Applied Information Technology And Computer Science Vol. 4 No. 2 (2023) 2074 - 2091
© Universiti Tun Hussein Onn Malaysia Publisher's Office

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

Streamlining Disaster and Incident Reporting using MyAssist

Fadilah Daud[1], **Norfaradilla Wahid**[1]*

[1]Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

DOI: https://doi.org/10.30880/aitcs.2023.04.02.116
Received 24 June 2023; Accepted 27 October 2023; Available online 30 November 2023

Abstract: Technological approaches were made to prevent and prepare in any case of emergency. The main objective of the project is to design and develop a mobile application for improving disaster preparedness where it emphasizes location monitoring, hazard alert, and sending complaints related to disasters. Furthermore, this project analyses other literary works by comparing existing applications and conducting a survey to analyse the requirements from different sources. Information gathered from the surveys is used to construct the nature of the applications by forming structured diagrams and designing wireframes for interfaces. The project development followed the Agile principle while implementing the modules and functions in Java language using Android Studio IDE. By the end of this project, the outcomes of the application are tested to show the reliability of the application where the result of the test shows the application meets the user satisfaction.

Keywords: Disaster preparedness, Geolocation, Crowdsourcing

1 Introduction

Efforts were put into place from a technological standpoint to proactively address unforeseen emergencies, acting as a gateway for information to be distributed widely. To emphasize it clearly, disaster awareness serves the purpose of informing the public about initial assumptions related to the event. Thus, Information and Communication Technologies (ICTs) play an important role in disaster prevention, response, and recovery.

In this project, MyAssist application for disasters and incidents is proposed to be developed. This application will improve preparedness, mitigation, and response to disasters in Malaysia by deploying features that are surely helpful during catastrophic situations.

The existing disaster management system encounters several challenges, highlighting the need for the proposed MyAssist application as a solution. It starts with limited exposure to different incidents occurring in the surrounding area causing unpreparedness and heavy traffic. Next, the lack of knowledge contributes to poor decision-making and finally causes delayed response for emergency services to arrive. Furthermore, the difficulty faced by people evacuating by locating the critical

*Corresponding author: faradilla@uthm.edu.my
2023 UTHM Publisher. All rights reserved.

facilities and emergency shelters causes longer exposure to possible harm. A properly well-planned self-motivated evacuation can reduce casualties to a great extent [1].

The objectives of this project are to analyze and design a mobile application for improving disaster preparedness. Next, is to develop the mobile application using an object-oriented approach and followed by the testing implementation on the developed application to determine whether it satisfies the user's needs.

The study will be focusing on the development of a mobile application using an object-oriented approach to improve disaster management. This application emphasizes location monitoring, hazard alert, and sending complaints related to disasters. The application is an Android application, therefore requiring the user to have a working mobile phone and an internet connection to have access to all features. This application consists of two types of users, the administrators obliged with maintaining the application activity, and general users. Supplementarily, the main targeted users for this application would be people that are in a state of emergency or people that are cautious about the risks of possible danger.

This paper consists of four sections which start with the first chapter containing the introduction to the study. The introduction for this project offers a preliminary discussion by stating the purposes, goals, and planning for the project. Then, the second section is the literature review which displays the knowledge and comprehension of scholarly literature on the given issue. The third section, the methodology is section that outlines the steps taken to examine a study issue and provides valid reasons for the use of standard methods or techniques used to collect and interpret data to comprehend the project and combines the procedure of gathering and evaluating information, detecting issues, and breaking down a system into its constituent parts and the process of designing its components or modules to meet specific needs. The fourth section comprises the implementation of the stated module by programming the designed components and analysing the results from the organized test. The final section would be the conclusion. This section concludes the current work and outlines the key points for future work.

2. Literature Review

This section presents the survey on the existing research which could be related to this project which focuses on the topic of centralized crowdsourcing. Supplementarily, the technological approach for this project evolves on the use of a Global Positioning System. Furthermore, a study on the existing systems is conducted for comparisons to be made with the proposed application.

2.1 Disaster Management

Disaster management is the practice of preparing for and managing natural catastrophes or other incidents. It entails carefully coordinating resources to mitigate the damage caused by calamities. It also entails a systematic strategy for handling catastrophe mitigation, readiness, response, and restoration duties. Planning and preventive measures strive to reduce the likelihood of loss and distress caused by events. Each country has an agency founded specifically to devise and handle disaster planning, the National Disaster Management Strategy (NDMS) of Malaysia is the backbone strategy to promote effective coordination and an integrated approach to cultivating a culture of prevention, protection, and public safety in the community [2].

Although disaster response may not avoid disasters, it may save them from growing worse by ignoring caused elements and controllable hazards. The phase focuses on dispersing interventions and helping as much as possible to save lives, community property, and animals through first responders actively providing services. Information management plays a crucial role in disaster response operations involving strategic planning to ensure timely and effective prioritization and allocation of resources amongst discrete response activities [3].

2.2 Crowdsourcing

Crowdsourcing is the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from the online community rather than from traditional employees or suppliers. The act of crowdsourcing is widely used in technologies through social media since the platform allows its users to share different types of pieces of information. The concept of crowdsourcing implemented into disaster management was already being studied by previous researchers and is said as recognized by the relief organization [4].

2.3 Geolocation

This project implements the use of GPS through Geolocation API. API stands for Application Programming Interface, and it is a software mediator that allows two programs to communicate with one another. A Geolocation API is a communication interface that connects a client device or service (client-side) and a server-side application or service to determine and provide comprehension of the client's geographical location. The implementation of GPS allows geocoding and geofence to be used. Geocoding can provide further information about a place, such as the local time, currency, or region. Geofencing is the virtual border created around a given geographic place in order to provide customized communications.

2.4 Comparison between Existing Application and Proposed Application

A comparison is made between the proposed application and the existing applications based on unique features shown in Table 1. Three existing applications chosen are SaveME 999 [5], Noonlight [6], and In Case Emergency [7]. The unique features are such as the types of operating systems, geolocation, online connectivity, and others. In Table 1, the symbol "√" indicates the presence of the features while "X" marks the absence of the features. There are also features that could only be found in the proposed application while others do not. The features are community guidelines, both offline and resource locator, and shelter features.

