UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

**UNDERGRADUATE SCHOOL**

****

Research and development

**BACHELOR THESIS**

By

Tung Son Vu

Information and Communication Technology

**ANDROID APPLICATION DEVELOPMENT FOR IMAGE, VOICE AND TEXT AQUISITIONS**

Supervisors: Dr. Tran Giang Son

Dr. Bruno Lescalier

***ICTLab, University of Science and Technology of Hanoi***

**Hanoi, July 2016**

Hanoi, September 2013

**Hà Nội, Sept. 2012**

INDEX OF CONTENTS

[ACKNOWLEDGEMENTS iv](#_Toc456082047)

[LIST OF ABBREVIATIONS v](#_Toc456082048)

[LIST OF FIGURES vi](#_Toc456082049)

[ABSTRACT vii](#_Toc456082050)

[I/ INTRODUCTION 1](#_Toc456082051)

[**1.** **Motivation** 1](#_Toc456082052)

[**2.** **Strategy** 2](#_Toc456082053)

[II/ ANALYSIS AND DESIGN 3](#_Toc456082054)

[**1.** **Technology and Literature Review** 3](#_Toc456082055)

[**1.1** **Android** 3](#_Toc456082056)

[**1.2** **Open Street Map** 4](#_Toc456082057)

[**1.3** **OSMDROID** 4](#_Toc456082058)

[**1.4** **Mobile Atlas Creator** 5](#_Toc456082059)

[**1.5** **Geocoder** 5](#_Toc456082060)

[**1.6** **Camera** 6](#_Toc456082061)

[**1.7** **AudioRecord** 6](#_Toc456082062)

[**1.8** **JSON** 6](#_Toc456082063)

[**1.9** **GeoJSON** 7](#_Toc456082064)

[**1.10** **File Transfer Protocol** 8](#_Toc456082065)

[**2.** **Analysis** 9](#_Toc456082066)

[**2.1** **Use Case 1: Add New Site** 9](#_Toc456082067)

[**2.2** **Use Case 2: Fill survey form** 10](#_Toc456082068)

[**2.3** **Use Case 3: Record Audio** 10](#_Toc456082069)

[**2.4** **Use Case 4: Take Pictures** 11](#_Toc456082070)

[**2.5** **Use Case 5: Write Transcript** 12](#_Toc456082071)

[**2.6** **Use Case 6: Send Data** 12](#_Toc456082072)

[**3.** **Architecture and Components** 12](#_Toc456082073)

[**4.** **UI Design** 14](#_Toc456082074)

[**4.1** **First UI Prototype** 14](#_Toc456082075)

[**4.2** **Second UI Prototype** 14](#_Toc456082076)

[**4.3** **Third UI Prototype** 15](#_Toc456082077)

[**4.4** **Mockups** 16](#_Toc456082078)

[III/ RESULTS AND EVALUATION 18](#_Toc456082079)

[**1.** **Screen Shots** 18](#_Toc456082080)

[**1.1** **Adding New Site** 18](#_Toc456082081)

[**1.2** **Survey Form** 18](#_Toc456082082)

[**1.3** **Recording Interface** 19](#_Toc456082083)

[**1.4** **Photo Interface** 20](#_Toc456082084)

[**1.5** **Transcript Interface** 21](#_Toc456082085)

[**1.6** **Send Data** 22](#_Toc456082086)

[**2.** **Evaluation** 23](#_Toc456082087)

[IV/ CONCLUSION 24](#_Toc456082088)

[REFERENCES 25](#_Toc456082089)

# ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my supervisors, Dr. Tran Giang Son and Dr. Bruno Lescalier or their patient guidance, enthusiastic encouragement, and useful comments through out my project. Without their assistance and advices, I would not keep my progress on schedule.

I also would like to thank my teachers, staffs and classmates from University of Science and Technology of Hanoi for the assistance and constant support provided by them on this project.

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| SWARMS | Say and Watch: Automated image/sound Recognition for Mobile Monitoring Systems |
| USTH | University of Science and Technology of Hanoi |
| ICT | Information and Communication Technology |
| IDE | Integrated Development Environment |
| SDK | Software Development Kit |
| UI | User Interface |
| OSM | Open Street Map |
| MOBAC | Mobile Atlas Creator |
| JSON | JavaScript Object Notation |
| FTP | File Transfer Protocol |

# LIST OF FIGURES

Figure 1: Architecture of Android Platform (Android Architecture, 2015) 3

Figure 2: SWARMS Application Use Cases 9

Figure 3: System Architecture 13

Figure 4: First UI Prototype 14

Figure 5: Second UI Prototype 15

Figure 6: Third UI Prototype 16

Figure 7: Mockup Using Basamiq 16

Figure 8: Mockup using Sketch 17

Figure 9: Adding New Site 18

Figure 10: Survey Interface 19

Figure 11: Recording Interface 20

Figure 12: Photo Interface 21

Figure 13: Transcript Interface 22

Figure 14: List of Sites 22

# ABSTRACT

The work of this project is a part of project SWARMS of the ICTLab University of Science and Technology of Hanoi. The purpose of this project is to build an Android application to gather information of place of interests (construction sites, pagoda, etc.). Photos, descriptive voice and text data of the designed places are recorded into Android devices before uploaded to project central server for further analysis. The project is developed by using Android Studio Integrated Software Development Kit (IDE) and Android Software Development Kit (SDK). The project used Android supporting libraries: OSMDROID, MOBAC, Apache Commons Net, GeoJSON and native Android libraries: Camera, Geocoder, AudioRecord, JSON to create the application with friendly and convenient User Interface (UI).

