SCHOOL OF SCIENCE AND ENGINEERING

EMERGENCY SITUATION HANDLER MOBILE APPLICATION
Capstone Project Final Report

Sara EL AMRANI ABOU EL ASSAD

Supervisor:
Dr. Violetta Cavalli Sforza

May 2nd, 2016
EMERGENCY SITUATION HANDLER MOBILE APPLICATION
Capstone Project Final Report

Approved by Supervisor
Acknowledgements

First, I would like to express my deepest gratitude toward my supervisor Dr. Violetta Cavalli Sforza for her constant guidance and support. I would like to thank her for investing her time and efforts, despite her busy schedule, to give me pieces of advice and guidance throughout the process of achieving the project goals.

I would like to thank Dr. Yassine Salih Alj who gave us support for our capstone projects. My thanks go also to my friends who showed moral support and unconditional encouragement. Last but not least, I would like to express my deepest gratitude to my parents for their presence in my life, and without them, I would never have gone this far.

El Amrani Abou El Assad Sara
# Table of Contents

I. Introduction ............................................................................................................................... 5

II. Project Plan and Process ....................................................................................................... 6
   1. Feasibility Study .................................................................................................................. 6
   2. STEEPLE Analysis ............................................................................................................. 7
   3. Methodology .................................................................................................................... 8
   4. Software Development Model .......................................................................................... 9

III. Requirements Analysis ........................................................................................................ 10
   1. Functional Requirements ............................................................................................... 10
   2. Nonfunctional Requirements ......................................................................................... 11

IV. Design Phase and Implementation ...................................................................................... 12
   1. Scenarios of Use of the Application ............................................................................... 12
   2. Technology Enablers ....................................................................................................... 14
   3. System Architecture ........................................................................................................ 15
      3.1. Local Storage .............................................................................................................. 17
      3.2. Implementation ........................................................................................................... 19

V. Results .................................................................................................................................. 24

VI. Conclusions, Challenges, and Future Work ....................................................................... 28

VII. Sources .............................................................................................................................. 29
I. Introduction

Human beings are usually concentrated when they face a dangerous situation. According to an article of Gayle Young of United Press International, people react differently to danger [1]. The common emergency behaviors are anxiety, fear, panic, and hyperactivity. However, they have a tendency to think better and simply in this kind of situations [1]. Governments invest on studies to understand human behavior in this kind of situations in order to implement a plan that would result in fewer damages. As most of us have a smart phone nowadays, it would be smart to use these devices in important matters such as to save our lives. It is true that mobile phones do not guarantee our safety, but they can be of good help in seeking it.

The goal of this project is to develop a mobile application that would enable persons in emergency situations to seek help from their relatives or organizations that could provide assistance. The user will provide the application with the necessary contact information that would be needed in emergency cases. The application would have the ability to handle different emergency cases including medical emergency situations, road accident emergencies, etc.

The user of the application will be able to set his or her own emergency situation handler(s) and enter the required pieces of information needed at an earlier time. In case of a dangerous situation, the user will be able to seek help with less effort from persons or organizations that are set in the list of contacts for the emergency.
II. Project Plan and Process

1. Feasibility Study

The project is feasible if we address the challenges in each phase of the development process. The main challenge consists in sending a message to seek help, providing the service across various platforms, and sending out the necessary information about the user.

First, the mobile application will have to send an SMS to the concerned people or institutions. It is feasible using various tools such as Twilio REST API. This API helps in sending text messages (SMS and MMS) and its requests are served over HTTPS to provide security and privacy when relaying messages.

Second, the application should be available for all users using different platforms, mainly Android and iOS. Therefore, the use of a cross-platform mobile development tool would be useful and efficient to avoid many complications for the developer, including having a good background in Java, Swift, Android APIs, and XCode directives. The use of a cross platform mobile development tool also relieves the developer from porting the application from a platform to another. Thus, the developer will not lose time and effort. The possible cross-platform development tools that can be used are Cordova from Apache or PhoneGap, both of which can translate code from HTML5, CSS, and JavaScript. Another tool to be used is Xamarin which uses a single shared C# codebase. For this project, we will use PhoneGap as the cross-platform tool because it using HTML5, CSS, and JavaScript, and also because JavaScript is used by the TWILIO API and other APIs used in the project.

