EVENT PARTICIPATION TRACKING FRAMEWORK

BY

CHEONG PEI BIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2016

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: _____
_____
_____

Academic Session: _____

I _____
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____
(Author's signature) (Supervisor's signature)

Address:
_____
_____ _____
_____ Supervisor's name

Date: _____ **Date**: _____

EVENT PARTICIPATION TRACKING FRAMEWORK

BY

CHEONG PEI BIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2016

DECLARATION OF ORIGINALITY

I declare that this report entitled "**EVENT PARTICIPATION TRACKING FRAMEWORK**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to express my gratitude and appreciation to my supervisor, Dr. Ooi Boon Yaik who has guided me throughout the process of completion of this report and also provided an Android device to me for development and testing purposes.

ABSTRACT

The objective of this project is to develop a framework that can track and analyze the patterns and participations of event attendee at location-based events.

This framework takes a different approach compared to existing popular event tracking system. Instead of using additional hardware, this framework aims to use only the device everyone has: smartphones. Besides, this framework not only can track participant attendance, but also their mobility patterns inside the event venue.

This framework utilizes Wi-Fi capability of smartphones and existing Wi-Fi access points in the event venue, which allowing the tracking of the behavior of participants throughout the event. The framework allows event organizers to create POI (Point of Interest) inside the event venue. Once the participants walk through the POI, the framework can track this and send this data back to event organizers.

This framework also aims to eliminate the queuing time in the check-in process, by automatically launch the app in the background and start scanning for Wi-Fi access points. If any Wi-Fi access points set by the event organizers is found, the participant is considered checked-in.

After the development and testing, the framework is working and able to track users' attendance and participation pattern, such as duration of user staying in the event, visitation of POI and so on.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

POI	Point of Interest
WAP	Wi-Fi access point

CHAPTER 1 INTRODUCTION

1-1 Motivation and Problem Statement

Traditionally, event attendance tracking has been tedious, troublesome and slow, as it involves attendees queuing to wait for their turns in order to write down their names using pen and paper.

But now, almost every event attendance tracking system has been modernized. No more pen and paper. Event check-in counters often use a desktop PC or tablets to quickly scan through attendees' smartphone. Attendees' smartphone often contains special QR codes or ticket code which can be easily recognized by the check-in counters. This method has undeniably revolutionized the traditionally pen and paper check-ins. However, as we marked into a new age where our smartphones keep getting better and more powerful, there are certain expectations that current event tracking systems do not fulfill.

Firstly, although the current event tracking system uses technology to simplify the tasks, such as quickly identified an attendee from database and eliminate the need of a paper ticket, the speed and efficiency of the attendee check-in still depends on the number of check-in counter available. Attendees are still required to queue in front of the entrance of event venues. Which is, the current event tracking system has made the tasks of event organizers faster and more efficient, but not event attendee's.

Secondly, the current event tracking system uses additional hardware or device in order to track the attendance of participants. Desktop PC or tablets or KIOSK system are used. Attendees are required to show their custom made identity card or their smartphone. This situation is less ideal, because the cost of preparing the check-in counter is directly proportional to the number of participants of the event.

Thirdly, the current event tracking system only tracks for participant attendance, in other words, whether a particular people attends the event. This information is clearly inadequate for event organizers who want to analyze their participants and improve their events in the future. Most event organizers would want to know the engagement rate and participation pattern of their attendees, such as how long do they stay in the events, which section of the event is more engaging and the visitation pattern of the participants.

Thus, there is a need for a system to solve the problems stated above. The system should collect various kinds of data in order to address the problems. However, solving the problems isn't the only motivation to conduct this project.

Analysis can be performed on the data that the system collects to understand human crowd behavior. Understanding the pattern and engagement rate for event participation is not only interesting from the view-point of event organizing, but also has significant advance in the design of personalized event suggestion system.

Additionally, it is both interesting and motivated to understand human's attendance and participation pattern in location-based events in order to provide insights to the factors influencing the participation pattern.

1-2 Objectives

The objective of this project is to develop a framework that can track event participants' attendance securely and easily, which doesn't require user to launch mobile app and queuing up at the entrance. Besides, this project is also aims to be able to analyze the participations patterns of event attendee, such as duration of staying in the event, visitation pattern and popular event section, at location-based events.

Besides, the data collected using the framework can be analyzed to gain understanding the collective dynamics of user participation in organized events such as exhibition, concert, festivals. This can further provide critical insights that can aid in personalized event suggestion, venue layout planning and targeted advertising.

1.3 Proposed Approach

The main delivery of this project is not a front-end system but a back-end framework which can be easily implemented into various system applications. This approach is chosen as one application certainly cannot serve all kinds of possible use of the framework. Thus, the core libraries are packaged into one framework where other application developers can easily implement the framework into new projects and develop new functionalities.

The framework utilizes the Wi-Fi capability of smartphones and existing Wi-Fi access points in the event venue. The framework scans for Wi-Fi access points nearby and determine if the participant is in the event venue. This eliminates any additional hardware required to setup in the event venue, as the framework uses only existing Wi-Fi access points and user-carry smartphone. This approach also eliminates the need for participants to take out their smartphones and queuing at the entrance just for their QR codes in their smartphones to be scanned.

For the current event tracking system, a human is consider attended the event if he or she checks in. Event analysis so far has been limited to the number of check-ins without considering mobility aspects of users, especially during large-scale events.

Event analysis and tracking should be more efficient and cover more mobility aspects of participants, which provides valuable insights to the reason behind the user engagement and participation patterns.

The proposing framework can collect data more than just the attendance of a particular attendee. It can track various engagement data, such as when do users check-in and when do they leave, the visitation pattern, which section of the event where the users stay longer and more frequent, and so on. The framework allows event organizers to create POI (Point of Interest) inside their event venue. The framework can track which users walk through certain POI and sends the data to organizers. All of these are done without any

additional hardware being setup in the event venue, which provides ease of use and mobility to the event organizers and to the users.

1.4 Impact, Significance and Contribution

With the rise in popularity of location-based events such as concerts, car shows and electronics exhibition, event organizers are very keen to know more about their audience, how to promote engagement among participants, how to improve their event layouts, so that they can do a better job at their next events.

With such data collected and analyzed, it can contribute to the several important applications in marketing, event recommendation system and event planning.

CHAPTER 2 LITERATURE REVIEW

2-1 SocialTables

SocialTables (SocialTables 2016) is one of the many popular startups providing event management software services. SocialTables is recognized with their all-in-one event management software, including desktop web application and mobile app. One of the features they provide is mobile check-in. Event organizers can prepare check-in counter loaded with SocialTables mobile app. When the participants arrive, the check-in staffs search for their names. Once found, the check-in staffs mark the participant as arrived in the app. This approach is very simple, even for an untrained staff. However, this approach is not ideal. First, the security is weak as the system cannot securely authenticate the participant as there is no any digital or physically authentication needed. Second, the overall check-in process is slow, as the participants are required to queue at the check-in counter. Moreover, the speed of process is limited by the number of check-in counter available in the events. Third, the system only tracks for participant attendance. There is little data available for analyze for event organizers to improve their events. A better approach would be implementing digital security authentication into the systems. If not, the entrance of participants can be easily compromised by someone who knows their names. Besides, the speed of the check-in process should not be limited by the number of check-in counters available. The check-in should be as intuitive as if the check-in is automatic once the participants enter the event venues. Additionally, the tracking of attendees should not stop after entering the venue. The behavior and participation pattern of attendees is far more analytic and contains more information that can be utilized to generate meaningful reports.

2-2 AllianceTech

AllianceTech (AllianceTech 2016) is one of the popular options among mega event organizers, such as electronics exhibition show. AllianceTech is recognized for it's customized cards and badges powered with RFID technology. For event check-in, attendees scan their custom made cards or badges to the KIOSK counter. The custom made cards contain information such as the event, participant user information, ticket information in the RFID chip. This method is very simple as participants do not require to install any mobile app in their smartphones. It can also securely authenticate the identity of the participants as the custom made cards contain digital information about the users in RFID chip. However, this method is less ideal as each and every participant require a custom made badges in order to check-in to events. This is extremely cost-ineffective and unnecessary. A better approach would be utilizing the device where everyone has on them: smartphone. A smartphone has all the technical components needed to implement the event check-in system.