# I/ INTRODUCTION

The project is integrated with SWARMS (Say and Watch: Automated image/sound Recognition for Mobile Monitoring Systems) project held by ICTLab of USTH. The objective of SWARM is to achieve a flexible and real time network monitoring. Modeling and simulation of evolutionary scenarios help training and creating a decision-making support system or creating foreseen scenarios of a phenomenon to monitor. The application of SWARM includes epidemiology (monitoring mosquito larvae in ponds, flower pots, fields, etc.), and urban development modeling (monitoring risk sites, construction sites, etc.). The main idea was to test the possibility of a surveillance network based on smart phones in particular Android devices, where devices are not only used as passive sensor, but can also be used to upload pictures, audio and text information to the monitoring system, allowing participants to analyze and forecast information.

1. **Motivation**

Many field trips have been held to gather data for SWARMS. Participants used Android devices to take photo, record descriptive voice and fill in data form manually. To do those task, participants used 2 different applications: OSMTracker to take photos, get coordinators of the place of interest and Voice Recording to record audio file. This process has many disadvantages including: time consumption which caused by changing between applications, filling in form manually, OSMTracker works inefficiently, sometimes it takes 10-15 minutes to get GPS coordinators. Furthermore, data can not be sent automatically to project server, consequently participants must connect the device with computers and send data through email or external storage. It is not convenient for both participants and project managers.

The idea of this project is to create an Android Application which is user-friendly and take less time gathering all information needed for SWARMS, more importantly, can send data directly to project server for further analysis.

1. **Strategy**

The application is built based on native Android classes: Camera, AudioRecord (for photo and voice recording), GeoCoder (address information) and Android-support libraries: OSMDROID (Map visualisation), Apache Common Nets (send data to server). The UI of the application will be designed based on discussion with SWARMS managers for the best user experience. The mockup applications includes: Sketch, Basamiq are used to pre-design the UI prototypes.

Android Studio is used to create the prototype application. Any changes of the prototype application are based on the comments of the supervisors. After the UI is designed, functions of the application are decided on discussion with the supervisors and be added into the prototype application with the quality requirements from the supervisors. Functions including recording audio, taking photos should be in the highest quality. Map function should be tested in online and offline mode.

After finish creating application prototype, evaluation from supervisors and the application developer will be recorded for future and further development.

# II/ ANALYSIS AND DESIGN

## **Technology and Literature Review**

## **Android**

Androidis a mobile operating system (OS), developed by Google, based on the Open Linux Kernel and designed for touch screen mobile and handheld devices. Furthermore, it utilizes a custom virtual machine, designed to optimize memory and hardware resources in a mobile environment. Android Studio Integrated Software Development Kit allows developers to create mobile applications, taking advantages of all mobile hardware. Application can use any of the phone's core functionality for example: making calls, sending text messages, or using the camera, allowing developers to create best experiences for users.

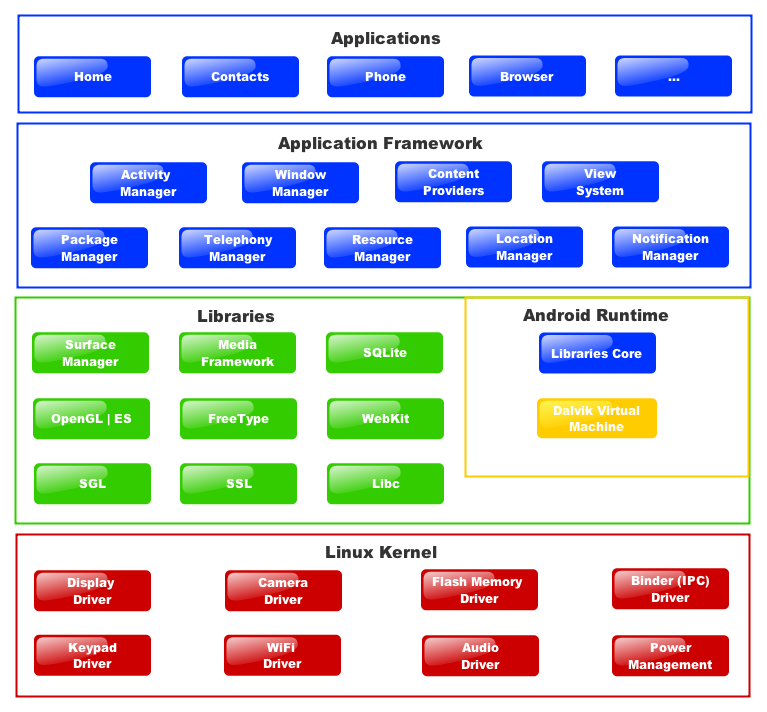


Figure 1: Architecture of Android Platform (Android Architecture, 2015)

Architecture of Android platform is showed in the Figure 1 above.

In the Applications layer, Android distributes with some core applications including SMS, Calendar, Map, Browser, etc. and they are developed in Java.

Application Framework below Applications enables reuse or replacement of components. For example, there is the component of displaying image in one application; other applications can make use of the component when they need.

Libraries include some set of C/C++ libraries used by some components of Android system. Developers can use those libraries through the application framework. The core libraries include SQLite, is lightweight database engine available to all applications.

Android is chosen to develop the application because Android is the platform is used in 80% of all smartphones around the world. The best part of Android is that it is easily to learn and accessible to any programmer with knowledge of Java and the Android SDK. The Android SDK can be easily accessed by detailed documents that are provided by Google Android developers. Android SDK has the potential to create some of the most creative mobile applications.

## **Open Street Map**

OSM is a world online map services and open contents. OSM's purpose is providing geographic data collected from collaborated members around the world. Often called "Wikipedia of maps" with the inspiration of Wikipedia, a website allows collaborative modification its content and structure (Rethams, 2011). The project OSM is created mainly to compete with other companies and government agencies which are providing geographic data under strict used terms.

Members contribute data from their GPS receiver, or draw from the air with supporting application. They upload GPX route data and modified vector data using GIS programs for browsers, desktops, and mobile phones. OSM is also constantly updated through helps of companies, government institutions that often send information of new locations or corrections of the existing ones.

