022). Predicting climate factors based on big data analytics based agricultural disaster management. *Physics and Chemistry of the Earth, Parts A/B/C*, *128*, 103243.

Jindal, T. (2016). Importance of Testing in SDLC. *International Journal of Engineering and Applied Computer Science (IJEACS)*, *1*(02), 54-56.

Kaur, P., Stoltzfus, J., & Yellapu, V. (2018). Descriptive statistics. *International Journal of Academic Medicine*, *4*(1), 60-63.

Knopf, J. W. (2006). Doing a literature review. *PS: Political Science & Politics*, *39*(1), 127-132.

Kopacz, J. R., Herschitz, R., & Roney, J. (2020). Small satellites an overview and assessment. *Acta Astronautica*, *170*, 93-105.

Krichen, M., Abdalzaher, M. S., Elwekeil, M., & Fouda, M. M. (2023). Managing natural disasters: An analysis of technological advancements, opportunities, and challenges. *Internet of Things and Cyber-Physical Systems*.

Krishna, V. V., & Gopinath, G. (2021). Test automation of web application Login Page by using selenium ide in a web browser. *Management*, 713-732.

Kubíček, P., & Staněk, K. (2006). Dynamic visualization in emergency management. Proceedings of First international conference on cartography and GIS. Sofia: Sofia Univerzity.

Kuechler, W., & Vaishnavi, V. (2012). A framework for theory development in design science research: multiple perspectives. *Journal of the Association for Information systems*, *13*(6), 3.

Kumpulainen, T. (2021). Web application development with Vue. js.

Kute, S. S., & Thorat, S. D. (2014). A review on various software development life cycle (SDLC) models. *International Journal of Research in Computer and Communication Technology*, *3*(7), 778-779.

Langer, A. M., & Langer, A. M. (2008). System development life cycle (SDLC). *Analysis and Design of Information Systems: Third Edition*, 10-20.

Latvakoski, J., Öörni, R., Lusikka, T., & Keränen, J. (2022). Evaluation of emerging technological opportunities for improving risk awareness and resilience of vulnerable people in disasters. *International Journal of Disaster Risk Reduction*, *80*, 103173.

Lettieri, E., Masella, C., & Radaelli, G. (2009). Disaster management: findings from a systematic review. *Disaster Prevention and Management: An International Journal*.

Li, L., & Ulaganathan, M. N. (2017). Design and development of a crowdsourcing mobile app for disaster response. 2017 25th International Conference on Geoinformatics.

Li, X., & Da Xu, L. (2020). A review of Internet of Things—Resource allocation. *IEEE Internet of Things Journal*, *8*(11), 8657-8666.

Li, Z. (2022). Design of ordering system based on spring boot framework. *International Core Journal of Engineering*, *8*(5), 579-588.

Lim, W. M., Kumar, S., & Ali, F. (2022). Advancing knowledge through literature reviews:'what','why', and 'how to contribute'. *The Service Industries Journal*, *42*(7-8), 481-513.

Long, L., He, F., & Liu, H. (2021). The use of remote sensing satellite using deep learning in emergency monitoring of high-level landslides disaster in Jinsha River. *The Journal of Supercomputing*, *77*(8), 8728-8744.

Macaulay, M. (2017). *Introduction to web interaction design: With Html and Css*. Chapman and Hall/CRC.

Martin-Lopez, A., Segura, S., & Ruiz-Cortés, A. (2022). Online testing of RESTful APIs: Promises and challenges. Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering.

Maryam, H., Shah, M. A., Javaid, Q., & Kamran, M. (2016). A survey on smartphones systems for emergency management (SPSEM). *International Journal of Advanced Computer Science and Applications*, *7*(6).

Medium. (2023). *How does the SDLC play a role in the success of product development and launch?* https://medium.com/agileinsider/how-does-the-sdlc-play-a-role-in-the-success-of-product-development-and-launch-a17baaac1054.

Meechang, K., Leelawat, N., Tang, J., Kodaka, A., & Chintanapakdee, C. (2020). The acceptance of using information technology for disaster risk management: A systematic review. *Engineering Journal*, *24*(4), 111-132.

Méndez Fernández, D., & Wieringa, R. (2013). Improving requirements engineering by artefact orientation. Product-Focused Software Process Improvement: 14th International Conference, PROFES 2013, Paphos, Cyprus, June 12-14, 2013. Proceedings 14.

MetService. (2024). *Weather on your mobile*. https://about.metservice.com/our-company/ways-to-get-the-weather/weather-on-your-mobile/.

Mizutori, M. (2019). From risk to resilience: Pathways for sustainable development. *Progress in Disaster Science*, *2*, 100011.

Mobaraki, A., Mansourian, A., Malek, M., & Mohammadi, H. (2007). Application of mobile GIS and SDI for emergency management. *Revue Francaise de Photogrammetric et de Teledetection*, *185*, 95-100.

