

European Project Nº: **FP7-614154**  
Brazilian Project Nº: **CNPq-490084/2013-3**  
Project Acronym: **RESCUER**

Project Title: **Reliable and Smart Crowdsourcing Solution  
for Emergency and Crisis Management**

Instrument: **Collaborative Project**  
European Call Identifier: **FP7-ICT-2013-EU-Brazil**  
Brazilian Call Identifier: **MCTI/CNPq 13/2012**

## Deliverable D2.5.1 Development of the Mobile Solution 1

Due date of deliverable: PM10  
Actual submission date: January 21, 2015



Start date of the project: **October 1, 2013 (Europe) | February 1, 2014 (Brazil)**  
Duration: **30 months**  
Organization name of lead contractor for this deliverable: **MTM Tecnologia**

Dissemination level		
PU	Public	✓
PP	Restricted to other program participants (including Commission Services)	
RE	Restricted to a group specified by the consortium (including Commission Services)	
CO	Confidential, only for members of the consortium (including Commission Services)	

# **Executive Summary**

## **Development of the Mobile Solution 1**

This document presents deliverable D2.5.1 (Development of the Mobile Solution 1) of project FP7-614154 | CNPq-490084/2013-3 (RESCUER), a Collaborative Project supported by the European Commission and MCTI/CNPq (Brazil). Full information on this project is available online at <http://www.rescuer-project.org>.

Deliverable D2.5.1 provides the results of the first iteration of Task 2.5 (Development of the Mobile Solution). In this task, we develop the RESCUER component called Mobile Crowdsourcing Solution taking into consideration the specifications received from the other tasks of Work Package 2 (Mobile Crowdsourcing Solution) and the most feasible mobile technologies for the RESCUER Project. The main result of this task in the first project iteration is the alpha version of the Mobile Crowdsourcing Solution with some basic features to be evaluated during the FIFA World Cup 2014.

### **List of Authors**

Paulo Soares – MTM

Gustavo Perez – MTM

### **List of Internal Reviewers**

Paulo Jr. – UFBA

Kai Breiner – Fraunhofer

Karina Villela – Fraunhofer

# Contents

- 1. Introduction..... 4
  - 1.1. Purpose ..... 4
  - 1.2. Partners’ Roles and Contributions ..... 4
  - 1.3. Document Overview ..... 4
- 2. Project Decisions ..... 5
  - 2.1. Hybrid Frameworks ..... 5
  - 2.2. Phonegap ..... 7
- 3. Specifications vs Implementation..... 8
- 4. Mobile App Screens..... 12
- 5. Conclusion ..... 14
- Bibliography..... 15

# 1. Introduction

The purpose of the Mobile Crowdsourcing Solution in the RESCUER Project is to provide an interface to enable crowds in large-scale events or industrial parks to send information about an incident to a command and control centre and to receive warnings and directions on how to proceed.

Deliverable D2.1.1 (Conceptual Model of Mobile User Interaction in Emergencies) studied the human behaviour in emergency situations in order to allow the design of the Mobile Crowdsourcing Solution with regards to the gathering of information provided by the crowd spontaneously or as a result of a known process (Deliverable D2.2.1: Crowdsourcing Information Gathering). The result was the specification of the information gathering features of the Mobile Crowdsourcing Solution, which have been implemented in a project demonstrator.

## 1.1. Purpose

The purpose of this deliverable is to describe the main decisions and results of the development of an alpha version of the Mobile Crowdsourcing Solution for the first project iteration, which is only concerned with information gathering. The presented results have been available since the due date of this deliverable and were evaluated during the FIFA World Cup 2014. This document focuses in the mobile application and makes no reference to the RESCUER's backend.

## 1.2. Partners' Roles and Contributions

The partner responsible for this deliverable is MTM. Fraunhofer contributed to this deliverable by conducting a one-week workshop for specification and prototyping of the Mobile Crowdsourcing Solution for the first project interaction and by conducting regular project monitoring meetings. It is in the scope of MTM's work the research of the most appropriate technologies to be applied in the project development and in the distribution of the mobile applications, taking into consideration the strategy and guidelines defined in D1.4.1 (Portability & Variation Management Strategy 1).