In terms of SWARM project, the participant need to go to not only big cites but also small and new cities. The OSM service have more precise information of places around the world which can be a big advantage while Google map focuses mainly big cities (García, 2013).

## **OSMDROID**

OSMDROID library is an Android library including tools and views to interact with OSM (OSMDROID). It is a good alternative of Android Google Map tools. OSMDROID helps to build custom MapView, adding custom marker, polyline into MapView. OSMDROID using map tiles to display map. Map tiles are map pictures on different zoom levels. OSMDROID provides tools to work with tile providers, load tiles from offline storage or load tiles from tiles server. OSMDROID assists offline and online mapping at the same time, loading missing map tiles of offline storage from tiles server for user full map experience.

OSMDROID developers provides long range of supporting documents and tutorials which help users to implement the library properly. Other alternative library for OSMDROID is MAPSFORGE. Its map tiles have better quality, but it only supports offline mapping and harder to implement due to lacking of supporting documents.

In this project, OSMDROID is used to illustrate map in the application. User can add marker at the place of design. Brief information of place, including site id and site address, appears when the marker is selected.

## **Mobile Atlas Creator**

Mobile Atlas Creator (MOBAC) is open source program provides offline map tiles for handheld or mobile phone application (Welcome to Mobile Atlas Creator). OSMDROID offline map tiles could be created by MOBAC, using large number of online maps providers including OSM map provider (MAPNIK, MAPQUEST, etc.). Maps from MOBAC can be consisted of multiple layers and different map sources and map resolution. Map of specific location, location parameters, zoom level, can be created in user choice.

MOBAC is the only application that allows user to create custom map with map tiles of their design. In this project, MOBAC is used to create Hanoi (capital of Vietnam) offline map with zoom levels from 13 to 18. The offline map file is put under the OSMDROID folder and loaded when the application need to show the map.

## **Geocoder**

Geocoder is a class in the Android framework API. Geocoder provides a way to handle geocoding and reverse geocoding. Reverse coding is getting address of the location from coordinate (latitude, longitude). Reverse geocoding information could not be exact location (may contain only city name or full street name of close by building).

Geocoder is one of wide range geocoding libraries including: MapQuest, Google Geocoding, etc. Geocoder is not the best geocoding service available. However, in the project’s intention, geocoding is only use to get the address of the site. Therefore, Android native class Geocoder is chosen, no need to add libraries from other geocoding service providers.

## **Camera**

The Camera class is managed to taking the picture, start/stop preview. It is a client for the Camera service, which manages the actual camera hardware. It provides only point and shoot, give no control of how the camera work. This is suitable for the project 's intention, getting information only, not working with raw image capture.

## **AudioRecord**

AudioRecord is an Android class to handle voice recording. The output can be raw audio signal, high quality raw audio file (PCIM). It can be easy implemented and converted into other audio file format (WAV by adding headers or other compressed audio file format MP3, ACC, etc. by wide range of support library). AudioRecord is used to create record of site information with high quality.

## **JSON**

JSON, JavaScript Object Notation, a data format to follow a certain rule that most programming languages are readable (Lengstorf, 2009). JSON can act as a record database.

JSON string is enclosed by braces {}. One object started by open bracket { and end with close bracket }. The key is followed by the value and they are separated a colon. Both key and value are put it in double quotes. Single quote is not the correct standard of JSON. Comma is used to separate between two objects.

JSON format example

{

{"firstName":"John", "lastName":"Doe"},  
{"firstName":"Anna", "lastName":"Smith"},  
{"firstName":"Peter", "lastName":"Jones"}

}

One array is one set of values, One array started by open tag [ and ends with the closing tag ]. The values are separated by commas.

JSON Array Example:

{

"employees":[  
{"firstName":"John", "lastName":"Doe"},

{"firstName":"Anna", "lastName":"Smith"},

{"firstName":"Peter", "lastName":"Jones"}  
]

}

One value can be one string in double quotes, or one number, or true or false or null, or is 1 object or 1 arrays.

JSON is used as application database which only stores the current active site ID and sites’ information to display in the map. It is simple and easy to implement. Text file with JSON string format has information of the site that currently need to gather information or edit the site information. Another text file with JSON string format has coordinators and address of recorded sites to show as markers in the map.

## **GeoJSON**

GeoJSON is a format for encoding geographic data structures (GeoJSON). Basically GeoJSON follows JSON format but it has their specific key (type, geometry, properties). Type could be Feature GeoJSON object (location) or can be Feature Collection GeoJSON array (locations) which contains of Feature type g GeoJSON object. Geometry key is position of the data (Point, MultiPoint, LineString, Polygon, etc.). Properties is a JSON object where user can freely add needed information of the location.

GeoJSON Example:

{ "type": "FeatureCollection",

"features": [

{ "type": "Feature",

"geometry": {"type": "Point", "coordinates": [102.0, 0.5]},

"properties": {"prop0": "value0"}

},

{ "type": "Feature",

"geometry": {

"type": "LineString",

"coordinates": [

[102.0, 0.0], [103.0, 1.0], [104.0, 0.0], [105.0, 1.0]

]

},

"properties": {

"prop0": "value0",

"prop1": 0.0

}

}

]

}

After the survey form is filled in, the data is saved into text file with GeoJSON format. The GeoJSON format can be used by wide range of applications in the internet including Google Map. It is also easier for SWARMS managers to gather specific information.

## **File Transfer Protocol**

FTP is a protocol with a set of rules that networked computers use to talk to one another. FTP is the language that computers on a TCP/IP network (internet) use to transfer files to and from each other (Beal). The FTP server may support Active and Passive connections. In an Active FTP connection, the client opens a port and listens and the server actively connects to it. In a Passive FTP connection, the server opens a port and listens and the client connects to server through port. In term of this project, FTP client library (Apache Common Net) help uploading data from the application to FTP server.