Table 1: Comparison between existing applications with the proposed system

	SaveME 999	**Noonlight**	**In Case Emergency**	**MyAssist (PSM)**
Android operating system	√	√	√	√
Community Guidelines	X	X	X	√
Customizable Alert	X	X	√	√
Resource locator	X	X	*X*	√
Information Hub	X	X	√	√
Provide info on emergency numbers	√	X	X	√
Is available to be used in Malaysia	√	X	X	√
Shelter features	X	X	X	√
Crowdsourcing features	X	X	X	√
Geolocation features	√	√	X	√

3. Methodology

This section describes the approach that is specifically chosen for this project is the Agile Model. Agile methodology refers to a set of approaches and processes that encourage adaptable strategy,

developmental growth, and continuous integration. This section also includes the phases used in an Agile Model such as planning, design, development, testing, and deployment.

3.1 Agile Model

The agile model is used in this project to provide a changing environment for the project to adapt. Based on Figure 1, shows an iteration that includes planning, design, development, testing, and deployment. The outcome is sent to the user to be utilized and evaluated after each iteration. Any updates that improve the application are welcomed by the client at whichever stage of the process and are incorporated. The agile model reduces risks of development as the incremented mini software is delivered to the customers after every short development cycle and feedback is taken from the customers, it warns developers about the upcoming problems which may occur at the later stages of development [8].

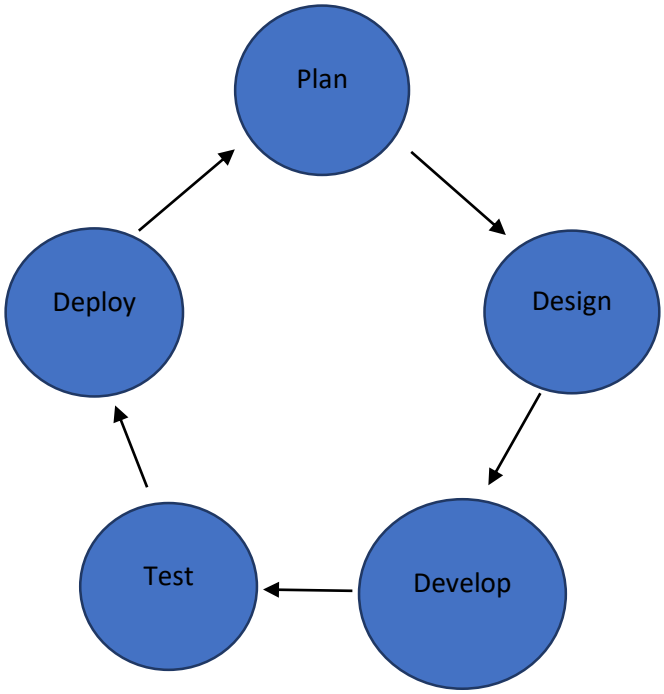


Figure 1: Example of an iteration from Agile Model [9]

3.1.1 Planning

The planning phase initializes the scope of the project by determining the possible solution to the existing problems and the documentation of the project schedule. This phase includes a great range of activities and tasks; however, they may be divided into three major categories. The first step is to create a Gantt Chart using Agile. The second step is to define all the project quality criteria within a simple summarization of the project proposal to help identify the key points of this project while the literature review asses the related studies and applications in comparison to the proposed application. Finally, analysing the expectations of the targeted users, a survey conducted through questionnaires is distributed. The result shows that the targeted users look forward to features the most such as live location updates and alerts for when there is a disaster.

3.1.2 Design

The outcomes from the analysis are then used for designing the proposed application to cater the requested features mentioned by the users. The design phase is a critical stage in the life cycle of the project. The goal of this phase is to transform the requirement specification into a structure or plan [10]. Therefore, in this phase database and user interfaces of the proposed application are designed. To avoid

destroying data in a database server, modifications must be meticulously planned. A Class Diagram is made to model the classes for passing the objects or data from the database. On the other hand, using wireframes is a great way of designing user interfaces digitally.

3.1.3 Develop

After the project was carefully designed, the proposed application is developed using Java as a programming language and Extensible Markup Language (XML) to script the design of the user interface. The development is performed with the Android Studio Integrated Development Environment (IDE) since it allows the software to be executed through a connected device or emulator to run the proposed application.

Then, the database chosen to relate to the proposed application is Firebase. By using Firebase, different types of databases are provided with different levels of security and purposes.

3.1.4 Testing

After the development of the proposed application, the application needs to undergo different types of testing to prove the non-functionality requirement stated is met and to test whether there are bugs that need to be fixed. Therefore, the application is to be tested with functional testing to focus on inspecting the output and dismissing the system's steps in the process. Moreover, the application undergoes the User Acceptance Test to measure user satisfaction when using the application.

3.2 System Development Workflow

The system development workflow is the summarization of the activities from each iteration and their deliverables. Based on Table 2 and Table 3 shows only two iterations which comprise all the phases and activities plan.

Table 2: Workflow of the project

Iteration	Phases	Activity	Deliverables
1	Planning	1. Establish a proper work plan. 2. Outline project quality criteria and identify project key points. 3. Analyse the expectations of targeted users.	1. Gantt Chart 2. Proposal 3. Literature Review 4. Questionnaire
	Design	1. Perform application analysis 2. Design class relationship	1. Use case diagram 2. Sequence diagram 3. Activity Diagram
	Develop	1. Develop a module.	1. Sign Up and Login
	Testing	1. Conduct a test on the module.	1. Test for database connection and usability of the module.
2	Planning	1. Analyse the limitation and outcomes.	1. System functionality and non-functionality requirement.
	Design	1. Design the wireframe.	1. Wireframe
	Develop	1. Develop the application prototype.	1. MyAssist prototype.

Table 3: (cont)

Iteration	Phases	Activity	Deliverables
2	Testing	1. Test for error and dysfunctionality in the application. 2. Conduct a test plan among users.	1. Improve the application. 2. Test for user overall satisfaction with the application.

3.3 Analysis and Design

Analysis and design done in this project to specify the application purposes and focuses on ways to accomplish the objectives of the application. Therefore, this section specifies the use case diagram, a sequence diagram, and a class diagram only due to the limitation of the paper where the data dictionary, system prototype design, and activity diagram could not be included.

