DESIGN AND IMPLEMENTATION OF HEALTH-BASED EMERGENCY REPORTING MOBILE APPLICATION

**SHOEWU O.O, AYANGBEKUN O.J., LAMINA I.S.**

Department of Electronics and Computer Engineering, Lagos State University, Epe Campus, Nigeria. 

Department of Computer Science, Redeemer’s College of Technology & Management, Ogun State. Nigeria

Received: 11th November 2020
Accepted: 25th January 2021
Published: 24th March 2021

**ABSTRACT**

Every human being in distress or in health emergency cases will try to contact any available emergency and rescue services to get help from rescue services using their phone to call for help. However, the constraints of using the conventional method exits when they are in shock and they cannot communicate effectively to explain their position and condition. Thus, mobile applications are developed due to the fact that it can help people who are in emergency cases. This paper provides an application that can help any individual to immediately contact the direct emergency rescue unit of the health center. The application automatically adds the individual online and extract the user details embedded in their SIM and also provides message alert to the contact for urgent medical aids. This is built on a cross platform mobile app that runs on both android and iOS.

**Keywords:** Android, Application, Emergency, Report, Health, Mobile

**1. INTRODUCTION**

Emergency-based health response is a mobile application that is used to help individuals when they are in a critical circumstance. For instance, when a person is in a state of a health emergency, during the extraordinary circumstance, they can simply click one button from the mobile phone and the individual’s information will be sent as an emergency message to the health Centre system. The individual data of the user, the location of the user current area and the emergency message selected by the client will be included in the message to be sent. What's more, the application will assist the user in informing somebody when the emergency function is activated. Besides, the applications will assist the user to learn about certain health concepts like a disease. Along these lines, this is how the application work to help individuals.

Also, the possibility of the emergency-based health mobile application that we build in this research is brisk, simple and viable. In this way, we choose to set just one quick launch button to let the user click. For instance, if the user needs to use the application, they simply need to click one button to activate the major capacity in the application. This will make the user feel less weight when they have to use the application since quick and snappy is the most significant thing during an emergency circumstance. The fundamental idea of the mobile application that we build in this project will be that when the user launches the application for the first time, the application will spring up a screen and request that user input some personal data. The individual data will incorporate the name of the user, date of birth, gender, email address, personal contact number, address, and some personal message. Also, the application will require some emergency contact number from the user. The contact number can be the relative of the user or even the friend of the user. The application will send all the data or even call emergency contact during the emergency circumstance. To trigger the emergency function, the user will launch the app and click the emergency button on the mobile phone.

**Background Study**

These days, the smartphone is one of the electronic devices that individuals cannot lack, this is on the grounds that smartphone has many types of application that can help individuals complete a ton of errand. A mobile application is a software or program that is built by a software developer to take care of a problem or engaging people. In the present market, a mobile application can be separated into three kinds; native applications, mobile web applications, and hybrid applications.[1] Native application is a mobile application that can permit user access through the icon, it can be downloaded from the website or mobile applications store. It can work when offline. Mobile web application is not exactly a mobile
application, this is because it needs a browser to run and it is like a webpage or website, more often than not users are accessible through the bookmark of the webpage or website. Hybrid application is a mobile application that contains the piece of native application and part of mobile web application, this is because the mobile application is running on the device and it is written in web technology, for example, HTML5, JavaScript and so forth.

Furthermore, concurring the description of the kinds of mobile application above, the type that we focus on developing an emergency-based health response mobile application in this research is native applications. The reason for choosing native type of application is because native application can get to all the elements in the device itself and the application can work with or without internet connection. To develop this application, we have to get to the GPS location system in the smartphone. The GPS location system is used to get user’s location. Also, this application will get to the message application and phone call application in the mobile phone, in order to send message and call the particular emergency contact that is being set by the user. Along these lines, native application is the suitable type in developing an emergency mobile application for this research.

2. LITERATURE REVIEW

Today’s mobile cross-platform development is full of compromise. Developers are compelled to choose either to develop the same app numerous times for various operating systems or to embrace the lowest common denominator solution that handles native speed and portability accuracy. The cross platform mobile development framework that will be considered for this research is Flutter (An Open-Source SDK by Google).