## **Analysis**

Figure 2 describes SWARMS Application Use Cases. User and FTP Server are two main actors. User interacts with all main use cases while FTP Server only interacts with Send Data Use Case. Each Use Case will achieve its specific goal which is described further in Use Case document.

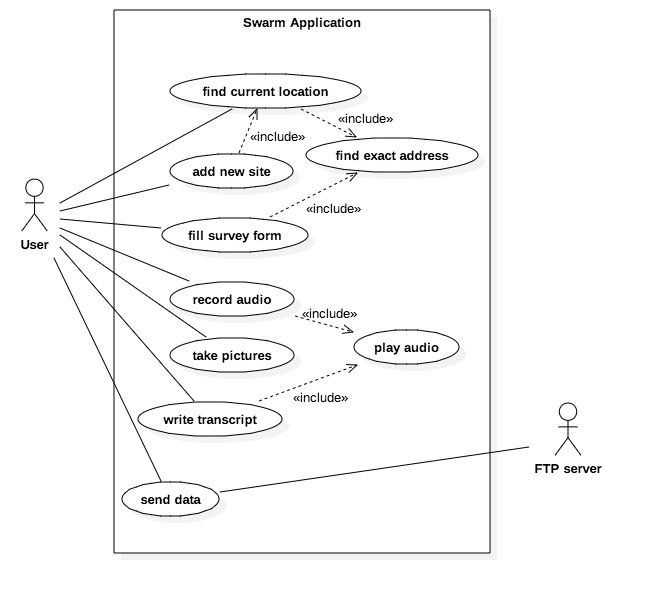
****

Figure 2: SWARMS Application Use Cases

## **Use Case 1: Add New Site**

**Primary Actor: User**

**Goal:** The user to add a new site to gather information

**Trigger:** The user press add-new-site button

**Precondition:** The user is on Map with Function Buttons Interfaces

**Basic Flow:**

1. Checking GPS

Case 1a: GPS is not turn on

1a.1 A dialog appears ask to turn on GPS

1a.2 A GPS setting appear, user chooses to turn on or do not turn on the GPS

Case 1b: GPS is turn on

1. New Site is active

Case 2a: connected to Internet

2a.1 the address obtained automatically

2a.2 show address in the toolbar, marked as active

Case 2b: not connected to Internet

2a.1 show “New Site” in the toolbar, marked as active

## **Use Case 2: Fill survey form**

**Primary Actor: User**

**Goal:** The user to add text information of the site

**Trigger:** The user press survey button from Map Interface with Function Buttons or swipe from Recording Interface

**Precondition:** The user is on Map with Function Buttons Interfaces or Recording Interface

**Basic Flow:**

1. User fill in the form
2. Save information when user un-focused edit text or choose a radio button
3. Update the address
4. Show saved information to the survey form

## **Use Case 3: Record Audio**

**Primary Actor: User**

**Goal:** The user to add a descriptive audio file

**Trigger:** The user press survey button from Map Interface with Function Buttons or swipe from Survey Interface or Photo Interface

**Precondition:** The user is on Map with Function Buttons Interfaces or Survey Interface or Photo Interface

**Basic Flow:**

1. User press record button
2. User speak to microphone
3. User press stop button
4. User decide to save file or not

Case 4a. No file recorded

User press save button, the file is saved

User press cross button, the file is not saved

Case 4b File recorded previously

User Press Save Button, a dialog asks if user want to overwrite the previous file

User Press Cross Button, the old file would not be overwritten

## **Use Case 4: Take Pictures**

**Primary Actor: User**

**Goal:** The user to add photos of the place of interest

**Trigger:** The user press photo button in Photo Interface

**Precondition:** The user is on Map with Function Buttons Interfaces, or in Recording Interface or in Transcript Interface

**Basic Flow:**

1. The user is brought to the "Take Photo" screen.
2. The user selects the “Take Photo” button in order to capture an image.
3. ***Case 3a:***

The cook presses the “Save Photo” button.

3a.1 The image is saved photo list and the user is returned to the Photo List screen.

***Case 3b:***

The cook presses the “Do Nothing” button.

3b.1 The image is saved photo list and the user is returned to the photo List screen.

***Case 3c:***

The cook presses the “Back” button.

3a.1 The image is saved photo list and the user is returned to the photo List screen.

## **Use Case 5: Write Transcript**

**Primary Actor: User**

**Goal:** The user to add a transcript from audio recording

**Trigger:** The user press transcript button from Map Interface with Function Buttons or swipe from Photo Interface

**Precondition:** The user is on Map with Function Buttons Interfaces or Photo Interface

**Basic Flow:**

1. User fill in text into the edit text box
2. Application save text into text file when user touch outside the edit text box

## **Use Case 6: Send Data**

**Primary Actor: User**

**Goal:** The user chooses data to central server

**Trigger:** The user press List of Site in the application drawers

**Precondition:** The user press drawer menu

**Basic Flow:**

1. User choose sites from the list view
2. Press send button in the toolbar
3. A dialog appears and shows upload in progress
4. A notification when all upload progress is finish

