**New Cairo Technological University**

**Technological College of Industry and Energy**

**Information and Communication Technology Department**

**Community Verified icon**

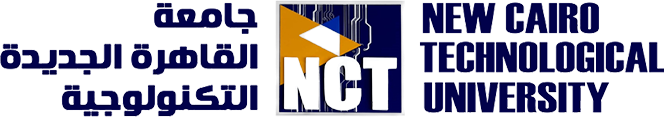
**Bimar App**

A health care solution

**Team members:**

**Project supervisor:** Dr. Eman Monir & Dr. Iman Ahmed El Sayed

**Graduation Project (2024/2025)**

**New Cairo Technological University**

**Technological College of Industry and Energy**

**Information and Communication Technology Department**

**Community Verified icon**

**Bimar App**

A health care solution

**Team members:**

|  |  |
| --- | --- |
| 1- | () |
| 2- | () |
| 3- | () |
| 4- | () |
| 5- | () |
| 6- | () |

1. ()

**Project supervisor:** Dr. Eman Monir & Dr. Iman Ahmed El Sayed

# Acknowledgment

Firstly, the success of our project happens when we believe in ourselves, believe in doing the right thing to help other people, and the help of many people around us. Secondly, we would like to thank our supervisor, Dr. Eman Monir & Dr. Iman Ahmed El Sayed for their encouragement, patience, guidance, and support when our project starts until we finished the project; thank you, Dr. Eman Monir ;thank you, Dr. Iman Ahmed El Sayed , We feel that we have really been lucky to be working with someone like them.

Finally, we are very grateful to the many people who helped us with their contributions to complete this project.

# Abstract

Bimar is a smart healthcare platform with a mobile app for patients (Android & iOS) and a web portal for doctors. It enables doctor booking, secure medical record management, AI-powered diagnosis predictions, and real-time communication between patients and doctors.

Patients can store and manage medical and personal health records, including family history, allergies, chronic diseases, medications, surgeries, and prescriptions, while controlling doctor access. Doctors can chat with patients, request medical records, analyze clinic performance, and manage their practice.

The app also features AI-driven insights for symptom-based diagnosis, automated reminders for medications and appointments, and a privacy-focused data-sharing system. With intelligent health tracking and seamless doctor-patient interaction, Bimar enhances diagnosis accuracy, treatment management, and healthcare accessibility.

# Table of Contents:

[Table of Contents: 3](#_bookmark0)

[List of Tables and Figures 5](#_bookmark1)

[Chapter 1: Introduction 6](#_bookmark2)

* 1. [Purpose 6](#_bookmark3)
  2. [Problem statement 6](#_bookmark4)
  3. [Project Scope 8](#_bookmark5)
  4. [Methodology 11](#_bookmark6)

[Chapter 2: Software Requirement Specification 13](#_bookmark7)

* 1. [Functional Requirements 13](#_TOC_250006)
  2. [Non-Functional Requirements 15](#_TOC_250005)
     1. [Performance Requirements 15](#_TOC_250004)
     2. [Safety and security requirements 16](#_TOC_250003)
     3. [Reliability 16](#_TOC_250002)
     4. [Usability 16](#_TOC_250001)
     5. [Availability 17](#_TOC_250000)

[Chapter 3: System analysis 18](#_bookmark8)

1. [Use Case Diagram 18](#_bookmark9)
2. [Class Diagram 20](#_bookmark10)
3. [Sequence Diagram 21](#_bookmark11)
4. [Activity Diagram 22](#_bookmark12)

[Chapter 4: External Interface Requirements 23](#_bookmark13)

1. [User Interface 23](#_bookmark14)

[Chapter 5: Implementation 37](#_bookmark15)

[Chapter 6: Conclusion & Future Work 40](#_bookmark16)

[Appendix 41](#_bookmark17)

[References 41](#_bookmark18)

# List of Tables and Figures

***Table 1 Functional Requirements*** *13*

***Table 2.1 Performance Requirements*** *15*

***Table 2.2 Safety and Security Requirements*** *16*

***Table 2.3 Reliability*** *16*

***Table 2.4 Usability*** *16*

***Table 2.5 Availability*** *17*

***Table 3 Results*** *39*

***Figure 1 Main Use Case Diagram*** *18*

***Figure 1.1 Camera Use Case Diagaram*** *19*

***Figure 1.2 Opening User Account*** *19*

***Figure 2 Class Diagram*** *20*

***Figure 3 Sequence Diagram*** *21*

***Figure 4 Activity Diagram*** *22*

***Figure 5 Splash Screen*** *23*

***Figure 6 Login Page*** *24*

***Figure 7 SignUp Page*** *25*

***Figure 8 Home Page*** *26*

***Figure 9 Menu bar*** *27*

***Figure 10 Camera Page*** *28*

***Figure11 My Profile Page*** *29*

***Figure 12 Post Page*** *30*

***Figure 13 Emergency Page*** *31*

***Figure 14 Hospitals Page*** *32*

***Figure 15 Notification Page*** *33*

***Figure 16 Help Page*** *34*

***Figure 17 Contact Us Page*** *35*

***Figure 18 Community Page*** *36*

# Chapter 1: Introduction

## Purpose

The purpose of Bimar is to enhance healthcare accessibility, efficiency, and accuracy by leveraging digital technology and artificial intelligence. The platform aims to streamline doctor-patient interactions, medical record management, and diagnosis prediction, providing a seamless, secure, and user-friendly experience for both patients and healthcare providers.

For patients, Bimar offers a centralized platform to book doctors, store and manage medical records, receive AI-powered diagnosis predictions, and track medications and appointments. It ensures that individuals have full control over their health data privacy, allowing them to grant or revoke access to doctors as needed.

For doctors, Bimar provides a comprehensive web portal to manage patient interactions, request and review medical records, analyze clinic performance, and enhance decision-making with AI-driven insights.

By integrating real-time communication, AI-driven diagnosis assistance, and automated health management, Bimar seeks to improve the quality of care, reduce delays in diagnosis, and empower both patients and doctors with advanced healthcare solutions.

## Problem statement

The increasing burden of healthcare management presents a significant challenge to patients and medical professionals alike. **The lack of an integrated, user-friendly system for storing, accessing, and sharing medical records hinders efficient healthcare delivery, leading to delays in diagnosis, treatment inefficiencies, and potential medical errors.** Patients often struggle to maintain comprehensive health histories, making it difficult for doctors to obtain crucial information for accurate assessments. Additionally, the traditional methods of booking appointments and seeking medical consultations are often time-consuming and inconvenient, further impacting patient care.

The fragmented nature of healthcare data results in **disorganized record-keeping**, where critical information such as allergies, chronic diseases, medications, previous surgeries, and vaccination histories may be scattered across multiple sources. This fragmentation not only affects patient safety but also **burdens healthcare providers**, who must navigate complex systems to retrieve essential information, sometimes leading to **misdiagnoses or redundant tests**.

Beyond inefficiencies in medical record management, the healthcare industry is yet to fully harness the potential of **artificial intelligence (AI) in diagnosis, treatment recommendations, and predictive analytics**. AI-driven solutions have the capacity to **analyze patient history, detect patterns, and provide clinical decision support**, yet many existing healthcare platforms lack these advanced capabilities. The absence of AI-driven insights contributes to a reactive rather than proactive approach to patient care, delaying early interventions for potentially serious conditions.

Moreover, the **rising global demand for accessible, remote healthcare services** further highlights the limitations of traditional systems. Patients, especially those in rural or underserved areas, face difficulties in connecting with specialized doctors, resulting in delayed care and worsening health conditions. In emergency situations, **rapid access to a patient’s complete medical history is crucial for timely and effective treatment**, yet current systems fail to provide seamless, instant access to such data.

In light of these challenges, **there is a pressing need for a modernized, AI-integrated healthcare system that offers centralized medical record management, seamless doctor-patient communication, and intelligent diagnostic support**. By addressing these issues, **Bimar** aims to bridge the gaps in healthcare accessibility, enhance efficiency for both patients and medical professionals, and ultimately improve overall healthcare outcomes.

## Project Scope

The Bla3'ny project is an innovative mobile application designed to revolutionize disaster management and provide timely support during emergencies. It addresses the critical gap between affected communities and essential resources by harnessing advanced technology and real-time

data. The app delivers critical information, facilitates communication, and mobilizes resources during natural disasters or human-made incidents. With its responsive design, it ensures seamless access and usability across various devices. By embracing the power of technology and fostering collaboration, the Bla3'ny app has the potential to make a significant difference in disaster response, saving lives, and building resilient communities.

#### The scope of this project includes the following deliverables:

* High performance and easy to use android application with a user interface.
* detect disasters on a real-time base.
* alert emergency responders on a community.
* Communicate on a spread wise basis through Facebook Community in addition to app community.
* that uses machine learning and camera technology to detect disasters in real- time.
* use image recognition algorithms to analyze photos captured by user’s smartphones and identify signs of an impending disaster such as flooding, wildfires, accidents or earthquakes.
  + **Project timeline:** The project is expected to take approximately 5 months to complete, as following:
    - Month 1: Collecting requirements, planning, and design
    - Month 2-3-4: Development and testing
    - Month 5: Project launch

#### Project resources:

The project team have knowledge in front-end development, back-end development, flutter development, UI/UX, database development, NLP, and ML.