Android relies on Linux, an open-source mobile software package created by the Google-led Open handset Alliance to make android device app to get started. Collection of tools is being used and enclosed within the android SDK. The android Open source Project (AOSP) presently runs the maintenance and development cycle for android.[5]

The table below indicates android API's timetable between 2008 and 2019. Androids keep change their software from android one in Nov 2007 till the foremost recent version of android update to Android 10.0 (Android Q) as of 3rd Sept 2019 is published (Table 1).

Android developed Google Play, which is a digital application distribution platform. Google Play was developed on October 23, 2008, and is known as Android Market. In Google Play, it has several parts such as Play Books, Play Magazines, Play Movies & TV and Applications. Meanwhile, Android Resource Manager provide access to non-code resources; a Notification Manager that enables all applications to display custom alerts in the status bar and an Activity Manager that manages the life cycle of applications.[6]

The iOS operating system manages the device hardware and provides the technologies needed to implement native apps. The operating system conjointly ships with numerous system apps, like Phone, Mail, and Safari, that give standard system services to the user. Native apps are engineered using the iOS system frameworks and Objective-C language and run directly on iOS. not like web apps, native apps are installed physically on a device and are so always accessible to the user, even when the device is in flight mode.[7]

The Table 2 shows the versions of iOS from February 2010 to September 2019.

**Cross-platform**

In this research, the cross-platform technology is being considered amongst various cross-platform mobile technologies is the flutter. Flutter is an open-source, cross-platform mobile development framework from Google. It allows high-performance, beautiful applications to be built for iOS and Android from a single code base. It is also the development platform for Google’s upcoming Fuchsia operating system. Additionally, it is architected in a way that it can be brought to other platforms (Fig. 1), via custom Flutter engine embedders. [8]

Flutter is distinct from most totally different mobile app development alternatives because it takes a different approach in an attempt to make mobile development better. It provides a framework application developer work against and an engine with a portable runtime to host applications. The framework builds upon the Skia graphics library, providing widgets that are actually rendered, as opposed to being just wrappers on native controls. This approach gives the flexibility to build a cross-platform application in a completely custom manner like the web wrapper option provides, but at the same time offering smooth performance. Meanwhile, the rich widget library that comes with Flutter, along with a wealth of open-source widgets, makes it a very feature-rich platform.
to work with. Put simply, Flutter is the closest thing mobile developers have had for cross-platform development with little to no compromise.

Table 1: Android Versions (https://en.wikipedia.org/wiki)

<table>
<thead>
<tr>
<th>Code name</th>
<th>Version number</th>
<th>Linux kernel version</th>
<th>Initial release date</th>
<th>API level</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Codename</td>
<td>1.0</td>
<td>2.1</td>
<td>September 23, 2008</td>
<td>1</td>
</tr>
<tr>
<td>Petit Four (internal)</td>
<td>1.1</td>
<td>2.6</td>
<td>February 9, 2009</td>
<td>2</td>
</tr>
<tr>
<td>Cupcake</td>
<td>1.5</td>
<td>2.6.27</td>
<td>April 27, 2009</td>
<td>3</td>
</tr>
<tr>
<td>Donut</td>
<td>1.6</td>
<td>2.6.29</td>
<td>September 15, 2009</td>
<td>4</td>
</tr>
<tr>
<td>Éclair</td>
<td>2.0 – 2.1</td>
<td>2.6.29</td>
<td>October 26, 2009</td>
<td>5 – 7</td>
</tr>
<tr>
<td>Froyo</td>
<td>2.2 – 2.2.3</td>
<td>2.6.32</td>
<td>May 20, 2010</td>
<td>8</td>
</tr>
<tr>
<td>Gingerbread</td>
<td>2.3 – 2.3.7</td>
<td>2.6.35</td>
<td>December 6, 2010</td>
<td>9 – 10</td>
</tr>
<tr>
<td>Honeycomb</td>
<td>3.0 – 3.2.6</td>
<td>2.6.36</td>
<td>February 22, 2011</td>
<td>11 – 13</td>
</tr>
<tr>
<td>Ice Cream Sandwich</td>
<td>4.0 – 4.0.4</td>
<td>3.0.1</td>
<td>October 18, 2011</td>
<td>14 – 15</td>
</tr>
<tr>
<td>Jelly Bean</td>
<td>4.1 – 4.3.1</td>
<td>3.0.31 - 3.4.39</td>
<td>July 9, 2012</td>
<td>16 – 18</td>
</tr>
<tr>
<td>KitKat</td>
<td>4.4 – 4.4.4</td>
<td>3.10</td>
<td>October 31, 2013</td>
<td>19 – 20</td>
</tr>
<tr>
<td>Lollipop</td>
<td>5.0 – 5.1.1</td>
<td>3.16</td>
<td>November 12, 2014</td>
<td>21 – 22</td>
</tr>
<tr>
<td>Marshmallow</td>
<td>6.0 – 6.0.1</td>
<td>3.18</td>
<td>October 5, 2015</td>
<td>23</td>
</tr>
<tr>
<td>Nougat</td>
<td>7.0 – 7.1.2</td>
<td>4.4</td>
<td>August 22, 2016</td>
<td>24 – 25</td>
</tr>
<tr>
<td>Oreo</td>
<td>8.0 – 8.1</td>
<td>4.10</td>
<td>August 21, 2017</td>
<td>26 – 27</td>
</tr>
<tr>
<td>Pie</td>
<td>9.0</td>
<td>4.4.107, 4.9.84, and 4.14.42</td>
<td>August 6, 2018</td>
<td>28</td>
</tr>
<tr>
<td>Android 10 Queen Cake</td>
<td>10.0</td>
<td></td>
<td>September 3, 2019</td>
<td>29</td>
</tr>
</tbody>
</table>