Third, the application relies a lot on the server side, since the user will have to enter personal data, plus the persons and/or organizations information to be contacted when using it.
Therefore, the choice will be made between SQLite and Firebase. SQLite is a good fit for use in cellphones as it provides support for the storage, security and performance of data of mobile applications. It also stores databases as single disk files in a cross-platform format to allow their use on different machines. Firebase is also a good database as it allows developers to store data and authenticate users in the mobile application with just few lines of code and across different platforms. For this application, we will use SQLite as a database because it is the most used convenient open source database for our project thanks to its efficient range queries and storing of each database as a single disk file in a cross-platform format [2].

2. STEEPLE Analysis

SOCIAL:

The social impact of ESHMA is to enable people to avoid the worst in a dangerous situation and decrease the rate of crime and accidental death.

TECHNOLOGICAL:

The project will make use of one of the latest mobile application development tools, Hybrid mobile app development tools that enable the support of various platforms using technologies applied to web development. The application will also use already existing APIs and systems to achieve the goal of the project.

ECONOMIC:

The mobile application will not have direct economic impacts.

ENVIRONMENTAL:

The application does not have a direct impact on the environment.

POLITICAL:
The application has no political vision.

**LEGAL:**

The application has no legal impact; however, it will abide by license agreements for the tools employed.

**ETHICAL:**

The application will ensure the privacy of the users’ data.

### 3. Methodology

As we have checked the feasibility of the project, we proceeded in choosing the process that would best fit to realize the project with time and requirements constraints.

The best approach was to first determine the different features that are offered as services by already implemented solutions that help people in difficult situations. As previously discussed, the existing solutions offer first aid help to users as form of steps to follow in each specific situation. In our solution, we offer another kind of service which is different from the already existing ones, as the users of the application will seek help from other people or organizations. Therefore, we need to conduct requirements analysis in order to include these new features that are mandatory to implement.

The next step is to get the non-functional requirements that we would like to provide along with the functional requirements, which include the design of the user interface and the additional features that would offer better services to the client.

The design phase would be the next step. This step would be delicate as the application will be used in different platforms including Android, iOS, and Windows Phone 8. To realize this,
we will use a cross-platform development tool that is a native container that wraps the code of a web application written in JavaScript, HTML, and CSS. This hybrid model to implement mobile applications has many advantages. They include the ability to have access to native APIs of the different platforms, which is not usually possible through browsers [3]. Another advantage is that a developer does not need to learn about the different programming languages and tools to develop a single application for each platform.

4. Software Development Model

In order to create a hybrid application, it is necessary to test the implementation of the different features to make sure that they are working. That is why we decided to choose a software development model that would enable testing units or modules at different stages of the process. Therefore, the best model to follow would be the V-model, shown in Figure 1, which is an advanced Waterfall model.

![Figure 1: V-Model Phases](image-url)
The advantage of this model is that it emphasizes the validation and verification of the deployment of the units of the project singly and the deployment of the project as a whole [4]. Like the Waterfall model, the phases of the V-model are completed in a chronological method, except that the testing of the code is done in parallel with the development phase.

III. Requirements Analysis

In this section, we will specify the functional and non-functional requirements of the project.

