**Mobile Assistant for Disasters**

**Mobile Client**

**Design Document**



Andre Lukito {andre.lukito@iis.sinica.edu.tw}

# Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| Ver | Author | Date | Change log |
| 1.0 | Andre Lukito | 2014/05/20 | Initial version |

Table of Contents

[Change Log i](#_Toc388965345)

[Table of Figures ii](#_Toc388965346)

[1. Mobile Client Architecture 1](#_Toc388965347)

[1.1. Maps Feature 2](#_Toc388965348)

[1.2. Image Feature 3](#_Toc388965349)

[1.3. List Feature 4](#_Toc388965350)

[1.4. Settings Feature 5](#_Toc388965351)

[2. Requirement 6](#_Toc388965352)

[3. Data Format 7](#_Toc388965353)

[4. Future Works 9](#_Toc388965354)

[5. Package Reference 10](#_Toc388965355)

# Table of Figures

[**Figure 1** – General Architecture in Mobile Client. 1](#_Toc388965336)

[**Figure 2** – Navigation process with Google APIs Map. 2](#_Toc388965337)

[**Figure 3** – Show Image process. 3](#_Toc388965338)

[**Figure 4** – Show List of Facilities process. 4](#_Toc388965339)

[**Figure 5** – Details of Facilities process. 5](#_Toc388965340)

[**Figure 6** – Download and Delete data process. 6](#_Toc388965341)

[**Figure 7** – Saved Map into Image. 7](#_Toc388965342)

[**Figure 8** – Data Structure with JSON format. 8](#_Toc388965343)

[**Figure 9** – Data Structure with RDF/XML format 8](#_Toc388965344)

# 1. Mobile Client Architecture

The arrangement of IS and POS is actually a hybrid of centralized and distributed architecture, and can benefit from both aspects. While POS-s rely on IS to give them data and manage their updates, it can then operate on its own, providing data to mobile clients. Multiple POS-s can ease the burden of IS or even the original data source by lessening the probability that a sudden surge of client requests will overwhelm the server software. In fact, all MAD client application should never go directly to IS for data, it should first try to access POS instead.

The applications aims to enable the general public to easily retrieve and store useful disaster preparedness and response information on their phones even when cellular networks and internet are no longer functional. The early version of MAD mobile application already developed in Android as the ecosystem. The source code this early version of MAD mobile application can be seen on github with this link: "http://gitlab.iis.sinica.edu.tw/gitlab/sp4\_vr/mobileapplication-git".



**Figure 1** – General Architecture in Mobile Client.

As Figure 1 shows, this is the general architecture used by MAD mobile application for early version. MAD mobile client will have 4 main feature, such as Maps, Image, List, and Settings. Maps feature will be used to display user location and navigation from user location to facility location using Google APIs for Map. Image feature will show maps that saved as image for user use where there is no cellular networks or internet. List feature will show all facility that downloaded from POS or IS and from the list of facility, user can read detail of facility and start the navigation by using navigation in Maps feature. Settings feature will be used only for this early version, where user can decide location of POS or IS with IP Address or URL to download the data. Each of main feature will be explained more detail in different subsection.

## 1.1. Maps Feature

As brief explanation in section 1, this feature will display user location and user can use the navigation from user location to facility location using Google APIs for Map.



**Figure 2** – Navigation process with Google APIs Map.

As Figure 2 shows, this is Navigation process using Google APIs Map in Map feature. Facility data will provided from List feature after user choose the facility and want to see the location in Google Maps. When user want to do the navigation, it will send the latitude and longitude data of user location as source and facility location as destination to the Google APIs, and then Google APIs will send the result for navigation data as JSON format. Navigation data that provided by Google APIs need to be parsed to get the each point of direction as polygon and draw it on Google Maps.

Because this feature will use navigation system using Google APIs with user location, every time user want to use this feature, location services settings in user mobile phone will be checked automatically. If the location services are not enabled, system cannot get the user location and cannot provide the navigation from user location to the facility location. So, user need to enable the location services setting in their mobile phone.

## 1.2. Image Feature

As brief explanation in section 1, this feature will have image that display saved maps and it is important when user cannot use the cellular network, internet, and location services. The saved maps are based on user location and point the facility location. User can go to the facility location when they see point on the image.



