Thesis Dissertation

e112 EMERGENCY APP FOR NATURAL DISASTERS

Katerina Ioannidou

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

May2024

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

e112 EMERGENCY APP FOR NATURAL DISASTERS

Katerina Ioannidou

Supervisor

Dr. Marios D.Dikaiakos

Thesis submitted in partial fulfillment of the requirements for the award of degree of bachelor's in computer science at University of Cyprus

Acknowledgments

I would like to thank Professor Marios Dikaiakos for his support and guidance in completing my project. It was a great learning experience.

Abstract

In the contemporary world, the traditional methods during emergencies are becoming outdated and facing numerous challenges and limitations. In this document, we present the e112 which is an innovative emergency app designed to enhance more effective communication. The application allows users to ask for help by activating the SOS button and selecting their category which is then sent to the admins. The operators receive these requests and can communicate with the person in need via private chat. Furthermore, users can report an incident by giving related information. The system dynamically groups them by category and location, maximizing the administrator's ability to efficiently handle them and communicating the basic guidelines, initiating group chats, and providing evacuation plans. Another important feature is to provide alerts and latest updates based on the user's location. Finally, another feature is the forum where they can exchange posts, gather volunteers, and do anything that can be useful in such situations. The forum and group chats are also extended to admins, who analyze user discussions to gain a deeper awareness of the event. The development section of the system includes a full architecture consisting of the mobile app, the admin panel, and several middleware. The app is developed in Flutter and is available on both Android and iOS. On the other hand, the admin dashboard is built using standard web technologies such as HTML, CSS, and JavaScript, where admins can interact with the system through the API acting as a mediator. The backend server is built using Express.js and Node.js and is hosted on Vercel. The middleware layer includes several important components that are useful in the app such as GoogleMaps API for location, Twilio API for number verification, and Firebase API for push notification. The evaluation and conclusion section mark out the tactic we used to validate the effectiveness of the application and discussed the results of the users.

Contents

Chapter 1

Introduction

1.1 Background

1.1.1 Importance of Emergency Management System

Natural disasters constitute serious threats to humanity and must not be neglected. The predictions for the twenty-first century show an increased number of disasters due to factors such as climate change and urbanization [1][2] based on recent trends. To mitigate these impacts, proper and effective emergency management plays an important role. The ability to respond immediately and accurately to emergencies can mainly save human lives and land areas. Establishing an emergency management system can ensure that communities are better prepared to handle disasters effectively and protect public safety. Some critical elements of effective emergency management are customized notifications, effective communication, quick response, evacuation plans, and basic instructions which we can achieve through the technology achievements. Technology, especially mobile applications, has a key role in increasing the efficiency of emergency management. With the advent of smartphones, we have access to real-time information, communication channels, real-time location, interactive maps with the feature of step-by-step evacuation routes, and much more. Modern technology amplifies the efficiency and

effectiveness of emergency management by equipping communities with all the required tools to remain safe during critical circumstances.

1.2 Problem Statement

According to statistics, the fatalities and injuries caused by natural disasters increased rapidly to 11,494 and 6,354,195 respectively between 1980 and 2015 compared to 1950 and 1979 [3]. Nevertheless, despite the technological advances that we have been experiencing these days we are still lacking proper and effective methods to react to natural disasters leading us to delayed responses [4]. It is evident that in situations like these, communication plays a vital role. A major concern is the lack of real-time and reliable information and updates as inaccurate notifications can lead to negative consequences since they can cause anxiety and fear to the affected people [5]. A great example occurred in Hawaii in 2018 when the country raised an alarm over false information on Twitter, causing the receivers to be stressed for up to 7 days after it was released [5]. Consequently, false alarms are a crucial issue that should be treated urgently [5]. Except for this, another dominant problem that we are currently facing is the inability of rescuers to assess the prevailing situation immediately. Without a doubt, rescuers require quick and efficient access to information to be able to assess the incident and provide the most effective assistance to the affected population [6]. Therefore, to achieve this goal, the information is transferred from organization to organization until it reaches the right destination. However, failures and mistakes during this process can cause catastrophic consequences and incompetence in dealing with the disaster, resulting in loss of life and property. Finally, it is worth mentioning that the rescuers may face disruptions due to the overload of the network from the increased communication efforts, triggering interruption in the network. As a result, the rescuers cannot exchange information and the people, on the other hand, cannot contact them to ask for help or report an incident [6].

1.3 Motivation and Mission Statement

The motivation for developing the e112 emergency app for natural disasters comes from the need to solve all the problems that I mentioned above to enhance disaster response and, at the same time, take advantage of advanced technologies such as smartphones. Natural disaster situations require timely and fast management since every minute counts for saving either human life or part of the affected land area. This app addresses the above-mentioned problems by delivering targeted information that the end-users need based on their location without making them feel confused and anxious. The e112 will not only benefit the end users but also the administrators by providing effective ways to handle all available information that is coming during a critical time, easily and quickly.

The mission of this app is to enhance preparedness before, during, and after critical times, reducing stress and information overload.Empowerment of a strong community is achieved through forums and public live group chats where they help each other and provide reliable and accurate information that comes up with essential services such as customized evacuation routes, basic safety guidelines, and live tracking of user's requests.

1.4 Use Case Scenario

Every year, one of the biggest festivals, well known as the Fringe Festival, in Edinburgh takes place, and a quarter of a million people attend this event from all over the world [7].

As a hypothetical scenario, let’s imagine that during this festival, a man in one of the food stalls forgets the kitchen towel above the pot, but he doesn't realise it and goes outside to enjoy the wonderful festival until the food is ready. Unfortunately, a fire started in the towel, and the prevailing winds on that day were not in their favor. As a result, the fire started spreading like lightning in different directions, causing chaos. Similarly, panic spread among the crowd very quickly, with people trying to contact the emergency services by phone, which wasn’t effective because everyone was doing the same thing, leading to a landline overload, and as a result, most of them were unable to reach the emergency services. On the other hand, the ones who managed to communicate couldn’t give the exact location of the incident due to their

nervousness or because they were in this region for the first time, causing responders to be clueless about the exact location and severity of the incident. This could have caused serious consequences because the rescuers didn't know where the incident was taking place or all the affected coordinates. As if this were not enough, there was no effective way to inform the affected users and share with them an exit route that would navigate them to a safe place. Even though this is a hypothetical scenario, it is not as far from reality as we think. Mistakes happen due to negligence or unintentional, and such situations may have a serious impact, including fatalities and the destruction of hectares of land unjustly, while at the same time causing a serious impact on climate change [8].

1.5 Objectives

The primary goals of this project are:

- A User-friendly interface without a technological background.
- Easy navigation through the app, so that the users will found asap the information looking for
- Provide accurate, up-to-date information with all necessary notifications.
- Smooth user experience without delays in response time.
- Responsive User Interface across different sizes for a better experience.

Chapter 2

Related Work

2.1 Existing Emergency Systems

Currently, the only viable solution in these cases is the 112, which relies entirely on verbal communication, where individuals are required to call an operator to report an emergency and will be sent some instructions during a disaster. Despite the fact that calling is a **direct** means of quick communication, it has its limitations when it comes to an actual emergency. High overflow, causing network disruption, and language barriers are some of the constraints of the existing solution.

Before the introduction of smartphones, people relied mostly on landlines or public telephones to seek assistance, report an incident, or even stay updated with the latest alerts in their region. This process was time-consuming and less effective. With the advent of smartphones being an opportunity, we can take advantage of getting more accurate information in less time via an application, as well as communicate directly with authorities, leading to huge progress over the old phone call methods. Unlike the existing solution, e112 offers written communication through text messaging with the opportunity for multimedia use. Younger generations find messaging more attractive and convenient since it is already embedded in their lifestyle, making them feel more comfortable reporting an incident or engaging in group communications. Moreover, the current norm for youngsters is the use of smartphones for sharing information and visual elements, making this app more appealing to them.

2.2 User Experience in Emergency Systems

One of the essential components that we need to keep in mind when we are developing a system is the user experience (UX). The UX is closely connected with users' experiences when they interact with the system. The end goal is to provide positive satisfaction [9].

2.2.1 Importance of User Experience

The user experience (UX) is of vital importance when we are dealing with emergency applications. Under such circumstances, the primary goal is to make sure that the users will have access to all the necessary information without effort. A user-friendly interface can minimize navigation time and improve the effectiveness of the system. Poor UX can cause stress and confusion. For example, an abstruse interface will lead users to face difficulties during the access of the necessary information and might not be able to effectively report a situation. For that reason, providing a seamless and easily accessible UX will enhance the user's experience and facilitate smooth access to crucial information.

2.2.2 Best Practices in User Experience Design

Following the best UX practices will ensure the attainability and simplicity of the system. Some of them are [11]:

1) Become Your Users:

- Listen to the user's thoughts, needs, and requirements.
- Creating personas for different users will be useful for making more accurate decisions.
- Give them a real-life scenario and watch how they will interact when using the system, through their facial expressions.
- Try to put yourself in their situation and understand their feelings and thoughts.

2) Focused on the design first:

- Adopt inspiration from other successful system designs.

3) Trust no one:

- Through continuous testing, we will achieve improvements in the app's functionality.

By using these best practices, developers can develop user-friendly and effective interfaces that meet the user's needs.

Chapter 3

Requirement Analysis

3.1 Target Users

3.1.1 Overview of Target Users

The whole system is designed to serve the primary group of users: the end users and the administrators- operators who manage the system. The end users are people of the general public who will use the application to receive alerts, reports, and access to all necessary information. However, operators are responsible for handling the app's functionality through the admin panel, to ensure that the information provided through the app is updated, accurate, and managed correctly.

3.1.2 End Users Needs

The end users require a user-friendly interface that can easily navigate through the features. An app that empowers them with customized alerts and the latest updates based on their locations without causing them stress for no reason. Furthermore, the capability to report an incident or ask for help easily, and quickly without facing any kinds of difficulties. They need the adroitness to access basic safety guidelines and evacuation safe routes in case of need. Also, the end users need to be a part of a community with their neighbors so that during critical times to not feel alone and have support. The users also expect the app to offer robust, easy and less timeconsuming user management features including sign-in, sign up, phone number verification, and customizable settings and preferences such as language.

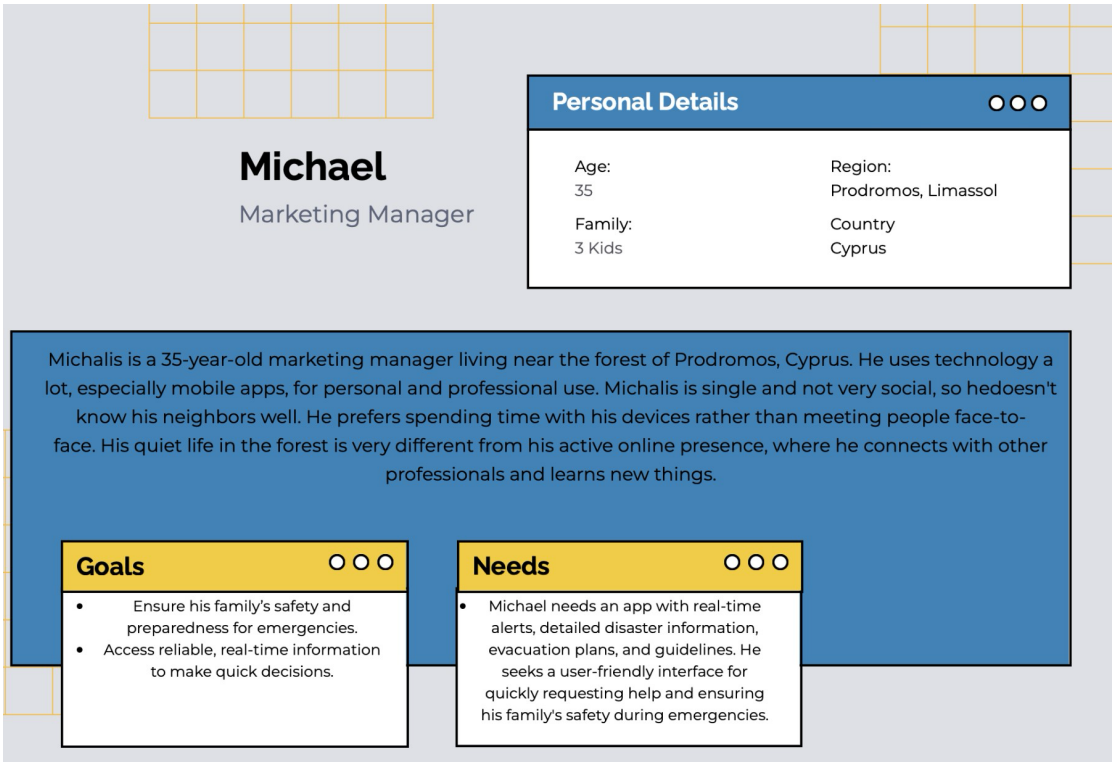
3.1.3 Operators Needs

An understanding of the operator's needs has a vital role in the app's effectiveness. Operators need a friendly interface that provides up-to-date information on disaster events allowing them for quick actions and accurate decision-making. Furthermore, any type of communication and a way to spread the information is essential. The primary need is to understand the current state of the situation so that it can handle it efficiently by providing all the necessary alerts, guidelines, and evacuation routes for all the affected users. By addressing the above mentioned we can achieve improvement in response time during natural disasters.