1. Functional Requirements

- The User Interface is designed to make it very quick to send an alert for one of the following emergencies:
  1. General Emergency: to get help by sending location to a list set by the user
  2. Medical Emergency
  3. Road Emergency
  4. Assault/Crime Emergency
  5. Terrorism Emergency
  6. Fire Emergency
  7. Natural Disaster Emergency
  8. User Customized Emergency

- Actions available to user:
  1. Modify
     1.1. General Emergency
     1.2. Modify Medical Emergency
1.3. Modify Road Emergency
1.4. Modify Assault/Crime Emergency
1.5. Modify Terrorism Emergency
1.6. Modify Fire Emergency
1.7. Modify Natural Disaster Emergency

2. User Customized Emergency
   2.1. Create Emergency
   2.2. Delete Emergency
   2.3. Modify Emergency

➢ The user will also have the choice to search for first aid as the following in the sliding menu:

Other Services:
   1. Search in First Aid Dictionary
   2. First Aid Kit
   3. Show Nearest Police stations

➢ In addition, the user will be able to check the history of calls for help using SMS messages by date

2. Nonfunctional Requirements

➢ The application should have a user friendly interface
➢ English is the language used by the application
➢ The project should be finalized by the end of the semester
- The application will run on a mobile phone. Thus we should consider memory and storage space, battery life, ability to adapt to difference screen sizes and orientations of the mobile phone.

- The application should ensure the privacy of user information and conform to open source software standards.

IV. Design Phase and Implementation

1. Scenarios of Use of the Application

The first step is to design the method of access of the application. As the mobile app will be used in emergency cases, it should be designed in a way that the user will have no difficulties and no complicated step(s) to ask for help. Thus, we should determine and recognize the various situations in which the application will be used and differentiate the cases where it will be used in emergency situations. The envisioned scenarios of use are shown in Figure 2.
The application will also have a feature that will enable the user to set previously an emergency message that will be sent to the list of contacts of the application with a countdown time in multiple cases, as explained below. This feature will be available under the case of “User Customized Emergency”. To illustrate the main advantage of this feature, we can take as an example a student living alone who does not trust his or her surroundings and wants to go out for about half an hour to get something from the nearby shop. The student user can use this feature to alert family and / or friends that he or she is supposed to be at home after a maximum of 30 minutes so that if he or she is late, they will be informed that they need to take some action. Another example is of a person who suffers from a respiratory disease such as asthma. That
person may use this feature to alert family members and/or friends that he or she is taking a shower and if too much time elapses before that person reports back saying he or she is done, they again should take some action.

2. Technology Enablers

**SQLite:** It is a relational database management system that is widely used for mobile applications for local or client storage. Since that our application will mainly rely on local storage since it will not usually have access to internet, SQLite would be a good choice. Also, there exists a SQLite plugin for hybrid mobile developers that can be used along with PhoneGap. This plugin has tempting advantages as it ensures reliability and support of large databases sizes [5].

**PhoneGap:** It is a cross-platform development tool introduced by Apache Software Foundation. Used by numerous mobile developers, it reduces time spent on creating mobile applications by sparing the time spent on learning about technologies used to develop applications for each specific platform [6].

**MyEclipse:** It is a powerful IDE as it enables developers to work individually or on teams on Web applications and mobile applications. The good advantage of using MyEclipse is that it enables the development of cross-platform application with the use of PhoneGap framework integrated in it [7]. In addition, the IDE provides a better environment for developers to test their developed application as it offers virtual device services for simulations.

**JavaScript:** It is the main language deployed by websites along with HTML and CSS [8]. It is an interpreted programming language that is mainly used in web development and hybrid development tools such as Cordova and PhoneGap. Learning JavaScript is a great advantage, and
deploying it to build web application and mobile application has the main advantage of using one language to build projects for different platforms.

**HTML:** It is also one of the main markup languages used to develop web applications. It will be mainly used in this project to implement the user interfaces of the mobile application.

**CSS:** It is a language to design styles of user interfaces written in markup languages such as HTML [9]. This language enables developers to separate between the presentation of the document and its content.

**JQuery Mobile:** It is a JavaScript library that creates a framework for mobile applications in order to have a better interaction and performance [10].