3.3.1 Use Case

Use case diagrams (UCDs) describe the functionalities of a system in terms of how its various users achieve their goals by using the system [11]. It collects the system's requests, which comprise both internal and external inputs. It summons people, use cases, and a variety of other things that hold actors as well as elements responsible for the development of use case diagrams. The use case for the proposed application can be seen in Appendix A, which contains all the features connected to specific users like users and administrators. The features that can be accessed by users are login, sign up, view statistics and information, tracking hazards, and shelters, reading incidents, commenting on incidents, submitting new incidents, deleting incidents, personalized user profile, registering for shelters, requesting emergency supply, contact authority and get user support. Although the administrator can also read incidents, track hazards, and log in, they can also exclusively update on shelters and post notices.

3.3.2 Sequence Diagram

Sequence diagrams are used to illustrate system situations as a collection of specified events across multiple components of the system, denoted by lifelines. Interactions are defined by event requirements that are arranged into a structure and managed by engagement users.

Figure 2 displays the sequence diagram for tracking incidents where the user use searchLocation() method to specify the address on the tracker at the homepage. Then, the method getGeolocate() will search the location with API and retrieve the location saved in the database. As location and data is located, method postDetails containing the message in variables of locationName, incidentTag and desc will display it on the tracker page.

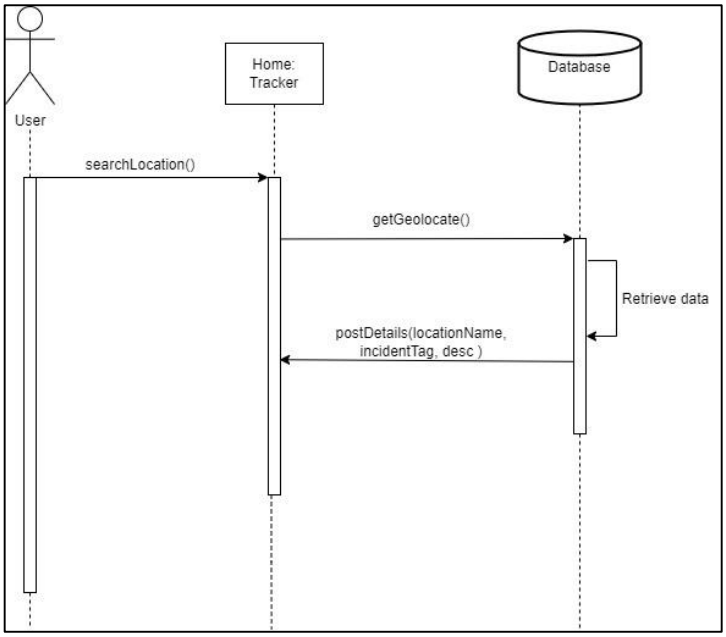


Figure 2: Sequence diagram for tracking

Here in Figure 3 shows the sequence diagram for posting the inicidents details in the proposed application. Users use the method selectPath() to choose features they want to take action with inside the Post page. If a user chose to submit a post, submitPost() will send all the details of the post to the database for saving. If a user chose to update a post, updatePost() will send the details to the database for saving. The method displayPost() will display the posts that users have made. Next, the method deletePost() is when the user opted to delete one of their posts where it will remove the data from the database. Then, if the user opted to report a post, reportPost() is the method that submits the request to the database.

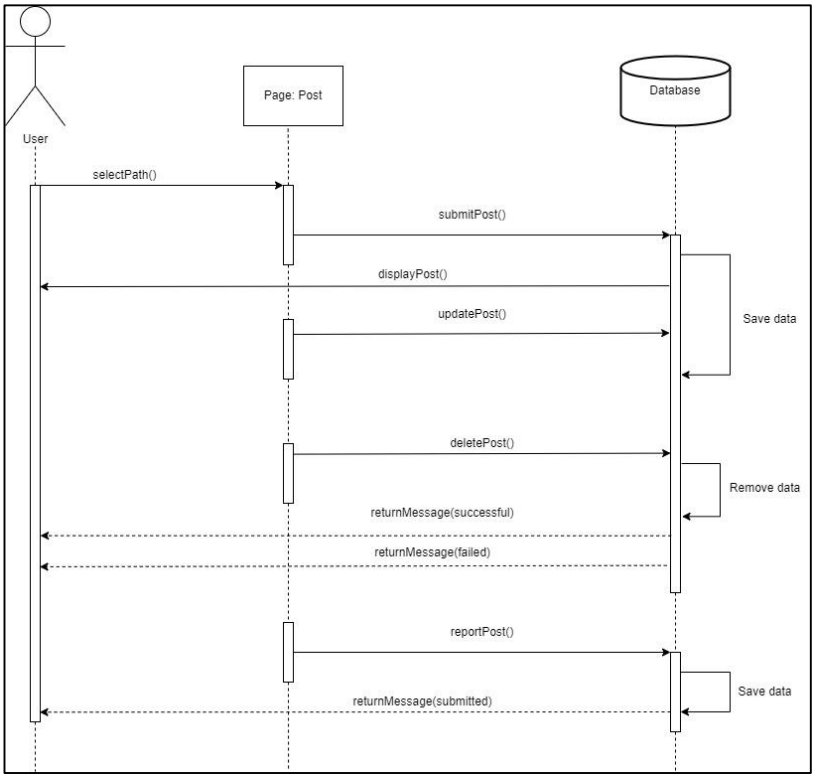


Figure 3: Sequence diagram for posting

3.3.3 Class Diagram

The class diagram is a structural diagram that depicts system attributes, processes, and relations within classes. It operates on the basis of object-oriented concepts. This orientation explains how entities interact with one another. The class diagram of the proposed application entails variables and methods in every class like User, Administrator, Hotline, Info, Tracker, Shelter, Register, Login, and Post. The class diagram can be seen in Appendix B.

4. Result and Discussion

This section presents the implementation of the system using the programming language selected which is Java and the user interface in the MyAssist application. Besides that, the result from the conducted test on User Acceptance Test and functional testing is further discussed to improve the outcomes of the project.