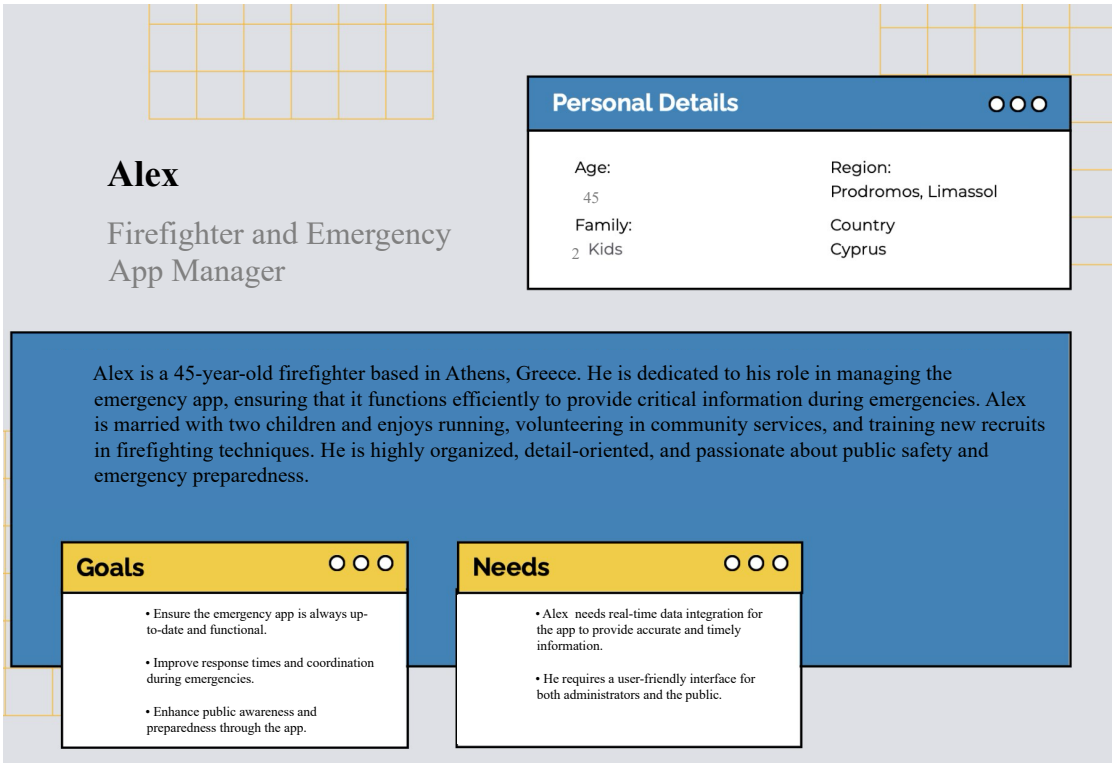
3.2 Persona and Use Case

3.2.1 User Personas

3.2.1.2 End User Persona



3.2.1.2 Administrator Persona



3.2.2 Use Case Scenario

3.2.2.1 End User Use Case Scenario

Preconditions:

- The app is installed on Michael's mobile phone.
- Michael has created an account using his phone number
- Allowed the app to use his location

One day, Michael notices from his house's window a fire in the forest. He quickly opens the app to report it. He enters the details about the fire, including its location and some notes that will be useful, and submits the report. When the operators processed his request, then he took his family to a safe place based on an evacuation plan through the app.

3.2.2.1 Administrator Case Scenario

Preconditions:

- Logged in as Admin

One day, Alex receives a forest fire alert, verifies it, sends detailed emergency basic guidelines, creates a group chat, and adds in the system all the affected and safe coordinators for the app to display to all the affected users the safe exit route.

Chapter 4

Functionality

4.1 Customer Journeys

One of the crucial parts that we had kept in mind was the importance of customer experience and therefore we focused on the customer journey map (CJM) to achieve a customer-focused solution that would effectively manage the natural disaster [12]. The customer journey as you can see in Figure 1, includes four phases. Initially, the potential user searches for existing emergency apps that will offer them full safety guidance during critical times and keep them updated. The next phase is the registration phase, where the users expressed a need for a straightforward sign-up process that allowed them to set up their profiles in less time. Then, in the sharing phase, users share the app with others, intending to push around the safety. Furthermore, at the beginning of the journey, we can see the main goals and needs, including staying informed, ensuring safety, and maintaining communication, while common struggles revolve around a lack of information and unclear evacuation routes. Finally, at the end of the customer journey appears the outcomes including what the user can do now, what they are avoiding and what changed in their environment.

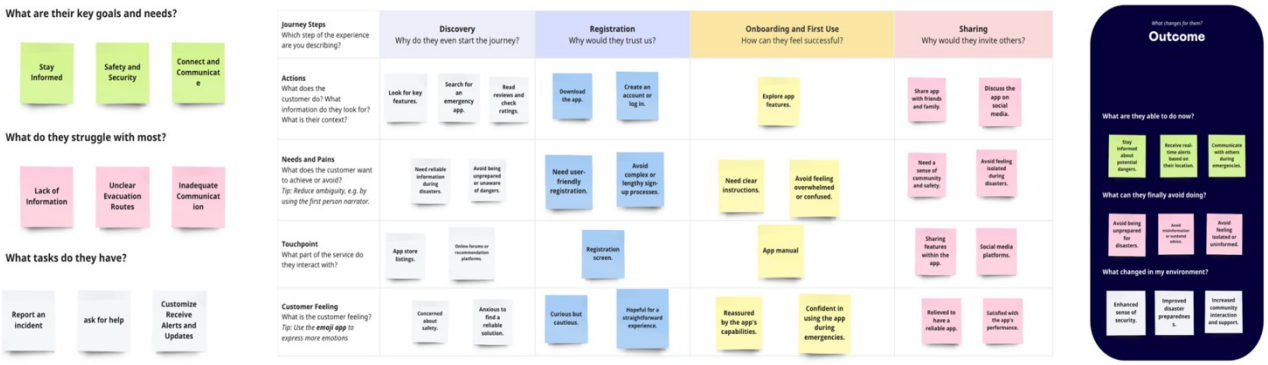


Figure 1 Customer Journey

4.2 Interaction Diagrams

During the software design, one of the crucial components is the interaction diagrams which are a graphical representation of all parts of the systems interacting with each other to achieve functionality [13].

4.2.1 Report an Incident

The interaction diagram in Figure 2 illustrates the user flow, for one of the basic functionalities of the app, reporting an incident. The process begins when the user first opens the app, navigates to the report incident screen, and press the "Report" button to initiate the process. After pressing the button, a dialogue appears, displaying all the incident categories and letting the users choose the most suitable for their case. Upon selecting a category, will automatically navigate into the report incident form which can provide additional material to help admins have a clearer picture of the situation. After submitting the report, the app displays to the user a message with the confirmation code, enhancing the user's confidence and minimizing the fear if they get the request.

After the report submission, the app automatically navigates the user to the Help Request Log, where they can view the progress of their request. In that section the user can see any responses provided by the administrator, and can join available group chats related to their request, fostering community support and

communication. Beyond of that, the user also can read the basic guidelines and instructions provided by the admins for safety.

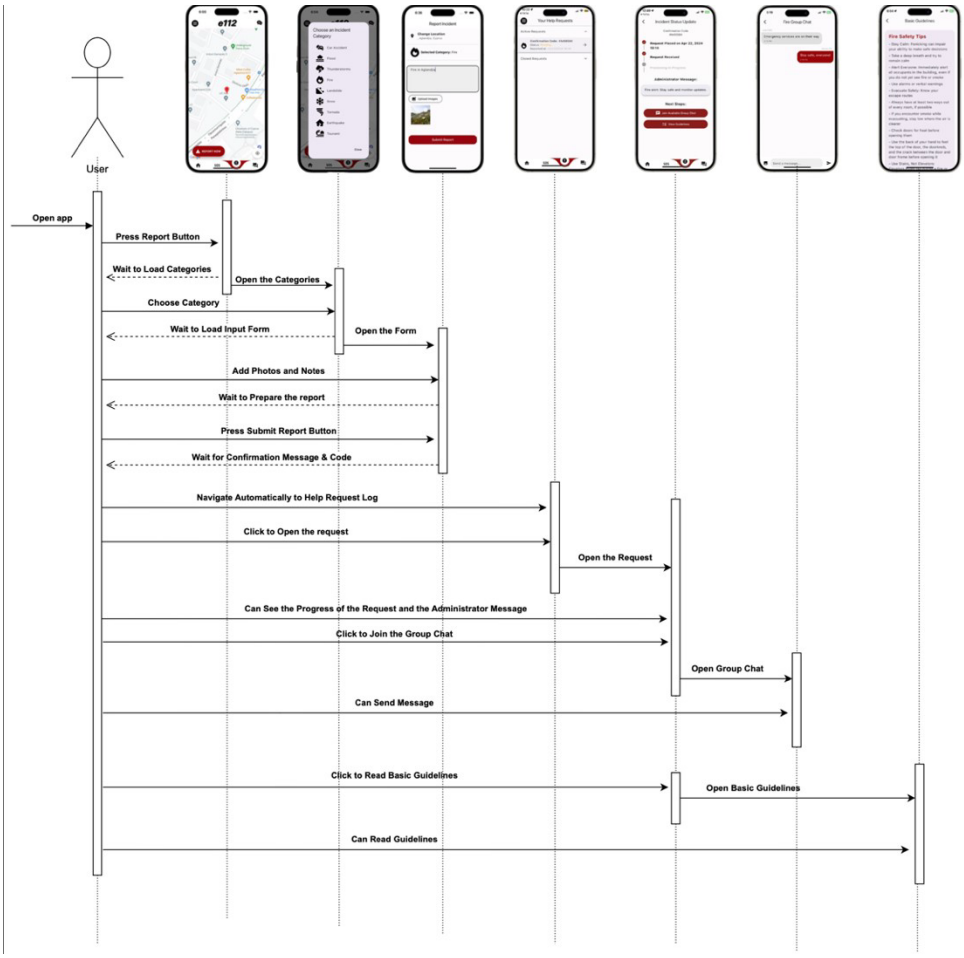


Figure 2 Report an Incident

4.2.2 Ask for Help

The interaction diagram displayed in Figure 3 focuses on the help request feature within the e112 emergency app. The process begins with the user opening the app navigating to the help request screen and pressing the SOS button. Upon pressing the SOS button, a dialogue with the available categories appears and lets the user choose one of them. After the user selects a category, the app processes the request and navigates the user to the Help Requests Logs screen, where the users can view the progress of their request and any messages from the administrator. Also, the user can join an available private group chat with the admin.

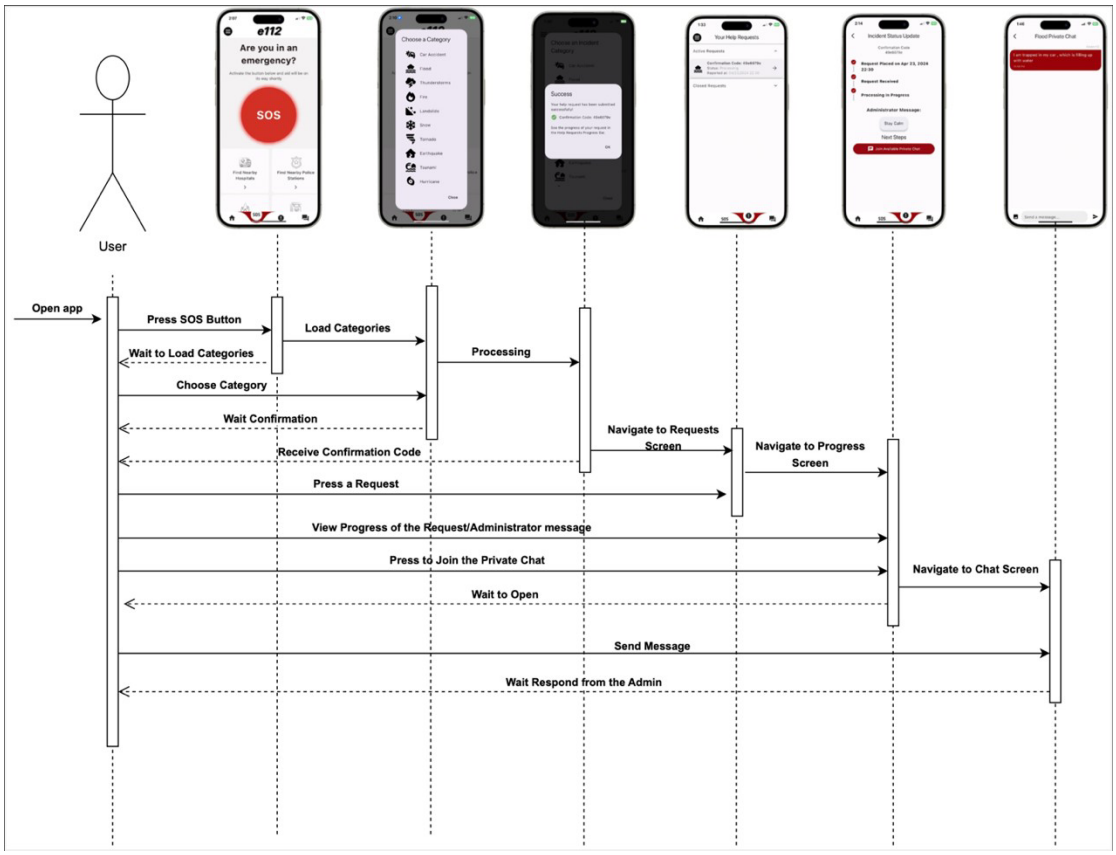


Figure 3 Ask for Help

4.2.3 Alerts and Latest Updates

The interaction diagram displayed in Figure 4, illustrates the alerts and the latest updates functionality. The user first opens the app, which navigates automatically to the home page and includes alerts and the latest updates. The user can either open the latest update and read the contents or click the alert which inside may include basic guidelines, group chats and an evacuation map. If the user presses the group chat, it opens the chat which is connected to all users in the affected area and can communicate with each other. If the user presses the basic guidelines will display basic guidelines for that specific incident of the alert, and if they press the evacuation map, this will open a map. The map displays all the safe points in green and all affected points in red. If the user presses the “find exit route” button, the nearest safe route will be displayed.