Mojtahedi, M., Sunindijo, R. Y., Lestari, F., & Wijaya, O. (2021). Developing hospital emergency and disaster management index using topsis method. *Sustainability*, *13*(9), 5213.

MoldStud. (2024). *Developing Android Apps for Natural Disaster Preparedness*. https://moldstud.com/articles/p-developing-android-apps-for-natural-disaster-preparedness#:~:text=Real%2Dtime%20Alerts%20and%20Updates,action%20to%20ensure%20their%20safety.

Monte, B. E. O., Goldenfum, J. A., Michel, G. P., & de Albuquerque Cavalcanti, J. R. (2021). Terminology of natural hazards and disasters: A review and the case of Brazil. *International Journal of Disaster Risk Reduction*, *52*, 101970.

Mukhopadhyay, B., & Bhattacherjee, B. (2015). Use of information technology in emergency and disaster management. *American Journal of Environmental Protection*, *4*(2), 101-104.

Munir, A., Aved, A., & Blasch, E. (2022). Situational awareness: techniques, challenges, and prospects. *AI*, *3*(1), 55-77.

National Emergency Management Agency. (2024). *About National Emergency Management Agency*. Retrieved from https://www.civildefence.govt.nz/about/about-nema.

National Library of Medicine. (2023). *Improvements and Persisting Challenges in COVID-19 Response Compared with 1918–19 Influenza Pandemic Response, New Zealand (Aotearoa)*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10461674/#:~:text=New%20Zealand%20(Aotearoa%2C%20the%20commonly,%2DJune%202023%20(3).

Nawaz, S. (2024). Distinguishing between effectual, ineffectual, and problematic smartphone use: a comprehensive review and conceptual pathways model for future research. *Computers in Human Behavior Reports*, 100424.

Neene, V., & Kabemba, M. (2017). Development of a mobile GIS property mapping application using mobile cloud computing. *International Journal of Advanced Computer Science and Applications*, *8*(10).

Nelson City Council. (2023). *Red Cross Hazard App*. https://www.nelson.govt.nz/services/apps/hazard-app-new-zealand-red-cross/#:~:text=The%20Hazard%20App%20is%20a,to%20hazards%20in%20New%20Zealand.

New Zealand Civil Defence. (2023). *National Emergency Management Agency*. https://www.civildefence.govt.nz/.

New Zealand Government. (2021). *Get information in an emergency or disaster*. https://www.govt.nz/organisations/national-emergency-management-agency/get-information-in-emergency-or-disaster/.

New Zealand Red Cross. (2022). *Hazards - Red Cross*. https://play.google.com/store/apps/details?id=com.cube.gdpc.nzl.hzd.

New Zealand Red Cross. (n.d.). *Prepare for a disaster – get Good and Ready*. https://www.redcross.org.nz/get-help/emergencies-and-disasters/prepare-for-a-disaster-get-good-and-ready/.

Nobre, G. G., Muis, S., Veldkamp, T. I., & Ward, P. J. (2019). Achieving the reduction of disaster risk by better predicting impacts of El Niño and La Niña. *Progress in Disaster Science*, *2*, 100022.

Nowak, M. M., Dzió識, K., Ludwisiak, Ł., & Chmiel, J. (2020). Mobile GIS applications for environmental field surveys: A state of the art. *Global Ecology and Conservation*, *23*, e01089.

Nurbekova, Z., Grinshkun, V., Aimicheva, G., Nurbekov, B., & Tuenbaeva, K. (2020). Project-based learning approach for teaching mobile application development using visualization technology. *International Journal of Emerging Technologies in Learning (iJET)*, *15*(8), 130-143.

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). Outline of a design science research process. Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology.

Olorunshola, O. E., & Ogwueleka, F. N. (2022). Review of system development life cycle (SDLC) models for effective application delivery. Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces.

Ozdemir, E. (2020). A general overview of RESTful web services. *Applications and approaches to object-oriented software design: emerging research and opportunities*, 133-165.

Pachika, A. (2022). *Review of Satellite Image Analysis in Infrastructure Management and Disaster Management*.

Pacific Disaster Center. (2024). *Disaster Alert*. https://play.google.com/store/apps/details?id=disasterAlert.PDC&hl=en_NZ&pli=1.

Pal, A., Raj, M., Kant, K., & Das, S. K. (2020). A smartphone-based network architecture for post-disaster operations using WiFi tethering. *ACM Transactions on Internet Technology (TOIT)*, *20*(1), 1-27.

Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T., & Ozkaya, I. (2013). Parallel worlds: Agile and waterfall differences and similarities. *Software Engineering Institute*, *1*(1), 1-3.