## **Architecture and Components**

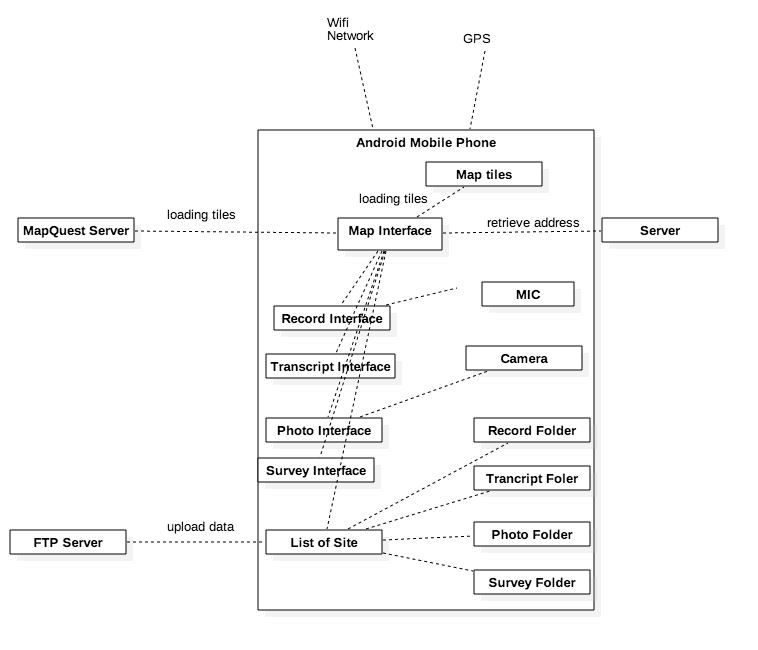


Figure 3: System Architecture

The diagram above presents the general architecture of the prototype. The application is built with an assumption that users use their Android phones in the environment with wireless network, 3G network, having the ability of getting GPS data and can work in offline situation. Map Interface is the core and the start of application. Map Interfaceimports Map Quest’s map tiles and retrieves information of address from GeoCoder server. Map Interface calls map overlayto add marker to the map. From Map Interfaceusers can go to other Interface to have more functions such as taking photos, recording audio and so on.

Recoding Interface interacts with the microphone of the device, records descriptive voice audio files and saves the files to Record Folder. Photo Interface interacts with the camera of the device, taking photo files and saves the files to Photo Folder. Survey Interface interacts with the edit text box, save data to text files and saves the files to Survey Folder. Transcript Interface interacts with the edit text box, save data to text files and saves the files to Transcript Folder. List of Site Interface upload all folders and the data files inside to FTP Server.

## **UI Design**

## **First UI Prototype**

The First UI Prototype is shown in Figure 4 below

The application starts with adding site interface where user adding new site to a list of sites. For each site of the list, user can choose whether to delete the site or send the site to Server. Any action pops up a dialog to conform user’s intention. When click to any site, a Site Control Interface appears. Site Control Interface contains of big buttons to access other interface to gather information include: Record Interface, Photo Interface, Map Interface, Transcript Interface, Survey Interface. User can come back to Site Control Interface any time with Back button.

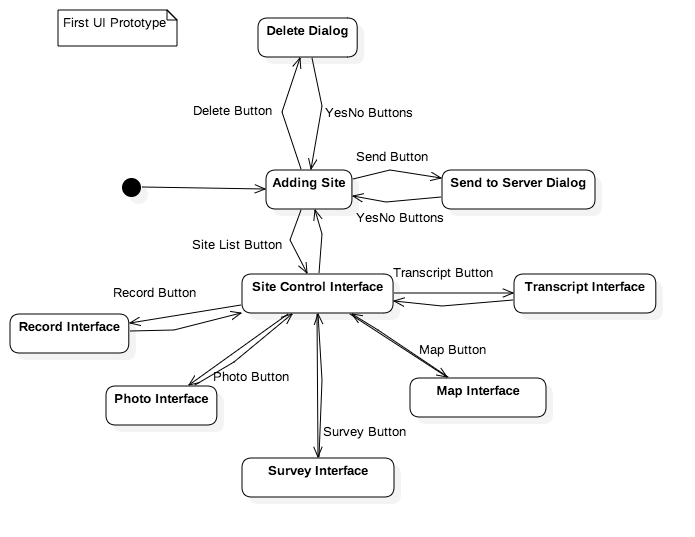


Figure 4: First UI Prototype

## **Second UI Prototype**

The Second UI Prototype is shown in Figure 5 below

The application starts Introduction Wizards where users could get information of the project then user is required to add some of user’s team information. After Enter Team Information, user can go to Adding Site with similar functions as in the First UI Prototype.

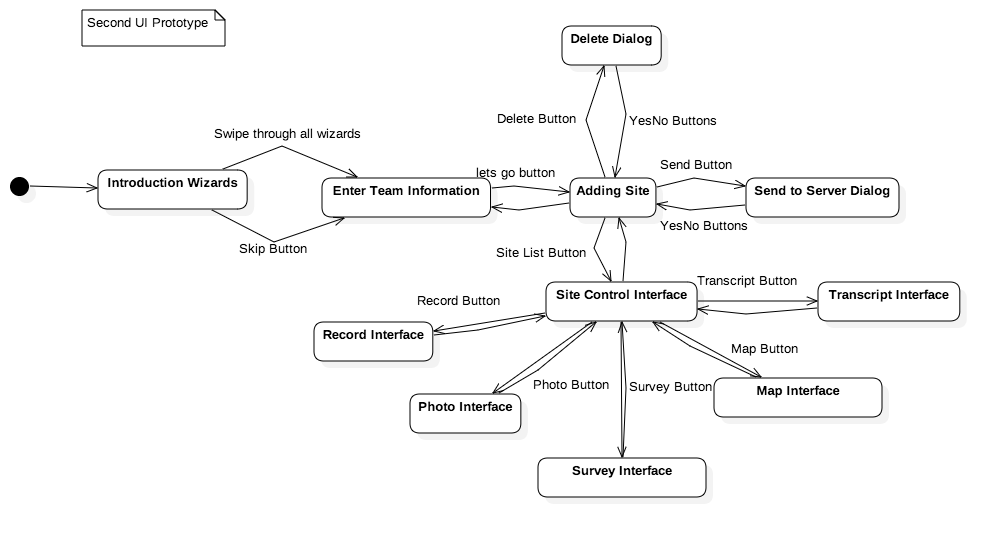


Figure 5: Second UI Prototype

## **Third UI Prototype**

The Third UI Prototype is shown in Figure 6 below

The Third UI Prototype is a new approach. Second UI Prototype has many buttons, and is not the best user UI experience. Third UI Prototype uses swipes along with buttons for faster interaction between users and the system. The application starts with Map Interface with Login Button. Login Interface appears when user clicks on Login Button in the Map Interface. After logging in to the system, user can interact with Map Interface with Function Buttons where user can gather specific information. From each function interface, user can swipe left or right to go to other function. When user press the menu button at any time, a drawer appears and allows user to access to List of Site where user can choose sites to upload data to server.