4.1 Implementation

Implementing the MyAssist application requires translating the stated specifications and requirements into actual codes through integrating libraries and frameworks. The application is developed using Android Studio which is an integrated development environment (IDE) fitting for an Android application. This IDE organizes the project structure for the design layout using XML and performs tasks of operation using Java programming language. Android Studio also incorporates Firebase which is a database service through plugins to handle the application's data.

```
hospital.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        StringBuilder stringBuilder = new StringBuilder("https://maps.googleapis.com/maps/api/place/nearbysearch/json?");
        stringBuilder.append("location=" + lat+"," +lng);
        stringBuilder.append("&radius=1000");
        stringBuilder.append("&type=hospital");
        stringBuilder.append("&sensor=true");
        stringBuilder.append("&key=" +"AIzaSyCeuhYBrLcC3R1FP6_J3g4WomPRhOqKbGQ");
        String url = stringBuilder.toString();
        Object dataFetch[]=new Object[2];
        dataFetch[0]=mMap;
        dataFetch[1]=url;

        FetchData fetchData=new FetchData();
        fetchData.execute(dataFetch);
    }
});
```

Figure 4: Utilizing Google Maps API in Android Studio

Figure 4 depicts the code segment where an asynchronous job (FetchData) is launched that retrieves data from the Google Places API based on the supplied location using latitude and longitude, radius, and type (hospital). The retrieved information is most usually used for presenting or interpreting data about neighboring hospitals on a map (mMap). Therefore, this function allow when button hospital is clicked, the nearest hospital 1000 meters to user current location will be shown. This function applied to shelters and police.

```
public void getIncidentLocation(String incidentAddress, String incidentType) {
    Geocoder geocoder = new Geocoder(MainActivity.this);

        List<Address> addressList = null;
        try {
            addressList = geocoder.getFromLocationName(incidentAddress,  maxResults: 1);
            if (addressList == null){
                Toast.makeText( context: MainActivity.this,  text: "Data is null in geocoder", Toast.LENGTH_LONG).show();
            }
            if (addressList != null && addressList.size()>0) {
                Address targetLocate = addressList.get(0);
                String iress = addressList.get(0).getAddressLine( index: 0);
                LatLng latLng = new LatLng(targetLocate.getLatitude(), targetLocate.getLongitude());
                if(incidentType.equals("Flood")){
                    addMarkerpurple(latLng);
                    addCircle(latLng, GEOFENCERADIUS);
                }
                if(incidentType.equals("Landslide")){...}
                if (incidentType.equals("Accident")){
                    addMarkerblue(latLng);
                    addCircle(latLng, GEOFENCERADIUS);
                }
                if(incidentType.equals("Fire")){
                    addMarkerPink(latLng);
                    addCircle(latLng,GEOFENCERADIUS);
                }

                mMap.setOnMarkerClickListener(new GoogleMap.OnMarkerClickListener() {
                    @Override
                    public boolean onMarkerClick(Marker marker) {
                        onMarkerClicked(marker,incidentAddress,incidentType, iress);
                        return false;
                    }
                });

            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
```

Figure 5: Processing marker and geofence on incidents

In addition to the tracker module, 'getIncidentLocation()' as shown in Figure 5 is a method passing the data like address and type of incidents from Firebase where incidents detail were stored. This method also processes the geocoding of the address into longitude and latitude for mapping purposes. Moreover, this method calls the geofencing functions where each type of incident gets a different virtual mapped radius.

Therefore, calling this function returns the mapping for the incidents in real-time as well as the geofence for when users appear in the virtual radius, notification will be sent to notify them. The result for the tracker module is shown in Figure 6 where a virtual radius is built around the location of the incident marking it as a sensitive area that a user should avoid at best. The interface contains other methods implemented in the module such as searching for location, locating nearby resources like hospital, police and shelters.