#### Stakeholders:

* **End-users**: The people who will be downloading and using the app to detect potential disasters in their area. They are the primary stakeholders.
* **Developers**: The team of software developers who will be responsible for designing, developing, and maintaining the app.
* **Machine Learning Engineers**: The experts in machine learning who will be responsible for developing the algorithms that enable the app to detect disasters.
* **Emergency responders**: The emergency response teams who will receive alerts from the app when disasters are detected.
* **Government agencies**: The local or national government agencies responsible for disaster management and response.
* **Investors**: The individuals or organizations who provide funding for the development and maintenance of the app.
* **Data providers**: The organizations or individuals who provide data for the app to use in its machine learning algorithms, such as satellite

imagery or weather data.

* **Privacy advocates**: Individuals or organizations who are concerned about the privacy implications of the app and may provide feedback or input on privacy policies.
* **Media and journalists**: The media and journalists who may report on the app and its potential impact on disaster detection and response.
* **Non-governmental Organizations (NGOs)**: NGOs that work in disaster relief and recovery may be interested in the app as a tool to help with their work.
  1. **Methodology *(tools)***

1. **Flutter**: Flutter is an open-source UI software development kit (SDK) created by Google. It is used to build high-performance, cross-platform mobile applications for iOS, Android, and web. Flutter uses the Dart programming language and offers a rich set of customizable widgets, tools, and libraries that allow developers to create beautiful and responsive user interfaces. Some of the advantages of using Flutter include fast development, hot reloading, and the ability to build natively compiled applications that can run on multiple platforms.
2. **Adobe XD**: Adobe XD is an all-in-one design tool that provides a complete set of design tools for designers to create and prototype user interfaces. It includes features such as wireframes, high-fidelity designs, and interactive prototypes, all in one place. Adobe XD also allows designers to collaborate with other members of the team and share their designs with developers for implementation.
3. **Firebase**: Firebase is a mobile and web application development platform that provides tools and services for building and managing applications. It offers features such as real-time database, authentication, cloud storage, hosting, and more. Firebase allows developers to build and deploy applications quickly and easily, without worrying about server infrastructure or backend development. It also provides features for app analytics and user engagement, making it easier for developers to monitor and optimize their applications for better performance and user experience.
4. **Dart**: Dart is a general-purpose programming language developed by Google for building web, desktop, and mobile applications. It is an object-oriented language with syntax similar to C. Some of the advantages of using Dart include fast development, easy syntax, and the ability to compile to native code. Dart is the language used by Flutter and is also used in other Google projects.
5. **TensorFlow**: TensorFlow is an open-source machine learning framework developed by Google that allows developers to build and train machine learning models. Some of the advantages of using TensorFlow for machine learning include flexibility, scalability, and ease of use. TensorFlow offers a high degree of flexibility in designing and implementing machine learning models, allowing

developers to build models from scratch or use pre-built models and customize them as per their requirements. TensorFlow is also highly scalable, making it suitable for large-scale machine learning projects. TensorFlow is used in a wide range of applications, including image and speech recognition, natural language processing, and more.

1. **GitHub** : is a web-based platform used for version control and collaboration on software development projects. It provides a platform for developers to store, manage, and share their code with others. GitHub offers features such as version control, issue tracking, project management, and collaboration tools that make it easier for developers to work together on projects. GitHub is built on top of Git, a distributed version control system that allows developers to keep track of changes to their code and collaborate with others. It provides a graphical user interface for managing Git repositories and offers features such as pull

requests, code reviews, and continuous integration and deployment.

**The project will follow the agile model**, the Agile model is an iterative and incremental approach to software development that focuses on collaboration, flexibility, and continuous improvement. It is based on the Agile Manifesto, which outlines four values and twelve principles for Agile software development. The Agile model involves short iterations called sprints, where development teams work closely with end-users and the product owner to deliver small sets of features or functionality in a working state. This approach allows for flexibility and adaptability, as the team can quickly respond to changes in requirements or user feedback.

# Chapter 2: Software Requirement Specification

## Functional Requirements

***Table 1 Functional Requirements***

|  |  |
| --- | --- |
| Functional Requirement  No. | Function Requirement Description |
| R-F1 | The system must provide user authentication through a login feature to ensure that only authorized users can access the app's  features. |
| R-F2 | The system must provide user authentication through a sign-up  feature to allow new users to create an account. |
| R-F3 | The system must allow users to create and manage their  profiles, including personal information and contact details. |
| R-F4 | The system must allow users to manage their preferences, such  as settings and notification preferences. |
| R-F5 | The app should be able to access the rear camera of the mobile  device to capture images of the disaster-affected area. |
| R-F6 | The app should be able to access the front camera of the mobile device to capture images of the user's face for authentication  purposes. |
| R-F7 | The app should have machine learning algorithms integrated to analyze the images captured by the rear camera and detect any  signs of a disaster. |
| R-F8 | The app should have machine learning algorithms integrated to analyze the images captured by the front camera and  authenticate the user. |
| R-F9 | The app should be able to identify different types of natural disasters, such as earthquakes, floods, fires, etc., and provide  relevant information to the user. |
| R-F10 | The app should be able to identify different types of man-made disasters, such as terrorist attacks, and provide relevant  information to the user. |

|  |  |
| --- | --- |
| R-F11 | The app should have a notification system that alerts users  about any potential disasters in their area. |
| R-F12 | The app should have a notification system that provides updates  related to any disasters the user has reported. |
| R-F13 | The app should allow users to report any natural disasters they  detect using the camera, including location, type, and severity. |
| R-F14 | The app should allow users to report any man-made disasters they detect using the camera, including location, type, and  severity. |
| R-F15 | The app should allow users to add emergency contacts to their  profiles, which can be contacted in case of an emergency. |
| R-F16 | The app should allow users to add medical information to their profiles, which can be accessed by emergency responders in  case of an emergency. |
| R-F17 | The app should be able to track the user's location and provide  relevant information about any natural disasters in their area. |
| R-F18 | The app should be able to track the user's location and provide relevant information about any man-made disasters in their  area. |
| R-F19 | The app should securely store all user data, including login  credentials and profile information, in a cloud-based database  or server. |
| R-F20 | The app should securely store all disaster reports in a separate  cloud-based database or server. |

## Non-Functional Requirements

* + 1. Performance Requirements
    2. Safety and security requirements
    3. Reliability
    4. Usability
    5. Availability

#### Performance Requirements:

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| PR-1 | The system must respond to user/admin request in less than 3 seconds. |
| PR-2 | The system must provide high accuracy in disaster detection to ensure reliable alert generation. |
| PR-3 | The system must provide fast data processing and transmission with minimal latency to ensure timely response to disasters. |
| PR-4 | The system must be able to handle high user traffic  during emergencies to ensure that all users can access the app's features. |
| PR-5 | The system must consume minimal power to ensure prolonged use during emergencies when power  sources may be limited. |
| PR-6 | The system must be able to handle large data sets for historical analysis to provide insights and  improve disaster response in the future. |

***Table 2.1 Performance Requirements***

#### Safety and security requirements:

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| SSR-1 | The system must provide safe login and logout through session. |
| SSR-2 | Login process must be secured by using an encryption technique. |
| SSR-3 | The system database must be secured from SQL injections to prevent leak/loss of user data. |
| SSR-4 | The system must conduct regular security audits  and updates to detect and prevent potential security breaches. |

***Table 2.2 Safety and Security Requirements***

#### Reliability:

|  |  |
| --- | --- |
| Requirements ID | Requirements Description |
| R-1 | The system must provide high uptime and availability to ensure that the app can be accessed  during emergencies. | |
| R-2 | The system must provide robust backup and recovery mechanisms to prevent data loss in case of system failures or disasters. | |
| R-3 | The system must conduct regular testing and quality assurance to ensure optimal performance and functionality, including load testing, stress testing, and usability testing, among others. | |

***Table 2.3 Reliability***

#### Usability:

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| U-1 | The system must provide an intuitive and user- friendly interface that is easy to navigate and use. |
| U-2 | once the user has learned how to use the system, he should perform his tasks on the system efficiently. |

|  |  |
| --- | --- |
| U-3 | The system must provide clear and concise instructions for disaster response and emergency management to ensure that users can respond  effectively to disasters. |

***Table 2.4 Usability***

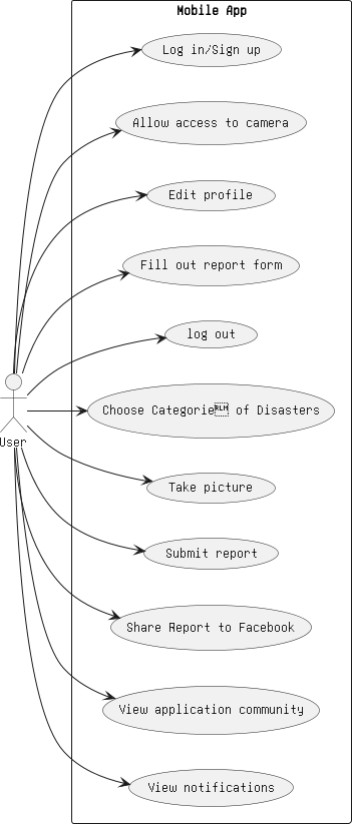
#### Availability:

|  |  |
| --- | --- |
| Requirement ID | Requirement Description |
| A-1 | The system must be available 24/7 |
| A-2 | The system must be compatible with different  network types and speeds, such as 3G, 4G, and 5G, for seamless access in both urban and remote areas. |
| A-3 | The system must provide high uptime and availability to ensure that the app can be accessed  during emergencies. |