**Figure 3** – Show Image process.

As Figure 3 shows, this is process to show saved maps as image and display it to user. Application will check whether the image file are exists or not in the mobile device storage. If the image file are not exists, image feature will display image as blank and it is indicate that there is no data in the application at that time.

## 1.3. List Feature

As brief explanation in section 1, this feature is the place where user can see all facility that already listed in their area. This facility are important to user when disaster occur, so, they can choose the facility that near within their location, see the details of facility, and then user can navigate to the facility location from their location using Maps feature.



**Figure 4** – Show List of Facilities process.

As Figure 4 shows, this process will generate list of facilities from data that downloaded from POS or IS and show it to user. Like in Image feature, application will check whether the data file are exists or not in the mobile device storage. If the data file are not exists, list feature will display empty list and it is indicate that there is no data in the application at that time. If the data file are exists, the data file need to parsed by application. As for now, data that can be parsed are data with JSON or RDF/XML format. Application will check if the data file is in JSON or RDF/XML format, and parse it. After that the parse result will show it to user as list.



**Figure 5** – Details of Facilities process.

As Figure 5 shows, this is process to show the details of selected facility to user. If there is no data file in application, then there is no details of facility that can be shown to the user. When user select one of facility in the list, the application will find the details of facility and then show it to user. From the details of facility, user can go directly into Maps feature and then Google Maps in Maps feature will point out the facility location and show the details too. So, basically, the data from details of facility will be pass it into Maps feature and with this data, application can point out the facility location in Google Maps and user can use the navigation from their location into selected facility location.

## 1.4. Settings Feature

As brief explanation in section 1, this feature only for this early version and in the future, download data will be done automatically and user does not need to input the IP Address or URL of POS or IS. The early version still underdevelopment, so, download and delete data will be done manually by input the IP Address or URL of POS or IS.

As Figure 6 shows, this process are download and delete data files (image and data file (JSON or RDF/XML format)). User does not need to find the IP Address or URL of POS or IS, because at the first time, IP Address or URL of POS or IS will be provided as initial value. So, the only user need to do in Settings feature are download and delete data.



**Figure 6** – Download and Delete data process.

When user want download data files, application will make request or connection and then download data files. Data files are consist of image file and data file with JSON or RDF/XML format. When download are success, downloaded data will be saved into mobile device storage and used by other feature. When user want delete data files, application will delete all data files that stored in mobile storage. At this time, application will not have any data and user need to download data files to use other feature.

# 2. Requirement

This section will describe requirement that needed to develop mobile device application in Android devices:

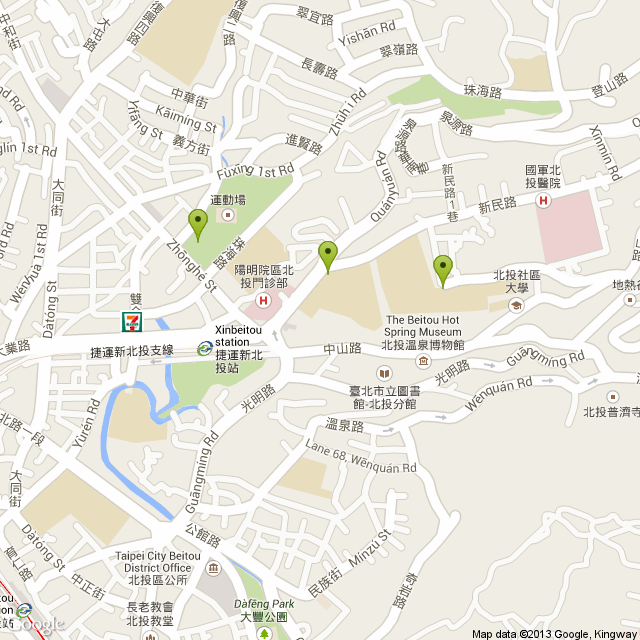
1. Operating System : Windows, Mac OS X, and Linux.
2. IDE : Android Studio or Eclipse.
3. SDK : Android SDK and Google Play Services.
4. Devices : Android Devices (Smartphones or Tablet).
5. Key : Google APIs Key.