Pamudji, A., Susilorini, R., Ismail, A., & Amasto, A. H. (2020). The Effectiveness of Mobile Application of Earthquake and Tsunami Early Warning System in Community Based Disaster Risk Reduction. *Int. J. Eng. Res. Technol.*, *13*.

Paracha, M. T., Chandrasekara, B., Choffnes, D., & Levin, D. (2020). A Deeper Look at Web Content Availability and Consistency over HTTP/S. 2020 Network Traffic Measurement and Analysis Conference (TMA'20).

Paul, J., & Criado, A. R. (2020). The art of writing literature review: What do we know and what do we need to know? *International Business Review*, *29*(4), 101717.

Paul, J. D., Bee, E., & Budimir, M. (2021). Mobile phone technologies for disaster risk reduction. *Climate Risk Management*, *32*, 100296.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, *24*(3), 45-77.

Pine, J. C. (2017). *Technology and emergency management*. John Wiley & Sons.

Posey, B. (2024). *Apple iOS*. https://www.techtarget.com/searchmobilecomputing/definition/iOS#:~:text=Apple%20iOS%20is%20a%20proprietary,use%20with%20Apple's%20multitouch%20devices.

Pour, F. S. A. (2021). *Application of a blockchain enabled model in disaster aids supply network resilience* Old Dominion University.

Pour, F. S. A., & Gheorghe, P. N.-M. (2021). A blockchain-enabled model to enhance disaster aids network resilience. *Romanian Cyber Security Journal*, *3*(2).

Pukdesree, S. (2017). The comparative study of collaborative learning and SDLC model to develop IT group projects. *TEM Journal*, *6*(4), 800-809.

Ragunath, P., Velmourougan, S., Davachelvan, P., Kayalvizhi, S., & Ravimohan, R. (2010). Evolving a new model (SDLC Model-2010) for software development life cycle (SDLC). *International Journal of Computer Science and Network Security*, *10*(1), 112-119.

Raj, G., Singh, D., & Bansal, A. (2014). Analysis for security implementation in SDLC. 2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence).

Ritchie, H., & Rosado, P. (2022). *Natural Disasters - How many people die from disasters, and how are these impacts changing over time?* https://ourworldindata.org/natural-disasters.

Robertson, B. W., Johnson, M., Murthy, D., Smith, W. R., & Stephens, K. K. (2019). Using a combination of human insights and 'deep learning'for real-time disaster communication. *Progress in Disaster Science*, *2*, 100030.

Rothenberger, K. P. M., & Kuechler, B. (2012). Design Science Research in Information Systems.

Sakurai, M., & Murayama, Y. (2019). Information technologies and disaster management–Benefits and issues. *Progress in Disaster Science*, *2*, 100012.

Sarker, M. N. I., Peng, Y., Yiran, C., & Shouse, R. C. (2020). Disaster resilience through big data: Way to environmental sustainability. *International Journal of Disaster Risk Reduction*, *51*, 101769.

Sarker, M. N. I., Wu, M., Chanthamith, B., & Ma, C. (2020). Resilience through big data: natural disaster vulnerability context. Proceedings of the Fourteenth International Conference on Management Science and Engineering Management: Volume 1.

Sarrion, E. Master Vue. js in 6 Days.

Scroggins, R. (2014). SDLC and development methodologies. *Global Journal of Computer Science and Technology*, *14*(7-C), 21.

Senarath, U. S. (2021). Waterfall methodology, prototyping and agile development. *Tech. Rep.*, 1-16.

Shah, I. A., Jhanjhi, N. Z., & Rajper, S. (2024). Use of Deep Learning Applications for Drone Technology. In *Cybersecurity Issues and Challenges in the Drone Industry* (pp. 128-147). IGI Global.

Sharma, K., Anand, D., Sabharwal, M., Tiwari, P. K., Cheikhrouhou, O., & Frikha, T. (2021). A Disaster Management Framework Using Internet of Things-Based Interconnected Devices. *Mathematical Problems in Engineering*, *2021*(1), 9916440.

Sharma, M. K. (2017). A study of SDLC to develop well engineered software. *International Journal of Advanced Research in Computer Science*, *8*(3).

Sharma, S. K., Misra, S. K., & Singh, J. B. (2020). The role of GIS-enabled mobile applications in disaster management: A case analysis of cyclone Gaja in India. *International Journal of Information Management*, *51*, 102030.

Solokha, M., Pereira, P., Symochko, L., Vynokurova, N., Demyanyuk, O., Sementsova, K., Inacio, M., & Barcelo, D. (2023). Russian-Ukrainian war impacts on the environment. Evidence from the field on soil properties and remote sensing. *Science of the Total Environment*, *902*, 166122.

Srivastava, A., Verma, R., & Srivastav, V. (2020). *Weather Finder: An Application for weather forecasting* (2516-2314).