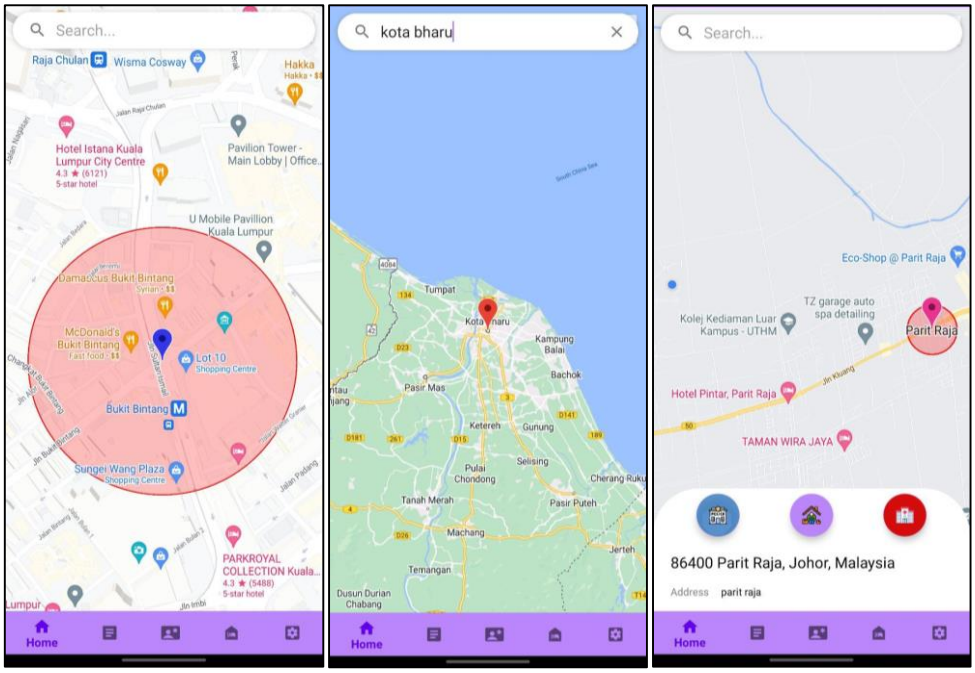


Figure 6: Tracker module interface

The data used for each marker is collected through users collectively performing the posting on the incidents. The data were uploaded into Firebase through 'uploadData()' function as depicted in Figure 7 where the Firebase connection was initialized to add all the necessary details of incidents given by users. Figure 8 shows the interface used for users to fill in the fields that best describe incidents in their posts and the results for each posting to be available for all users to read and perform other activities like commenting on the post, delete or report on the post.

```
private void uploadData(final String titl, final String description, final String tag, final String inciAddr) {
    // show the progress dialog box
    pd.setMessage("Publishing Post");
    pd.show();
    final String timestamp = String.valueOf(System.currentTimeMillis());
    String filepathname = "Posts/" + "post" + timestamp;
    Bitmap bitmap = ((BitmapDrawable) image.getDrawable()).getBitmap();
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.PNG, quality: 100, byteArrayOutputStream);
    byte[] data = byteArrayOutputStream.toByteArray();

    StorageReference storageReference1 = FirebaseStorage.getInstance().getReference().child(filepathname);
    storageReference1.putBytes(data).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

            Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
            while (!uriTask.isSuccessful()) ;
            String downloadUri = uriTask.getResult().toString();
            if (uriTask.isSuccessful()) {
                // if task is successful the update the data into firebase
                HashMap<Object, String> hashMap = new HashMap<>();
                hashMap.put("uid", uid);
                hashMap.put("uname", name);
                hashMap.put("uemail", email);
                hashMap.put("udp", dp);
                hashMap.put("incidentType",tag);
                hashMap.put("title", titl);
                hashMap.put("description", description);
                hashMap.put("incidentAddress", inciAddr);
                hashMap.put("uimage", downloadUri);
                hashMap.put("ptime", timestamp);
                hashMap.put("pcomments", "0");

                DatabaseReference databaseReference = FirebaseDatabase.getInstance( url: "https://fyp2-e811b-default-rtdb.asia-southeast1.firebasedatabase.app/").getReference( path: "Posts");
```

Figure 7: Tracker module interface

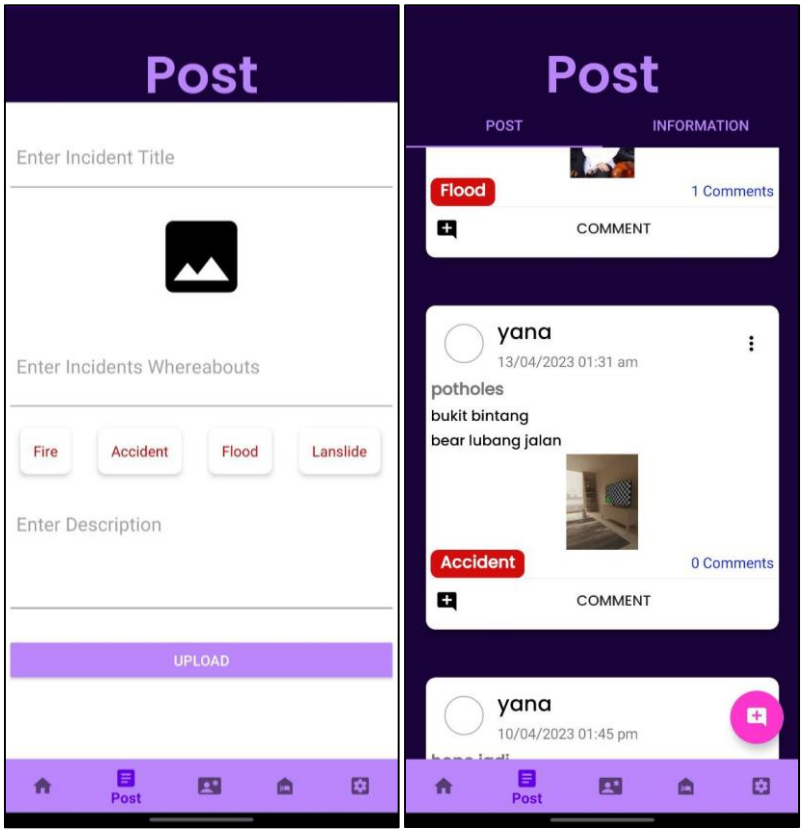


Figure 8: Incident reporting module interface

MyAssist application also implements a module for shelters where relocation sites were presented to users to register for refugees or see the requests made by each shelter for emergency supplies. Here in Figure 9 shows the interface for the shelter module where only the currently opened evacuation sites were presented, and the form needed to be filled in by users who seek refugees in the selected shelters. To prevent unnecessary data in database, user id was traced before each register as shown in Figure 10.