## 1.3. Document Overview

The remainder of this document is structured as follows:

- Chapter 2 describes project decisions and selected technologies.
- Chapter 3 presents screenshots of the first version of the Mobile Crowdsourcing Solution.
- Chapter 4 presents the conclusion of this document.

## 2. Project Decisions

An important decision when developing a mobile app is the approach to be used. There are mainly three kinds of approaches for mobile application development: Native, Hybrid and Web App.

The native approach requires the same system to be developed independently for each mobile platform to be supported. It consumes a lot of effort to develop and maintain those different source codes. The Web App approach uses pure web technology to develop the system. Despite the fact that it uses one source code for all platforms, the technology behind it does not give access to core features of the devices that are important to the RESCUER Project, like GPS, camera, accelerometer, notification, background threads, among others. The hybrid approach is able to access the core features of the device and share major parts of the source code between different platforms. Hence, this was the approach chosen for the development of the Mobile Crowdsourcing Solution.

Section 5.3 (Development Approaches) of deliverable D1.4.1 (Portability & Variation Management Strategy 1) presents a study about the three approaches for mobile development.

### 2.1. Hybrid Frameworks

The next step after choosing the development approach was to identify the best frameworks to use for hybrid development. Three frameworks were evaluated for use in the project: Phonegap[1], Xamarin[2] and Titanium[3].

Xamarin has different versions available. One of them is free, but is limited and has some constraints like the size of the compiled application. Xamarian is not open source and this deliverable opted for an open source framework that provides the maximum level of reuse.

Titanium's source code is not entirely reusable between platforms [13]. Approximately 70% - 80% of the code can be reused between platforms. This results in the need for native code for each platform, reducing the advantage of using a hybrid approach.

Phonegap supports access to all main core features of the devices, as presented in Figure 1, and the entire code can be reused between platforms, except when the solution requires the use of native libraries.

	iPhone / iPhone 3G	iPhone 3GS and newer	Android
Accelerometer	✓	✓	✓
Camera	✓	✓	✓
Compass	X	✓	✓
Contacts	✓	✓	✓
File	✓	✓	✓
Geolocation	✓	✓	✓
Media	✓	✓	✓
Network	✓	✓	✓
Notification (Alert)	✓	✓	✓
Notification (Sound)	✓	✓	✓
Notification (Vibration)	✓	✓	✓
Storage	✓	✓	✓

✓ - supported feature  
X - unsupported feature due to hardware or software restrictions

**Figure 1: Core features supported by Phonegap [14]**

In addition, Phonegap has a big community and a good documentation and online resources. The Phonegap development uses web technologies like HTML5, CSS and JavaScript. Some frameworks to create mobile application components with HTML5, CSS and JavaScript were analyzed including Ionic [6] with AngularJS [7], Sencha Touch [8], jQuery Mobile [9] with Backbone [10], and Kendo[11]. These frameworks have the web components adapted for the use in mobile devices, and reduce the work and time of implementation of mobile applications.

## 2.2. Phonegap

Phonegap works like a wrapper. It encapsulates a web page renderer into a native application; therefore it can access the features of the mobile device and still ensure reuse of code. The graphical interface of the application is developed with HTML5, CSS and JavaScript, and the resulting code can be interpreted in different platforms that have a web renderer.

Figure 2 shows the operation of Phonegap and the components used to generate the executable code for the various supported platforms.

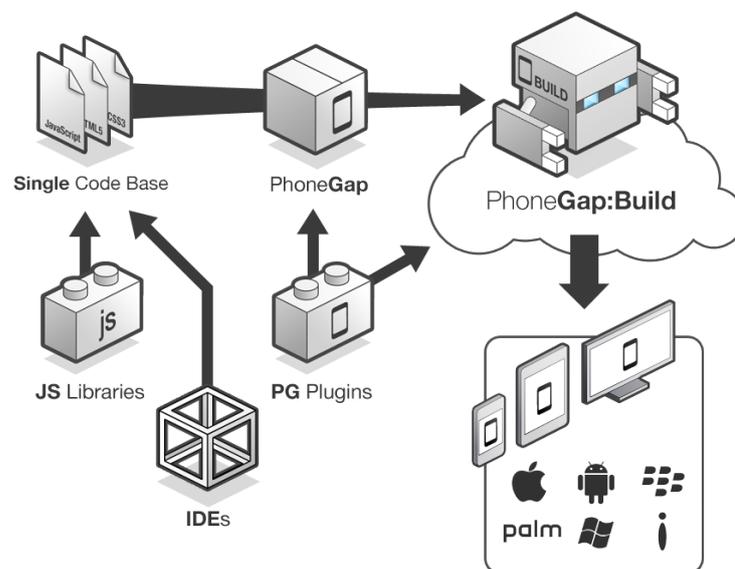


Figure 2: Phonegap development components [15].