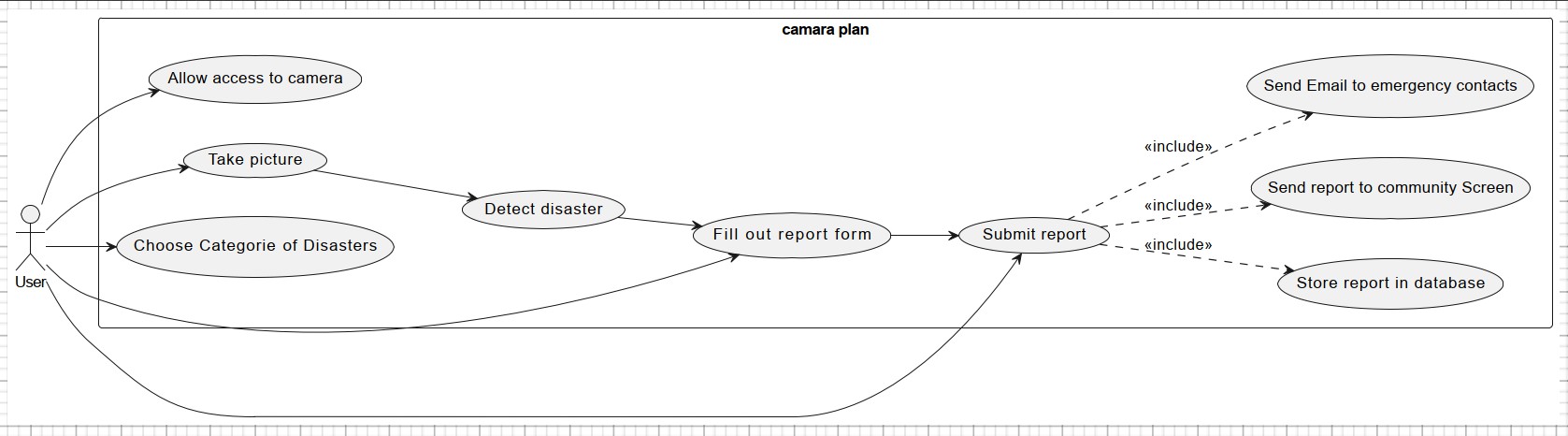
***Table 2.5 Availability***

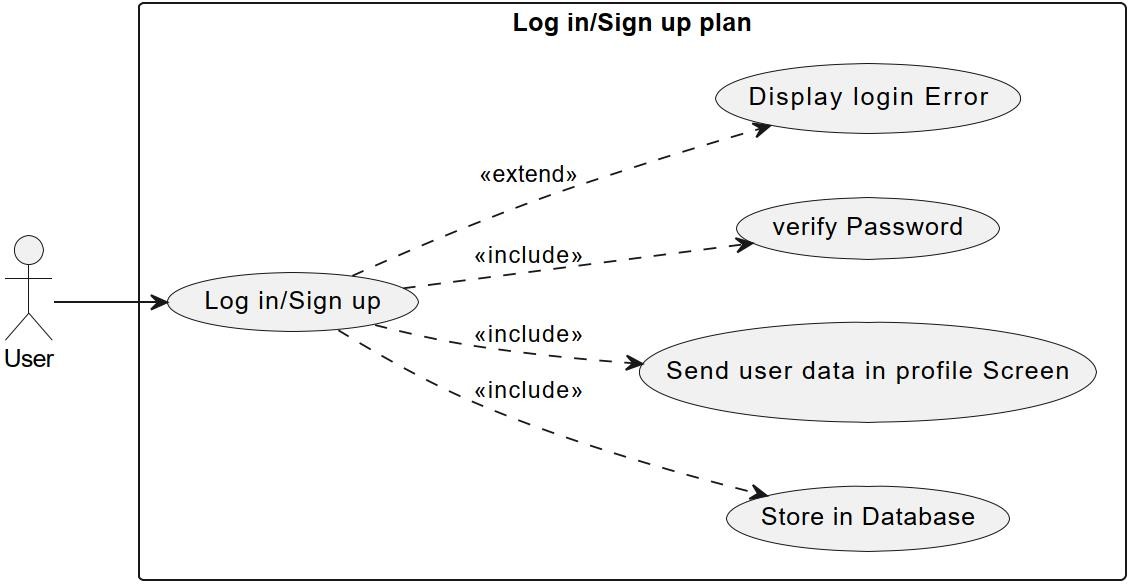
# Chapter 3: System analysis

### Use Case Diagram



**Figure 1. Main Use Case Diagram**



**Figure 1.1 Camera Use Case Diagram**

**Figure 1.2 Opening User Account**

### Class Diagram

**Figure 2. Class Diagram**

### Sequence Diagram

**Figure 3. Sequence Diagram**

### Activity Diagram

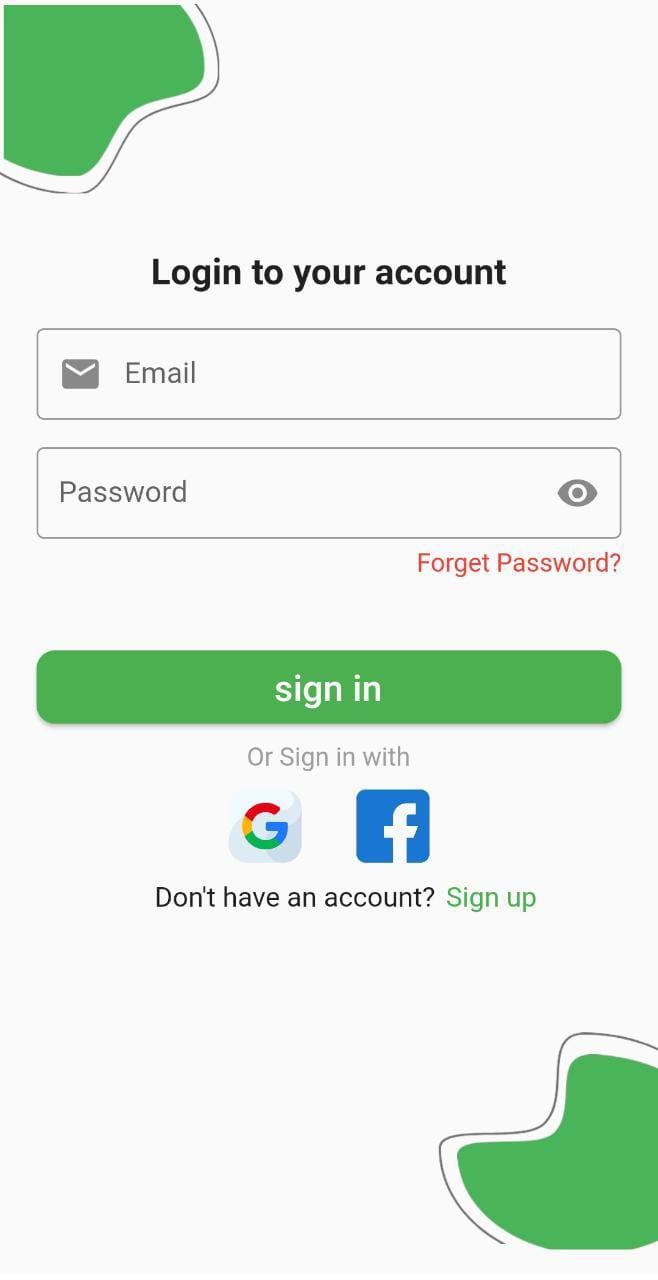
**Figure 4. Activity Diagram**

# Chapter 4: External Interface Requirements

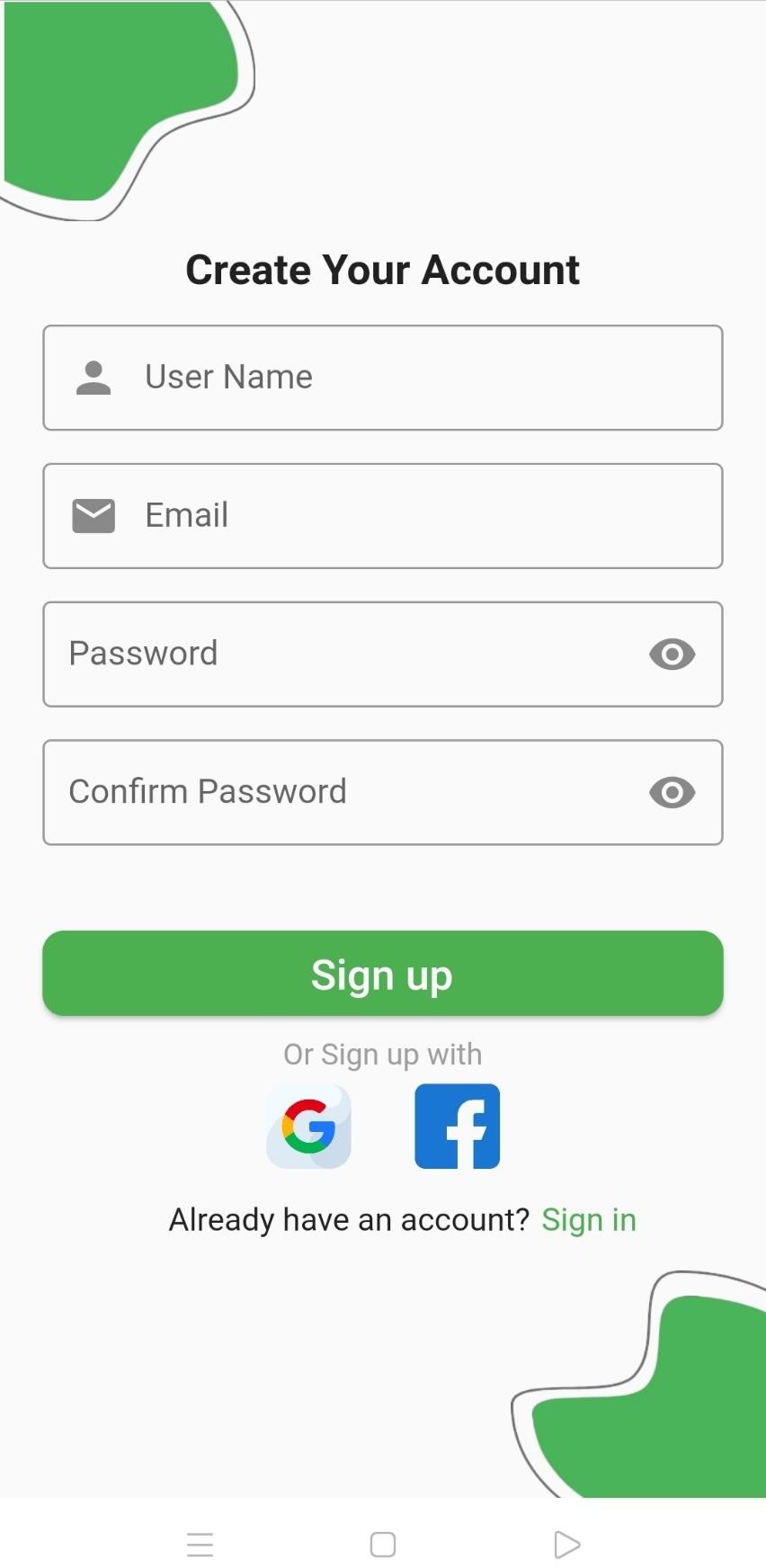
### User Interface



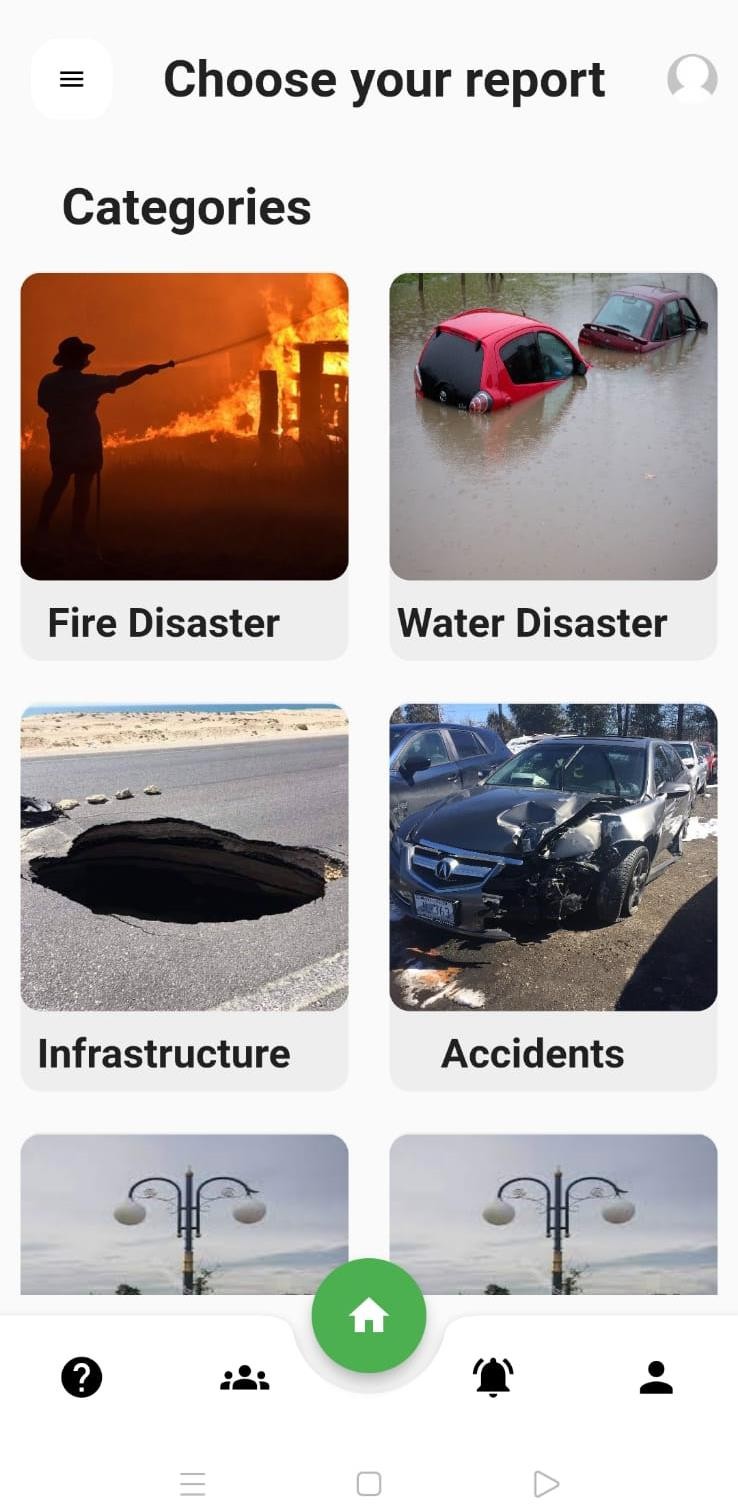
**Figure 5. Splash Screen**



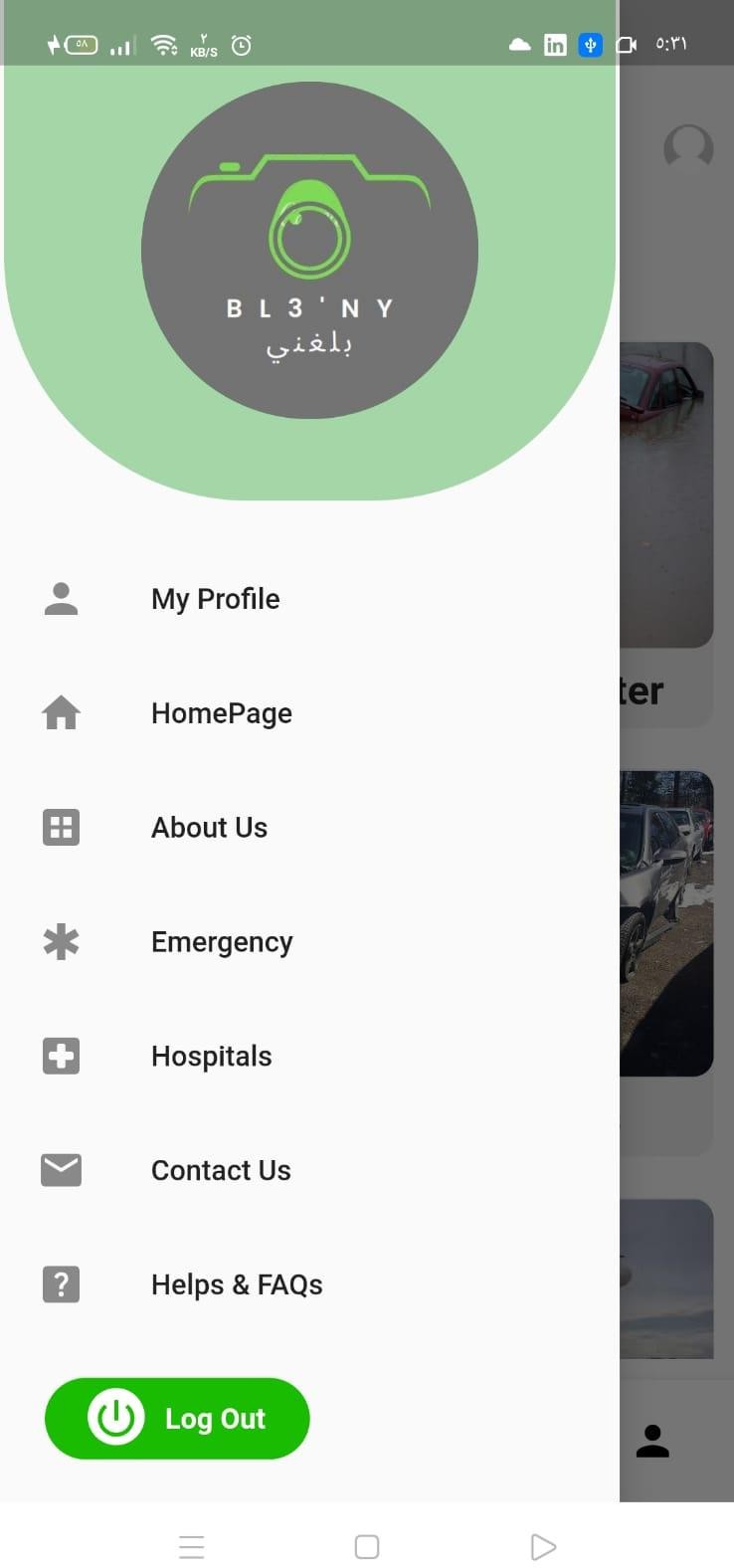
**Figure 6. Login Page**



**Figure 7. SignUp Page**



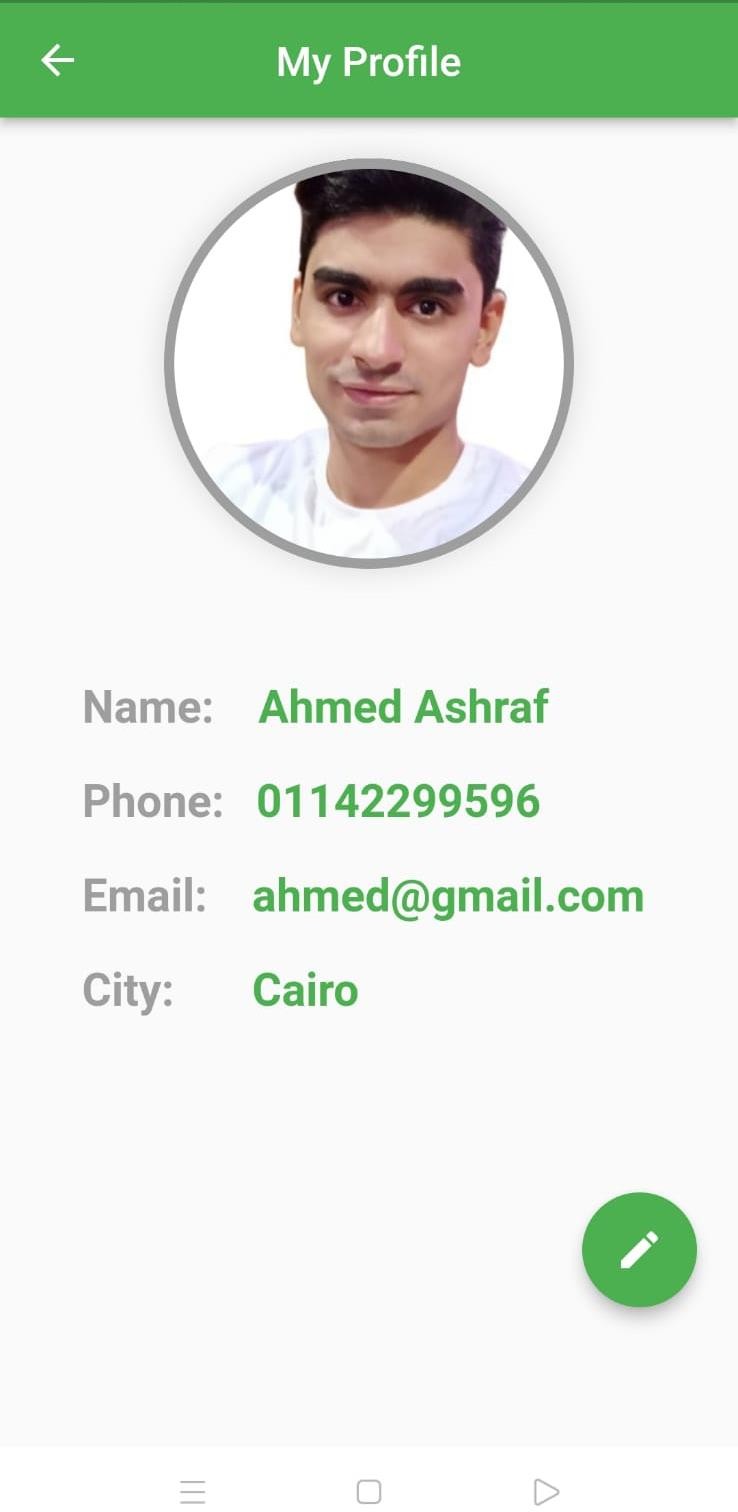
**Figure 8. Home Page**



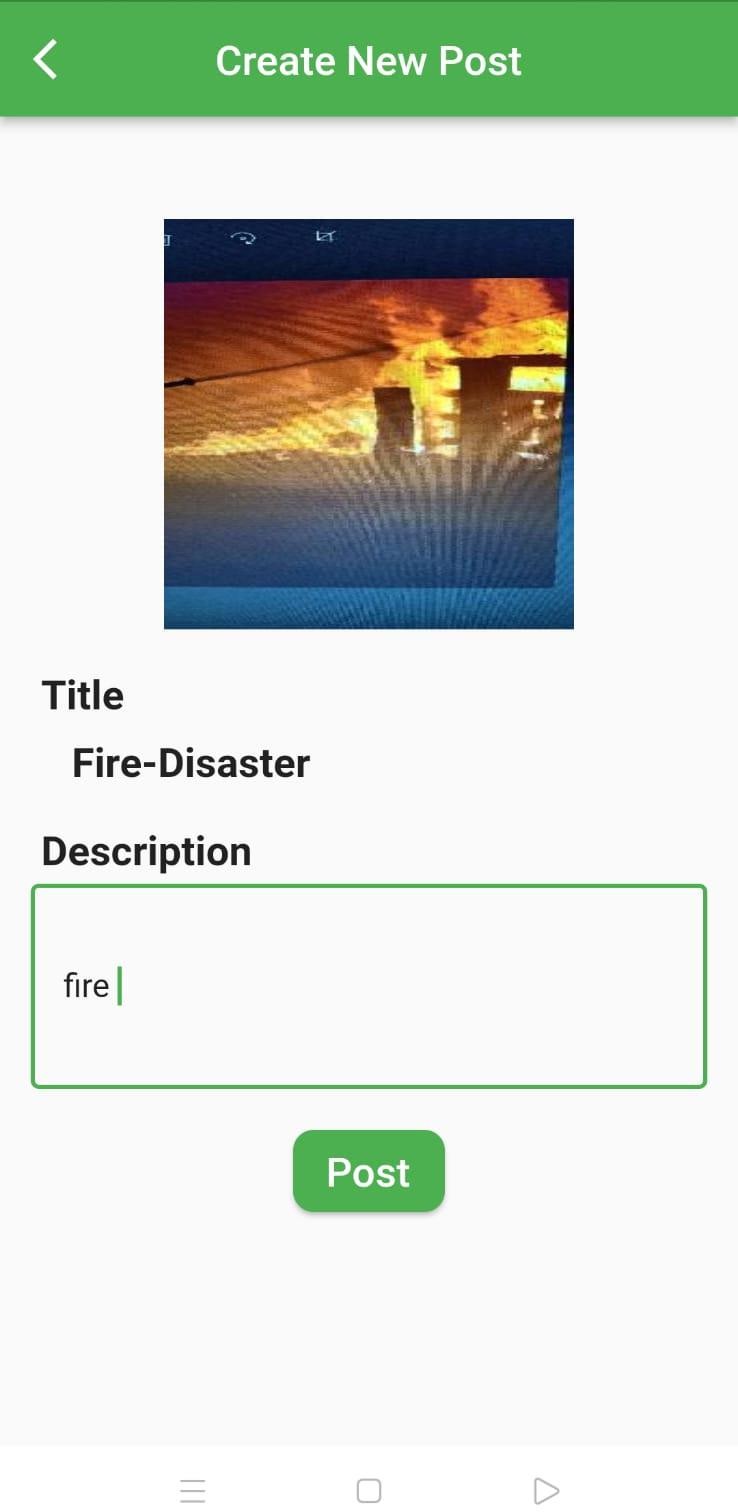
**Figure 9. Menu bar**



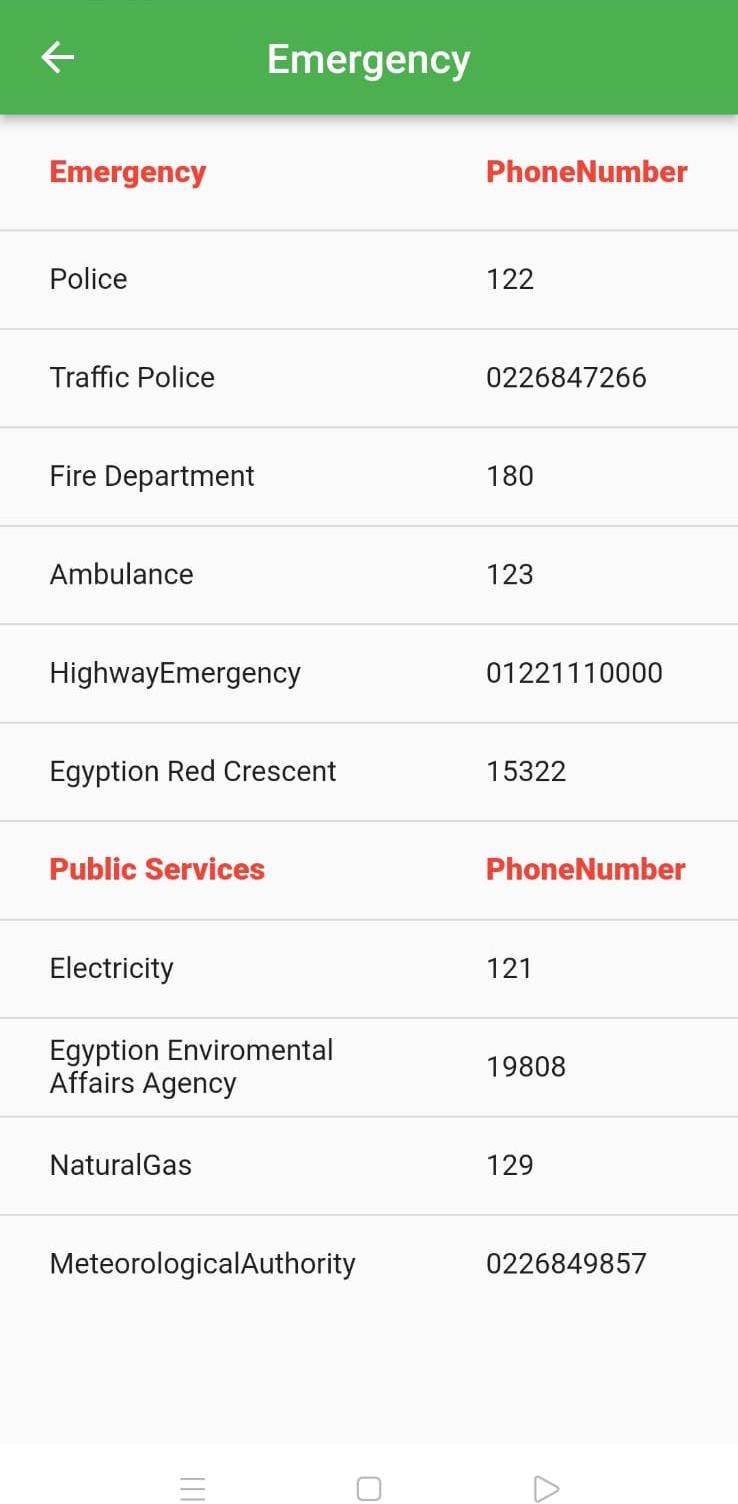
**Figure 10. Camera Page**



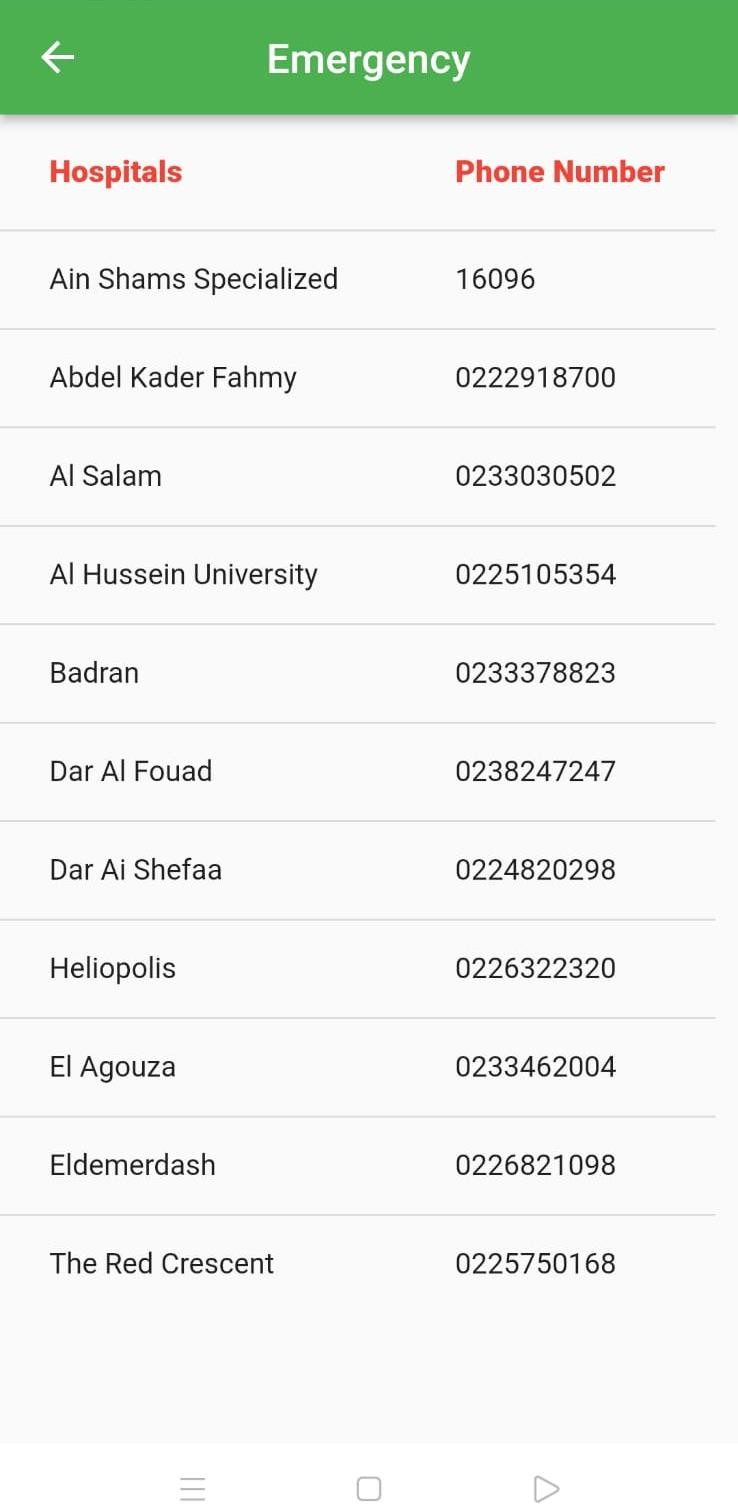
**Figure 11. My Profile Page**



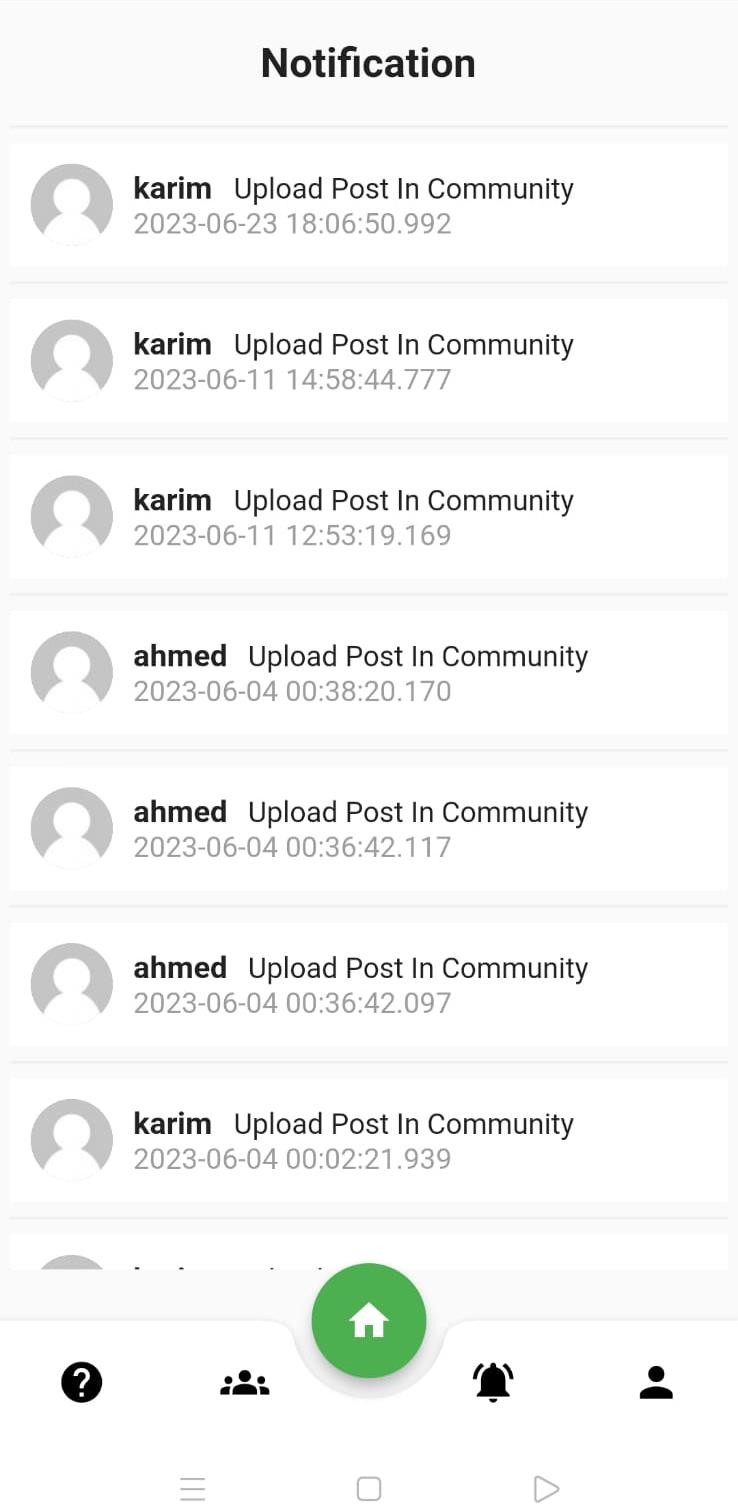
**Figure 12. Post Page**



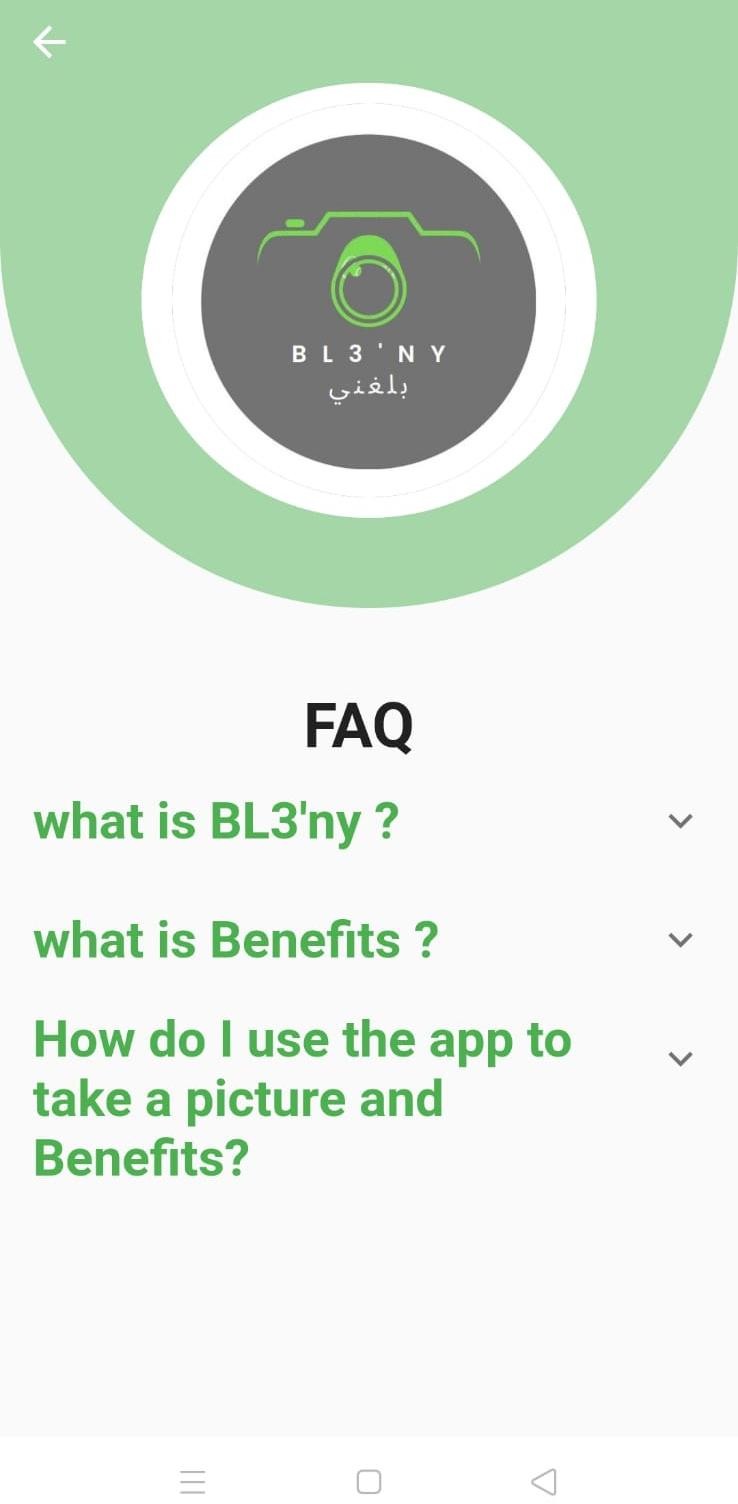
**Figure 13. Emergency Page**



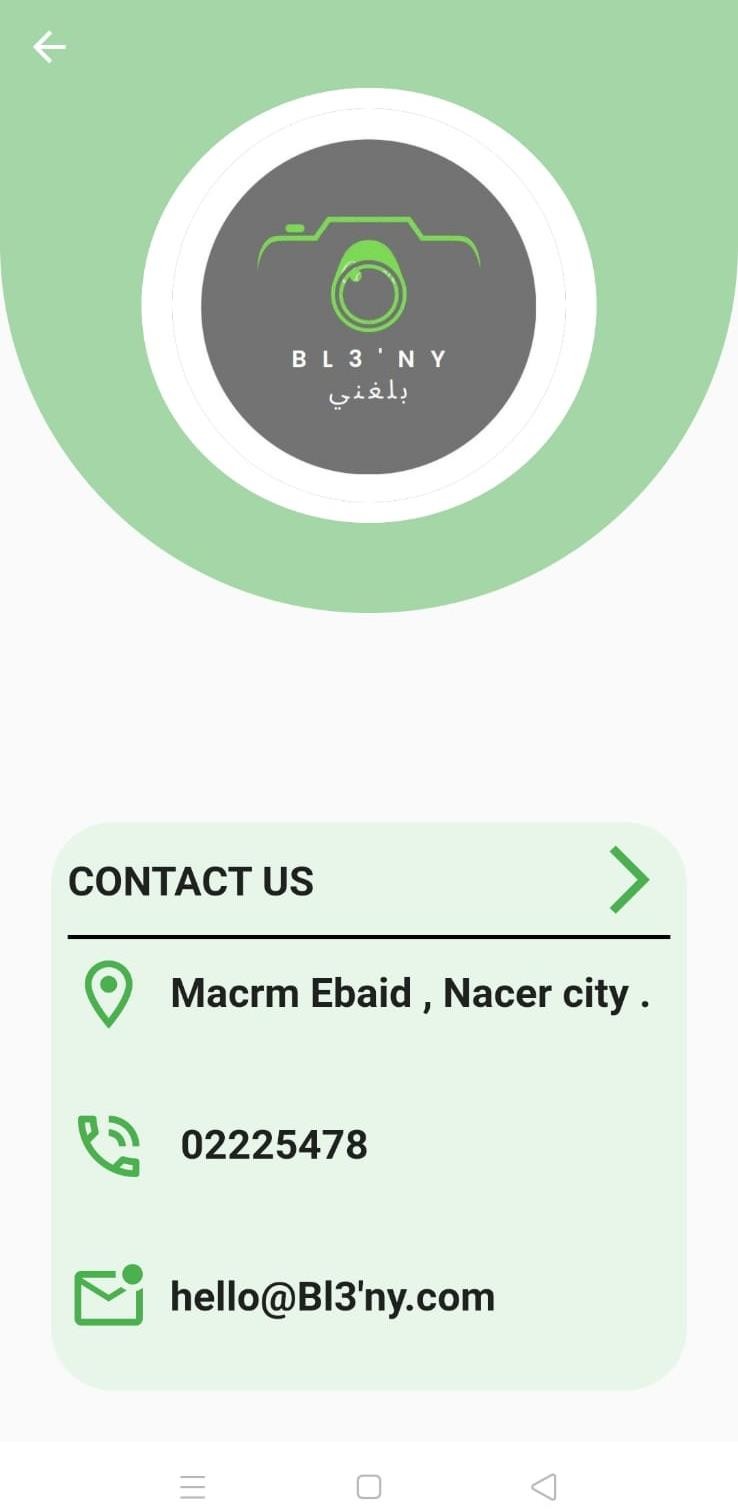
**Figure 14. Hospitals Page**



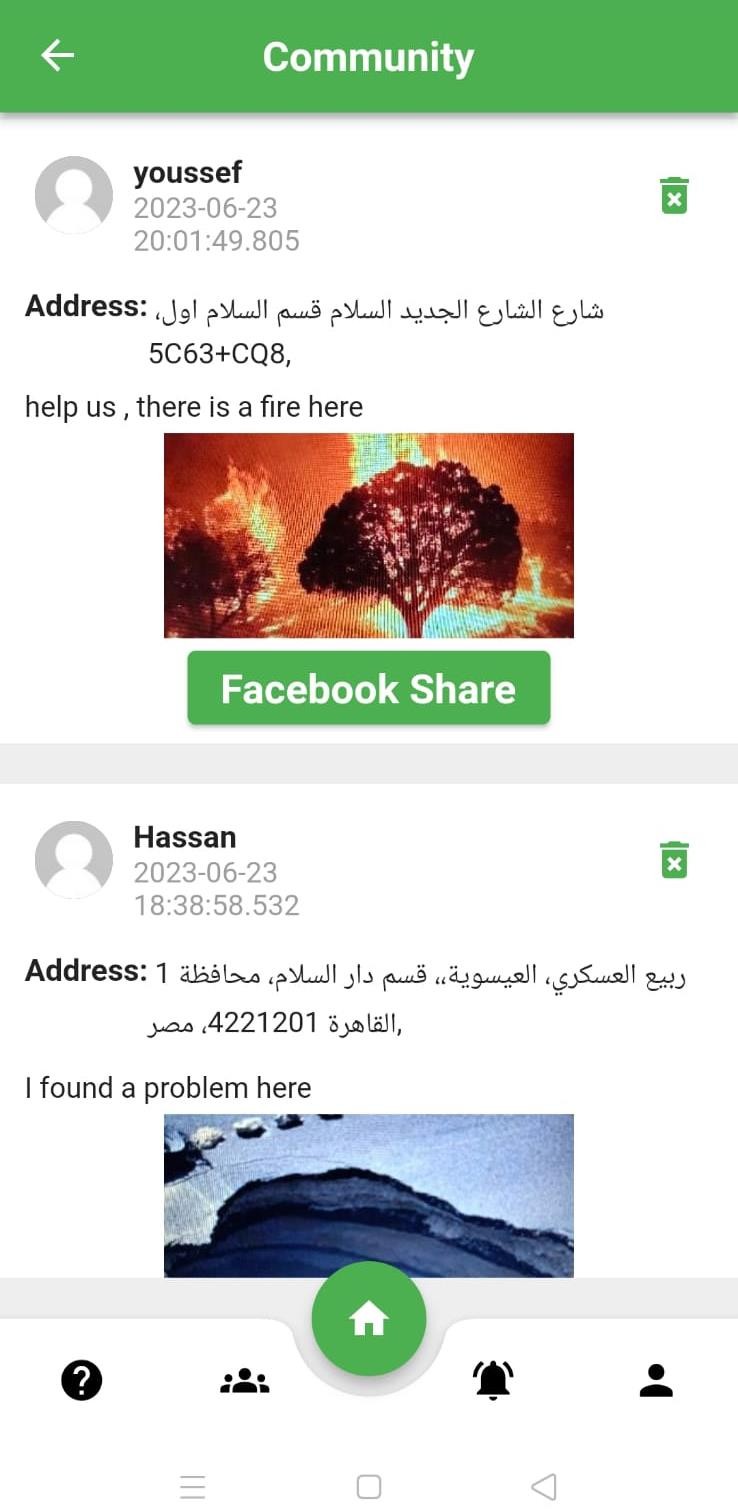