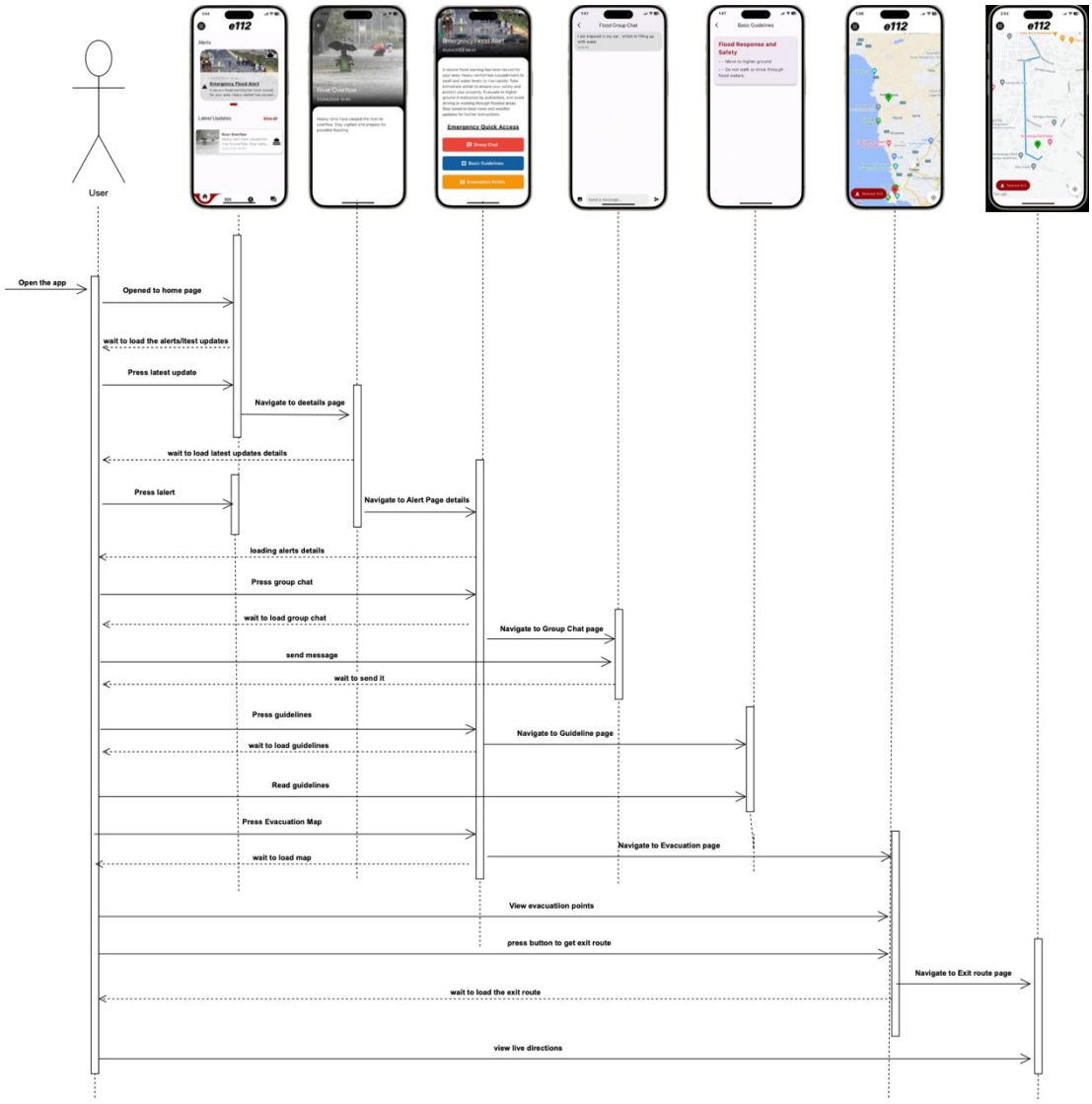


Figure 4 Alerts and Latest Updates

Chapter 5

User Interface

5.1 Overview User Interface

One of the significant components of the emergency app is the User interface (UI), and therefore we dedicated a lot of time to the design to provide a smooth userfriendly experience during critical times.

5.2 Customer Empathy Map

The main purpose of the customer empathy map is to help us gain a deeper understanding of the user's experience and needs during an emergency. Specifically, figure 5 displays the customer empathy map of a young woman's experience and needs.

In the map, we notice that the young woman in the Figure 5, measures quick help, reliable and accurate information, and community support when we are dealing with critical times. Her pains include slow response times, outdated information, and complex navigation interfaces. Furthermore, she recognizes services that

provide mainly support and keep her updated with the newest updates without any delay. As we can see from the figure, she is currently relying on communication channels such as news, social media, and friends and family, however, she is not fully satisfied with that because she is facing delays in emergency responses. This led her to look for alternative ways to get help.

Her gains are centered around saving time, feeling supported, and connecting with users in her area.

The customer empathy map helps us align our services to meet her requirements, with the aim of a positive user experience and satisfaction.

Figure 5: Customer Empathy Map

5.3 Flowchart

The design of a flowchart was pivotal during the creation of the mobile app as it represents a visual representation of the structure and functionality of the app. Also, it provides the developers, end users, and designers with a straightforward clean roadmap ensuring that everyone is on the same page.

The flowchart in Figure 6 illustrates the user journey for the e112 mobile app. Initially, the process begins when the user opens the app, either logging in if they have an account or signing up if they don't. In case the user forgets the password, they can request a new one, which can be reset by adding their phone number. There are two paths the user can follow, one to use the app with an account where allows them to access all the features the app provides, and the second one to skip the authentication process and have limited access to the features
Specifically, will have access only for viewing the alerts, latest updates, basic guidelines, and safe exit routes.

When the users are logged in the app will automatically navigate them to the home screen, where they can access alerts, latest updates, report incidents, ask for help, and join the forum. Reporting incidents involves several steps, including adding images, categories, notes, and location details, followed by confirmation, and tracking of the incident's progress which also includes group chat, basic guidelines for the specific incident, and exit routes. The help request feature is approximately the same process with the difference that the progress bar is also provided just a private chat, where the user can have live communication with the admins.
The flowchart ensures that all features of the app are easily accessible from all end users providing a seamless user experience.

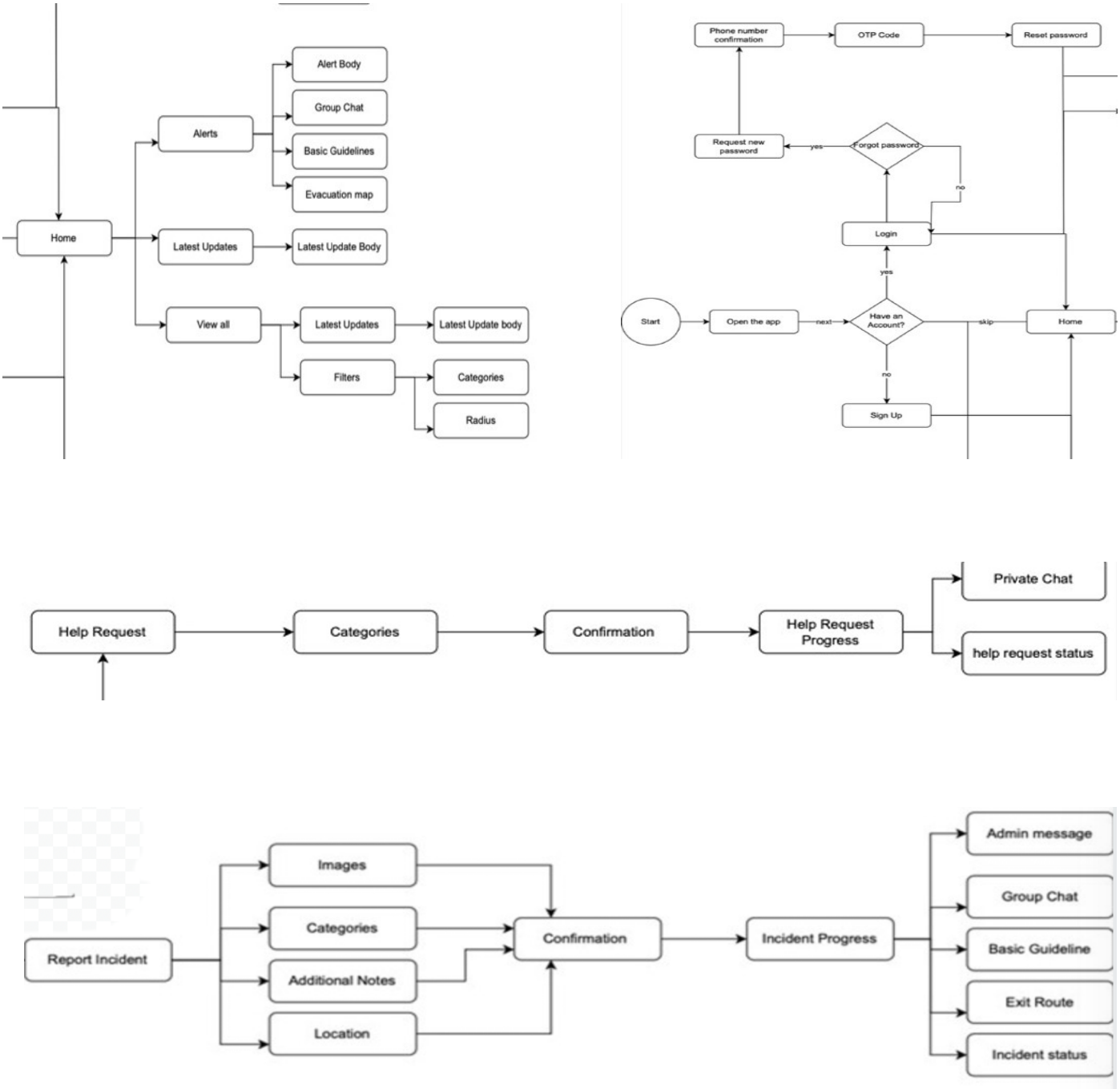


Figure 6: Flowchart

5.4 UX Screenshots of the App

In this section, the user interface of the mobile app is displayed. This offers several features including viewing customized alerts and the latest updates, joining a group and private chats, reading basic guidelines, viewing the evacuation map with exit route if needed, interacting in the forum, report an incident by providing additional material if you wish and watch in real time the progress of your request and finally ask for help with live tracking of the progress of your request.