Steele, J., & To, N. (2010). *The Android developer's cookbook: building applications with the Android SDK*. Pearson Education.

Stute, M., Maass, M., Schons, T., Kaufhold, M.-A., Reuter, C., & Hollick, M. (2020). Empirical insights for designing information and communication technology for international disaster response. *International Journal of Disaster Risk Reduction*, *47*, 101598.

Sun, W., Bocchini, P., & Davison, B. D. (2020). Applications of artificial intelligence for disaster management. *Natural hazards*, *103*(3), 2631-2689.

Suzianti, A., Wulandari, A. D., Yusuf, A. H., Belahakki, A., & Monika, F. (2020). Design thinking approach for mobile application design of disaster mitigation management. Proceedings of the 2020 2nd Asia Pacific Information Technology Conference.

Syukron, M., Madugalla, A., Shahin, M., & Grundy, J. (2024). A Comprehensive Study of Disaster Support Mobile Apps. *arXiv preprint arXiv:2407.08145*.

Tan, M. L., Prasanna, R., Stock, K., Hudson-Doyle, E., Leonard, G., & Johnston, D. (2017). Mobile applications in crisis informatics literature: A systematic review. *International Journal of Disaster Risk Reduction*, *24*, 297-311.

Tashildar, A., Shah, N., Gala, R., Giri, T., & Chavhan, P. (2020). Application development using flutter. *International Research Journal of Modernization in Engineering Technology and Science*, *2*(8), 1262-1266.

Tilley, S. R., & Rosenblatt, H. J. (2017). *Systems analysis and design: Tilley, Rosenblatt (Eleventh edition.)*.

Tiun, S., Mokhtar, U., Bakar, S., & Saad, S. (2020). Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text. journal of Physics: conference series.

Toledo, M. P., Sarvida, J. C. Y., Patiten, B. K. S., Mitamura, D. C. R., & Guadaña, R. R. H. (2017). SakunAPP: A framework for mobile application development in disaster awareness, preparedness and response. TENCON 2017-2017 IEEE Region 10 Conference.

Torani, S., Majd, P. M., Maroufi, S. S., Dowlati, M., & Sheikhi, R. A. (2019). The importance of education on disasters and emergencies: A review article. *Journal of education and health promotion*, *8*.

Turner, A. (2024). *How Many Smartphones Are In The World?* https://www.bankmycell.com/blog/how-many-phones-are-in-the-world.

UNDRR. (2020). *Human Cost of Disasters 2000-2019 Report*. https://www.undrr.org/publication/human-cost-disasters-overview-last-20-years-2000-2019.

Van Tilburg, C. (2017). First report of using portable unmanned aircraft systems (drones) for search and rescue. *Wilderness & environmental medicine*, *28*(2), 116-118.

Venable, J., Pries-Heje, J., & Baskerville, R. (2016). FEDS: a framework for evaluation in design science research. *European journal of information systems*, *25*(1), 77-89.

Venable, J. R., Pries-Heje, J., & Baskerville, R. L. (2017). Choosing a design science research methodology.

Wahyudi, D. (2022). Disaster Risk Reduction for Natural Disaster using Mobile Learning Application to Improve the Students Disaster Mitigation Literacy in Elementary School. GMPI Conference Series.

Walls, C. (2015). *Spring Boot in action*. Simon and Schuster.

Wang, Y., Li, J., Zhao, X., Feng, G., & Luo, X. (2020). Using mobile phone data for emergency management: a systematic literature review. *Information Systems Frontiers*, *22*, 1539-1559.

Weather Underground. (2024). *Weather Underground*. https://play.google.com/store/apps/details?id=com.wunderground.android.weather&hl=en_NZ.

Wich, M., & Kramer, T. (2015). Enhanced human-computer interaction for business applications on mobile devices: a design-oriented development of a usability evaluation questionnaire. 2015 48th Hawaii International Conference on System Sciences.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

Winter, R., & vom Brocke, J. (2021). Teaching Design Science Research. ICIS.

Yu, E. S., & Mylopoulos, J. (1994). Understanding" why" in software process modelling, analysis, and design. Proceedings of 16th international conference on software engineering.

Yu, M., Yang, C., & Li, Y. (2018). Big data in natural disaster management: a review. *Geosciences*, *8*(5), 165.

Zhang, F., Sun, G., Zheng, B., & Dong, L. (2021). Design and implementation of energy management system based on spring boot framework. *Information*, *12*(11), 457.

Zhao, X., Pan, S., Sun, Z., Guo, H., Zhang, L., & Feng, K. (2021). Advances of satellite remote sensing technology in earthquake prediction. *Natural Hazards Review*, *22*(1), 03120001.

Zohud, T., & Zein, S. (2021). Cross-platform mobile app development in industry: A multiple case-study. *International Journal of Computing*, *20*(1), 46-54.