**Figure 15. Notification Page**



**Figure 16. Help Page**



**Figure 17. Contact Us Page**



**Figure 18. Community Page**

# Chapter 5: Implementation

## (Explanation of each stage of implementation)

1. **Gather data**: In this stage, data in the form of images of various disasters, such as fires, floods, and earthquakes, is collected from various sources. The images can be obtained from satellites, drones, or other sources. The data collected

should be relevant to the disaster being detected, such as images of flooded areas or burned forests.

1. **Train ML model**: The gathered images are used to train the machine learning model. The model is trained to recognize various disaster scenarios by analyzing the visual features of the images. This involves using machine learning algorithms to

teach the model how to identify patterns and features in the images that are indicative of different types of disasters.

1. **UI Design**: The user interface (UI) for the mobile app is designed. The UI should be intuitive and easy to use, with features such as a camera viewfinder and a map view. The design should also take into account the different types of

disasters being detected and the information that users will need to see in order to take appropriate action. For example, the UI may display the detected disaster type, the location, and the

severity of the disaster.

1. **Integrate the model into flutter**: The trained machine

learning model is integrated into the Flutter framework, which is a popular mobile app development platform. This involves

writing code to connect the model to the app and to handle

inputs from the camera and other sensors. The model analyzes

the images captured by the camera in real-time and provides the app with information about the type of disaster detected.

1. **Integrate Firebase into the app**: Firebase is a platform that provides various services, including cloud storage,

### authentication, and real-time database. It can be used to store user data and app data. This stage involves integrating Firebase into the app to enable features such as user authentication and data storage. For example, the app may store user preferences and settings in Firebase.

1. **Integrate the app with Facebook community**: The app can be integrated with a Facebook community to enable users to

share information about disasters, such as photos, videos, and location data. This involves writing code to connect the app to

the Facebook API and to handle user data securely. For example, users may be able to share images of disasters they have

encountered and provide information about the location and severity of the disaster.

1. **Implement the app logic**: The app logic is implemented, which includes the algorithm for detecting disasters, as well as other features such as push notifications, alerts, and user settings. This stage involves writing code to handle different types of inputs from the camera and other sensors, and to trigger alerts and notifications based on the results of the machine learning model. For example, the app may notify users of nearby disasters and provide them with information on how to stay safe.