Legend: | Old version | Older version, still supported | Latest version | Latest preview version |

One big advantage native application have is they can adopt brand new technologies Apple and Google create in beta immediately if desired, without having to wait for any third-party integration. The main disadvantage to building native applications is the lack of code reuse across platforms, which can make development expensive if targeting iOS and Android. Flutter applications are written in Dart, which is a programming language originally developed by Google. Dart is an object-oriented language that supports both ahead-of-time and just-in-time compilation, making it well suited for building native applications, while providing for efficient development workflow with Flutter’s hot reloading. Flutter recently moved to Dart version 2.0 as well.[9][8]
Table 2: iOS Versions [https://en.wikipedia.org/wiki/iOS_version_history]

<table>
<thead>
<tr>
<th>Version</th>
<th>Build</th>
<th>Processor support</th>
<th>Application support</th>
<th>Kernel</th>
<th>Release date</th>
<th>Device end-of-life</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.3</td>
<td>7E18</td>
<td>32-bit ARM</td>
<td></td>
<td></td>
<td>February 2, 2010</td>
<td>iPad N/A, iPhone 1st gen, iPod 1</td>
</tr>
<tr>
<td>4.2.1</td>
<td>8C148</td>
<td></td>
<td></td>
<td></td>
<td>November 22, 2010</td>
<td>3G 2</td>
</tr>
<tr>
<td>5.1.1</td>
<td>9B206</td>
<td></td>
<td></td>
<td></td>
<td>May 7, 2012</td>
<td>1st gen N/A 3</td>
</tr>
<tr>
<td>6.1.6</td>
<td>10B500</td>
<td></td>
<td></td>
<td></td>
<td>February 21, 2014</td>
<td>iPad N/A, iPhone 3GS 4</td>
</tr>
<tr>
<td>7.1.2</td>
<td>11D257</td>
<td></td>
<td></td>
<td></td>
<td>June 30, 2014</td>
<td>iPhone 4 N/A</td>
</tr>
<tr>
<td>9.3.5</td>
<td>13G36</td>
<td>32/64-bit ARM</td>
<td></td>
<td></td>
<td>August 25, 2016</td>
<td>2, 3, Mini N/A 5</td>
</tr>
<tr>
<td>9.3.6</td>
<td>13G37</td>
<td></td>
<td></td>
<td></td>
<td>July 22, 2019</td>
<td>4S N/A</td>
</tr>
<tr>
<td>10.3.3</td>
<td>14G60</td>
<td></td>
<td></td>
<td></td>
<td>July 19, 2017</td>
<td>4 5C N/A</td>
</tr>
<tr>
<td>10.3.4</td>
<td>14G61</td>
<td></td>
<td></td>
<td></td>
<td>July 22, 2019</td>
<td>5 N/A</td>
</tr>
<tr>
<td>12.4.1</td>
<td>16G102</td>
<td>64-bit ARM</td>
<td></td>
<td></td>
<td>August 26, 2019</td>
<td>Air, Mini 2, Mini 3 5S, 6 6</td>
</tr>
<tr>
<td>13.0 Beta 8</td>
<td>17A5572a</td>
<td></td>
<td></td>
<td></td>
<td>August 21, 2019</td>
<td>N/A</td>
</tr>
<tr>
<td>13.1 Beta 2</td>
<td>17A5831c</td>
<td></td>
<td></td>
<td></td>
<td>September 4, 2019</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Legend: | Discontinued | Current | Beta

With Flutter, we tend to believe we have an answer that offers you the most effective of each world: hardware-accelerated graphics and UI, powered by native ARM code, targeting each standard mobile operating system.