2-3 i-Attend

i-Attend (i-Attend 2016) is an event attendance tracking system, which provides many ways of attendance tracking, such as barcode scanners, smartphone app and RFID scanners. One unique feature they have is the check-in systems is not only limited to one type of check-In method. Which means participants may use various kinds of check-in methods, such as barcode, QR code, RFID, to interact with one type of check-in counter systems. This approach provides flexibility to the participants as the participants may choose whichever suit their needs. However, this approach still requires the preparation of check-in counters and the speed of the check-in process still depends on the check-in counters available.

2-4 Eventbrite

Eventbrite (Eventbrite 2016) is a mobile app which allows users to create, promote and buy tickets for events. It also allows users to discover popular and recommended events. Eventbrite app is also a mobile check-in system. Upon registering for an event, the users are granted a QR code, which then can be used as a check-in ticket. The check-in counters use a QR code scanner plugged into a computer to scan the QR code on the Eventbrite app. This approach is easy and secure, as only the valid participants has the valid QR code. However, event participants are required to launch the Eventbrite mobile app in their smartphones and queue for their turn. A better approach would be eliminating the need of taking out and presenting the smartphone to check-in counters, by processing the check-in in the background once the participants enter the event venue.

2-5 Zkipster

Zkipster (Zkipster 2016) is an event management software which mostly used in high profile fashion shows. Zkipster is particularly popular with fashion shows and wine party thanks to it's modern website and marketing technique. Zkipster also provides event attendance tracking system, which they called 'Guest List management' themselves. It works by check-in counter staffs holding an iPad installed with Zkipster guest list management app, searching for guest list's names on the iPad, and marked them as checked-in. One unique feature is the guest list management app works even the device has no internet connection. However, the same problems persist, where there is no security authentication implemented, and the speed of the check-in process is limited to the number of iPads available at the check-in counter. A better approach would be provide an easy and secure way to check-in participants without them queuing at the entrance.

2-6 Discussion

Features	SocialTables	AllianceTech	i-Attend	Eventbrite	Zkipster	Proposed framework
Attendance tracking	✓	✓	✓	✓	✓	✓
Require additional hardware to detect check in	✓	✓	✓	✓	✓	✓
Speed of check in depends on number of hardware to detect check in	✓	✓	✓	✓	✓	X
Require custom made check in device	X	✓	✓	X	X	X
Require mobile app on attendee's smartphone	X	X	X	✓	X	✓
Require attendee's launch of mobile app during check in	X	X	X	✓	X	X
Attendee identity authentication	X	✓	✓	✓	X	✓
Tracking for POI	X	X	X	X	X	✓
Tracking for attendee mobility patterns	X	X	X	X	X	✓
Track leaving of attendee	X	X	X	X	X	✓
Real time data viewing	✓	✓	✓	✓	✓	✓
Report generating	Attendance report	Attendance report	Attendance report	Attendance report	Attendance report	Attendance report & Participant mobility pattern

Table 2-6-T1 Comparison between existing system and proposed framework

In general, the proposed framework will take a different approach than the reviewed systems. The framework works by scanning the existing Wi-Fi access points in the event venue, which means it does not require any additional hardware other than participants' smartphones. By utilizing the Wi-Fi access points, the framework can also track participants' mobility patterns in the event venue. All of these processing do not require users to launch their app before or during the event.

CHAPTER 3 SYSTEM DESIGN

3-1 Design Specifications

The main delivery of this project is a framework for developing Android mobile applications written in Java programming language. The framework will be open-sourced and can be easily plugged-in into any Android project with a minimum Android SDK API level equal or greater than 16.

This framework requires Android smartphone's Wi-Fi capability and ability to receive Google Cloud Messaging (GCM).

This framework will be able to launched the app in background (even if the app is shut down), triggered by GCM notifications, and start the Wi-Fi access points scanning. The scanning is scheduled on 30 seconds interval, which lies in the balance between device's battery life and frequency of tracking.

This framework also utilizes device's internet connection in order to retrieve event information and to upload tracking data to the mobile backend server.

This framework will store events data into local database. This approach allows the framework to fetch the event information from local database without retrieving it from mobile backend server every time, consuming time and internet quota.

3-2 System Context Diagram

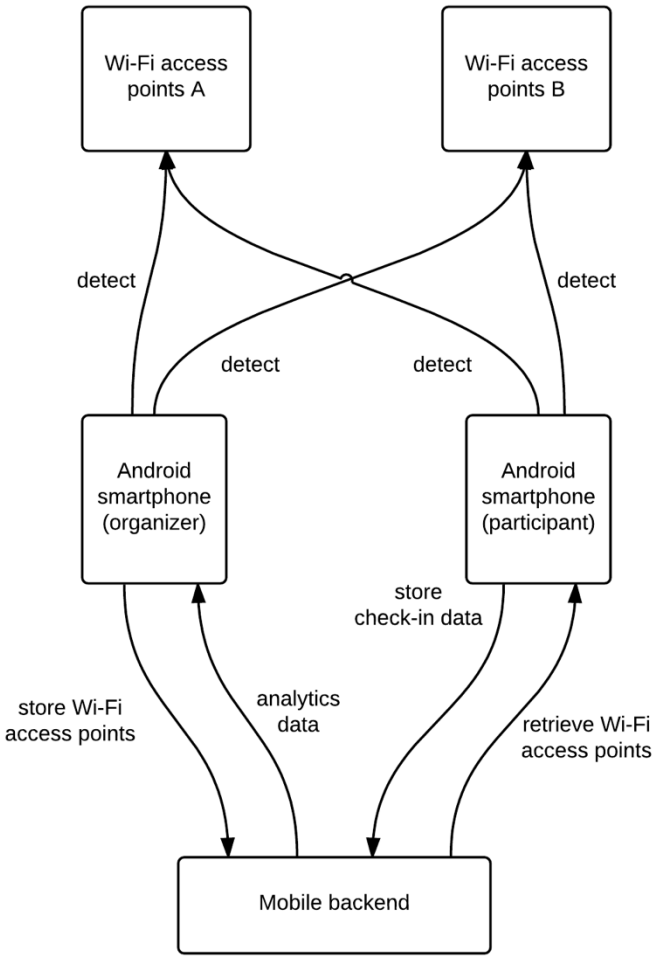


Figure 3-2-F1 System Context Diagram

The framework in the mobile applications scans and detects for Wi-Fi access points in the event venue. After processing, the framework uploads the information to mobile backend server. The framework also retrieves event and it's Wi-Fi access points configuration data from mobile backend server.

3-3 System Flowchart

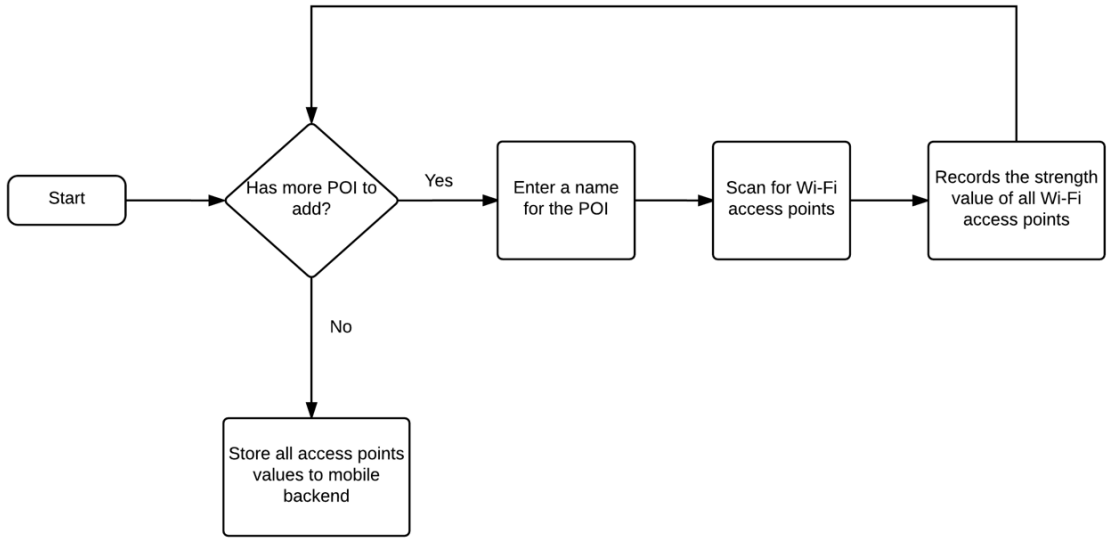


Figure 3-3-1 System Flowchart (Event organizer)