### 3. System Architecture

The application will be using the Model-View-Controller (MVC) architecture, shown in Figure 3. The MVC paradigm is an approach used to develop software programs in a maintainable and clear way. This architecture has many advantages as it separates concerns by programming independent elements (Model, View, and Controller) that are interchangeable and reusable, and results in a clear well-structured program [11].
The interaction between the components of the MVC design pattern goes as follows: The Model is the central element of the pattern because it represents the domain logic of the application, which gives meanings to the data and information on which it operates. The view is the representation of the output as it renders the model to provide a better form of interaction with the user. We can have multiple views of an application as the output can represented using more than one form. The last element of this architectural pattern is the Controller. It gets the input from the user and transforms it into a set of commands for the model to manipulate it and make changes in it. It also makes changes on the views of the model content because the views are a representation of the model [11].

Thus, the architecture of the project is shown in Figure 4.
3.1. Local Storage

The storage of data will be done locally to enable both users with access to internet and users without access to internet to use the services of the application. The latter will also have the benefit of increasing service efficiency and not relying necessarily on the presence of internet.

In this project, we will be using along with the queries in the application the SQLite Manager to manage the database. PhoneGap also offers a SQLite plugin to connect with the
database. In addition, we will be handling queries to access data in the database in the model of the architecture as shown in Code Snippet 1:

```javascript
$scope.openDB = function (){  
  var db;
  if(window.cordova){
    db = $cordovaSQLite.openDB({name: this.db});
  }
  else{
    db = sqlitePlugin.openDatabase(this.db, '1.0', 'my', 65536)
  }
}
```

**Code Snippet 1: Opening the Database**

The piece of code above allows opening the database. The opening of the database can be done through the device or through the browser (for testing) as we are building a hybrid application.

Similarly, creating and storing data is done using a JavaScript function in the model of the architecture that executes the queries, as shown in Code Snippet 2.

```javascript
function execQuery($queryToexec,callback){
  try{
    if(sqlitePlugin.openDatabase){
      db.transaction(
        function(tx){
          tx.executeSql($queryToexec,[],function(tx,result){
            if(typeof(callback) == "function"){
              callback(result);
            }else{
              if(callback != undefined){
                eval(callback+"(result)"));
              }
            },function(tx,error){});
        });
      return rslt;
    }
  }catch(e){}
}
```

**Code Snippet 2: Creating and Storing Data**
An example of a query to be executed is shown in Code Snippet 3.

```javascript
function insertval()
{
    var img = document.getElementById('image');
    var sql = 'INSERT INTO SmsTable (F_name, i_name, phone_number, emergency_type) VALUES ('+fname+','+i+
execQuery(sql,function(results){alert(results)});
}
```

**Code Snippet 3: Querying the Data**

### 3.2. Implementation

When a user wants to use the mobile application for an emergency, he or she is not required to login or authenticate in order to use the services of the app, since they may not have the time to do so. However, the user is asked to enter login credentials when he or she is modifying the settings and data of the application, because the latter is confidential (Figure 5).
The user is required to enter the necessary contact list for at least one emergency, which is the General emergency case. The user is prompted to enter the first, last name, phone number, and priority for each contact entered.

In an emergency situation, the user clicks on the button referring to a specific situation. He or she has two choices:

1. Call a number put in at high priority by the user
2. Send SMS messages to the list to contact put by the user
When choosing the first choice, the call is triggered using Twilio API. This API using POST requests to Twilio Server in order to make calls or send and receive SMS messages. First the Twilio client connects with the server using HTTP requests, as in Code Snippet 4:

```
7 var tokenRequest = new XMLHttpRequest();
8 tokenRequest.onload = function() {
9     Twilio.Device.setup(tokenRequest.responseText);
11
```

**Code Snippet 4: Twilio Request**

The client side sends HTTP requests with unique credentials to use the services. In order to use Twilio, a developer can buy an API key to get a Token in order to use the service. In our case, we used a free key for the trial version. When the Twilio client gets the required token, it triggers the call with the JavaScript function shown in Code Snippet 5.

```
14 $("#dialButton").click(function() {
15     params = { "tocal" : $('#phoneNumber').val()};
16     connection = Twilio.Device.connect(params);
19 });
```

**Code Snippet 5: Call Triggering Using Twilio**

The function called is implemented in the plugin used in the mobile application. The POST request made to the server side is as shown in Code Snippet 6.

```
12 tokenRequest.send();
```

**Code Snippet 6: POST Request to Twilio Server**
The String “AC127cf62099e4123b023497b66a331193” is the SSID we got when we created an account to use Twilio API for trial; it is used to make the POST requests to the server.