Development Environment

Flutter offers incredible flexibility with regard to tooling. Applications can just as easily be developed from the command line along with any editor, as they can be from a supported IDE like Visual Studio Code, Android Studio, or IntelliJ. The approach to take depends largely on developer preference. The flutter framework consists of some tools that can enable the developer’s productivity.

Android Studio offers the most features, such as a Flutter Inspector to analyze the widgets of a running application and well as monitor application performance. It also offers several refactoring that are convenient when developing a widget hierarchy.
VS Code offers a lighter development experience in that it tends to start faster than Android Studio or IntelliJ. Each IDE offers built-in editing helpers, such as code completion, allowing exploration of various APIs as well as good debugging support. The command line is also well supported through the flutter command, which makes it easy to create, update and launch an application without any other tooling dependency beyond an editor.

3. METHODOLOGY

The design phase cut across different platforms - Android and iOS. To realize this, cross-platform development tool (FLUTTER) framework is being employed. This provides ability to have access to native APIs of the different platforms, which is not usually possible through browsers. Also, the developer does not need to learn about different programming languages and tools to develop a single application for each platform.[8]

3.1 Software Development Model

Iterative and incremental development methodology is used for the development phase to produce a good quality and reliable system as well as to increase the functionality of the system [3], [4].

In iterative approach, every time the developer review or revisit the system, some modifications or improvements will be made in that iteration. Moreover, the developer can get users feedback in order to make quality improvement in the system. In incremental approach on the other hand, is able to develop and analyze the system several times at different modules, new feature of the system will be added incrementally until the whole system is completely developed. Besides, the system design, testing and implementation will also be performed incrementally until the system is ready. At the end of the development phase, the developer check the system completion and make sure it fulfils the user requirement.

3.2 System Requirements

**Functional Requirements**: The User Interface is designed to make it very quick to send an alert for emergency: to get help by sending location and information to the health Centre as being set by the user

- User should be able to add to the personal information to the system.
- User should be able to add emergency contact to the system.
- User should be able to add emergency message to the system.
- User should be able to login to the system.
- User should be able to perform change on personal information.
- User should be able to trigger the emergency function by clicking the button

**Non - Functional Requirements**: User-friendly interface for the application

- English as the language used by the application.
- The application will run on a mobile phone. Memory and storage capacity, battery life, ability to adapt to difference screen sizes and orientations of the mobile phone being considered.
- The application should ensure the privacy of user information and conform to open-source software standards.
3.3 System Design

**Use Case Diagram:** The use case diagram is used to describe how the system works and interacts. The significance of the Use Case Diagram is because it shows clearly the overall system in the perfect view including every function and method in the system.

User is the person who uses the emergency mobile app, when the user creates an account, personal information like the email, full name and gender will be requested (Fig. 2). After setting up personal information, the user needs to add emergency contact(s) and also emergency message(s). In addition, the user can edit emergency information, which is personal information, emergency contact and emergency message. User can edit the content inside this information.

![Use Case Diagram](image)

**Activity Diagram:** The Activity diagram show each of the functions process in a detailed view. The reason for choosing Activity Diagram is because it can show how each of the functions operate step by step.

**Set up Personal Information**

![Activity Diagram of setup personal information](image)
When the user sets up the personal information, the user login with his/her password. After that, the system validates the password but if the password is wrong, the system display error message. If the password is correct, the user update his/her personal information from the account registration. Then proceed to add emergency contact(s) and emergency message(s); then the supplied information is saved to the database (Fig. 3).

Add Emergency Contact

![Activity diagram of add emergency contact](image)

After logging – in, the user adds emergency contact and being prompted to enter the contacts name and phone number that will be accessible to the emergency unit (Fig. 4) in case they need to let someone know of the current situation. This contact will then be saved in to the database.