### 3. Specifications vs Implementation

This chapter presents the features implemented in the first project iteration. The objective was to evaluate the first version of the solution during the FIFA World Cup 2014, in Brazil. Thus, the implementation focused on the features to be used by visitor of large events. Figure 3 presents the interaction diagram with the features implemented in the first version of the Mobile Crowdsourcing Solution.

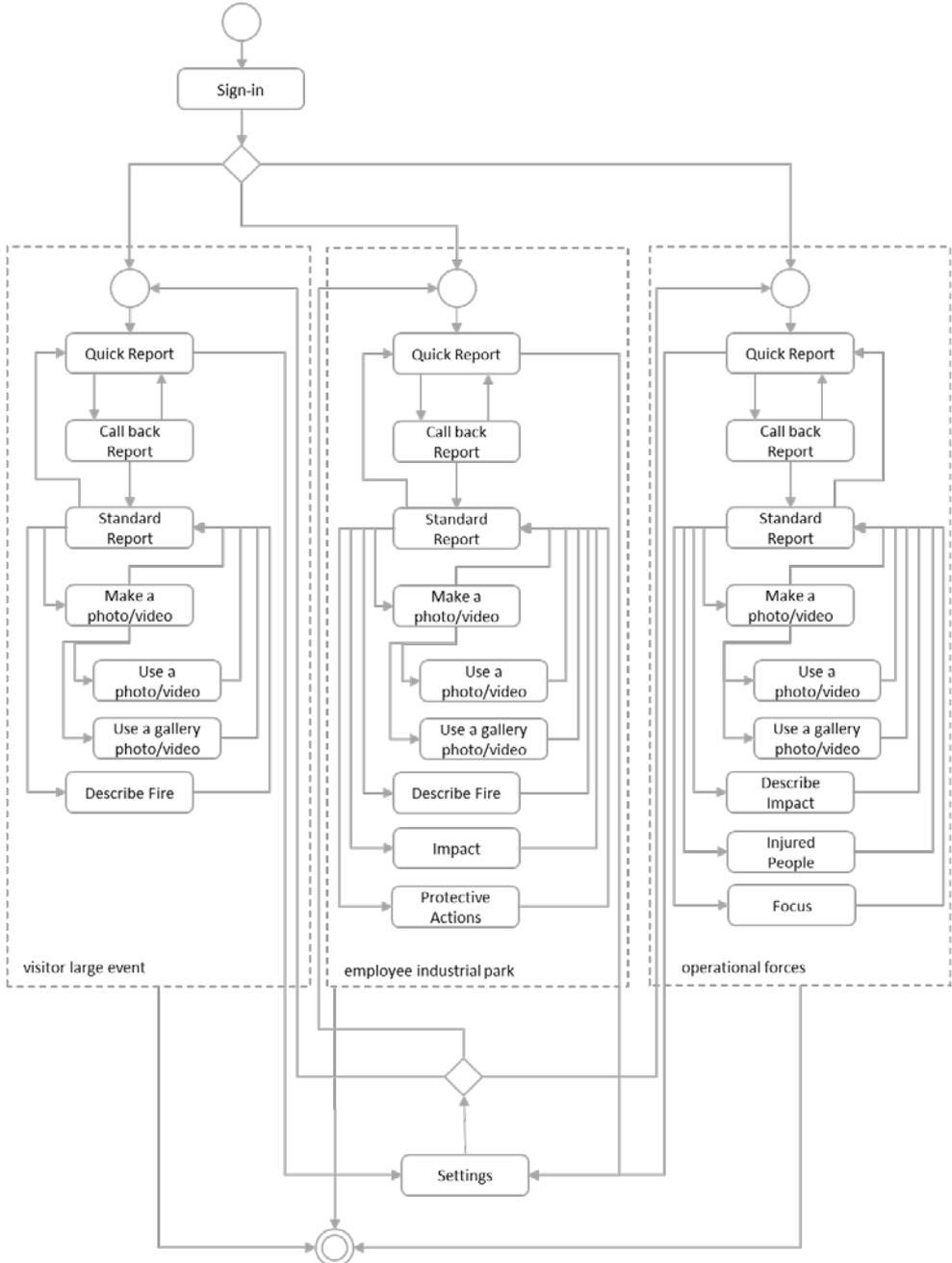


Figure 3: Interaction case diagram of RESCUER app

Table 1 describes the status of features defined for the mobile app and their status.

**Table 1: Implementation status of the specified features**

Features	Description	Implemented	Partially Implemented	Not Implemented	Comments
<b>Sign-In Screen</b>	The user set his profile to get profile based information			x	The server structure to register the profile of the user has not been implemented yet.
<b>Very quick report</b>	The user can send a very quick report, which is just a notification of an emergency situation of a specific type (e.g, fire). This action generates a new emergency report.	x			
<b>Call back report</b>	The user can call back the incident notification in the case of a false alarm.		x		Just the modal window has been implemented. The functionality is not available.
<b>Standard report for large-scale event (fire)</b>	The user can describe basic information about the emergency situation in the context of large-scale events.	x			In the specification of the standard report screen, the “severity of fire” icons are displayed horizontally, but in the implemented version the items are displayed vertically (Figure 6(a)) due to a limitation of time to implement and display the icons horizontally (the default of the Ionic Framework component is vertical display.  The toggle buttons in the “Are people injured?” and “Flames” was changed to the Ionic Checkbox.  The button “send report” was added temporarily.
<b>Defining the fire spot on the map</b>	Map component that allow the users to change the location of the fire spot	x			Just the fire spot is shown on the map.  An open-source JavaScript library for mobile-friendly interactive maps, called Leaflet ( <a href="http://leafletjs.com/">http://leafletjs.com/</a> ), was used in the app. The library facilitates the implementation of the interaction, has several plugins developed by the community and provides a very detailed map.