Zwęgliński, T. (2020). The use of drones in disaster aerial needs reconnaissance and damage assessment–three-dimensional modeling and orthophoto map study. *Sustainability*, *12*(15), 6080.

Appendices

Appendix A – Ethical Approval Letter

Auckland University of Technology Ethics Committee (AUTEC)

20 May 2024

Krassie Petrova
Faculty of Design and Creative Technologies

Dear Krassie

Re Ethics Application: **24/128 Enhancing Disaster Preparedness: Development of an Android Mobile Application for Real-time Disaster Reporting, Notification, and Danger Zone Visualization.**

Thank you for your responses to AUTEC's conditions.

Your ethics application has been approved for three years until 20 May 2027.

Standard Conditions of Approval

1. The research is to be undertaken in accordance with the Auckland University of Technology Code of Conduct for Research and as approved by AUTEC.
2. All public facing documents must have the AUTEC approval number and be of a high standard of spelling and grammar. Dates on the Information Sheet(s) and Consent Form(s) must be consistent.
3. Any amendments to the project must be approved by AUTEC prior to being implemented.
4. A progress report is due annually on the anniversary of the approval date.
5. A final report is due at the expiration of the approval period, or, upon completion of project.
6. Any serious or adverse events must be reported to AUTEC, this includes unforeseen issues that might affect continued ethical acceptability of the project.
7. AUTEC grants ethical approval only. You are responsible for obtaining management permission for access from any institution or organisation at which your research is being conducted and you need to meet all ethical, legal, public health, and locality obligations or requirements for the jurisdictions in which the research is being undertaken.

The application number and title need to be referenced on all correspondence related to this project.

All forms are available online http://www.aut.ac.nz/research/researchethics

For any enquiries, please contact ethics@aut.ac.nz
(This is a computer-generated letter for which no signature is required)

The AUTEC Secretariat
Auckland University of Technology Ethics Committee

Cc: hanjunhj@hotmail.com

Appendix B – Tools

a. Participant Information Sheet

AUT
TE WĀNANGA ARONUI
O TĀMAKI MAKAU RAU

Participant Information Sheet

Date Information Sheet Produced:

20 May 2024

Project Title

Enhancing Disaster Preparedness: Development of an Android Mobile Application for Real-time Disaster Reporting, Notification, and Danger Zone Visualization.

An Invitation

My name is Jun Han. I am a Master's student at the School of Engineering, Computer, and Mathematical Sciences at AUT. I am currently working on my Master's Thesis. I would like to invite you to participate in my research project.

The primary focus of my research is the development of a mobile application aimed at enhancing disaster preparedness and response within our community. Your participation in this study would be greatly appreciated, as your insights and feedback will play a crucial role in improving the functionality and usability of this application.

Participation in the research will involve evaluating the mobile application and providing feedback through a short questionnaire. Your participation in this research is entirely voluntary, and you are free to withdraw at any time before the completion of the data collection, without facing any adverse consequence.

What is the purpose of this research?

The purpose of this research is to explore how mobile technology can be leveraged to enhance real-time disaster reporting, notification, and danger zone visualization, with the primary goal of improving disaster preparedness among the public. The motivation behind the study is to rapidly share detailed information about disaster events and danger zones with the public and emergency services, as this is crucial for minimizing the impact of disasters. This helps the public avoid entering hazardous areas, while also facilitating the quick dispatch of equipment and emergency supplies by public emergency services to the affected areas.

Furthermore, the research seeks to identify key features essential for an effective emergency disaster reporting system and identify the critical information that the public should be made aware of.

The findings of this research will be used for the discussion part of my thesis. I also intend to prepare and submit a journal paper that will present the findings of my research to the academic community.

How was I identified and why am I being invited to participate in this research?

You have been invited to participate in this research project because you represent the general public and have experience using mobile applications. Given that disasters can occur unexpectedly and affect anyone, your perspective as a member of the general public is invaluable to my research.

How do I agree to participate in this research?

Even though we are in the same church, please know that joining this study is entirely your choice. Whether you decide to take part or not, it won't change your relationship with me or your position in the church. Your decision is respected, and there won't be any consequences for your connection with me or the church.

If you are interested in participating, please express your interest by sending an email to the address provided in the advertisement. Alternatively, you can send your interest to hanjunhj@hotmail.com. Upon receiving your expression of interest, I will send you a consent form to sign. Digital signatures are accepted. Once I receive the signed consent form, I will contact you to confirm the venue and date for evaluating the mobile application.

Your participation in this research is voluntary (it is your choice) and whether or not you choose to participate will neither advantage nor disadvantage you. You are able to withdraw from the study at any time. If you choose to withdraw from the study, then you will be offered the choice between having any data that is identifiable as belonging to you removed or allowing it to continue to be used. However, once the findings have been produced, removal of your data may not be possible.