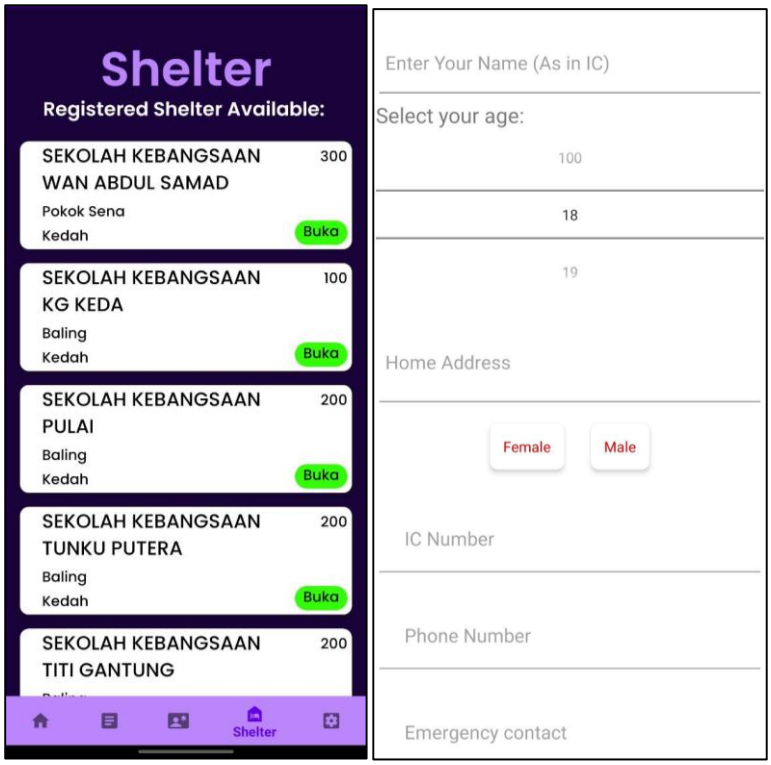


Figure 9: Shelter module interface

```
upload.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String ecmfon = "" + emcphone.getText().toString().trim();
        String icS = "" + ic.getText().toString().trim();
        String medics = ""+mediacal.getText().toString().trim();
        int tag = radioGroup.getCheckedRadioButtonId();
        int selectedAge = agePicker.getValue();
        radBut = findViewById(tag);
        gender = radBut.getText().toString();

        DatabaseReference bt = FirebaseDatabase.getInstance( url: "https://fyp2-e811b-default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference( path: "1mdpdQYSlxe7ITmXvHo0dPgTjGu109_skhSdP8gRGRbo").child( pathString: "Worksheet").child(bilRel).child( pathString: "residents");
        Query q = bt.orderByChild( path: "userid").equalTo(firebaseUser.getUid());
        q.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    Log.d( tag: "TAG", firebaseUser.getUid());
                    pd.dismiss();
                    Toast.makeText( context: RegisterShelter.this, text: "User already exists", Toast.LENGTH_LONG).show();
                } else{
                    uploadData(ecmfon, icS, gender, medics, selectedAge);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
```

Figure 10: Code segment to prevent registered user to re-register in selected shelter

4.2 Functional Testing

The purpose of the functional testing process is to uncover any functional faults or difficulties that may affect MyAssist's usability, dependability, or performance in terms of exception handling, the user

interface of MyAssist, and the interoperability of the Firebase database. Hence, a test plan is conducted as shown in Table 4 to assure the application's quality and the result is successfully passed.

Table 4: Test plan for application functionalities

Function	Test Case	Expected Output	Actual Output
Sign-in	Incomplete input	Display alert message of which text field found empty	Pass
	Password did not meet requirements	Display alert message of insecure password	Pass
	False pattern of inputs	Display alert message showing the right pattern of input	Pass
	Existing account	Display alert message on existing user	Pass
	Complete registration	User successfully registers an account and redirect to the login page	Pass
Login	Username did not match with password	Alert error message on incorrect password or username	Pass
	Incomplete input	Display alert message of which text field found empty	Pass
	Complete login	User successfully logs in and redirect to the home page	Pass
Recover password	Send email to user after user submits the email input	User received an email on changing the new password from the Firebase provider	Pass
	User can use the new password after resetting the password	New password can be used to log into the account	Pass
	New user using forgot password	Display error on no existing user with such an email	Pass
User profile	Update existing data	Updated data is successfully saved into the database every time the user updates	Pass
Add post	Incomplete input	Display alert message of which text field found empty	Pass
	Complete create post	Post successfully added to the post timeline and added marker to map	Pass
Registration for refuge	Incomplete input	Display alert message of which text field found empty	Pass
	User register despite already being registered in the shelter	Invalid error message indicating user already registered	Pass
	Complete registration for refuge	User successfully registered and redirect back to shelter page where request button is now available	Pass

4.3 User Acceptance Test

User Acceptance Testing (UAT) is a critical phase in the process of developing software in which users assess the software or system to see if it satisfies their needs and is suitable for launch. The UAT

for MyAssist application involved 5 individuals volunteering as end users from different age groups where they evaluate the application based on the assessment.

The data collected in Figure 11 and Figure 12 by using a Google Form answered from a different age group of people where more than half of the participants for UAT are mainly from the age group of 18 to 29. The data collected depicts that all the users are satisfied with user interface and functional features from MyAssist application. Furthermore, the data shows that user in the age group over 50 is highly satisfied with the user interface of MyAssist application. This data validates the engagement from a specific age group that targeted to be hardly using any form of mobile application due to aging and lack of technological exposure.