<b>Describing the fire</b>	The user can describe colour and direction of the smoke as well as if s/he can see flames.	x			The toggle buttons (on/off) in the specification of the “describe the fire” screen was changed to Ionic List (similar to radio button in html). It makes more sense because the user only can choose only one of the items, like “dark smoke” or “grey smoke” and not both.
<b>Making a photo or a video</b>	Camera component that allows the users to make photo or videos of the situation. This functionality can be accessed from each screen on the top bar of the user interface. The following sensor data will be recorded automatically, when making a photo or video: <ul style="list-style-type: none"> <li>• Position (GPS)</li> <li>• Timestamp</li> <li>• Gyroscope</li> <li>• Accelerometer</li> </ul>		x		Just take a photo has been implemented. Photos are not annotated with sensor data yet.

Features	Description	Implemented	Partially Implemented	Not Implemented	Comments
<b>Selecting a photo or a video from the device gallery</b>	<p>Component that allow users to upload photos or videos of their picture gallery of the smartphone. By using photos or videos of the mobile phone's gallery, only the following sensor data can be sent along with the selected photos or videos to the Data Analysis Solutions:</p> <ul style="list-style-type: none"> <li>• Position (GPS) – if allowed by the user in the settings of the device.</li> <li>• Timestamp – mostly expressed in the name of the pictures and videos files.</li> </ul>		x		Only the selection of the photo from the mobile phone's gallery has been implemented. The support to select a video from the gallery has not been implemented.
<b>User profile and permissions</b>	The user can edit his profile information as well as permissions related to the information monitoring and recording done by the app.			x	The server side of this feature has not been implemented yet.
<b>Automatic upload of reports</b>	<p>After 5 seconds without interaction with the elements of a screen, the screen information should be send to the server, since the user might be interrupted at any time. This approach reduces the possibility of losing already entered information. The following information are kept as a single report:</p> <ul style="list-style-type: none"> <li>• Report ID</li> <li>• GPS position</li> <li>• Timestamp</li> <li>• Report information (informed by the user through the interaction with the user interface)</li> </ul>			x	The server side of this feature has not been implemented yet.