What will happen in this research?

The evaluation process will take approximately 10 to 15 minutes to evaluate the mobile application prototype, and around 10 to 15 minutes to complete the questionnaire.

Here's how the evaluation and data gathering will proceed:

1. I will provide you with a smart phone already installed with the mobile application.
2. You will use the mobile application to simulate reporting a mock disaster event. You will also need to check if you can receive the event notification on the phone and view the event information including the danger zone on the map, through the mobile application. I will assist you throughout this process (If assistance is needed).
3. After the evaluation, you will provide your feedback by completing a provided questionnaire. You will be able to ask any questions you may have.

What are the discomforts and risks?

There are no expected discomforts and risks. However, if you think a question sensitive and prefer not to answer, you can choose not to.

How will these discomforts and risks be alleviated?

You have the option to decline to answer a question or to provide a brief answer without explaining your reasons.

What are the benefits?

The main benefit for the researcher is collecting high-quality data to successfully finish the project. Although you may not directly benefit, your involvement will help the wider community prepare better for disasters and reduce damage through the mobile application's timely information and resources. Additionally, this study will contribute to the researcher's academic qualification.

How will my privacy be protected?

The consent form is provided to you to ensure your rights are protected. Your name and contact details on the consent form will not be disclosed in the thesis or in any publication. Instead, your feedback on the questionnaire will be anonymized and referenced using an assigned index identifier, such as Participant-1, Participant-2, etc. The consent form and questionnaire will be stored securely at AUT for six years. After six years, they will be destroyed.

What are the costs of participating in this research?

There are no financial costs associated with this research. However, it will require approximately 30 minutes of your time.

What opportunity do I have to consider this invitation?

Your participation is completely voluntary. If you are unable to make an immediate decision, you may still reach out to me within one week of receiving the invitation.

Will I receive feedback on the results of this research?

The findings will initially be presented in the form of a thesis. If desired, you will receive an email containing the thesis URL once it is accessible online.

What do I do if I have concerns about this research?

Any concerns regarding the nature of this project should be notified in the first instance to the Project Supervisor, *Name: Krassie Petrova, Email: Krassie.pertrova@aut.ac.nz, and work phone number: 09-921-9999, x.5045*

Concerns regarding the conduct of the research should be notified to the Executive Secretary of AUTEC, *ethics@aut.ac.nz* , (+649) 921 9999 ext 6038.

Whom do I contact for further information about this research?

Please keep this Information Sheet and a copy of the Consent Form for your future reference. You are also able to contact the research team as follows:

Researcher Contact Details:

Name: Jun Han | Email: hanjunhj@hotmail.com

Project Supervisor Contact Details:

Primary Supervisor: Krassie Petrova | Email: Krassie.petrova@aut.ac.nz

Secondary Supervisor: Sarita Pais | Email: Sarita.pais@gmail.com

Approved by the Auckland University of Technology Ethics Committee on *20 May*, AUTEC Reference number *24/128*.

b. Consent Form

AUT
TE WĀNANGA ARONUI O TĀMAKI MAKAU RAU

Consent Form

Project title: ***Enhancing Disaster Preparedness: Development of an Android Mobile Application for Real-time Disaster Reporting, Notification, and Danger Zone Visualization.***

Project Supervisor: ***Krassie Petrova (Primary Supervisor), Sarita Pais (Secondary Supervisor)***

Researcher: ***Jun Han***

○ I have read and understood the information provided about this research project in the Information Sheet dated 20 May 2024.

○ I have had an opportunity to ask questions and to have them answered.

○ I understand that my opinions, data and comments expressed by me may be used as a part of the thesis work which is stated in the Information Sheet.

○ I understand that taking part in this study is voluntary (my choice) and that I may withdraw from the study at any time without being disadvantaged in any way.

○ I understand that if I withdraw from the study then I will be offered the choice between having any data that is identifiable as belonging to me removed or allowing it to continue to be used. However, once the findings have been produced, removal of my data may not be possible.

○ I agree to take part in this research.

○ I wish to receive a summary of the research findings (please tick one): Yes○ No○

Participant's signature: ....

Participant's name: ....

Participant's Contact Details (if appropriate):

....

....

....

....

Date:

Approved by the Auckland University of Technology Ethics Committee on 20 May AUTEC Reference number 24/128

Note: The Participant should retain a copy of this form.

c. Questionnaire

Mobile Application for Disaster Reporting and Danger Zone Visualization

User Feedback Questionnaire

Introduction:

Thank you for participating in my study. Your feedback is essential for evaluating the effectiveness of our mobile application for disaster reporting and danger zone visualization. Please take a few minutes to answer the following questions.