Figure 6: Third UI Prototype

## **Mockups**

Basamiq is a mockup tool helping pre-design UI:

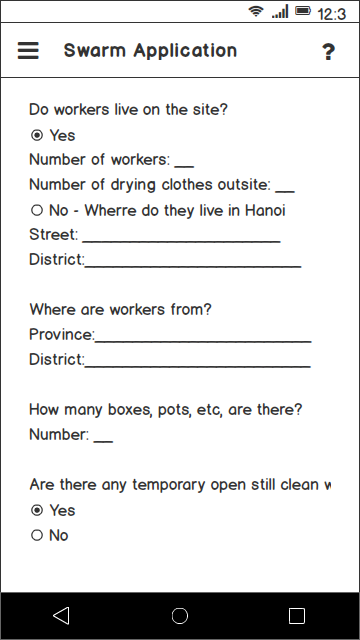
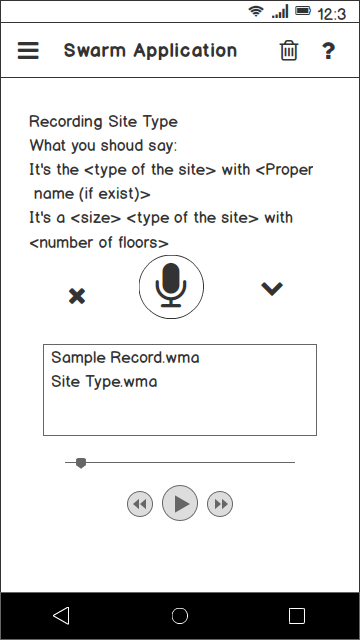
 

Figure 7: Mockup Using Basamiq

Sketch is a mockup tool helping pre-design application UI:

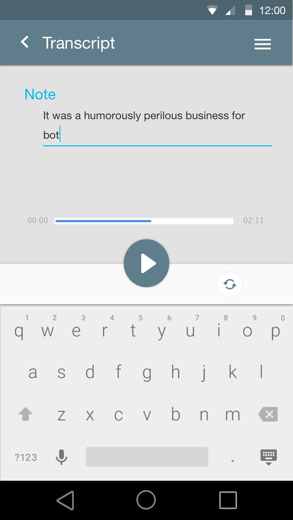
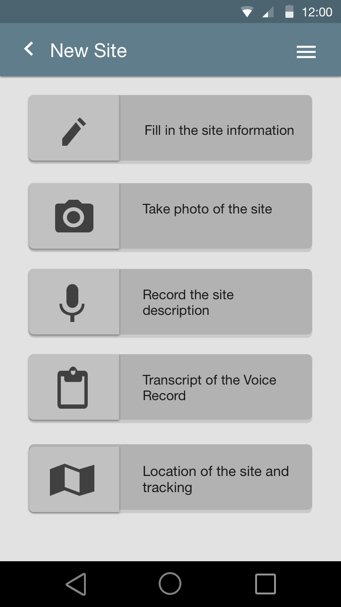
 

Figure 8: Mockup using Sketch

Balsamiq has a black-and-white design, it doesn't distract users in early project discussions and maintains lower expectations on fidelity. Balsamiq helps users create early UI design with low effort. Sketch has a colorful design and it is useful for higher fidelity mocks. Sketch helps users to create reusable designs for visual design and interactive prototyping. It also has a vector based, users have ability to scale up/down the designs.

In terms of this project, Balsamiq is used to create early visualization of UI Prototypes to discuss with the project managers and supervisors. After discussion, if no changes need to be made from the design, Sketch is used to create more detail visualization of UI Prototypes with colors before developing application with Android Studio Integrated Software Development Kit.

# III/ RESULTS AND EVALUATION

## **Screen Shots**

## **Adding New Site**

Map Interface with Function Button is shown in the Figure 9 below.

The left image shows the current address by pressing the Current Location Button. The right image shows the new site is added into the system when user click Add New Site Button in the bottom right of the screen. If the system can not retrieve the current address, then “New Site” appears in the pop-up marker and the tittle bar.

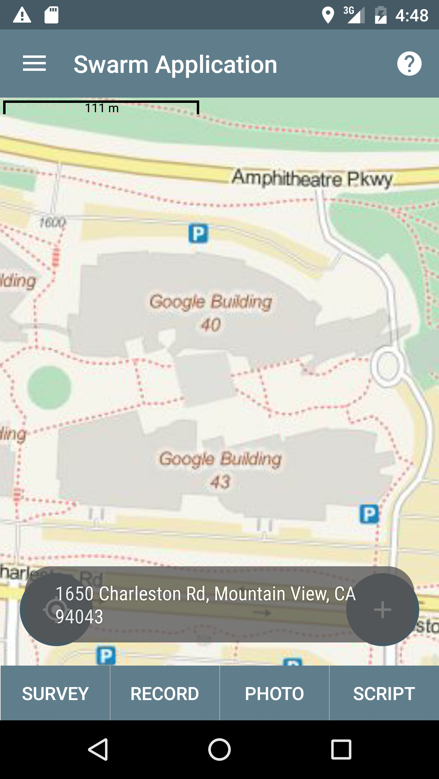
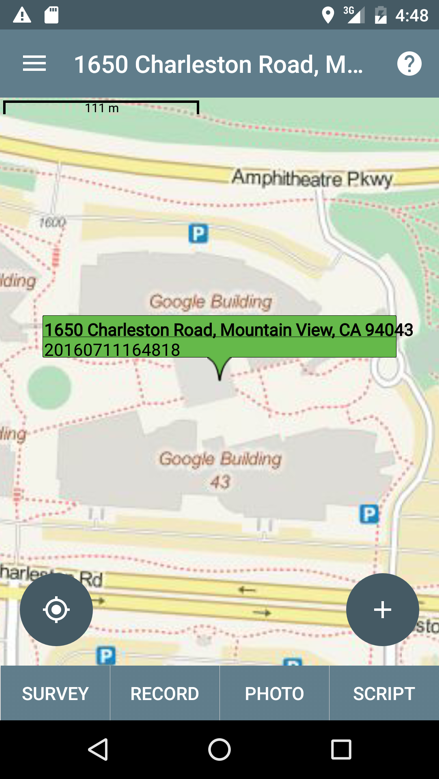
 