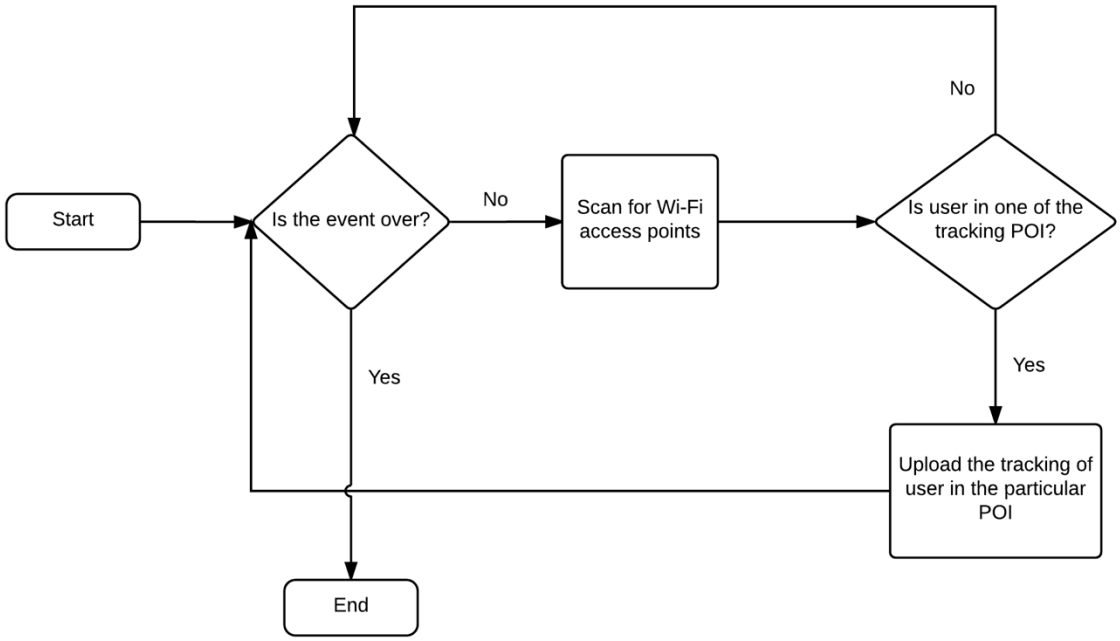


Figure 3-3-F2 System Flowchart (Event participant)

3-4 Data Flow Diagram

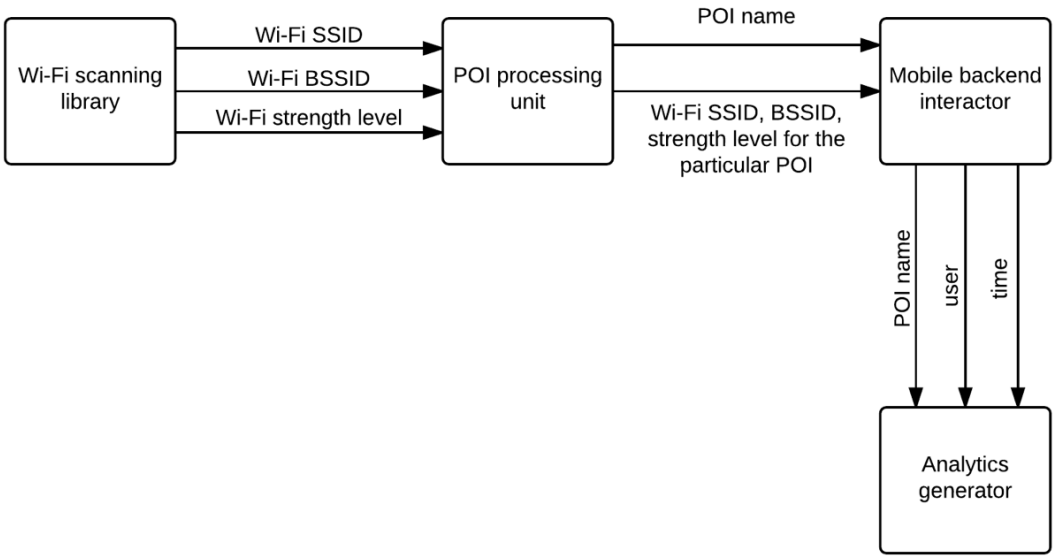


Figure 3-4-F1 Data Flow Diagram

Wi-Fi scanning library

This is the Android library responsible for scanning Wi-Fi access points available. It returns a list of access points, each access point is made up from SSID (Service Set Identifier), BSSID (MAC address of wireless access point) and strength level.

POI processing unit

This is the component responsible for processing the Wi-Fi access points. It generates POI based on multiple access points available. This component also responsible for determine if the user is in one of the POI, based on the Wi-Fi access points nearby.

Mobile backend interactor

This component serves as a gateway between the framework and the mobile backend server. It is responsible for uploading event, POI configurations to the mobile backend server. This component is freely customizable by other developers in case they wish to use their own choice of mobile backend service.

Analytics generator

This component retrieves event tracking and check-in data from mobile backend interactor and generate meaningful report and suggestion to the event organizers.

3-5 System Block Diagram

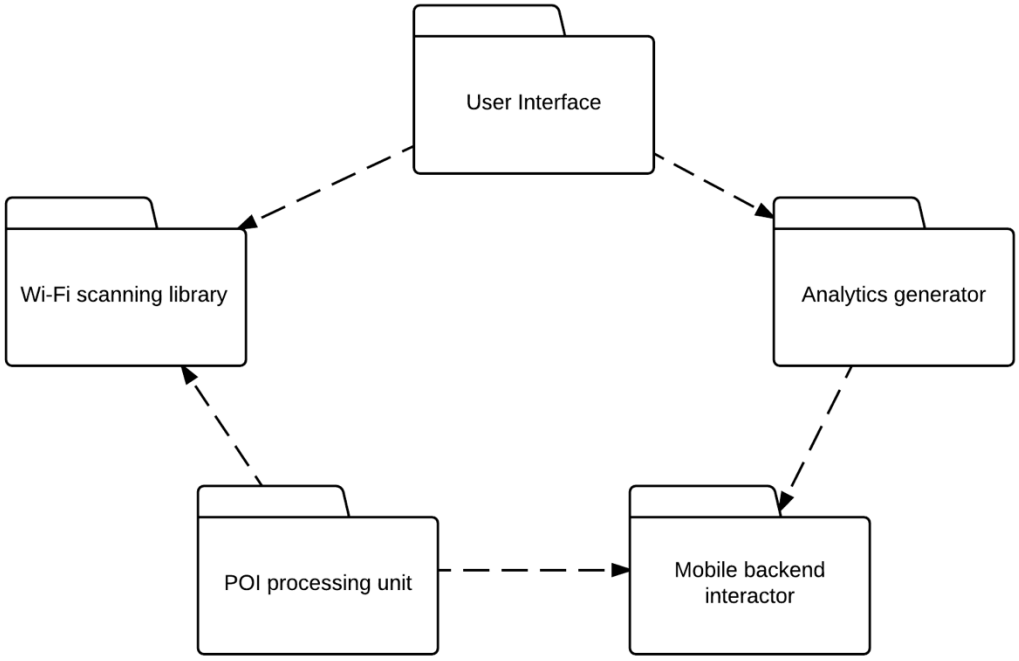


Figure 3-5-F1 System Block Diagram

User Interface needs to show event organizers which Wi-Fi access point is available.