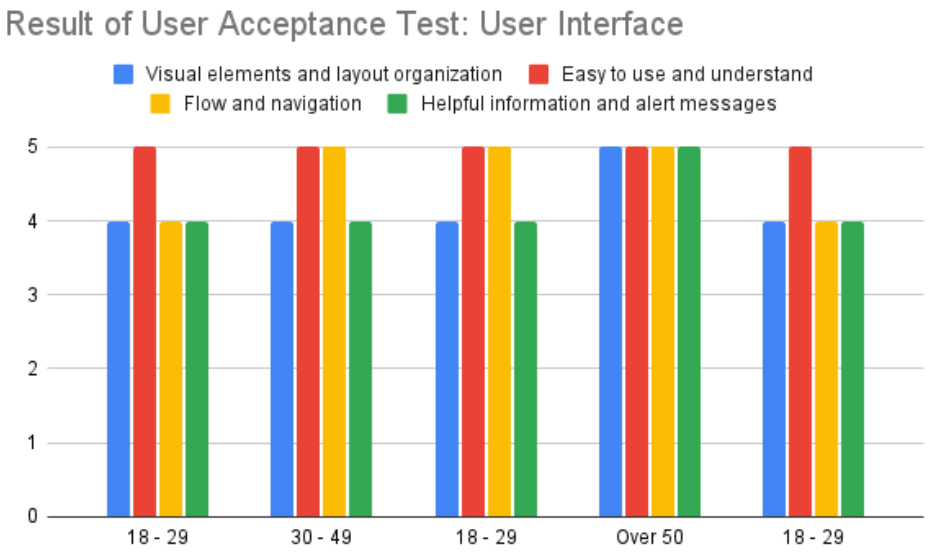


Figure 11: UAT result for user interface

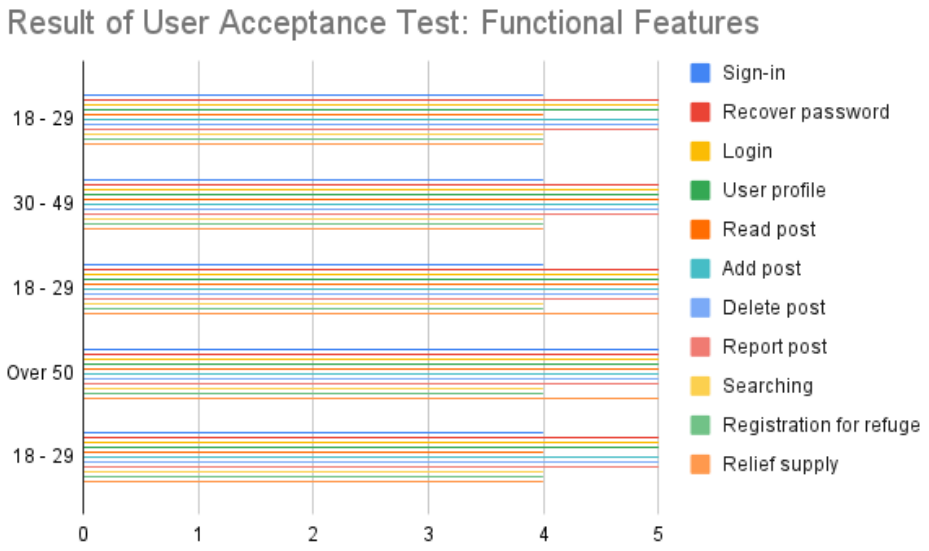


Figure 12: UAT result for functional features

5. Conclusion

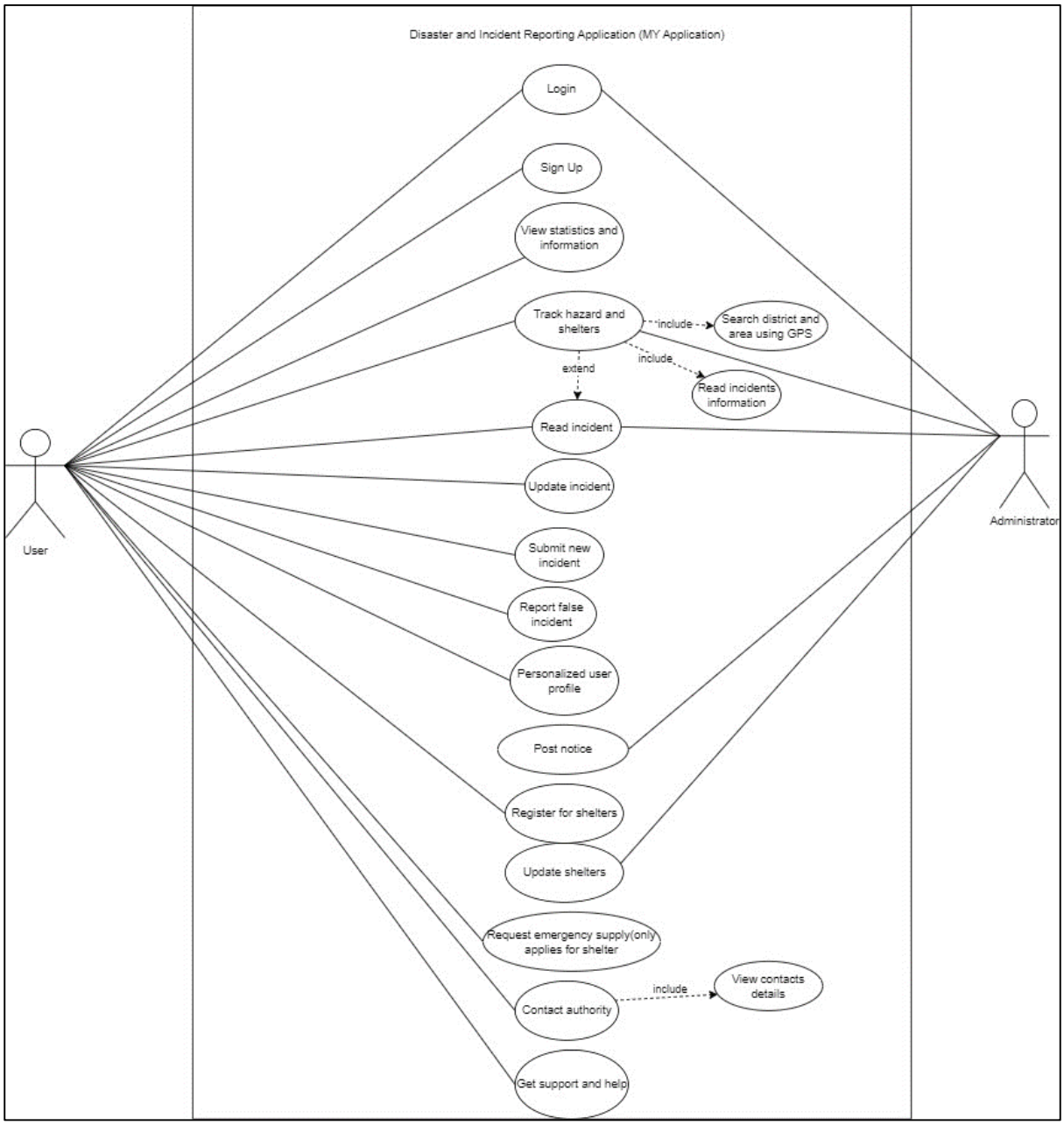
MyAssist accomplished its success through the development of a complete platform that covers communication demands during accidents and disasters. It delivers real-time information, allows for updates and dialogues, and includes useful features such as shelter data and incident monitoring on a map. Based on the test conducted, users have given the program favorable responses, recognizing its straightforward UI, ease, and overall experience with it through evaluation by rating it as satisfaction. The addition of a shelter feature increases its usefulness by aiding users in locating appropriate shelter choices for their safety and security amid emergencies.

It is critical to consider future proposals for additional enhancement in order to realize the intriguing possibilities of the MyAssist application. It is essential to strengthen privacy and security measures to secure user information and maintain confidentiality. Scalability issues linked to the geolocation function and data processes should also be addressed to reduce possible latency and delays. It is strongly advised that the program interacts with emergency help providers and interfaces with current emergency systems for increased functionality, enabling real-time data transfer, rapid interaction, and efficient event coordination. Furthermore, the employment of sophisticated algorithms or crowdsourcing approaches may substantially aid in the verification and filtering of user-reported events, assuring the quality and dependability of the data supplied. By taking these future recommendations into account, the MyAssist application can reach new heights of effectiveness and user satisfaction.