Figure 9: Adding New Site

## **Survey Form**

Survey Interface is shown in the Figure 10 below.

The retrieved address from Map Interface automatically add into site address in detail. Any changes from Site address and others fields will be saved in to text file.

Users can swipe up - down to go to other Survey Interface parts or swipe left – right to go to other functions.



Figure 10: Survey Interface

## **Recording Interface**

Recording Interface is shown in the Figure 11 below.

The application starts recording new audio file when user press the Record Button and ends when Stop Button is clicked. Only one file is allowed, so if user press Record Button again, the user will be asked whether they want to overwrite the old file. Users can play back the recorded audio any time they want and they can interact with the audio player through buttons and seekbar.

Users can swipe up - down to go to other Recording Interface parts or swipe left – right to go to other functions.

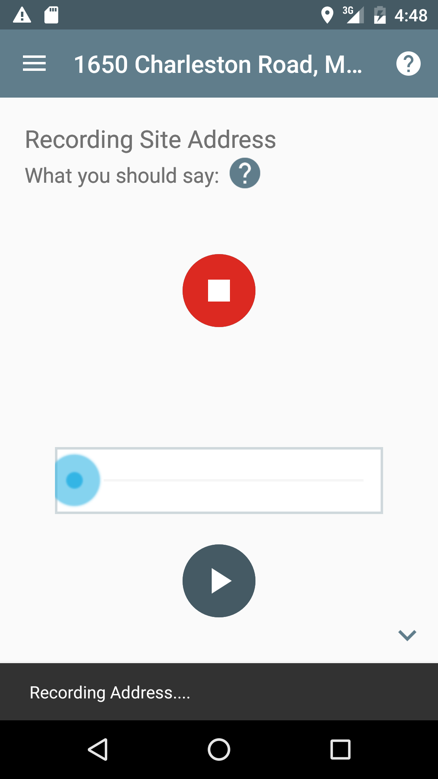
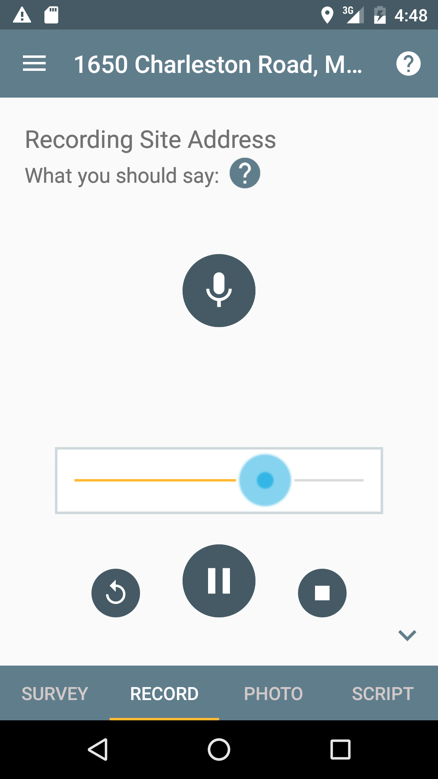
 

Figure 11: Recording Interface

## **Photo Interface**

Photo Interface is shown in the Figure 12 below.

The system camera preview appears when users click Taking Photo Button. When the photo is saved, a thumbnail appears in the grid view of Photo Interface.

Users can swipe swipe left – right to go to other functions.

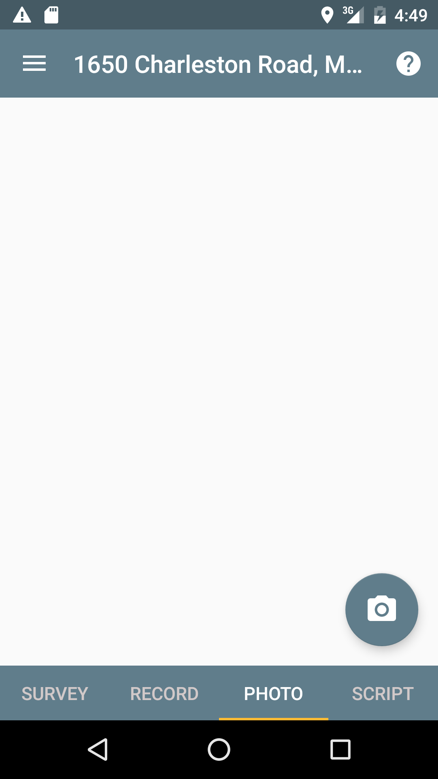


Figure 12: Photo Interface

## **Transcript Interface**

Survey Interface is shown in the Figure 13 below.

The users type in transcripts of recorded audio files. Any changes from Transcript edit text will be saved in to text files.

Users can swipe up - down to go to other Transcript Interface parts or swipe left – right to go to other functions.