User Interface depends on Analytics generator to provide event tracking report to event organizers.

During adding of POI, POI processing unit leverages access points data from Wi-Fi scanning library for the processing.

During the event, POI processing unit generates event check-in data and upload to mobile backend server through Mobile backend interactor.

Analytics generator retrieves event check-in data from mobile backend server through Mobile backend interactor.

3-6 Entity-Relationship Diagram

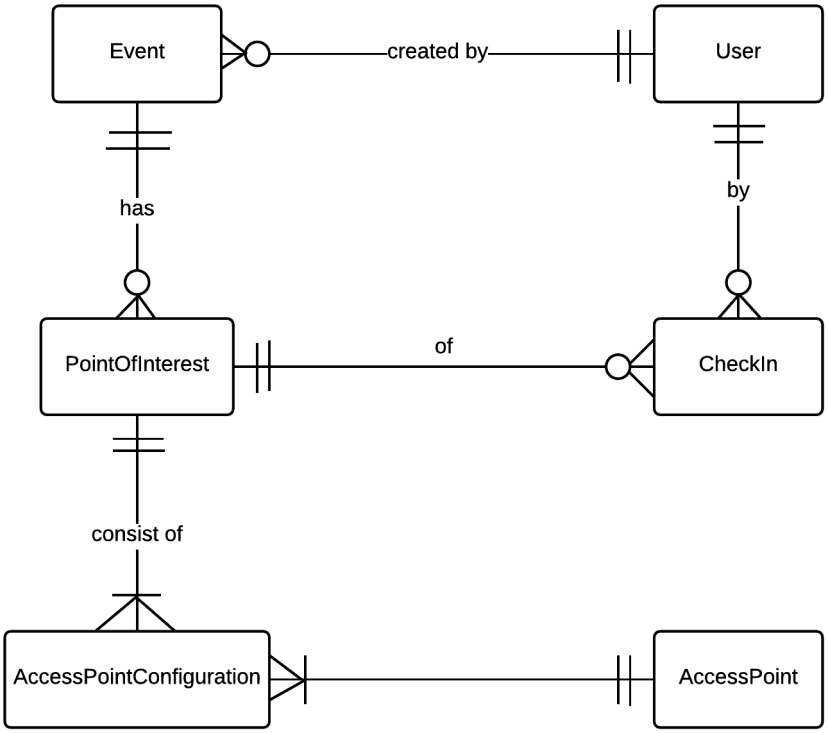


Figure 3-6-F1 Entity-Relationship Diagram

3-7 Class Diagram

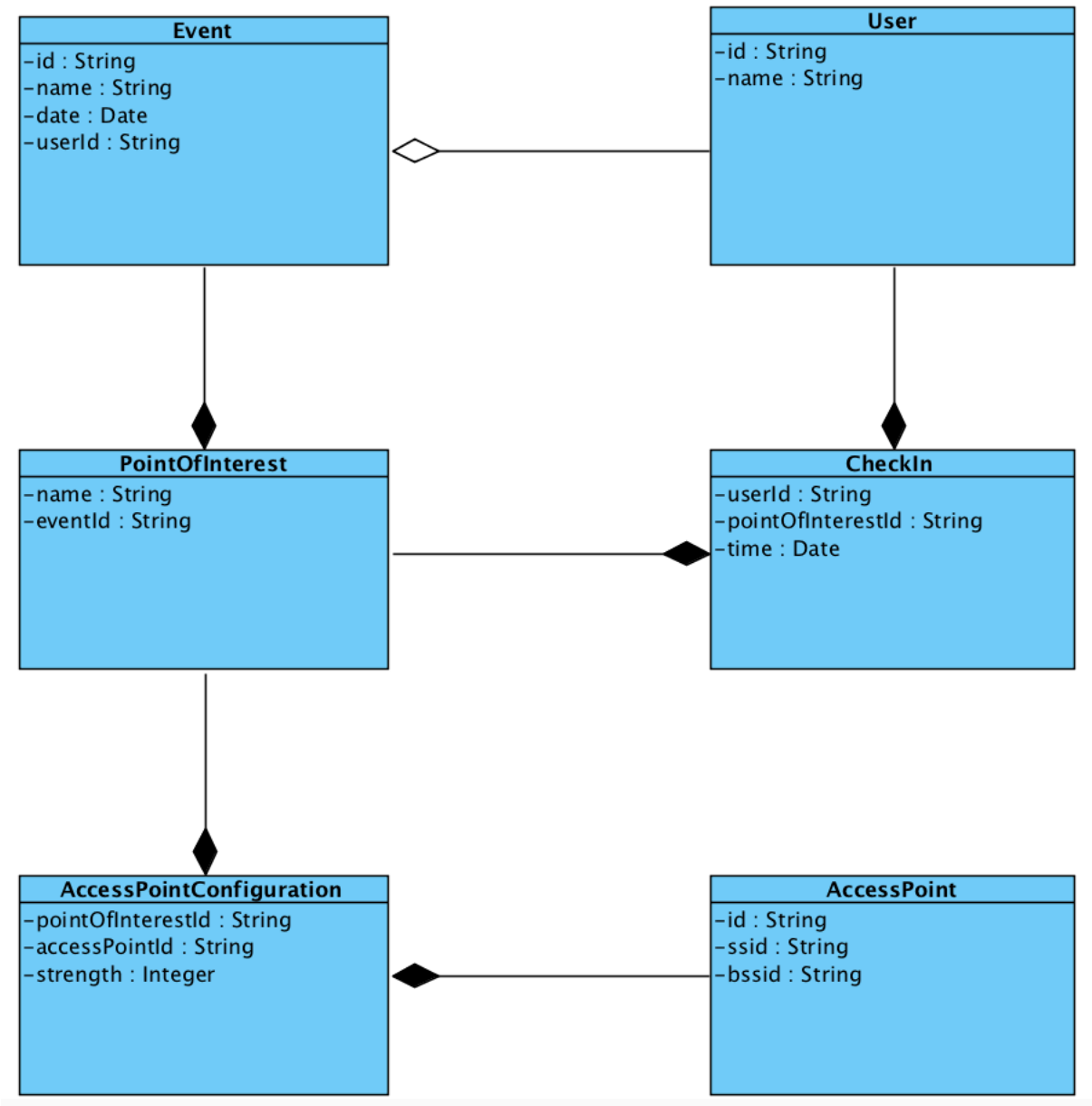


Figure 3-7-F1 Class Diagram

3-8 Use Case Diagram

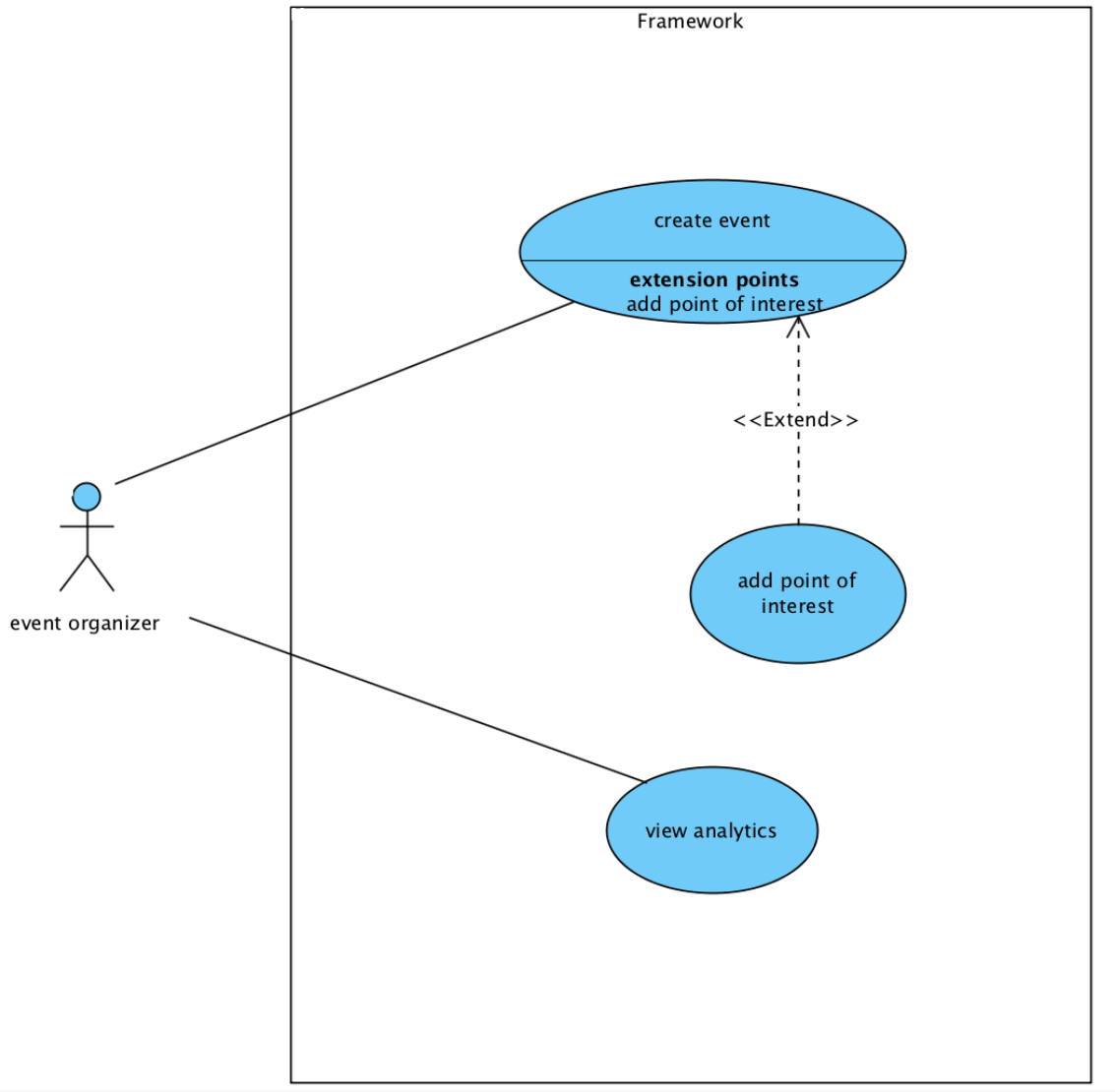


Figure 3-8-F1 Use Case Diagram (Event Organizer)

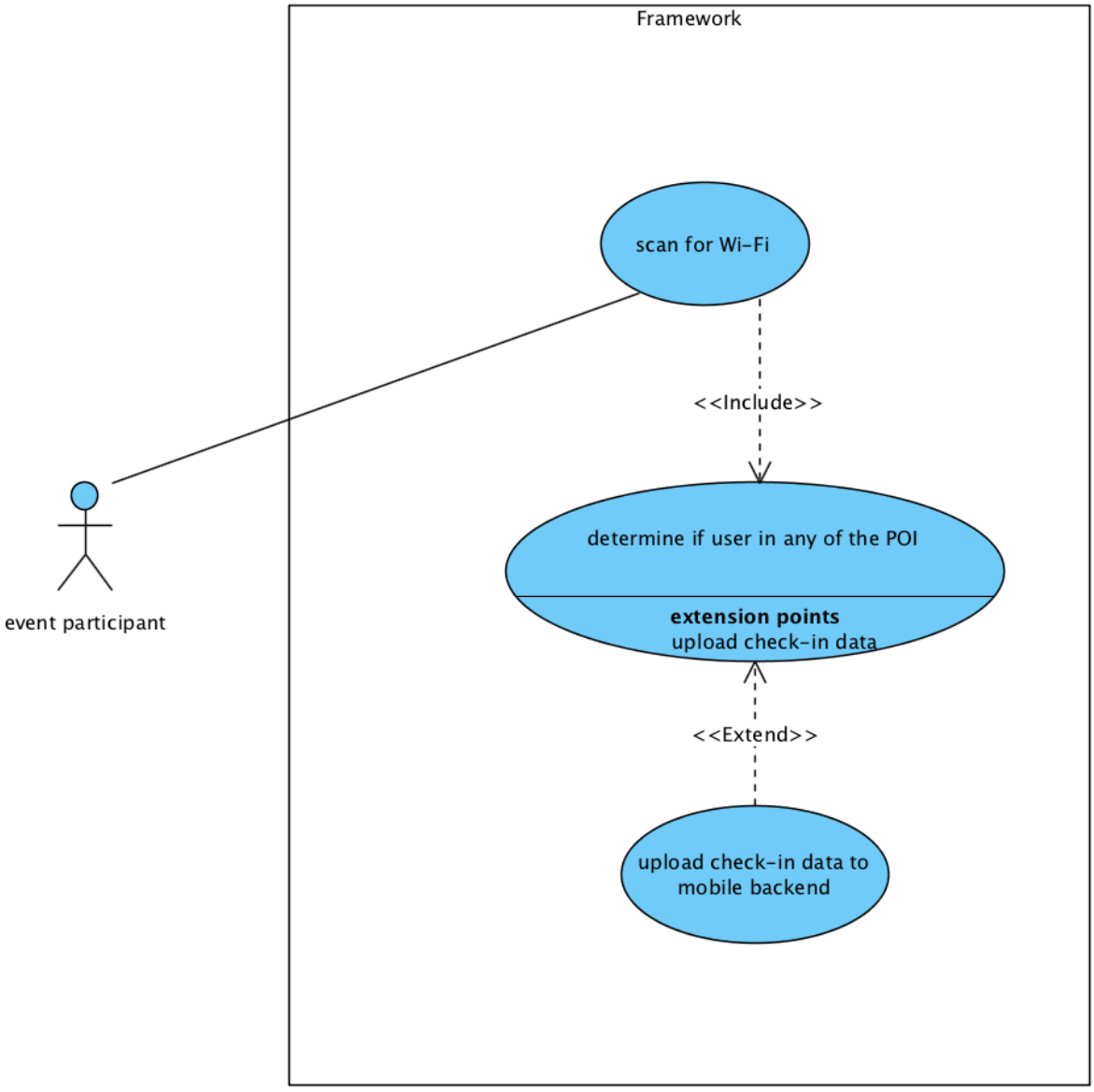


Figure 3-8-F2 Use Case Diagram (Event Participant)

3-9 Sequence Diagram

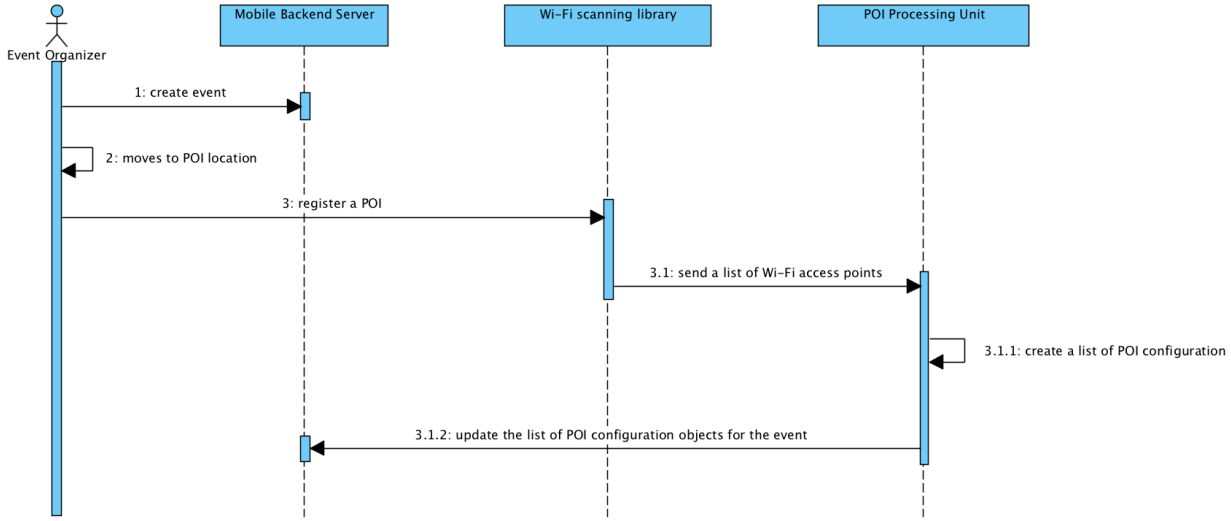


Figure 3-9-F1 Sequence Diagram (Event Organizer)