5.4.1 Authentication

Figure 7 Sign Up

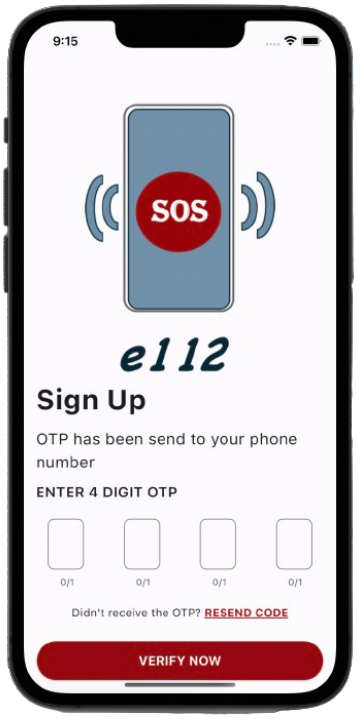


Figure 8 Verification

Figure 9 Sign In

5.4.2 Home Page

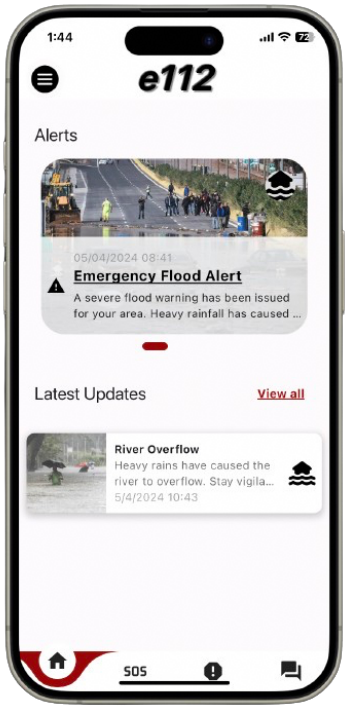


Figure 10 Home Page

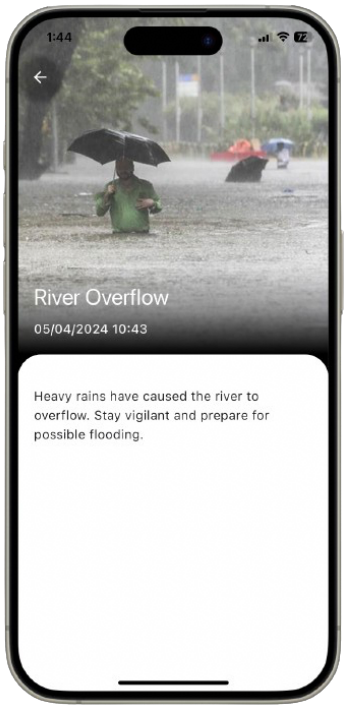


Figure 11 Latest Update Details

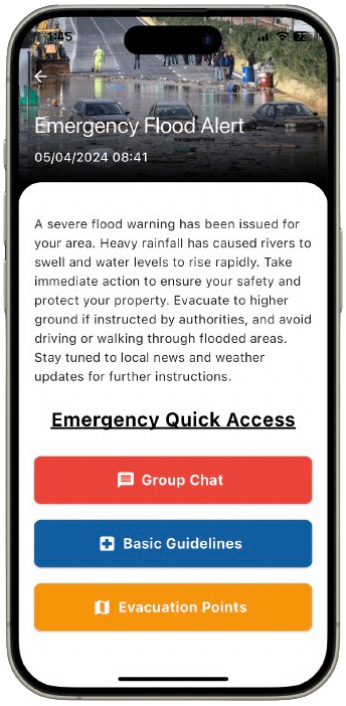


Figure 12 Alert Details

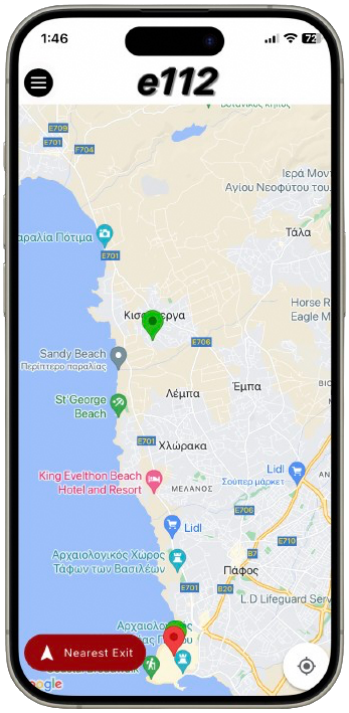


Figure 13 Evacuation Map

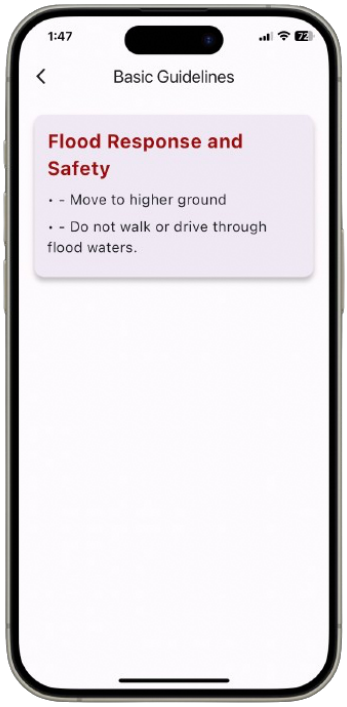


Figure 14 Basic Guidelines

Figure 15 Group Chat

5.4.3 Report an Incident

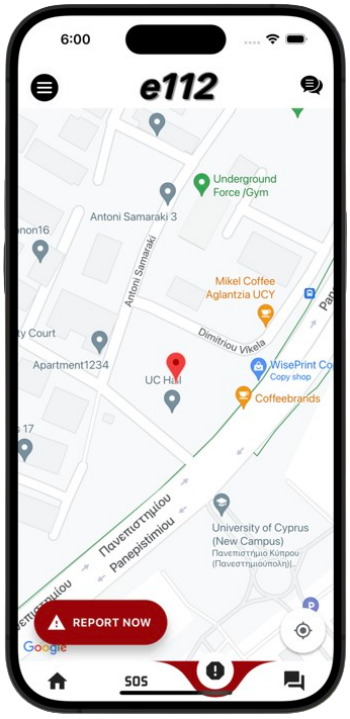


Figure 16: Report Incident Page

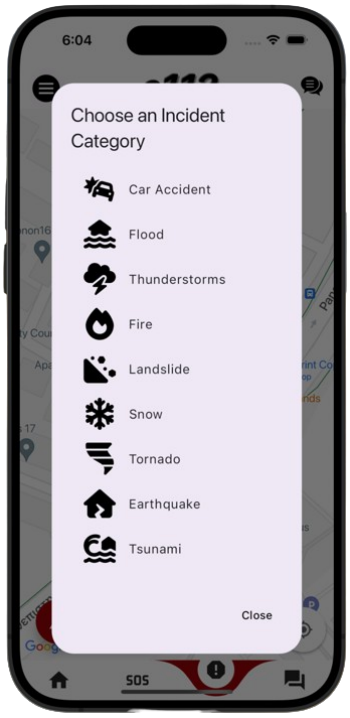


Figure 17: Incident Categories Dialog

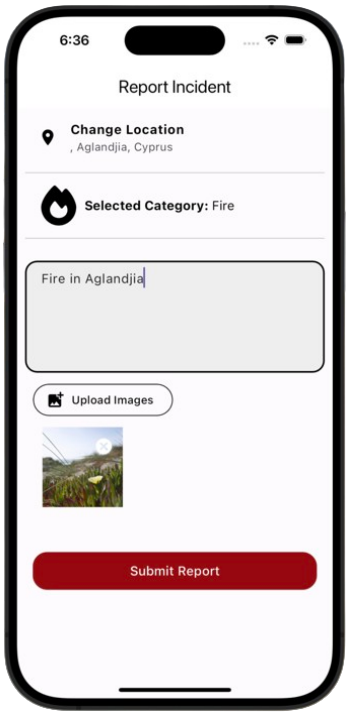


Figure 18: Incident Form

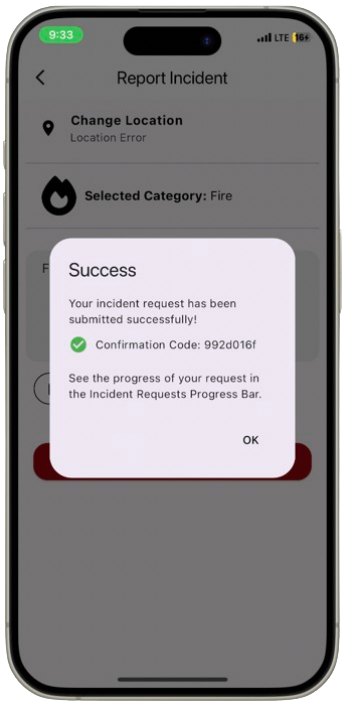


Figure 19 Confirmation

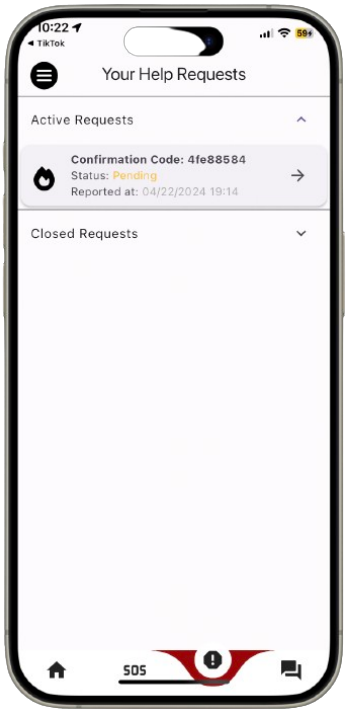


Figure 20 User's Help Requests

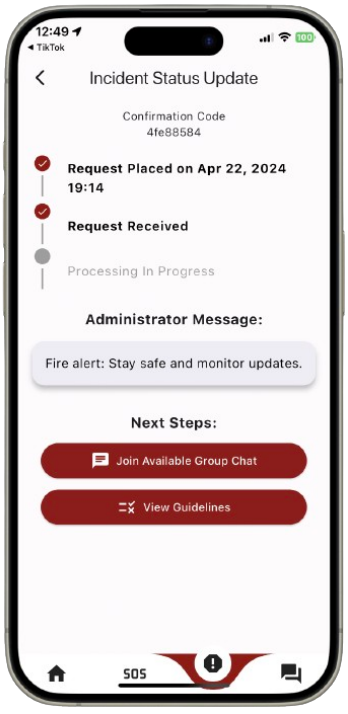


Figure 21 Incident Status Page

Figure 22 Group Chat Page

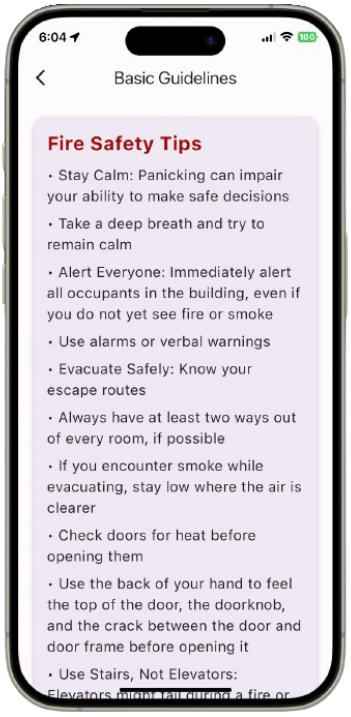


Figure 23 Basic Guidelines

5.4.4 Help Request

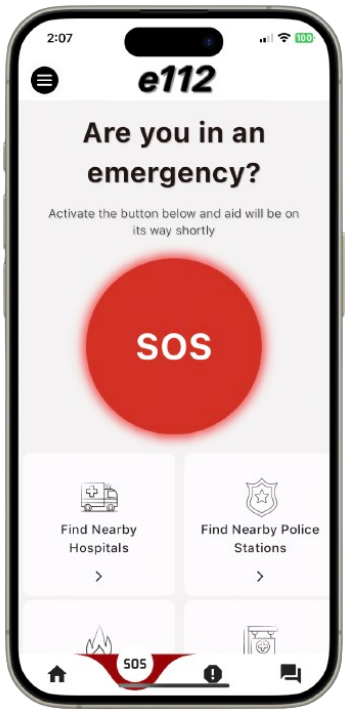


Figure 24 Help Request Page

Figure 25 Help Request Categories Dialog

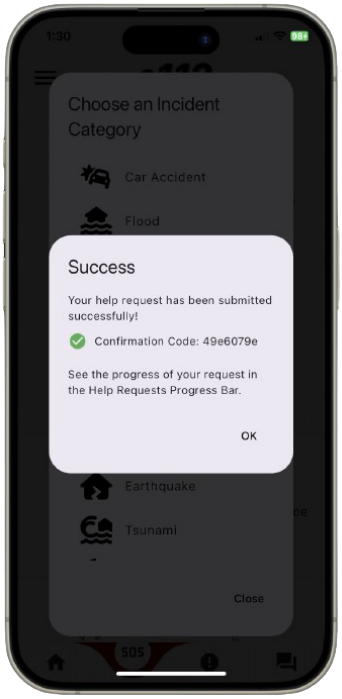


Figure 26 Confirmation Message

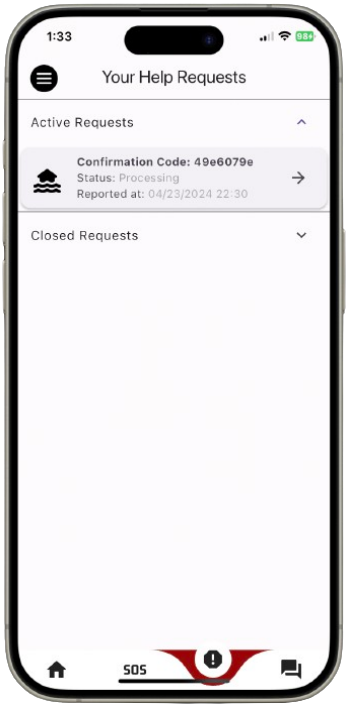


Figure 27 User's Help Requests

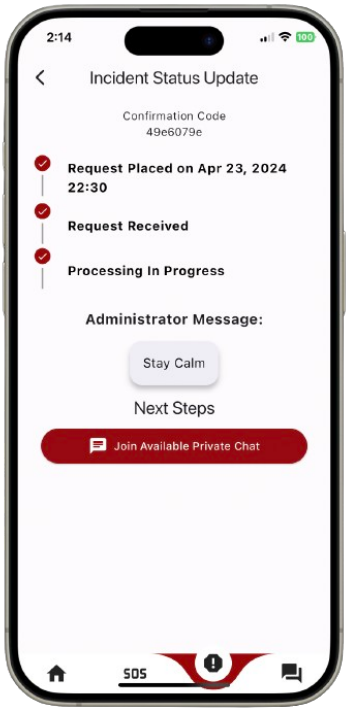


Figure 28 Help Request Update

Figure 29 Private Group Chat

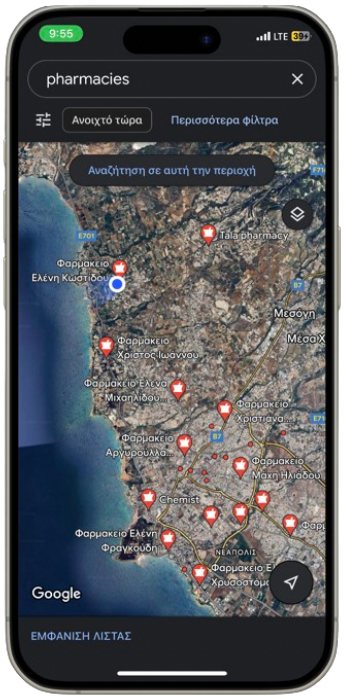


Figure 30 Nearest Hospitals

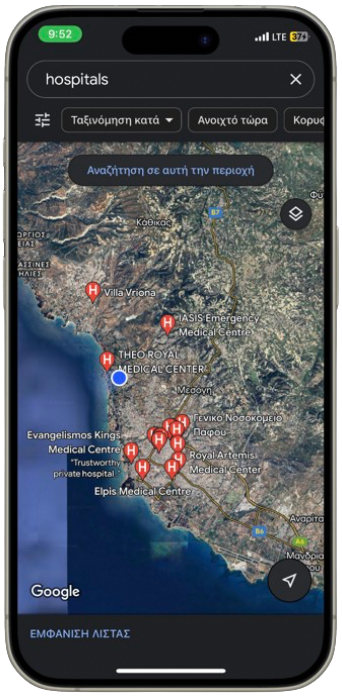


Figure 31 Nearest Police Stations

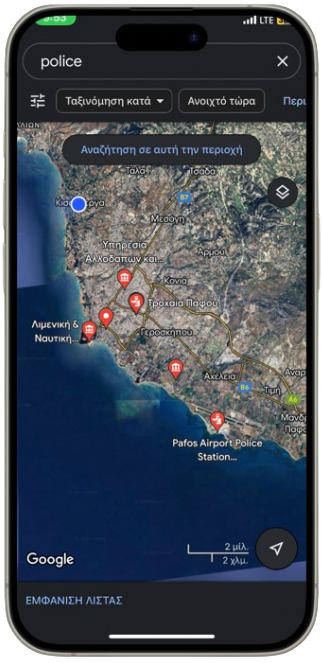


Figure 32 Nearest Pharmacies

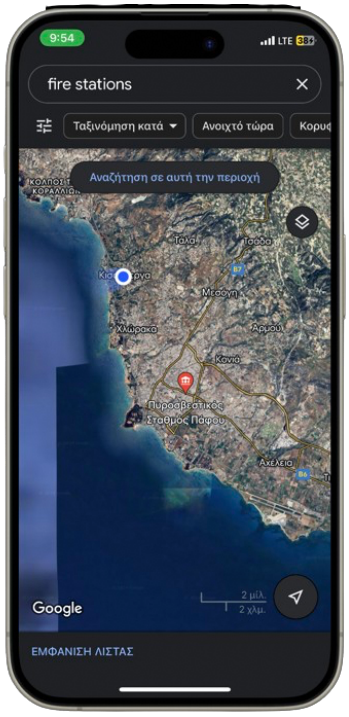


Figure 33 Fire Stations

5.4.5 Forum

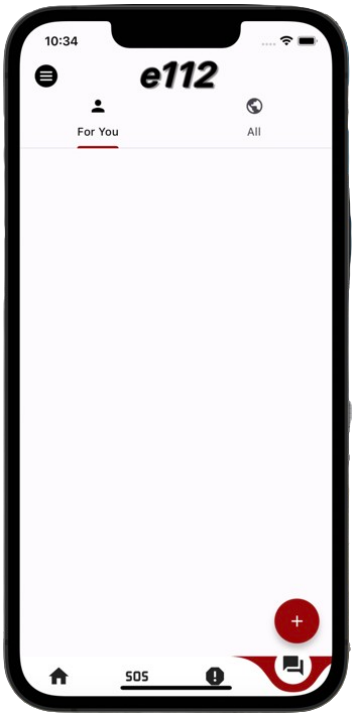


Figure 34 Forum Page

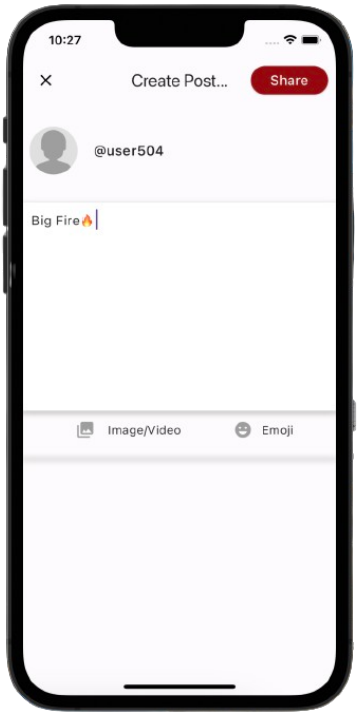


Figure 35: Create Post

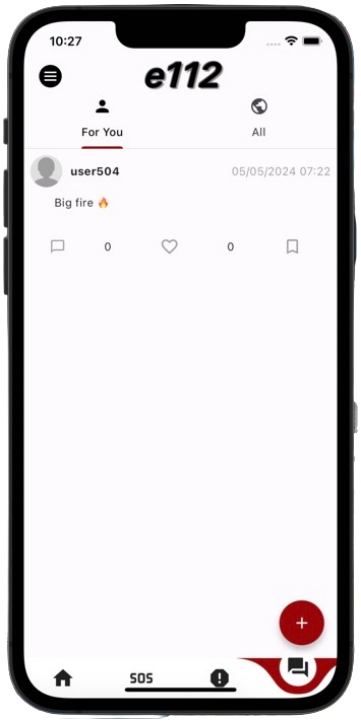


Figure 36: Forum Page

3.4.5 Account & Settings

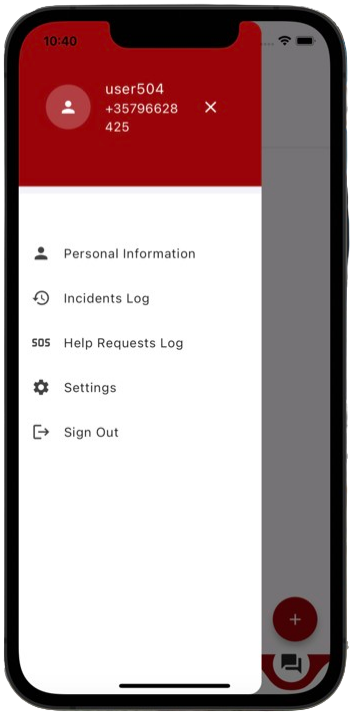


Figure 37 Drawer Menu

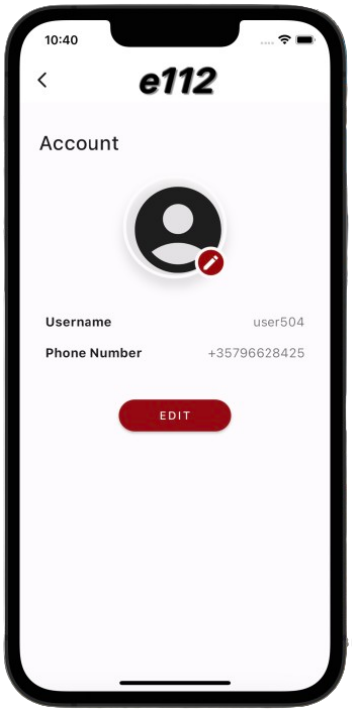


Figure 38 Account Page

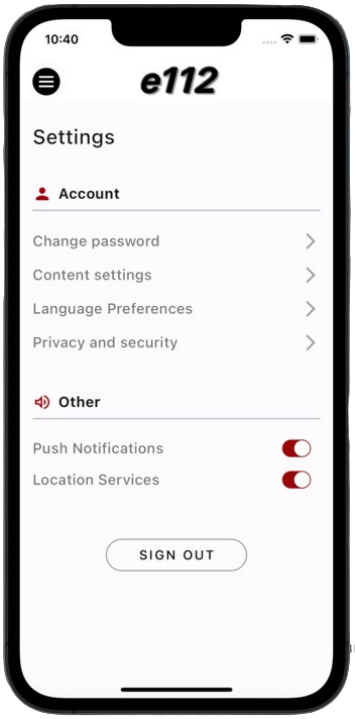


Figure 39 Settings Page

3.5 UI Style Guide

During the development of the UI, the interface style choices are crucial to provide effective use. The application's name is called e112 which aligns with the global emergency number 112 which emphasizes the purpose of the app. The primary color of the app is dark red which symbolizes urgency. All the colors of the app use the dark version of the color to be more restful for the eyes. All the colors are not randomly chosen and are analyzed in Table 1 below.

Color	Purpose	Rationale
Dark Red	Main app color, Group chat button, Urgency	Conveys urgency and aligns with the app's emergency focus

Red	Affected areas on the evacuation map	Highlights danger areas clearly
Dark Green	Safe areas on the evacuation map	Represents safety and guides users to secure zones
Dark Blue	Basic guidelines	Represents reliability and calm for essential safety information
Dark Yellow	Evacuation points	Conveys caution while being easily distinguishable

Table 1 App Colors

To achieve usability all the tappable buttons are large, clear, and legible text to be easily readable for users under stress or in a hurry. The overall design of the app focused on simplicity to ensure that the users would navigate without facing any difficulties.

Chapter 6

Software Design and Impelemtation

6.1 Software Design Overview

The software design phase is one of the most critical steps during the whole process to achieve software that is understandable, modifiable, and reusable. Most developers face a lack of understanding of the problem, facing flaws in the implemented system leading to poor design. The system to be considered as wellstructured requires meeting user's requirements. The software design deals with how the components and modules are organized and decomposed in the software system. Specifically, the software design is the translation of the user's requirements in a more representative way.

6.2 Architectural Design Overview

The architectural design represents a visual representation of the software system components as displayed in Figure 40. Firstly, at the high level, the system is structured into three layers: the hardware at the bottom, the middleware at the middle and the application at the top. The e112 mobile app was developed using Flutter which is open-source and allows

the development of a cross-platform application from a single codebase making it available on both Android and iOS.
On the other hand, the web-based Admin Dashboard is a control panel built using standard web technologies such as HTML, CSS, and JavaScript, where admins can interact with the system through the API acting as a mediator.