The second choice of the user will enable him or her to send SMS messages to the already existing list of contacts. To do that, we also used Twilio API to send messages. However, since that the user may not be connected to internet, Twilio API will not be used often. Thus, we used a PhoneGap plugin that enables user to SMS messages, which is Cordova-plugins-sms. The message is sent with the function shown in Code Snippet 7.

```javascript
var success = function () { alert('Message sent successfully'); };
var error = function (e) { alert('Message Failed:' + e); };
sms.send(number, message, options, success, error);
```

**Code Snippet 7: Function Call to Use the Plugin**

The function returns either a success message or an error message. The message used as a parameter includes the following information:

i. The location of the user by stating the longitude and attitude

ii. The type of emergency

iii. Any additional information that the user had previously entered

The location of the user can be found using the native sources of the device used. In this case we will be using GPS to find the location of the user through PhoneGap. This is shown in Code Snippet 8.
Code Snippet 8: Function to Get User Location

The location is then sent to the function where the message to be sent to contacts is assigned.

Besides the suggested emergencies in the menu, the user can create a customized emergency handler as mentioned in the cases of use of the application. The emergency is set either to send SMS messages immediately or after a period of time.

As stated in the functional requirements, the user can also check the nearest police stations on a map. This can be done using the function shown in Code Snippet 9.

Code Snippet 9: Get Map with Police Stations
V. Results

In this section, we will some screenshots of the main activities of the application.

The main activity that is seen by the user is the menu, displayed in Figure 6.

![Main user Interface](image)

**Figure 6: Main user Interface**

The first use of the application would enable the user to enter required information about contacts for at least one emergency case. It can be done by going through the sliding Menu of the application and choosing to modify an emergency, as shown in Figure 7.
The user clicks on ‘Modify an Emergency’, chooses an emergency, and enters the needed information about people or organizations to contact, as shown in Figure 8.
The user is ready to use the application after filling the required fields. Thus, during an emergency situation, when a user clicks on a specific emergency, he or she is directed to a page that will enable him to call or send SMS messages to seek help, as shown in Figure 9.
When the user sends an SMS, it displays in a popup message that it was sent successfully. The message sent to the contact(s) the location of user, type of the emergency, and/or any other pieces of information needed (which depends on type of emergency), as shown in Figure 10.

Figure 9: Emergency Case Handler of ‘General Emergency’

Figure 10: SMS Message Sent to User
VI. Conclusions, Challenges, and Future Work

With the help of this project, I immersed myself into the world of mobile development, which I describe as great experience. Building a hybrid application that uses technologies that I have never worked with and that I didn’t have the chance to learn about during my undergraduate studies was beneficial for me. Also I learned how to use the different native sources of devices using different platforms (mainly Android and iOS) was interesting and challenging.

The other challenge that I encountered was that most of the documentation about plugins used in my project is still under maintenance as developers of hybrid applications are trying to fix the resulting errors after running projects. Learning about JavaScript and CSS was also challenging given the time constraints.

There are some features in the project that are still not fully ready to work, such as having access to the First Aid Kit and searching on the First Aid Kit Dictionary. Once those features are ready to work, I plan to release an alpha version as a trial to check if there are any bugs and errors resulting from using the mobile application. However, for now, the project shall be used to test the results of combining web development tools with wrapping cross-platform tools like PhoneGap.
VII. Sources