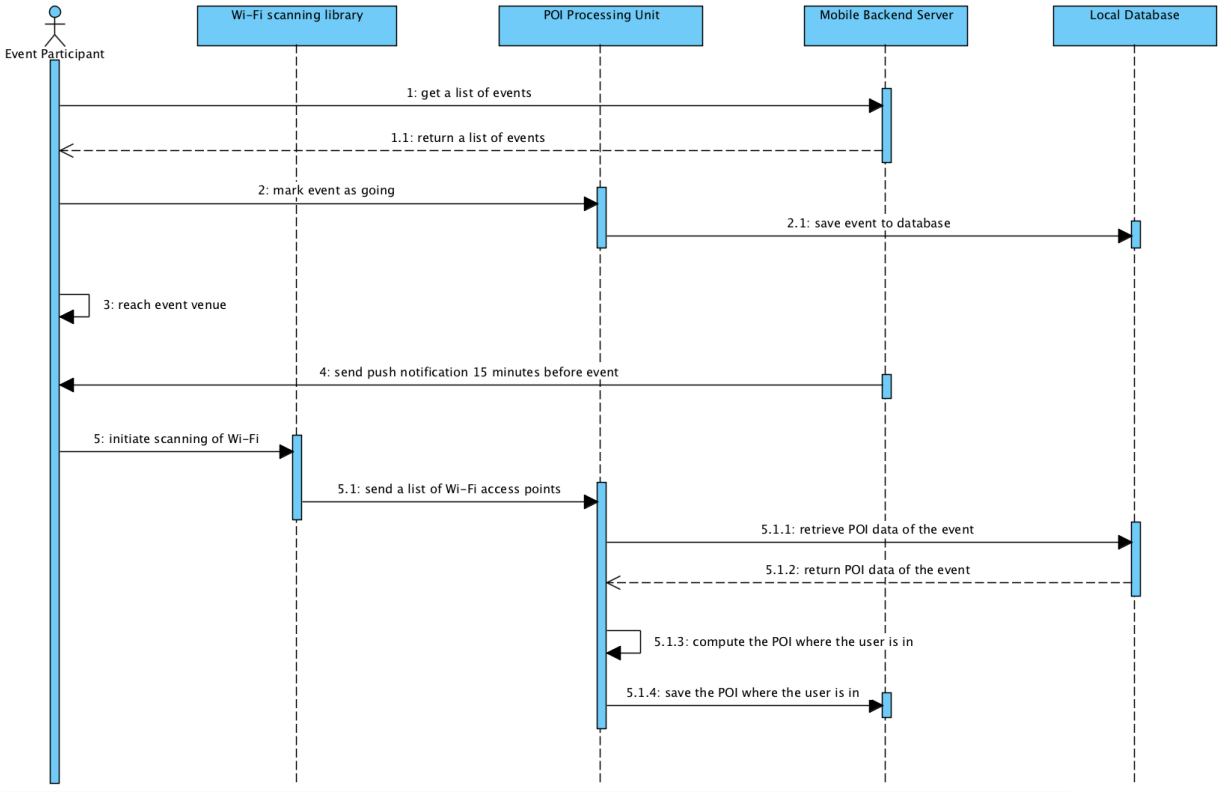


Figure 3-9-F2 Sequence Diagram (Event Participant)

CHAPTER 4 SYSTEM IMPLEMENTATION

4-1 Architecture Overview

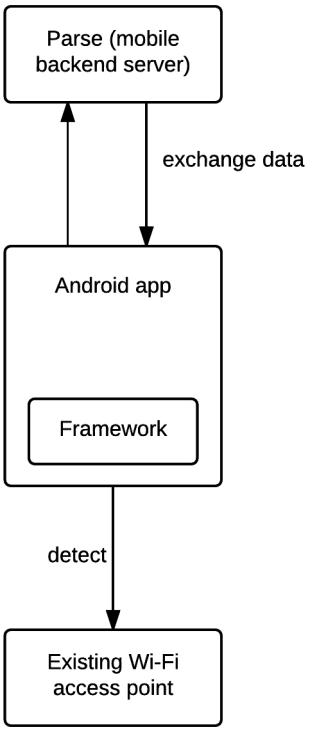


Figure 4-1-F1 Architecture Overview

The framework is embedded into an Android app by any developer who wants to use it. The framework acts as the core library for the Android app to interact with mobile backend server. The framework also include library to scan for Wi-Fi access points nearby.

4-2 Methodology and Tools

The Spiral methodology is used for the development of this framework. The iterative and incremental aspects of Spiral methodology suit the nature of this project, as it requires repetitive testing and adjusting the parameters until the correct balance is found.

For the IDE, Android studio 1.5.1 is used. Android Studio provides an integrated environment for the testing and debugging of an Android library. It also allows easy packaging the project into a framework where other developers can easily use.

4-3 Wi-Fi scanning library

```
public class WiFiScanning extends Service {

    private static long intervalSecond = 30;
    public static WiFiScanning singleton = null;
    Callback callback;
    WifiManager wifi;
    public List<ScanResult> results;
    boolean stopFlag = false;
    Handler scanHandler = new Handler ();
    Runnable UpdateUIResults = new Runnable () {
       public void run () {
          updateUI();
       }
       public void updateUI() {
           // update UI
       }
    };

    public static void setIntervalSecond(long second) {
        intervalSecond = second;
    }

    @Override
    public void onCreate() {
      wifi = (WifiManager) getSystemService(Context.WIFI_SERVICE);
      if (wifi.isWifiEnabled()) {
         Thread thread = new Thread(){
            int counter = 0;
            public void run(){
               while (counter % 2 == 0 && stopFlag==false) {
                  wifi.startScan();
                  results = wifi.getScanResults();
                  scanHandler.post (UpdateUIResults);
                  try {
                     //sleep for xx milliseconds
                     Thread.sleep(intervalSecond*1000);
                     counter += 2;
                  } catch (InterruptedException e) {
                     e.printStackTrace();
                  }
               }
            }
         };
         thread.start();
      } else {
         Toast.makeText(this, "Please enable WIFI network", Toast.LENGTH_LONG).show();
      }

      singleton = this;
    }
}
```

The library uses Android's WifiManager to get a list of ScanResult. ScanResult is the class representing a WAP. The WiFiScanning service is scheduled to run on an interval of 30 seconds by default. Developers can freely customized the interval value to suit the events.

4-4 Saving POI (Point of Interest)

```
private void saveAccessPoints() {
      ArrayList<AccessPoint> apList = new ArrayList<>();

      for (ScanResult result : WifiManager.singleton.results) {
          apList.add(new AccessPoint(result.SSID, result.BSSID, result.level);
      }

     // save and upload to mobile backend server
     ...
}

public class AccessPoint {
    private String ssid;
    private String bssid;
    private int level;

    public AccessPoint(String ssid, String bssid, int level) {
        this.ssid = ssid;
        this.bssid = bssid;
        this.level = level;
    }
}
```

This feature is used by the event organizers when creating event. Organizers can create multiple POI for an event. This feature requires at least one Wi-Fi access Point at the event venue. The more the number of Wi-Fi access points, the better the accuracy of the tracking.