Features	Description	Implemented	Partially Implemented	Not Implemented	Comments
<b>No save button</b>	People do not need to save or to explicitly send information to the RESCUER backend. Any report will be automatically sent after 5 seconds of no interaction.			x	Currently the information is send just when the user taps on a send button.
<b>Constant interaction</b>	During the upload of a picture/video the user can continue interacting with the mobile application.			x	The image and video are not being sent to the BLOB storage in this version of the app.
<b>Online/Offline capabilities</b>	The system has to work in both online and offline mode. Furthermore, the system requires a well-defined synchronization concept that works in both online and offline mode.			x	The app is just loading when the smartphone is online.
<b>Persistence</b>	As long as a report is often, the information has to persist on the screen. Just after leaving the screen, when starting a new report the report screen is reset.		x		The data are persisted, except photos and videos.

## Mobile App Screens

The screens of the Mobile Crowdsourcing Solution for Android are provided below.

The “Quick Report” screen (Figure 4) is where the user can inform that the incident occurred, by selecting the type of incident.

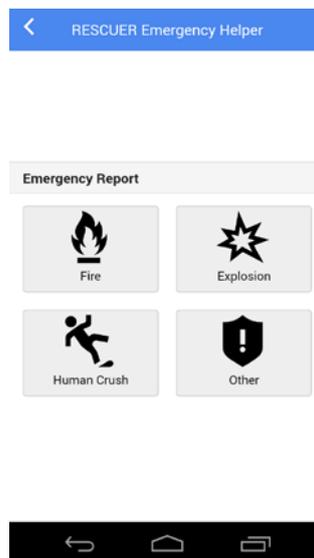


Figure 4: “Quick Report” screen

The screenshot in Figure 5 shows the popup that informs the user that the emergency has been reported and that he can continue giving more information about the incident, if s/he is in condition for doing so. S/He can call back the report in case it was reported by mistake. The quick report includes timestamp and the current GPS position of the user.

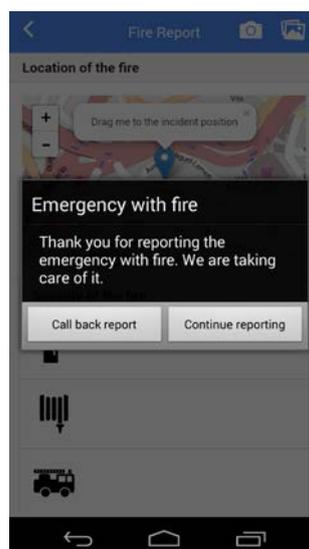


Figure 5: “Call-back” screen

Figure 6 shows the screen where the user provides additional information about the incident. S/he can inform the position of the incident (if different from his/her current position), the severity of the incident, and whether there are injured people or not. The user can also take pictures, or choose pictures from the device's gallery, and send the report.

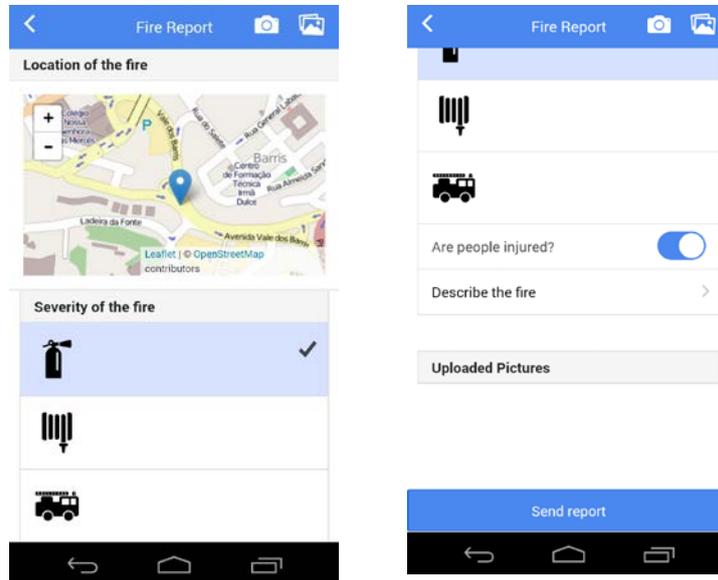


Figure 6(a) and 6(b): “Standard Report” screens

In the screen in Figure 7 the user can describe the fire, by providing the color of the smoke, the direction of smoke, and whether there are visible flames.