Developer need to create Google APIs key in Google API console to use Google Maps services. System that has been used to develop the early version of MAD mobile application are:

1. Operating System : Windows 8.1.
2. IDE : Android Studio (version 0.5.8 Build 135.1155795).
3. SDK : Android SDK (API level 19) latest update and Google Play Services (with SDK minimum version Android 3.0 (Honeycomb (API level 11)).
4. Devices : Android tablet Motorola Xoom with version 4.1.2 (Jelly Bean (API level 16)).
5. Key : Google APIs Key.

# 3. Data Format

Data used by MAD mobile application are saved map into image, file with JSON or RDF/XML format. Figure 7 shows example of saved map into image, where in image, there will be mark point to indicate the location of facilities in that area. With this image, user can still go to the facility location when there is no cellular network or internet.



**Figure 7** – Saved Map into Image.

List of facilities shown in the application are data downloaded from POS or IS, where the data format are in JSON or RDF/XML. The data will have same main structure between JSON and RDF/XML format. So, the application will parse the data and fit it into data structure for the facility. Example data structure with JSON format can be seen on Figure 8, for this example it will only use one data example.

[

{

"ID": "FAC663776",

"Name": "北投區區民活動中心",

"Type": "活動中心",

"Category": "Shelter(Indoor)",

"District": "北投區",

"Address": "臺北市新市街30號6樓",

"Telephone": "2891-2105#221",

"Latitude": "25.1324256",

"Longitude": "121.5030005",

"MoreInfo": "Null"

}

]

**Figure 8** – Data Structure with JSON format.

Example data structure with RDF/XML format can be seen on Figure 9, for this example it will only use one data example. The full example of data structure with JSON or RDF/XML format can be seen on github with this link: "http://gitlab.iis.sinica.edu.tw/gitlab/sp4\_vr/mobileapplication-git/tree/master/DataFormat".

<rdf:Description rdf:about="http://openisdm.com/MAD/facility/FAC560430">

<ns1:hasName>石牌國中</ns1:hasName>

<ns1:moreInfo>Null</ns1:moreInfo>

<ns1:hasType>學校</ns1:hasType>

<ns1:hasAddress>臺北市北投區石牌路1段139號</ns1:hasAddress>

<ns1:hasDistrict>北投區</ns1:hasDistrict>

<ns1:longitude>121.5141752</ns1:longitude>

<ns1:hasTelephone>28224682#250</ns1:hasTelephone>

<ns1:hasCategory>Shelter(Indoor)</ns1:hasCategory>

<ns1:latitude>25.1149915</ns1:latitude>

</rdf:Description>

**Figure 9** – Data Structure with RDF/XML format

The data structure will have property ID, Name, Type, Category, District, Address, Telephone, Latitude, Longitude, and More Info. For the early version, property will be used in the mobile application are Name, Type, Category, District, Address, Telephone, Latitude, and Longitude. Property Name, Type, District, Address, and Telephone are the details of facility that will be shown to the user. For property Category, it will be used to grouping several type of facility and use same logo for the picture to indicate the facility are same category. And for property Latitude and Longitude it will be used as facility location when user want to show it on Google Maps on Maps Feature.

Property Category are group for several type of facility that have same feature or function. There are 8 kind of category, they are:

1. **Shelter(Indoor)**. It is include school, gym, activity center, church, temple, community center, District office, etc.
2. **Shelter(Outdoor)**. It is include park, stadium, playground, river bank, etc.
3. **Medical treatment**. It is include hospital, pharmacy (drugstore), clinic, etc.
4. **Rescue**. It is include police station, fire station, army camp, etc**.**
5. **Livelihood**. It is include market, convenient store, grocery store, water supply site, etc.
6. **Communication**. It is include Wi-Fi area, booth, etc.
7. **Volunteer association**. It is include Red Cross, civilian rescue team, etc.
8. **Transportation**. It is include MRT, bus station, bike supply site, gas station, etc.

# 4. Future Works

The early version of MAD mobile application have main basic function for working MAD. But, there is some another feature that can bring more improvement for the application. The following list are the feature can be done for the new version of MAD mobile application:

1. Download data automatically from IS using cellular network and Wi-Fi connected to internet.

In the early version, MAD mobile application will only download data from POS with local Wi-Fi only. So in the future, user can choose to download data directly from POS or IS when cellular network or Wi-Fi connected to internet are available.