2. **Test the app**: The app is tested to ensure that it works correctly and reliably. Testing includes functional testing, performance testing, and user acceptance testing. This stage involves testing the app on different devices and in different scenarios to identify and fix any bugs or issues that may arise. The app should be tested to ensure that it accurately detects different types of disasters and provides users with relevant and useful information.

## Results of training machine learning algorithms on the gathered data :-

***Table 3 Results***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Fire** | **Water** | **Accident** | **Infrastructure** |
| **CNN** | 96% | 98% | 90% | 95% |
| **Random forest** | 86% | 83% | 84% | 72% |
| **KNN** | 78% | 75% | 78% | 79% |

# Chapter 6: Conclusion & Future Work

## Conclusion :-

* In conclusion, the mobile app using machine learning and camera technology for disaster detection can potentially be a valuable tool in disaster response and management. By leveraging the power of machine learning algorithms, such an app can help identify potential disasters and alert users to take necessary precautions or evacuate to safety.
* The app's ease of use and accessibility make it a valuable tool for anyone living in disaster-prone areas. With the ability to quickly detect and respond to disasters, communities can be better prepared and equipped to handle emergencies.

## Future work :-

* Synchronize app with different governmental states to facilitate for them how to quickly track and respond to different disasters.
* Synchronizing and Communicating with Hospitals, ambulance

..etc… in order to have a quick appearance in the place of action.

* Acting as a new community such as 911 in Egypt which is not applicable yet.
* Apply deep fake (deep learning technology) in our app
* Our goal is to create an app that everyone can use, including people with disabilities, by adding features that accommodate their needs.

# Appendix

### References

1. <https://www.javatpoint.com/uml-diagrams>
2. <https://www.tensorflow.org/> and <https://en.wikipedia.org/wiki/TensorFlow>
3. <https://flutter.dev/> and <https://en.wikipedia.org/wiki/Flutter_(software)>
4. <https://dart.dev/> and

<https://en.wikipedia.org/wiki/Dart_(programming_language)>

1. <https://firebase.google.com/>
2. <https://github.com/> and <https://en.wikipedia.org/wiki/GitHub>
3. <https://colab.research.google.com/?utm_source=scs-index>
4. <https://helpx.adobe.com/support/xd.html>
5. <https://www.kaggle.com/datasets/varpit94/disaster-images-dataset>