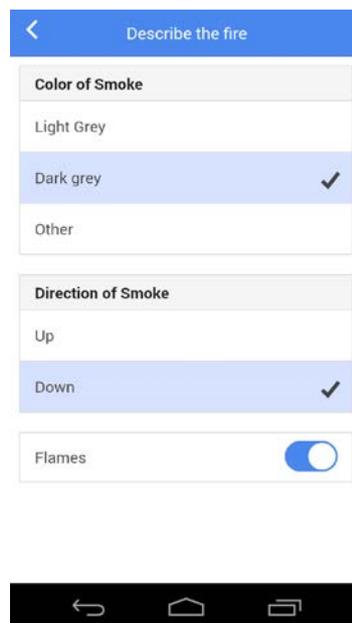


Figure 7: “Describe the fire” screen

## 4. Conclusion

This document and the prototype of the Mobile Crowdsourcing Solution comprise D2.5.1 (Development of the Mobile Solution).

The features planned for the current deliverable are listed below:

- Quick report of an incident;
- Call back a report that was sent;
- Standard report for large-scale events, which includes:
  - Locate the fire location on the map;
  - Describe the fire;
  - Take a picture of the incident;
  - Make a video of the incident;
  - Select a picture or a video from the device gallery, and
- Data persistence.

The use of a hybrid approach to implement the Mobile Crowdsourcing Solution presented some challenges that required more implementation time than it was initially planned. We faced performance issues because of the number of plugins required by the application. We had an issue with one of the plugins, which required the app to be on-line to be used. All these issues are being addressed in the next version.

The issues found during the development had an impact on the implementation of certain features. The call back feature could not be completed because the backend system did not implement the proper services to support this method. The video feature was not implemented until the end of the deliverable, only photo, which was not totally implemented because the sensor data to be stored with the photo was missing. Finally, the persistence of photo and video was not implemented until the release of this deliverable because of the issues faced by the development team during the implementation of the system using the chosen technology.

The prototype as presented here was evaluated during the 2014 FIFA World Cup and by employees of the Industrial Polo of Camaçari. The results of these evaluations are reported in D5.2.1 (Evaluation of the Mobile Crowdsourcing Solution 1).

## Bibliography

- [1] Phonegap. Date of Access: 14.08.2014. <http://phonegap.com/>
- [2] Xamarin. Date of Access:14.08.2014. <http://xamarin.com/>
- [3] Titanium. Date of access:14.08.2014. <http://www.appcelerator.com/titanium>
- [4] You half assed it. That is why your PhoneGap application sucks. Date of access:14.08.2014. <http://sintaxi.com/you-half-assed-it>
- [5] GPL License. Date of access: 14.08.2014 <http://www.gnu.org/copyleft/gpl.html> HYPERLINK ""]
- [6] Ionic Framework. Date of Access:14.08.2014. <http://ionicframework.com/>
- [7] AngularJS. Date of Access:14.08.2014. <https://angularjs.org/>
- [8] Sencha Touch. Date of Access:14.08.2014<http://www.sencha.com/products/>
- [9] JQueryMobile. Date of Access:14.08.2014. <http://jquerymobile.com/>
- [10] BackboneJs. Date of Access:14.08.2014. <http://backbonejs.org/>
- [11] Kendo UI. Date of Access:14.08.2014. <http://www.telerik.com/kendo-ui>
- [12] Xamarian Pricing. Date of Access: 11.09.2014. <https://store.xamarin.com/>
- [13] Titanium SDK. Date of Access: 11.09.2014. <http://www.appcelerator.com/titanium/titanium-sdk>
- [14] Phonegap supported features. Date of Access: 11.09.2014. <http://phonegap.com/about/feature>
- [15] Phonegap development. Date of Access: 11.09.2014. <http://www.metaltoad.com/service/phonegap-application-development>