Disaster Event Reporting:

1. **How easy was it to use the disaster event reporting feature within the mobile application?**
   - ☐ Extremely difficult
   - ☐ Difficult
   - ☐ Neutral
   - ☐ Easy
   - ☐ Extremely easy

2. **How useful do you find the mobile application in engaging users to report disaster events and danger zones?**
   - ☐ Not effective
   - ☐ Slightly effective
   - ☐ Moderately effective
   - ☐ Effective
   - ☐ Very effective

Emergency Notifications:

3. **Were you able to receive emergency notification through the application that a disaster event has occurred?**
   - ☐ Yes
   - ☐ No

Danger Zone Visualization:

4. **Were you able to easily identify the boundaries of the danger zone on the map using the application.**
   - ☐ Yes
   - ☐ No
   - ☐ Other ____________

5. **How useful was the visual representation of the danger zone in helping you understand the potential impact of the disaster?**
   - ☐ Not useful
   - ☐ Slightly useful
   - ☐ Moderately useful
   - ☐ Very useful
   - ☐ Extremely useful

Public Safety Impact:

6. **Do you believe that having access to this type of application could help the public avoid entering dangerous areas during the disasters? Please provide your comments.**

Overall Feedback:

7. **How satisfied are you with overall performance of the application in reporting disaster events, receiving emergency notifications and visualizing danger zones?**
   - ☐ Very dissatisfied
   - ☐ Dissatisfied
   - ☐ Neutral
   - ☐ Satisfied
   - ☐ Very Satisfied

8. **With regards to the application that you are evaluating, do you believe that (Select all that apply)):**
   - ☐ The application effectively provides real-time emergency notifications.
   - ☐ The application delivers accurate and up-to-date information about danger zones.
   - ☐ I find the maps and danger zone visualization features user-friendly.
   - ☐ The application offers valuable guidance for responding to emergencies or navigating danger zones.
   - ☐ It effectively notifies about nearby emergency services or resources.
   - ☐ The application's speed and responsiveness meet my expectations.
   - ☐ The application provides detailed information on potential dangers in specific areas.
   - ☐ Additional Comments (if your thoughts are not listed above):

   ____

9. **If you have used any mobile applications designed for emergency notification and danger zone visualization in the past, please share your thoughts on your experience with those applications and compare them to the one you are currently evaluating.**

   ____

10. If you were to use a mobile application to receive alerts about disaster danger zone, what features would you find most valuable? (Select all that apply)

 Real-time danger zone mapping.
 Two-way communication to report incidents.
 Navigation and routing to safe areas.
 Ability to share alerts with others.
 Notification with sound and visual cues.
 Other (please specify): _______________

11. What types of information are most important for you to receive in an alert about a disaster danger zone? (Select all that apply)

 Evacuation instructions.
 Emergency contact information.
 Estimated time of arrival for the danger zone.
 Description of the disaster event.
 Location and proximity of the danger zone.
 Other (Please specify). _____________

12. Based on your experience with the current application, what specific improvements or changes do you think would enhance its disaster event reporting functionality?

____

13. What additional features or functions, beyond those in the current application, would you like to see implemented to make it more effective for disaster reporting and visualization?

____

Conclusion:

Your responses are greatly appreciated and will help us enhance the mobile application's effectiveness, user engagement, and ethical considerations.

Thank you for your valuable feedback!

Appendix C – Source Code for Main Mobile Application Interface

- Login Interface

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="344dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="@drawable/bg">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="center_vertical"
                android:orientation="vertical">
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="260dp"
                    android:layout_gravity="center"
                    android:text="Disaster Emergency Events "
                    android:textColor="@color/black"
                    android:textSize="28dp"
                    android:textStyle="bold" />
                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_gravity="center"
                    android:text="Application"
                    android:textColor="@color/black"
                    android:textSize="28dp"
                    android:textStyle="bold" />
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="24dp">
        <com.google.android.material.textfield.TextInputLayout
            android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp">
    <EditText
        android:id="@+id/et_username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#00ffffff"
        android:hint="Username"
        android:inputType="textPersonName" />
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp">

    <EditText
        android:id="@+id/et_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#00ffffff"
        android:hint="Password"
        android:inputType="textPassword" />
</com.google.android.material.textfield.TextInputLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="24dp">

<Button
    android:id="@+id/btn_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:layout_weight="1"
    android:layout_marginTop="24dp"
    android:padding="12dp"
    android:background="@color/indigo_500"
    android:text="Login"/>

<Button
    android:id="@+id/btn_register"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
```

```
            android:layout_weight="1"
            android:layout_marginTop="24dp"
            android:padding="12dp"
            android:background="@color/indigo_200"
            android:text="Register"/>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

- Dashboard Interface

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            app:contentScrim="?attr/colorPrimary"
            android:background="@color/indigo_500"
            app:layout_scrollFlags="scroll|enterAlways|enterAlwaysCollapsed">