The backend server is built using Express.js and Node.js. The server is hosted on Vercel, which offers a serverless platform-as-a-service (PaaS) environment [14] and allows the deployment of applications to deliver high availability, easy scalability, and system reliability. The middleware layer includes several important components that help with the functionality of the system. One of the services we use is the GPS API which we use for live location tracking. The Google API is a crucial service for the e112 app, to know the exact location of the users when needed. Twilio, is a software-as-a-service (SaaS) platform [1] that provides communication services, including SMS. This is especially helpful for the user's phone verification using OTP. Firebase, a platform-as-a-service (PaaS) solution [14], is used to provide push notifications, for achieving real-time alerts and enhancing the user's experience. The system uses AWS Cloud for data storage. This includes Amazon RDS, for storing the system's data and it also utilizes a simple storage service (Amazon S3) where we store data and manage static files such as photos. This multilayered architecture provides scalability, reliability, and efficiency.

6.2.1.1 Hardware layer

In the bottom layer of the architecture is the hardware, which is an important part of the overall functionality of the system. This layer includes the server which in our case runs Express.js on Node.js which is responsible for handling the data processing and the API interactions. Another part of the hardware is the mobile devices, including smartphones running on both iOS and Android.

6.2.1.2 Middleware layer

The purpose of the middleware layer is to accelerate the communication between the hardware and application layers. In this layer, the express framework is deployed into Vercel. Vercel supports the deployment and scaling of our server simplifying the management and providing reliability. The cloud services from AWS are vital in this layer. The Amazon RDS purpose is to store a piece of critical information and data securely, the S3 provides scalable storage which our system uses to store static images. Furthermore, part of these layers are two critical thirdparty services, the Twilio and the Firebase. Firebase is a development platform developed by Google. This platform provides a lot of services such as NoSQL DB, authentication methods, etc., but in our system, we use just the push notification that is provided. Specifically provides Firebase Cloud Messaging (FCM), with the use of handling push notifications. With the FCM we can send notifications directly to the user's phones to alert them of any change in the app status such as tracking the request progress, when a new alert or latest update is released, or when a new group or private chat is opened. This functionality is crucial when we are dealing with emergencies for the users to stay up to date with the latest updates, request status, and messages. When a notification is established and ready to be sent the backend forms the message and sends it to the FCM service via API calls. This message includes the content, and the target such as the individual device tokens, user segments, and topics. Then, firebase processes these incoming messages to deliver them correctly to the corresponding device instantaneously.

Furthermore, we are using Twilio which is a cloud communications platform for phone number verification through the app.

Specifically, our system generates a random 4-digit code and sends it with the device token that needs to be sent. Twilio processes the request and sends it to the corresponding user.

In the background, FCM works by establishing a constant connection with devices through lightweight mobile or web applications that listen for messages from Firebase. When a notification needs to be sent, the backend sends the message to the

FCM Service via API calls. This message includes the content and target specifications such as user device token.

Furthermore, we are using Twilio which is a cloud communications platform for phone number verification through the app.

Specifically, our system generates a random 4-digit code and sends it with the device token that needs to be sent. Twilio processes the request and sends it to the corresponding user.

6.2.1.3 Application layer

The application layer consists of two main components: the mobile app which will be used by the end users and the admin dashboard for administrators. The mobile app, developed in Flutter, uses the Model View ViewModel (MVVM) architectural pattern, for a separation of concerns. On the other hand, the admin dashboard, built using HTML, CSS, and JavaScript, provides a web-based interface for administrators. It allows administrators to manage the functionality of the system.