Add Emergency Message

![Activity diagram of add emergency message](image)

When the user wants to add emergency message (Fig. 5), the system checks if the user is logged in or not, only then will it continue to add emergency message, the user will be prompted to enter the message title and message description that will be accessible to the emergency unit. This message is not the message that will be sent as the
emergency alert but the message that describes the user’s most likely health condition. This message will then be saved in to the database.

**Maintain Emergency Information**

![Activity diagram of maintain emergency information](image)

User can edit few of the information that he or she had saved at a previous time, the system will check if the user is logged in or not, only then will it allow the user to edit this information (Fig. 6) that will then be saved into the database.

![Class diagram of emergency call mobile application](image)

Figure 7. Class diagram of emergency call mobile application
The Figure above (Fig. 7) shows the relationships between each of the design attributes for the designed emergency call / report mobile application using class diagram representation.

4. SYSTEM OVERVIEW

![Class Diagram](image)

Figure 8. System Overview of Health-Based Emergency Reporting Mobile Application

The figure above Fig. (8) shows that the overall system of health-based emergency reporting mobile application, the diagram show that the application plays an important role in all the process. The user needs to create an account and the information will be stored in the database. The information will include the name, email and address of the user. The user is required to add one or more emergency contact to the emergency mobile application during the account registration process. Besides, the user is also required to add emergency message(s) after the process of account registration. When the call button is pressed, the emergency reporting mobile application will get the all the information from the database and send it to the health centre emergency contact.

5. SYSTEM IMPLEMENTATION AND TESTING

For the implementation of the emergency reporting mobile application, the mobile application is released as “.apk” format for the android platform and “.ipa” format for the iOS platform; and installed in the smartphone using android version 4.0.3 or later with at least dual core processor, 4 GB of ROM and at least 512 MB of RAM. The smartphone is also required to have data connection, GPS function.

The function and feature of the emergency reporting mobile application was tested accordingly the verification. The functions and features to be tested includes user account signup section, login section, messages section, contacts section, personal information section and emergency function section to validate the user requirements and functionality of the mobile health emergency app.

Splash Screen
This is the health-based emergency reporting mobile application splash screen, when the user taps on the app icon, the app will first launch the splash screen to show the graphic illustration and also the emergency text (Fig. 9). The splash screen will stay on the screen around three second in order to establish a connection with the database and then do a check to confirm if a user is logged in or not then only proceed to another page.
Home Page

The home page of an emergency mobile app is made up of three buttons (Fig. 10) performing different functions. It shows up after the user successfully logged in. The first button is “Emergency call” button which is to report the emergency. The second button is “Menu” button which shows several other functions to be done with the app. The third button is “History” button which directs to a page showing the user’s record of emergency reporting.

Account Creation Phase

This is the account creation page of the emergency mobile application. User needs to create an account before using the app. The user will only see this screen once after installing the app. There are five fields on the screen to get the user inputs, the field which is “Full name”, “Email”, “Gender”, “Password” and “Confirm Password” (Fig. 12). The user need to key in the password twice in order to be sure the password is correct. After entering all the information, the user can just press the “Sign up” button to create the account which will redirect to login screen. This page has been well validated to help the user understand every error.
Login Phase
This is the login page of emergency mobile application (Fig. 13). After the user creates an account, the user will be proceed to the login page. After the user successful login to the system, the main page of app will be show to the user. Each time the user wants to activate the emergency function or make some changes, he or she is require to login into the system.

Profile Phase
This is page that allow the user to enter their personal details (Fig. 14). User can also edit their personal information by just pressing the button to enter inside the information that user want to edit.

6. RESULTS
When the application is launched and emergency health situation is successfully reported (Fig. 15), the exact location of the user / victim is going to be displayed (Fig. 16) using the embedded GPS activator and response message will be synchronously received (Fig. 17).

7. CONCLUSION
This mobile app will work like an emergency assistant to aid user call for help during the critical condition. The app will only work when the user launches the app then taps on the call button. After the app is activated, all the necessary functions will work in background of the smartphone. The idea for this mobile app is for it to be quick, easy and effective. This makes all users to have an efficient and powerful mobile app to help people from urgent health situation. This tends to become a huge benefit to people. The proposed solution used to solve the existing problem will make the reporting of health emergency much easier and efficient by sharing the location of the victim.
REFERENCES