An UI element, Activity is implemented for developers to easily display in their apps and let users input the POI information. The Activity displays the nearby Wi-Fi access point's SSID, BSSID and strength level.

This framework is designed in a way that the users do not need to understand how the framework works, as the user interface is very self-explanatory. When users are ready to add a new POI, users move and stay at the particular POI. Then, users can tap on the 'Save' button to register a new POI and the framework will record multiple Wi-Fi access points and their respective strength level. Then, after users are finish with the adding, the framework uploads the configuration to the mobile backend server.

4-5 POI processing

This feature is used at the client app, which means it's for the event participants. This feature is activated at 15 minutes before the event starts and deactivated after the event ends. This component will get a list of nearby Wi-Fi access point through Wi-Fi scanning library every 30 seconds throughout the activation period in the background.

Then, the algorithm will determine if the user is in a particular POI set by the event organizers. How the algorithm works is: it acquires the current Wi-Fi access points and their strength levels, then it scans through the list of POI configuration that the event organizers set. If one of the POI configuration (consist of a list Wi-Fi access points and their strength levels) matches the current Wi-Fi strength level values, then the user is considered to had visited the particular POI.

```
public static int strengthLevelOffset = 8;

public void checkPOI(ArrayList<ScanResult> results, Event event,
 ArrayList<PointOfInterest> pointOfInterestList) {

    for (PointOfInterest poi : pointOfInterestList) {

    int numberOfAPMatched = 0;
    boolean foundUnmatched = false;

    for (ScanResult scanResult : results) {

        if (foundUnmatched) break;

        for (AccessPointConfiguration configuration : poi.getConfigurationList()) {

            if (scanResult.BSSID.equals(configuration.getAccessPoint().bssid)) {

                if (scanResult.level <= configuration.getAccessPoint().level +
strengthLevelOffset &&
                        scanResult.level >= configuration.getAccessPoint().level -
strengthLevelOffset) {
                    numberOfAPMatched++;
                } else {
                    foundUnmatched = true;
                    break;
                }
            }
        }
    }

    if (numberOfAPMatched == poi.getConfigurationList().size()) {
        // save a 'CheckIn' object
    }
}
```

4-6 Demo App: LAKE

To demonstrate the features and flexibility of the framework, a demo app is developed to showcase the usage of the framework. The app shows events in UTAR on the map. The UTAR map is redrawn from the ground up to provide better visual and accuracy compared to the original one in Google Map.

4-6-1 System Flowchart

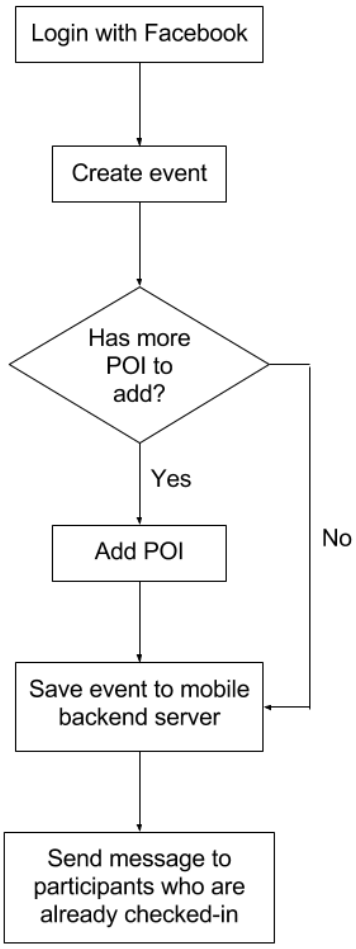


Figure 4-6-1-F1 System Flowchart (Event Organizer)

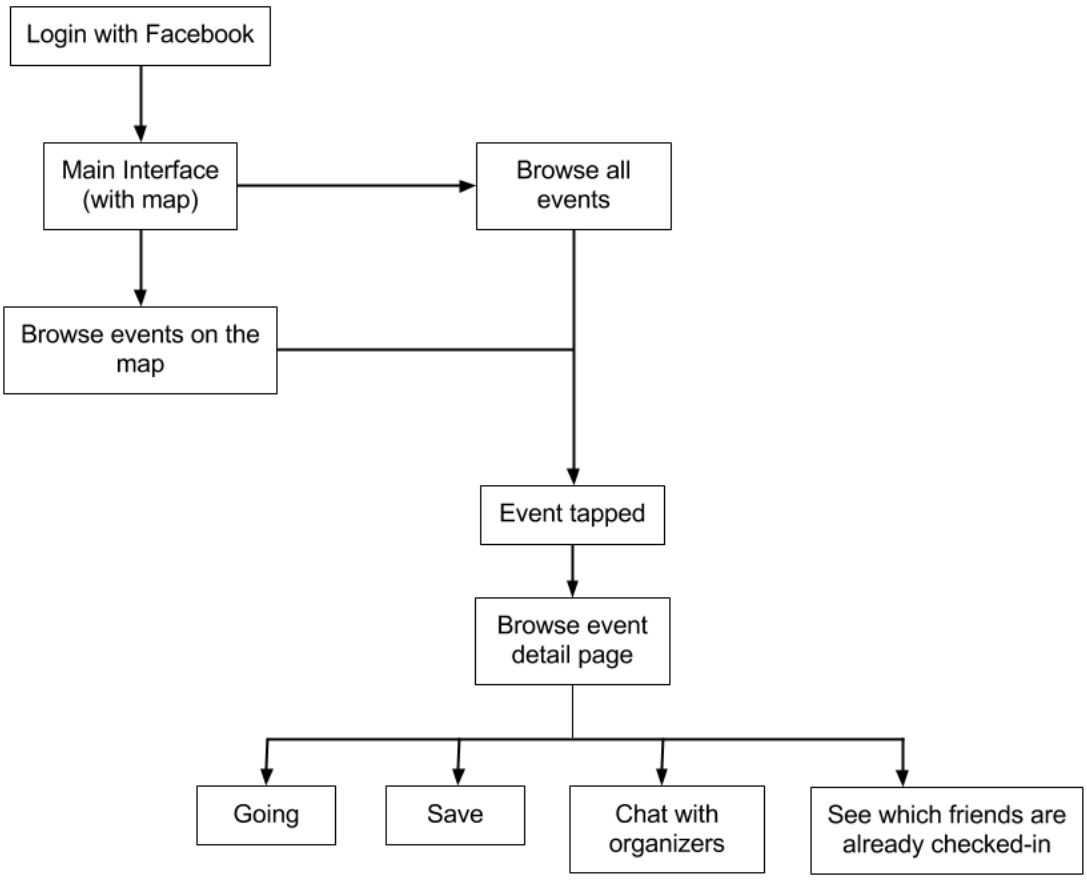


Figure 4-6-1-F2 System Flowchart (Event Participant)

4-6-2 App Screenshot

Figure 4-6-2-F1 Screenshot of demo app (Main Interface)

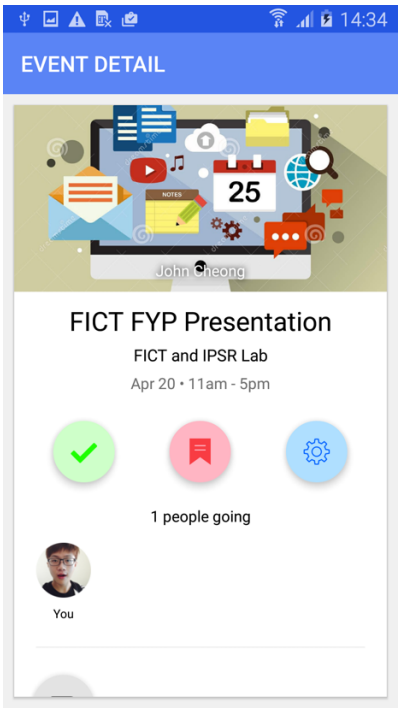


Figure 4-6-2-F2 Screenshot of demo app (Event Screen)