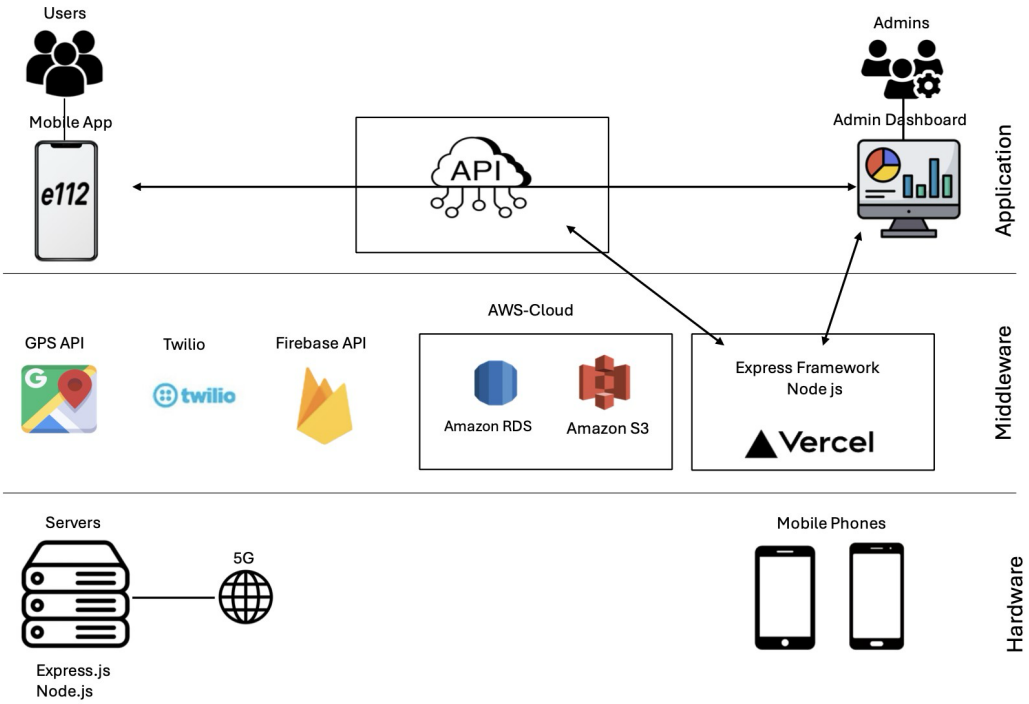


Figure 40: Architecture Diagram

6.3 Design Pattern

6.3.1 Model View ViewModel Design

In the mobile app, we utilize the pattern called Model View View model (MVVM). This pattern includes 3 primary components: the model, the view, and. This architecture has a primary objective to separate the business logic from the presentation logic.

To be more specific, the model represents the business logic and data of the application, the view includes all the UI components, and the ViewModel acts as a liaison between the model and the ViewModel, specifically transforming the data from the model into values that can be displayed in the view.

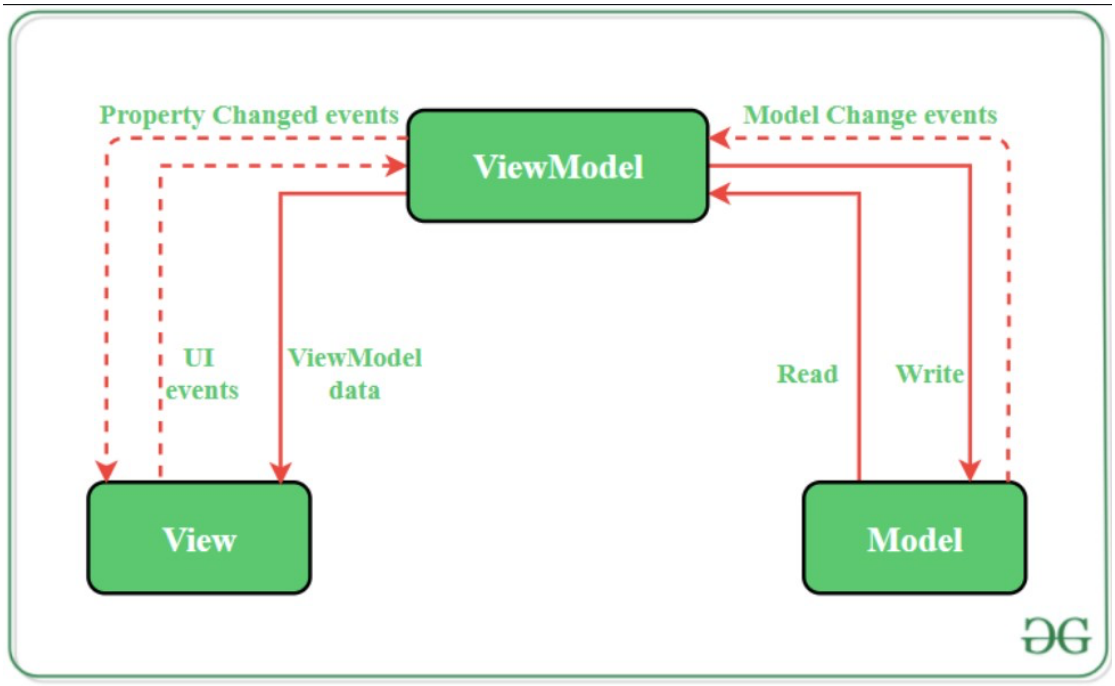


Figure 41: Model View ViewModel (MVVM) Pattern

6.4 Database Design

The primary entities of my system as displayed in Figure 42, are: User, Admin, Alerts, Incident, Help Requests, Group Chat, Basic Guideline, and Safe Exit Route.

User

Attributes:

- user_id: This is the primary key.
- username: representing the user's random generator name
- phone_number: storing the user's phone number.
- verification_status: indicating whether the user has been verified.
- created_at: shows when the user's account was created.

Admin

Attributes:

- admin_id: This is the primary key
- name: representing the admin's name.
- phone_number: storing the admin's phone number.
- role: shows the admin's role within the system.

Alerts

Attributes:

- alert_id: This is the primary key
- title: Representing the title of the alert.
- description: Storing the description of the alert.
- category: The category of the alert.
- main_location_lat: Representing the latitude of the main location affected by the alert.
- main_location_long: Rrepresenting the longitude of the main location affected by the alert.

- affected_location: Locations affected by the alert.
- affected_radius_km: The radius of the affected area in kilometers.
- isImportant: Indicating whether the alert is marked as important.
- created_at: The creation time of the alert.

Incident

Attributes:

- incident_id: This is the primary key
- category: The category of the incident
- latitude: The latitude of the incident location.
- longitude: The longitude of the incident location.
- reported_at: Indicating when the incident was reported.
- region: Specifyingthe region where the incident occurred.
- additional_notes: Additional Notes for the incident
- confirmation_code: A confirmation code of the incident.

Help Requests

Attributes:

- request_id: This is the primary key
- category: The category of the incident
- confirmation_code: A confirmation code of the help request.
- region: The region of the request.
- latitude: The latitude of the help request location.
- longitude: The longitude of the help request location.

Group Chat

Attributes:

- chat_id: This is the primary key
- chat_name: The name of the group chat
- category_icon: The category icon of the group chat.
- created_at: Indicating when the chat group was created.

Basic Guideline

Attributes:

- guideline_id: This is the primary key
- content: The guidelines.
- category: The category of guideline.
- category_icon: The category icon of the guideline

Safe Exit Route

Attributes:

- evacuation_point_id: This is the primary key.
- name: Name of the evacuation point.
- latitude: The latitude of the evacuation point.
- longitude: The longitude of the evacuation point.
- instruction: The instructions for the evacuation route.

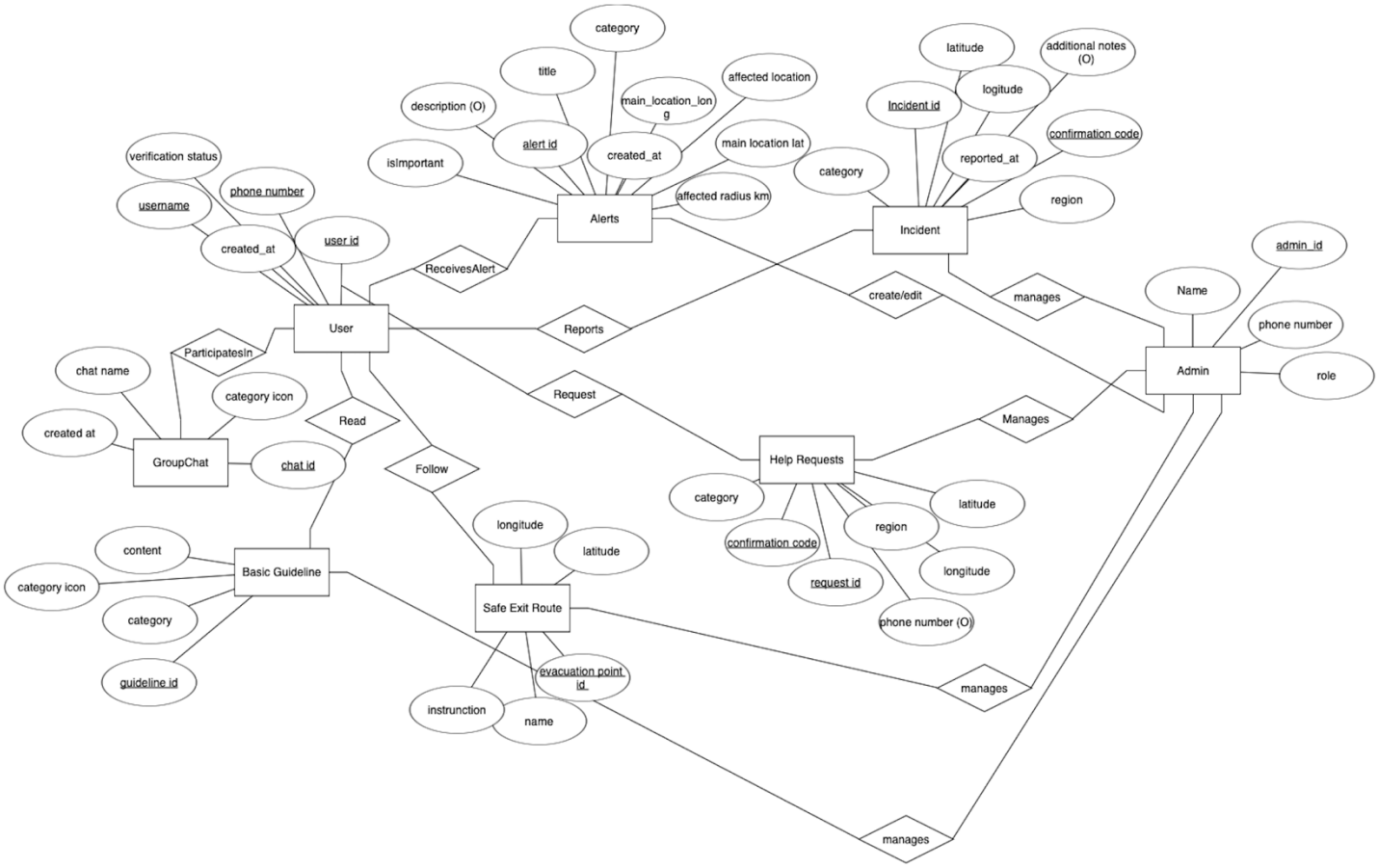


Figure 42: ER Model

Chapter 7

Evaluation

7.1 Evaluation Criteria for Performance Testing

In this chapter, we evaluate the performance of the admin panel which is a web application using key metrics and criteria. For the evaluation, we used a standard performance evaluation tool called Unlighthouse [15]. This tool provides a lot of insights into various aspects of the app's functionality, including page loading speed, accessibility, coding practices, and search engine optimization (SEO). To ensure that our system meets standards of usability, efficiency, and reliability, the following evaluation criteria are used:

7.1.1 Performance

One important aspect that we have to keep in mind is the performance score of our admin panel, as displayed in Figure 40. The performance specifically displays how quickly the webpage will be loaded to the operators for managing the requests. This criterion evaluates several key metrics which relate to the user experience [15]:

First Contentful Paint (FCP): FCP is a crucial metric for measuring a user's perception of load speed, highlighting the point when any content is rendered on the screen after a user navigates to a page [16]. As we can see from the figure, the value is 2.5 seconds which shows that the web application responds promptly.

Largest Contentful Paint (LCP): is a Core Web Vital metric that indicates when the main content of a web page has loaded, encouraging users to find it useful [17]. In our case the LCP is 2.8 seconds, which is good enough, the ideal is 2,5 seconds so can be improved more.

Total Blocking Time (TBT): Total Blocking Time (TBT) is a significant metric in web performance, measuring the time the main thread is blocked after First Contentful Paint (FCP), affecting user input responsiveness. A value of 0 milliseconds indicates which means excellent responsiveness.[18]

Cumulative Layout Shift (CLS): This metric measures non-expected layout shifts during loading [19]. A value of 0.005 shows that the layout is stable.

7.2.3 Best Practices

The best practices score of 96, shown in Figure 43, evaluates compliance with industry standards and coding guidelines. A high score in this category indicates that the application follows best practices, which will ensure security and code maintainability.

7.2.4 SEO

The SEO score of 82, presented in Figure 3, estimates how well the application is optimized for search engines. This score can be improved by using proper meta tags but is not the primary goal of the specific system because it will use it only to authorize people.