            <androidx.appcompat.widget.Toolbar
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize">

                <TextView
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginLeft="24dp"
                    android:layout_gravity="left"
                    android:text="Menu"
                    android:textSize="18dp"
                    android:textColor="@color/white" />

                <ImageButton
                    android:id="@+id/btn_back"
                    android:layout_width="36dp"
                    android:layout_height="36dp"
                    android:layout_marginRight="8dp"
                    android:layout_gravity="right"
                    android:padding="8dp"
                    android:background="@color/transparent"
                    android:scaleType="fitXY"
                    app:srcCompat="@drawable/app_logout" />
            </androidx.appcompat.widget.Toolbar>
        </com.google.android.material.appbar.AppBarLayout>
    </com.google.android.material.appbar.AppBarLayout>

```

```
    <TableLayout
        android:layout_width="404dp"
        android:layout_height="303dp"
        android:layout_gravity="center"
        android:gravity="center">

        <TableRow
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <Button
                android:id="@+id/btn_event"
                android:layout_width="139dp"
                android:background="@color/transparent"
                android:drawableTop="@drawable/map"
                android:text="New Event"
                android:textColor="@color/black" />

            <Button
                android:id="@+id/btn_list"
                android:layout_width="139dp"
                android:background="@color/transparent"
                android:drawableTop="@drawable/list"
                android:text="Current Events"
                android:textColor="@color/black" />

            <Button
                android:id="@+id/btn_news"
                android:layout_width="138dp"
                android:background="@color/transparent"
                android:drawableTop="@drawable/news"
                android:text="News"
                android:textColor="@color/black" />
        </TableRow>
    </TableLayout>
</LinearLayout>
```

- Disaster Event Reporting Interface

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/tv_field1"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

    <TextView
        android:id="@+id/tv_field2"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

    <TextView
        android:id="@+id/tv_field3"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

    <TextView
        android:id="@+id/tv_field4"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

    <TextView
        android:id="@+id/tv_field5"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

    <TextView
        android:id="@+id/tv_field6"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

    <TextView
        android:id="@+id/tv_field7"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />
```

```
<TextView
    android:id="@+id/tv_field8"
    android:layout_width="120dp"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center_vertical"
    android:ellipsize="end" />

<TextView
    android:id="@+id/tv_field9"
    android:layout_width="120dp"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center_vertical"
    android:ellipsize="end" />

<TextView
    android:id="@+id/tv_field10"
    android:layout_width="120dp"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center_vertical"
    android:ellipsize="end" />

<TextView
    android:id="@+id/tv_field11"
    android:layout_width="120dp"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center_vertical"
    android:ellipsize="end" />

<TextView
    android:id="@+id/tv_field12"
    android:layout_width="120dp"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center_vertical"
    android:ellipsize="end" />

<TextView
    android:id="@+id/tv_field13"
    android:layout_width="120dp"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center_vertical"
    android:ellipsize="end" />

<TextView
    android:id="@+id/tv_field14"
    android:layout_width="120dp"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center_vertical"
    android:ellipsize="end" />
```

```xml
    <TextView
        android:id="@+id/tv_field15"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

    <TextView
        android:id="@+id/tv_field16"
        android:layout_width="120dp"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        android:ellipsize="end" />

</LinearLayout>
```

- View Disaster Events

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/lay_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <com.google.android.material.appbar.AppBarLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:contentScrim="?attr/colorPrimary"
                android:background="@color/indigo_500"
                app:layout_scrollFlags="scroll|enterAlways|enterAlwaysCollapsed">


                <androidx.appcompat.widget.Toolbar
                    android:layout_width="match_parent"
                    android:layout_height="?attr/actionBarSize">

                    <ImageButton
                        android:id="@+id/btn_back"
                        android:layout_width="36dp"
                        android:layout_height="36dp"
                        android:layout_gravity="left"
                        android:background="@color/transparent"
                        android:padding="8dp"
                        android:scaleType="fitXY"
                        app:srcCompat="@drawable/app_back" />

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="left"
                        android:layout_marginLeft="24dp"
                        android:text="List"
                        android:textColor="@color/white"
                        android:textSize="18dp" />
```

```
                </androidx.appcompat.widget.Toolbar>
            </com.google.android.material.appbar.AppBarLayout>
        </com.google.android.material.appbar.AppBarLayout>
        <RelativeLayout
            xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:app="http://schemas.android.com/apk/res-auto"
            xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context=".ListViewActivity">
            <ListView
                android:id="@+id/lv_1_Id"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:listSelector="@drawable/list_liem"/>

        </RelativeLayout>
    </LinearLayout>
</androidx.drawerlayout.widget.DrawerLayout>
```