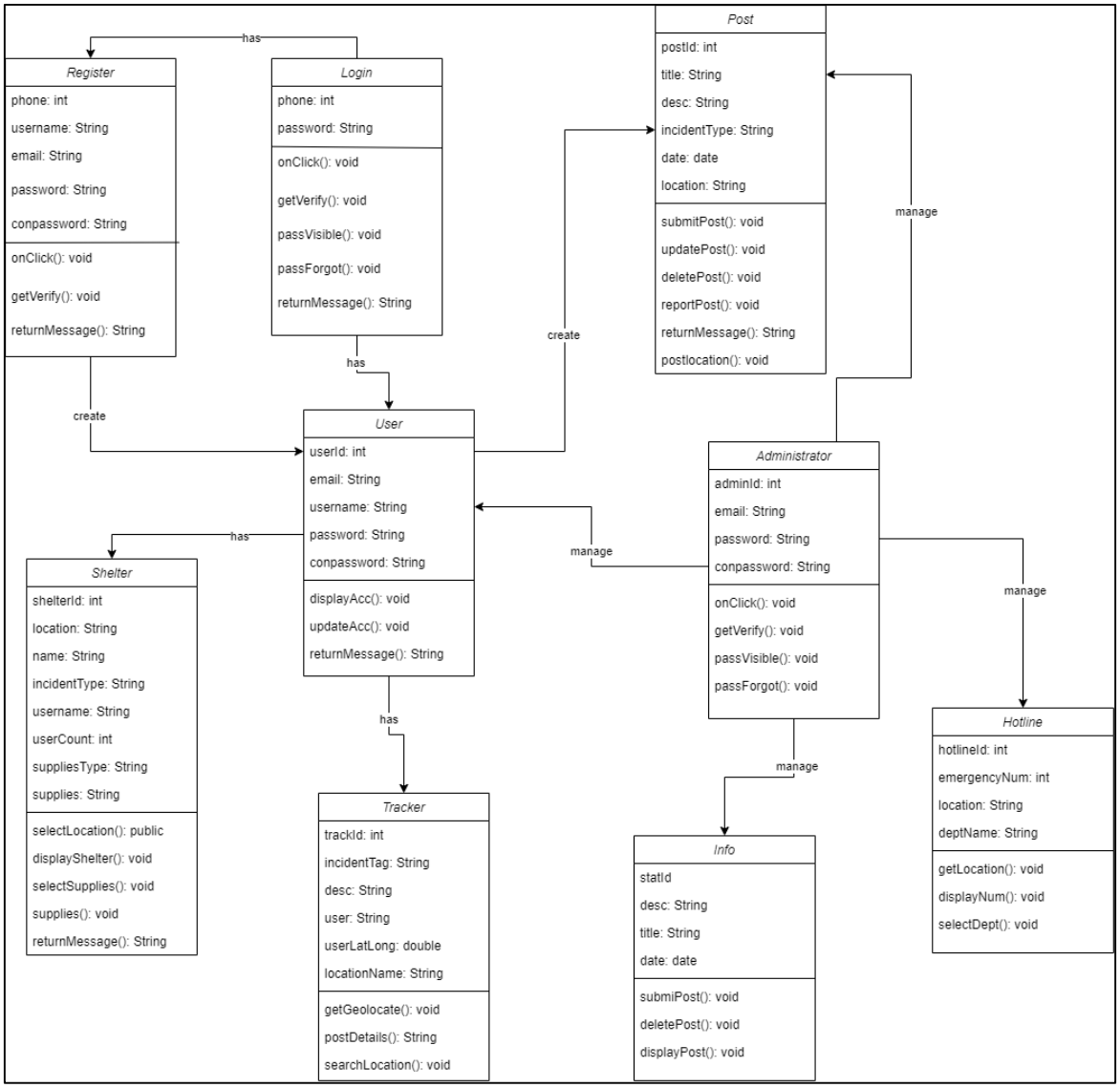
Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix A (Use Case for The Proposed Application)



Appendix B (Class Diagram for The Proposed Application)



References

[1] Y. Yao, Y. Zhang, Y. Taoyu, K. Wong, J. Y. Tsou, and Y. Zhang, "ISPRS International Journal of Geo-Information," Directory of open access journals, February 2021. [Online]. Available: https://doaj.org/article/231220b572594e6e8a82c2818fc8f522. [Accessed 19 October, 2022].

[2] Center for Excellence in Disaster Management and Humanitarian Assistance, Malaysia: Disaster management reference handbook, 2016. [E-book].

[3] Chaudhuri, N. Bose and Indranil, "Application of image analytics for disaster response in Smart Cities," in Proceedings of the Annual Hawaii International Conference on System Sciences, 2019, doi: 10.24251/hicss.2019.367.

[4] D. Auferbauer, and H. Tellioğlu, "Centralized crowdsourcing in Disaster Management," in Proceedings of the 8th International Conference on Communities and Technologies, pp. 173-182, June 2017, doi: 10.1145/3083671.3083689.

[5] (2022). Saveme 999 [Mobile app]. Retrieved from Saveme 999 polis, MERS 999. https://999.gov.my/saveme-999/saveme-999-polis/.

[6] (2022). Noonlight [Mobile app]. Retrieved from America's no. 1 safety app. https://www.noonlight.com/noonlight-app.

[7] (2022). ICE - in case of emergency [Mobile app]. Retrieved from Google Play Store. https://play.google.com/store/apps/details?id=com.lagache.sylvain.ice_android&hl=en&gl=US.

[8] S. Sharma, D. Sarkar, and D. Gupta, "Agile processes and methodologies: A conceptual study," May 2012. [Online]. Available: https://www.yashada.org/yash/egovcii/static_pgs/TC/IJCSE12-04-05-186.pdf. [Accessed Dec. 1, 2022].

[9] Aha!, "What is Agile Software Development? What Is Agile Software Development?", May 2022. [Online]. Available: https://www.aha.io/roadmapping/guide/agile/agile-software-development. [Accessed December 1, 2022].

[10] S. K. Dora, and P, Dubey, "Software development life cycle (SDLC) Analytical Comparison and Survey On Traditional And Agile Methodology," August 2013. [Online]. Available: https://www.researchgate.net/publication/319716548_Software_Development_Life_Cycle_Sdlc_Analytical_Comparison_And_Survey_On_Traditional_And_Agile_Methodology. [Accessed December 1, 2022].

[11] S. Friedenthal, A. Moore, and R. Steiner, A Practical Guide to SysML, Third Edition: The Systems ModelingLanguage (3rd ed.), United States: Morgan Kaufmann Publishers Inc, 2014.