6. [https://www.researchgate.net/publication/330061064\_An\_evaluation\_fram](https://www.researchgate.net/publication/330061064_An_evaluation_framework_for_disaster_risk_management_in_Egypt/figures?lo=1) [ework\_for\_disaster\_risk\_management\_in\_Egypt/figures?lo=1](https://www.researchgate.net/publication/330061064_An_evaluation_framework_for_disaster_risk_management_in_Egypt/figures?lo=1)
7. [https://www.researchgate.net/publication/275676593\_The\_early\_warning\_](https://www.researchgate.net/publication/275676593_The_early_warning_application_role_in_facing_the_environmental_crisis_and_disasters_Preliminarily_risk_management_strategy_for_the_greater_city_of_Cairo/figures?lo=1) [application\_role\_in\_facing\_the\_environmental\_crisis\_and\_disasters\_Prelimi](https://www.researchgate.net/publication/275676593_The_early_warning_application_role_in_facing_the_environmental_crisis_and_disasters_Preliminarily_risk_management_strategy_for_the_greater_city_of_Cairo/figures?lo=1) [narily\_risk\_management\_strategy\_for\_the\_greater\_city\_of\_Cairo/figures?l](https://www.researchgate.net/publication/275676593_The_early_warning_application_role_in_facing_the_environmental_crisis_and_disasters_Preliminarily_risk_management_strategy_for_the_greater_city_of_Cairo/figures?lo=1) [o=1](https://www.researchgate.net/publication/275676593_The_early_warning_application_role_in_facing_the_environmental_crisis_and_disasters_Preliminarily_risk_management_strategy_for_the_greater_city_of_Cairo/figures?lo=1)
8. <https://github.com/AhmedAshraf9999/Balghny>