User does not need to download data manually, but instead, application will check data version of POS or IS data with data version stored on mobile device storage. If data version of POS or IS are newer than data stored on mobile device storage, application will download automatically, backup the old data, and then use the new downloaded data.

So, Settings feature will be remove and implement the download data on the application background. Application will manage the data automatically as the application need and data still will be removed when user remove the application.

1. Peer to Peer chat between mobile devices.

MAD mobile application objective are to help user go to the facility location when disaster occur. Almost all user always carry their mobile devices whenever they go, MAD mobile application in their mobile devices can be mini POS for other people. When disaster occur, cellular network or internet can be not working properly and people need to get the latest data facility for MAD mobile application in their area. User can go to the near POS location and using local Wi-Fi to download latest data, but with this functionality, user can find nearby user and chatting without using cellular network or internet, but with Wi-Fi direct and they can share their data facility for MAD mobile application.

With this feature, user can make small community to communicate and know condition other facility and people who stay in the facility. So, when other facility need help, like medicine or food, user in other facility who have plenty food or medicine can go to the other facility who need it.

1. Offline Maps and GPS Navigation.

MAD mobile application using maps to show facility location and user can use navigation system to go to the facility location. But, when disaster occur and user cannot use cellular network or internet, maps feature will lost the functionality, because maps need internet connection to show the maps and navigate. With the current Google Maps version, user cannot get offline maps and the GPS navigation. As for now, Google still develop new feature to save maps into offline maps and can be used without internet connection. This can be good news when user need to go to the facility location but they don’t have internet.

For other options, there are offline maps application by Nokia HERE Maps, user can download directly full offline maps for country where they live and the GPS navigation can be done offline too. But this application, for now, only available on Windows Phone or other Nokia smartphones, and with other following news, Nokia HERE Maps will be developed into Android and iOS.

When this feature achieved, other feature such as Image feature can be obsolete and can be removed from the MAD mobile application.

1. MAD mobile application for iOS and Windows Phone.

As mention in the beginning, the early version of MAD mobile application are developed with Android. The goal of this application are can be used by many user with same or different mobile Operating System. So, this application need to be developed into other mobile Operating System, such as iOS and Windows Phone. The feature and structure will be same for mobile application on each mobile Operating System.

# 5. Package Reference

For other package reference, there is other application, library or SDK that can be used as reference to develop MAD mobile application:

1. Google APIs Key

To access the Google Maps servers with the Maps API, developer have to add a Maps API key to their application. The key is free, developer can use it with any of applications that call the Maps API, and it supports an unlimited number of users. Developer can obtain a Maps API key from the Google APIs Console by providing application's signing certificate and its package name.

Details can be seen on this link:

1. http://www.androidhive.info/2013/08/android-working-with-google-maps-v2/
2. https://developers.google.com/maps/documentation/android/start#installing\_the\_google\_maps\_android\_v2\_api
3. AllJoyn

Create peer-to-peer and multi-screen experiences by enabling your apps to connect, control and share resources with nearby apps and devices via the AllJoyn software connectivity and services framework. AllJoyn is an open, secure and universal framework and core set of services that enable interoperability among nearby devices, products and applications across platforms and operating systems.

Details can be seen on this link:

1. https://developer.qualcomm.com/mobile-development/create-connected-experiences/peer-peer-alljoyn
2. OpenGarden

With Open Garden's new SDK, enable your users with an entirely new way to play your games and interact with your applications, just like it does for Firechat. Unlock cutting-edge multi-player P2P connectivity. No user intervention required.

Details can be seen on this link:

1. http://opengarden.com
2. Multipeer Connectivity Framework

The Multipeer Connectivity framework provides support for discovering services provided by nearby iOS devices using infrastructure Wi-Fi networks, peer-to-peer Wi-Fi, and Bluetooth personal area networks and subsequently communicating with those services by sending message-based data, streaming data, and resources (such as files).

Details can be seen on this link:

1. https://developer.apple.com/library/ios/documentation/MultipeerConnectivity/Reference/MultipeerConnectivityFramework/\_index.html
2. http://www.appcoda.com/intro-multipeer-connectivity-framework-ios-programming/
3. http://www.appcoda.com/intro-ios-multipeer-connectivity-programming/