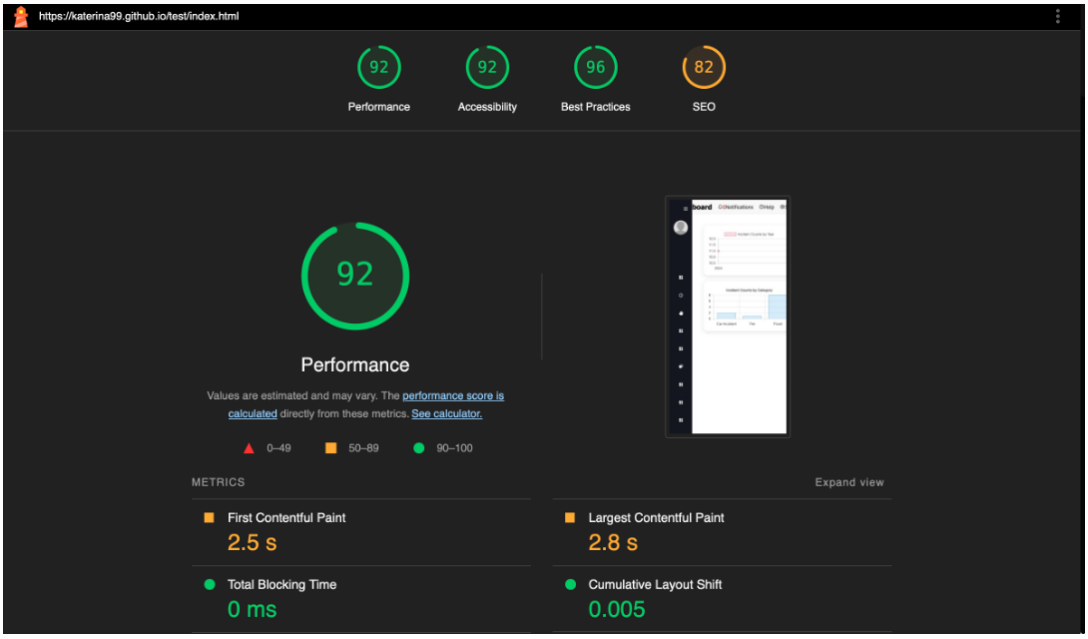


Figure 43: Admin Panel Performance

7.3 End-User Feedback

For the audience of our study, we chose a group of people aged between 16 and 65 from a local surf club. We gave them a hypothetical scenario of a flood happening here in our location. They were tasked to report this incident through the e112 mobile app. The following sections represent the results of users' responses

7.3.1 Questionnaire Design

To measure the user experience and efficiency of the e112 app, we designed a questionnaire that was given to a small group of people. The survey collected insights about the app's navigation and the effectiveness of several features. The questionnaire included targeted questions about different aspects of the performance of the app. The design of the questionnaire included different types of questions such as multiple-choice, and open-ended. The intention for creating this Questionnaire was primarily to gather insights that will help the improvement of the app.

7.3.2 Result Analysis

7.3.2.1 Question 1

The first question of the questionnaire was a demographic type of question that mainly gathered demographics about the users related to their age group. As you can see from the figure below, the majority of people that answered this survey were between 18-34.

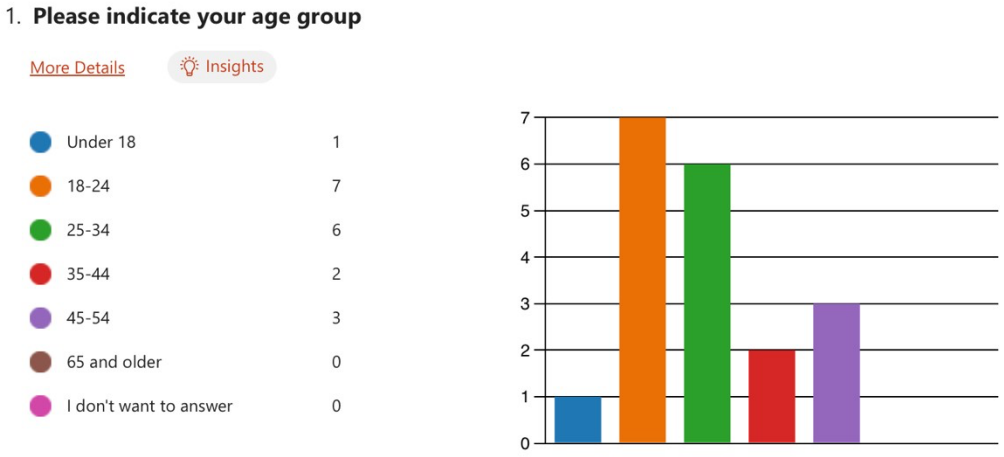


Figure 43: Question 1

7.3.2.2 Question 2

The second question was about the evaluation of the navigation of the app and whether it was easy to find what they were looking for. The results were very good, most responders found the app easy to navigate. Around 63% of the users answered with “Very easy” and 26% found it “Easy”. This positive feedback indicates that the User interface was user-friendly.

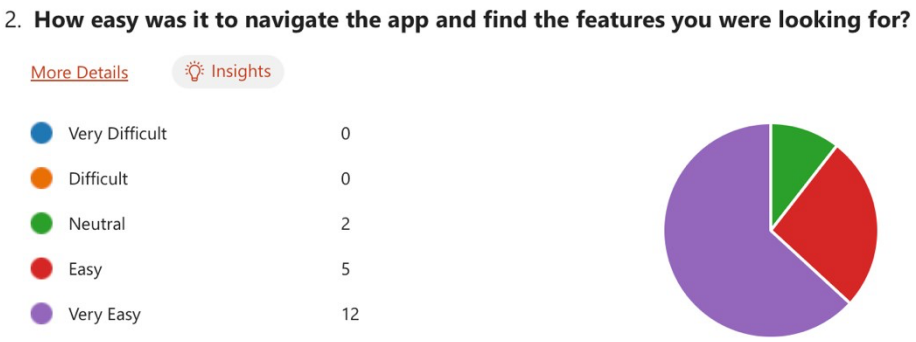


Figure 44: Question 2

7.3.2.3 Question 3/ Question 14

The third question was related to the alerts and latest updates. In particular, they evaluated how useful they think that feature is during emergencies. Almost 90% of the responders rated this feature as very useful (Figure 42). One of the other questions was to rate which of the features they think are more useful in the app and the majority of votes were for the customized alerts (Figure 43). This positive response highlights the importance of getting customized alerts and updates during critical times

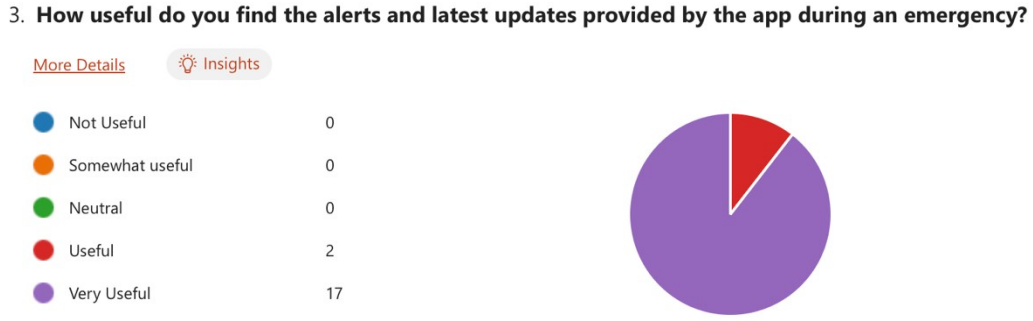


Figure 45: Question 3

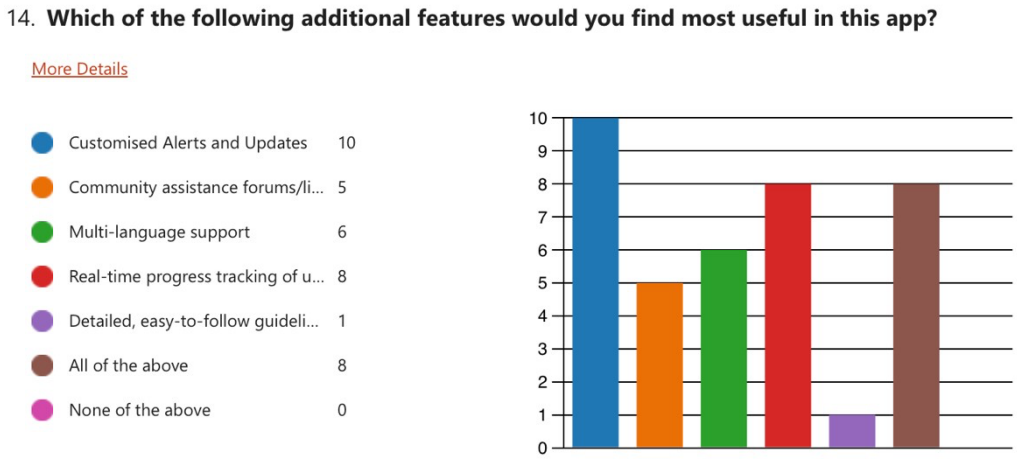


Figure 46: Question 14

7.3.2.4 Question 4/5

The fourth question asked the users if they were facing any difficulties while reporting an incident in the flood scenario and elaborated further if needed In Question 5. All participants who did this process answered no (Figure 44), which indicates that the app responded correctly to their needs without any struggles.

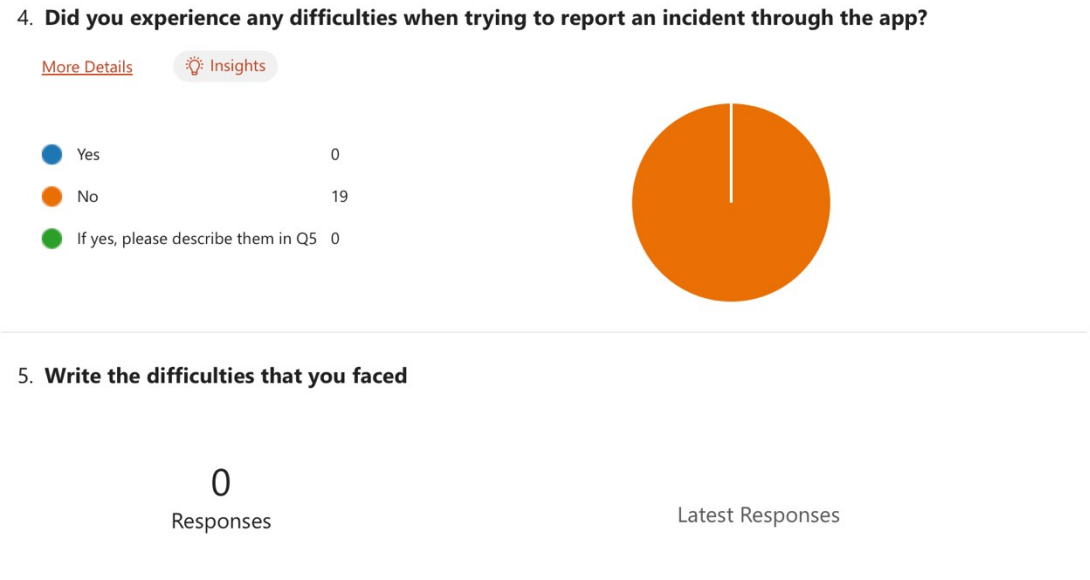


Figure 47: Question 4,5

7.3.2.5 Question 6

The sixth question was related to the satisfaction of the response time. The majority of it was that they were very satisfied. This indicates that the app responds fast, which is an essential part of such applications where the response time is very critical. Slightly less than 50% were “Very satisfied” and more than 40% were “Satisfied” (Figure 45)

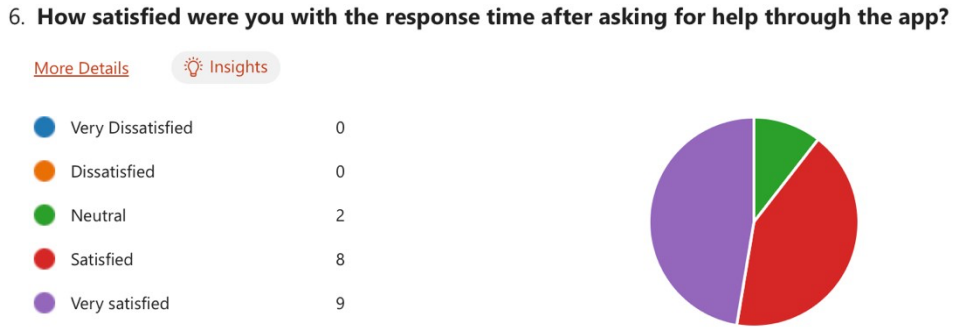


Figure 48: Question 6

7.3.2.6 Question 7

Question 7 was about the group chat that the app offers. Users were asked to evaluate the effectiveness of the group chat feature in providing support and gathering information during an emergency. Based on the results, 74% responded positively about this feature (Figure 46).

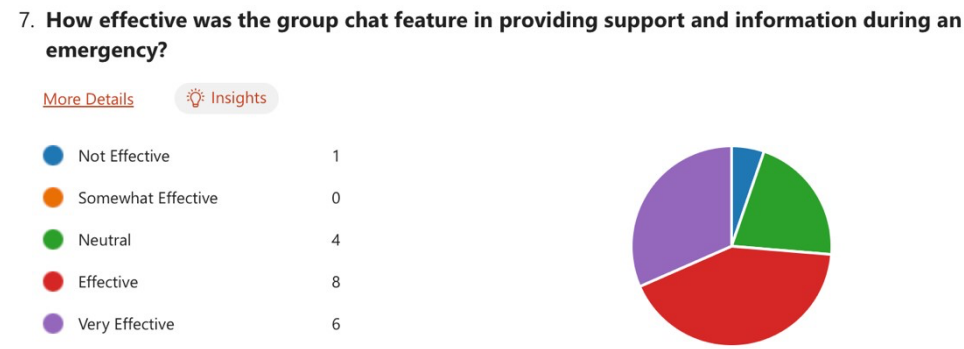


Figure 49: Question 7

7.3.2.7 Question 8

Question 8 considered the user's experience using the private chat during an emergency. "Excellent" and "Neutral" dominated the answers to this question. An interesting discovery showed that users between 34 and 45 years old found it "Neutral", whereas younger ages responded positively towards it. This indicates the benefit the younger generation gets through messaging. older people don't support a lot of the messages through app

Figure 50: Question 8

7.3.2.8 Question 9

Question 9 asked the users how useful the guidelines were, with most of them finding them clear and helpful, however, 26% didn't agree with it.