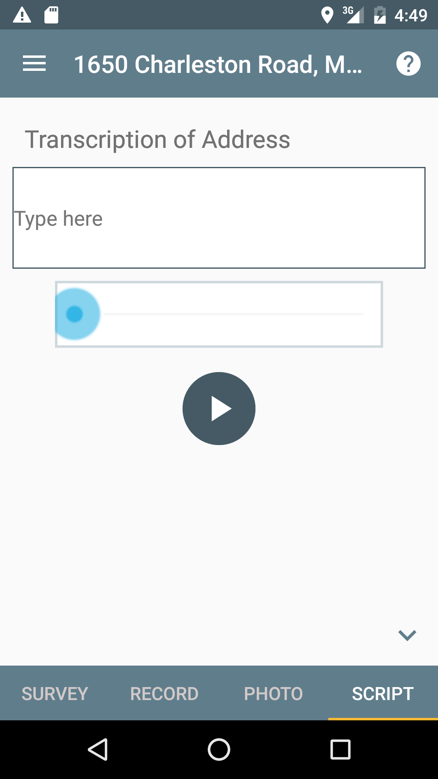


Figure 13: Transcript Interface

## **Send Data**

List of Sites Interface is shown in the Figure 14 below.

The List of Site is shown. Users can choose site to upload to server.

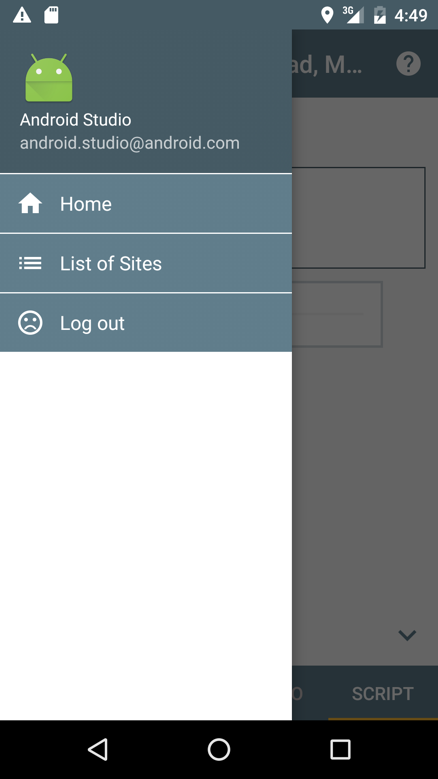
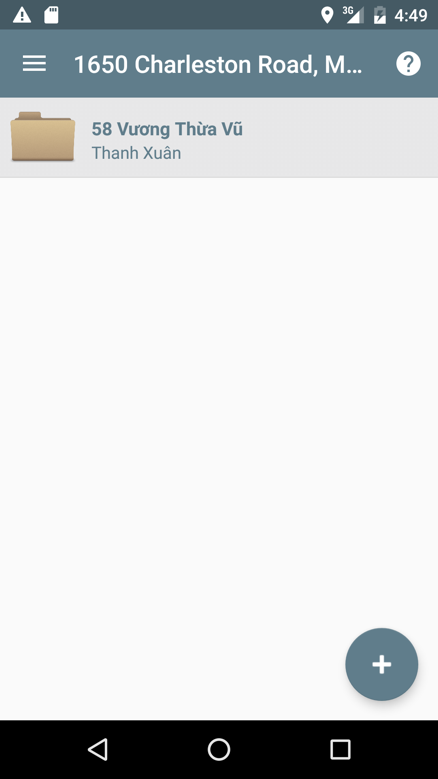
 

Figure 14: List of Sites

## **Evaluation**

Early on in the project, it was stated that the aim of this project was to develop a mobile which is easy to use and take less time consumption of user. By taking a test with real field trip, all data was recorded in good quality and user took short amount of time to gather information, it could be determined that the application and project succeeded in its goal.

However, the application need to be optimised in the future, some technique problems occurred rarely. For example: the application crashed when swipes between screens, the GPS sometimes take very long time to get coordinators.

Nonetheless, the map quality is not good, the application developer need to find another map source with better quality. The performance of both the ftp server and client has been fully tested with the local server, needed to test in online server.

# IV/ CONCLUSION

In conclusion, the application developed has successfully completed its goal, as can be determined by the success of the evaluation session.

This project has allowed me to apply the knowledge that I gained from the Human Computer Interaction course, the android application development I learnt through the Mobile development courses, and development skills I learnt throughout university life. One of the biggest skills developed through the project is time-management, including planning during the early stages of development, tracking the progress and adjusting schedule according to plan.

The application would be completed and published into the Android App Store after testing in FTP online server, with user logging-in function. It would be a useful tool as part of SWARMS project’s monitoring network.

# REFERENCES

*Android Architecture*. (2015, Febuary 3). Retrieved June 30, 2016, from Eazytutz: http://www.eazytutz.com/android/android-architecture/

Beal, V. (n.d.). *FTP - file transfer protocol*. Retrieved June 30, 2016, from Webopedia: http://www.webopedia.com/TERM/F/ftp.html

García, D. (2013, June 23). *OpenStreetMap: an alternative to Google Maps*. Retrieved July 2, 2016, from Webilop: http://www.webilop.com/openstreetmap-an-alternative-to-google-maps/

*GeoJSON*. (n.d.). Retrieved June 30, 2016, from GeoJSON: http://geojson.org

Lengstorf, J. (2009, July 2). *JSON: What It Is, How It Works, & How to Use It*. Retrieved June 30, 2016, from Copterlabs: http://www.copterlabs.com/json-what-it-is-how-it-works-how-to-use-it/

*OSMDROID*. (n.d.). Retrieved June 30, 2016, from http://wiki.openstreetmap.org/wiki/Osmdroid

Rethams, D. (2011, May 12). *What is OpenStreetMap?* Retrieved 6 30, 2016, from https://derickrethans.nl/what-is-openstreetmap.html

*Welcome to Mobile Atlas Creator*. (n.d.). Retrieved June 30, 2016, from Mobile Atlas Creator: http://mobac.sourceforge.net