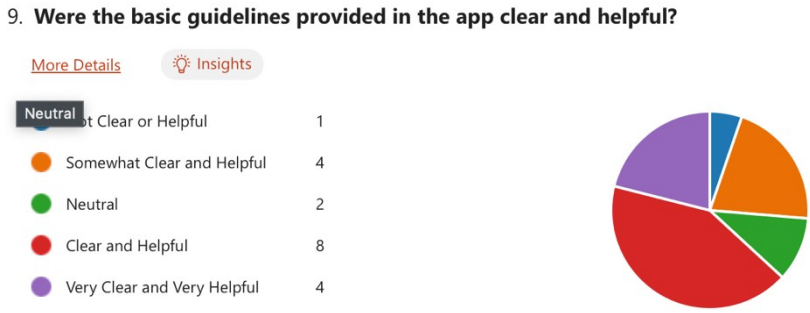


Figure 51: Question 9

7.3.2.9 Question 10

Question 10 asked the user how helpful they found the evacuation route during the critical time. “Neutral” got the most votes, which is something we attributed to the lack of familiarity with this feature.

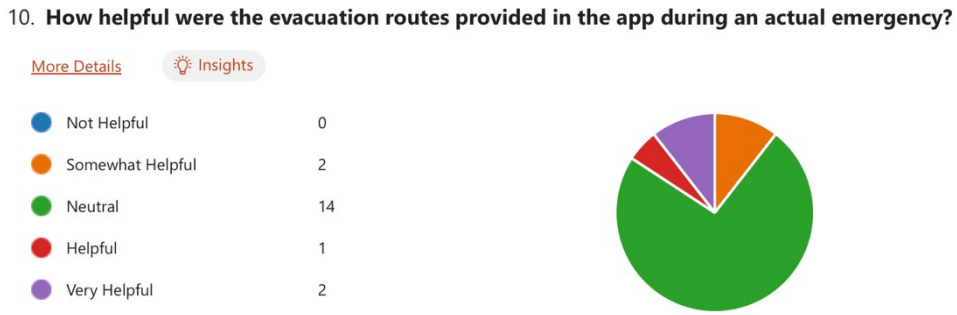


Figure 52: Question 7

7.3.2.10 Question 11

Question 11 asked them how easy it was to create a post in the forum using the mobile app. Except for five responders, the rest expressed a positive manner for this feature.

Figure 53: Question 10,11

7.3.2.11 Question 12

The overall score based on the users was 4.58/5. This shows that users are extremely satisfied with this application.

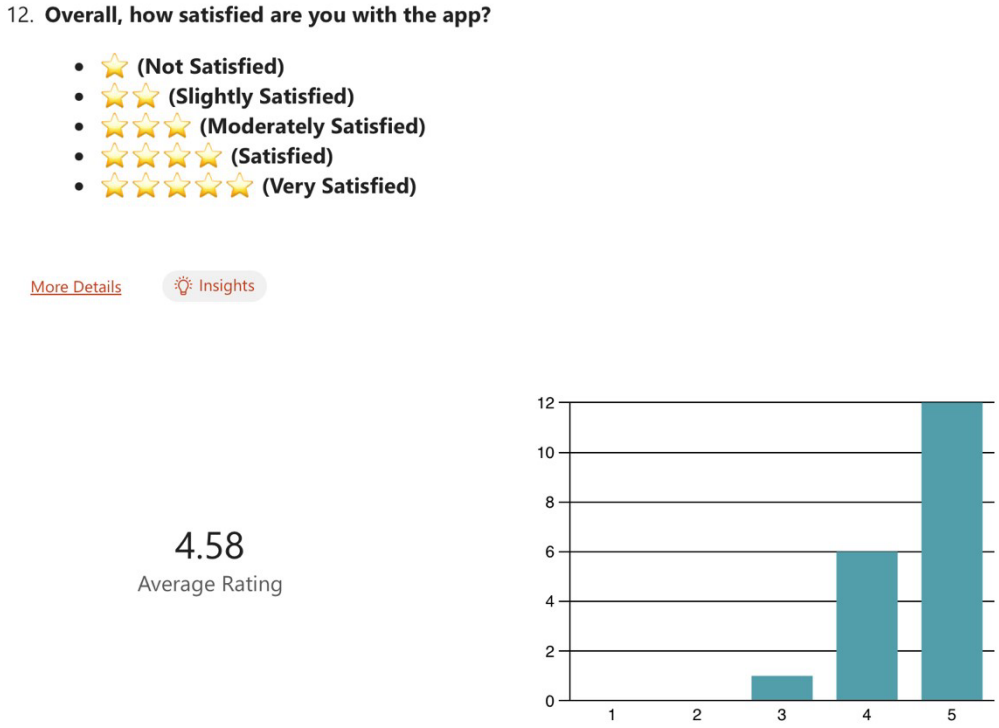


Figure 54: Question 12

7.3.2.12 Question 13

Question 13 asked the users how likely they were to recommend this app to others, with an overall good score of 4.63 out of 5.

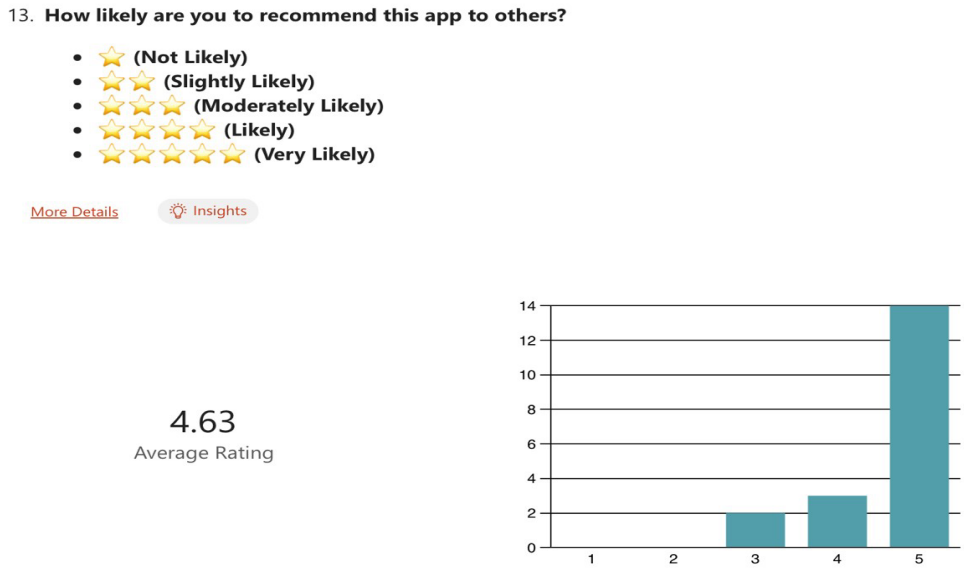


Figure 55: Question 13

7.3.2.13 Question 15

In question 15, we asked the group of users how likely we were to use the app during an emergency based on the features the e112 offers. The results were positive as all respondents agreed that it was either “Very likely” or “Likely”.

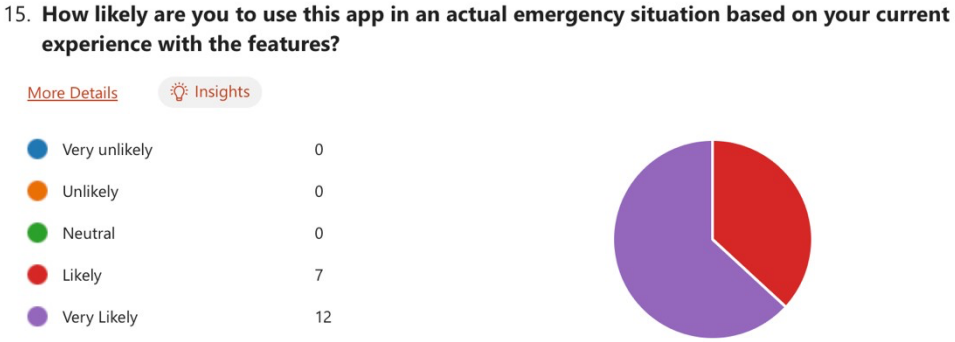


Figure 56: Question 15

7.3.2.14 Question 16

In the final question 16, we also asked them how necessary they think will need this app. Based on the results 13 out of 19 responded with “Very necessary” and 5 out of 19 answered with “Necessary”.

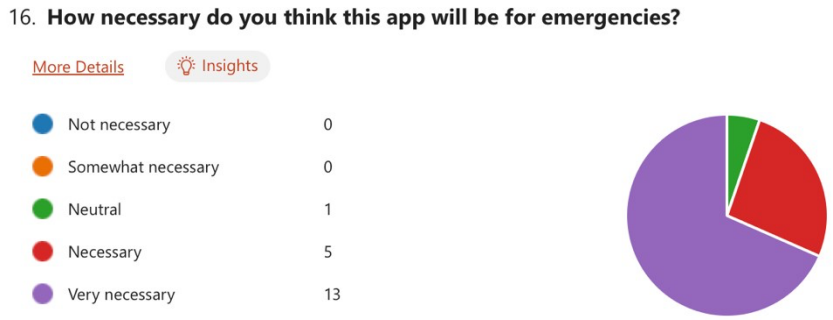


Figure 57: Question 16

7.3.2.15 General Conclusions

According to the users, navigation was easy, they didn’t face any difficulties during use, and they were very satisfied with the response time. It is important to mention some noticeable results regarding the age group 35 and 54, where they responded that the feature of a live chat was not necessary for the app, whereas people between 18 and 34 found it extremely useful. The overall score of the app was 4.58 out of 5 and 4.63 out of 5 suggested are likely to recommend the app to other people. Finally, regarding the score for the necessity of this app during critical times, responders said that it was very necessary.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In conclusion, e112 copes effectively in enhancing user safety, and preparedness through its features, including reporting an incident, asking for help, interacting in the forum and group/private chat, viewing the evacuation map, and finally getting a safe exit route and customized alerts. According to the results we gathered from the questionnaire, younger users found both private and public group chats effective, while older users did not believe they are beneficial, primarily due to their less frequent use of messaging platforms.

8.1 Future Work

Future enhancements will implement photo recognition and verification to verify the authenticity and prevent the spread of false information, enhance server scalability to handle increased traffic of requests, and also implement a feature where the users will have the opportunity to complete their medical info that will be verified by hospital authorities. This will help admins to handle requests more effectively, especially for users with specific health vulnerabilities.

References

[1] Banholzer, Sandra, James Kossin, and Simon Donner. “The Impact of Climate Change on Natural Disasters.” In *Reducing Disaster: Early Warning Systems For Climate Change*, edited by Ashbindu Singh and Zinta Zommers, 21–49. Dordrecht: Springer Netherlands, 2014. https://doi.org/10.1007/978-94-017-8598-3_2.

[2] World Bank. *Building Safer Cities: The Future of Disaster Risk*. Edited by Alcira Kreimer, Margaret Arnold, and Anne Carlin. The World Bank, 2003. https://doi.org/10.1596/0-8213-5497-3.

[3] Shen, Guoqiang, and Seong Nam Hwang. “Spatial–Temporal Snapshots of Global Natural Disaster Impacts Revealed from EM-DAT for 1900-2015.” *Geomatics, Natural Hazards and Risk* 10, no. 1 (January 1, 2019): 912–34. https://doi.org/10.1080/19475705.2018.1552630.

[4] Musani, Altaf, and Irshad Shaikh. “Preparedness for Humanitarian Crises Needs to Be Improved.” *BMJ* 333, no. 7573 (October 19, 2006): 843–45. https://doi.org/10.1136/bmj.38994.548125.94.

[5] Vera-Burgos, Catherine M., and Donyale R. Griffin Padgett. “Using Twitter for Crisis Communications in a Natural Disaster: Hurricane Harvey.” *Heliyon* 6, no. 9 (2020). https://www.cell.com/heliyon/pdf/S2405-8440(20)31647-9.pdf.

[6] “Case Studies of Communications Systems during Harsh Environments: A Review of Approaches, Weaknesses, and Limitations to Improve Quality of Service - Zayan EL Khaled, Hamid Mcheick, 2019.” Accessed April 27, 2024. https://journals.sagepub.com/doi/full/10.1177/1550147719829960.

[7] Ashcroft, Elizabeth. “Value Creation on an Open Access Platform: Performers’ Experiences at the Edinburgh Festival Fringe,” n.d.

[8] *Natural Disasters and Climate Change*. Accessed May 4, 2024. https://link.springer.com/book/10.1007/978-3-319-08933-1.

[9] *User-Experience-Design.pdf*. Available at: https://mu.ac.in/wpcontent/uploads/2022/10/User-Experience-Design.pdf.

[10] Abras, Chadia, Diane Maloney-Krichmar, and Jenny Preece. "1. Introduction and History," 2004.

[11] Gualtieri, Mike. "Best Practices In User Experience (UX) Design," 2009.

[12] Pantouvakis, Angelos, and Anastasia Gerou. "The Theoretical and Practical Evolution of Customer Journey and Its Significance in Services Sustainability." *Sustainability* 14, no. 15 (January 2022): 9610. https://doi.org/10.3390/su14159610.

[13] Parrow, Joachim. "Interaction Diagrams." In *A Decade of Concurrency Reflections and Perspectives*, edited by J. W. de Bakker, W. -P. de Roever, and G. Rozenberg, 477–508. Berlin, Heidelberg: Springer, 1994. https://doi.org/10.1007/3-540-58043-3_26.

[14] *Google Docs. "Saasbook.Pdf." Accessed May 8, 2024. https://drive.google.com/file/d/19dqWms9qUQ2Bi8JfsBhC5Q6zDeJd0s5o/view?us p= embed_facebook.*

[15] Unlighthouse. https://unlighthouse.dev

[16] *First Contentful Paint (FCP), web.dev* https://web.dev/articles/fcp

[17] *Largest Contentful Paint (LCP), web.dev* https://web.dev/articles/lcp

[18] Total Blocking Time (TBT), web.dev https://web.dev/articles/tbt