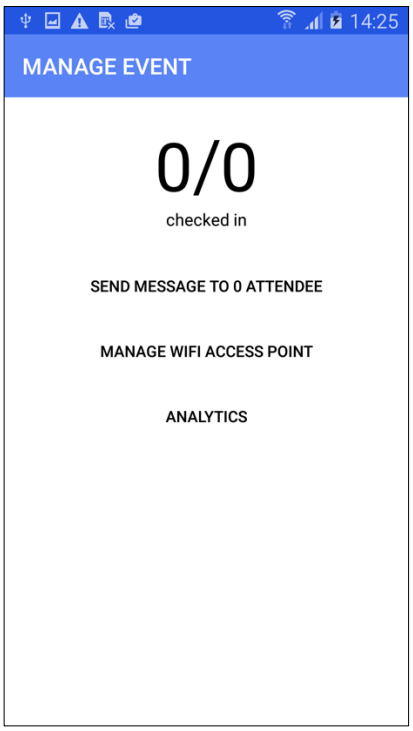


Figure 4-6-2-F3 Screenshot of demo app (Manage event)

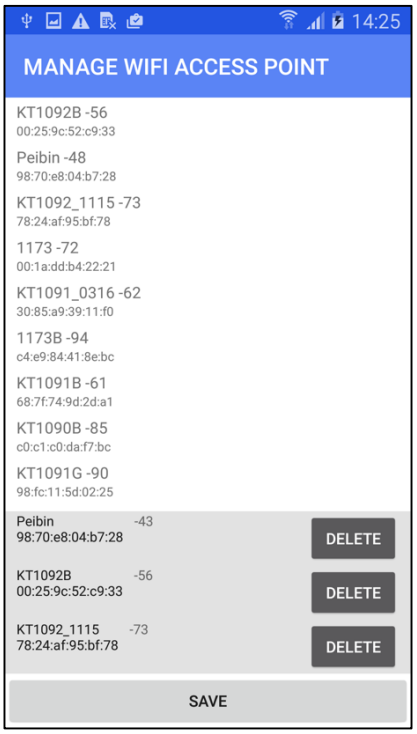


Figure 4-6-2-F4 Screenshot of demo app (Adding POI)

4-6-3 New Functionalities Build on Top of the Framework

By using the framework, there are multiple new features implemented by using the features provided by the framework:

i. Send messages to participants who are already checked-in (organizers)

This feature leverages data from the framework to generate a list of checked-in participants. It does this by grouping the 'CheckIn' object by 'userId' field, resulting in an unique list of checked-in users.

This feature allows organizers to send messages to checked-in participants to inform updates or emergencies. This approach is better than sending to all 'going' participants, as some of them may not arrived yet or they may leave the event already.

ii. See which friends are already checked-in (participants)

This feature also leverages data the same way as in (i). The only difference is it is initiated by the participants. When a participant check in, a notification is pop up and shows the user how many friends are already here. When the user opens the app, he or she can see which are the friends already checked-in.

This feature is useful especially for large events, where it is very difficult to find their friends in the large venue.

CHAPTER 5 TESTING

5-1 Testing app

In order to verify and test the algorithm in the framework, a demo Android app was developed and presented at the launch of UTAR Gallery at 25 March 2016. It's called UTAR Gallery Tour. This app also implements the exactly same framework. But, the way this app uses the framework is slightly different. Instead of tracking the attendance and the frequency of users visiting POI, the app tracks the completeness of users' visitation of POI in the gallery. The purpose of this app is to shows the users' percentage of their gallery tour and which section did they haven't visit. This app is not only useful for gallery visitors, but also to the gallery administrator. The administrator can analyze the data and know which part of the gallery where the visitor tends to stay longer. The administrator can also know the visitation pattern of the visitors. By utilizing the data, the administrator can improve the gallery layout to increase the tour completeness of visitors.

5-2 Test plan

To verify the framework, the app was used to save 9 POI around the gallery, including entrance and exit. Then, the smartphone is moved around the gallery like a normal visitor would.

Figure 5-2-F1 Test venue – UTAR Gallery (1)

Figure 5-2-F2 Test venue – UTAR Gallery (2)

5-3 Result

In the 10 iterations, the app successfully tracked all of the POI and shows the percentage of completeness of the gallery tour.

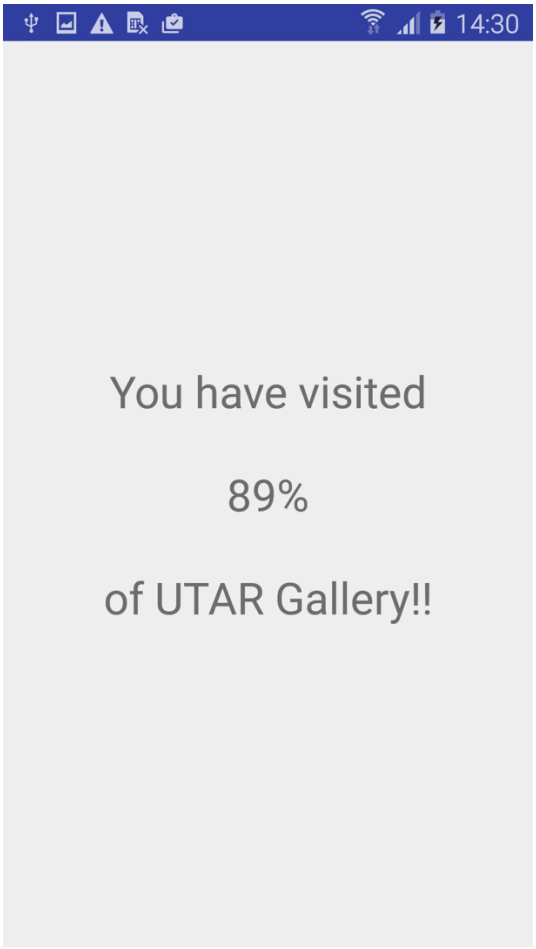


Figure 5-3-F1 Screenshot of demo app (Result)

CHAPTER 6 CONCLUSION

6-1 Conclusion

In a nutshell, the framework which can track event attendance and participation pattern is successfully developed. The Android framework can be easily implemented into any Android mobile applications. Mobile developers can develop new functionalities based on the features the framework provides. The demo apps are also successfully proved that the framework is well developed and tested, and also very flexible to implement into various kind of projects to provide new features.

In addition to the general analytics generator the framework provides, mobile developers can also develop their own analytics tools to help event organizers visualize and evaluate their events' attendance, engagement and participation pattern in the way suits their clients.

6-2 Future Direction

This framework will be a good guideline for mobile developers who wish to build event tracking system. As the framework will be open-sourced, the framework is expected to be improved by the developer community.

The framework is using Wi-Fi capability to detect user's mobility pattern, which may not suitable for events held in venue without Wi-Fi access points. Bluetooth Low-Energy (BLE) technology is also a great option to implement event tracking system. BLE has lower detection range, but it is also more reliable and less dependent on external hardware other than smartphones. . In the future, BLE can be tested and integrated into the framework to provide a more complete and accurate event tracking framework.

REFERENCE

SocialTables 2016, Event Check In App | Guest List App. Available at https://www.socialtables.com/product/check-in/ . [8 April 2016].

AllianceTech 2016, RFID Attendance Tracking for Events and Conferences. Available at http://www.alliancetech.com/solutions/attendee-tracking/intelligent-attendance/features . [8 April 2016].

i-Attend 2016, Event Registration, Event Management and Attendance Tracking Software. Available at http://www.i-attend.com/features_attendance_software.html . [8 April 2016].

Eventbrite 2016, Eventbrite - Local Events, Fun Things To Do Near Me & Event Planner. Available at https://itunes.apple.com/my/app/eventbrite/id487922291?mt=8 . [8 April 2016].

Zkipster 2016, Guest List App. Available at https://www.zkipster.com/guestlistapp/ . [8 April 2016].

Android

EVENT PARTICIPATION TRACKING FRAMEWORK

by Cheong Pei Bin

• TRACK MORE THAN •

JUST

ATTENDANCE

• ALSO TRACK •

POINT OF INTEREST

LEAVING

MOBILITY PATTERN

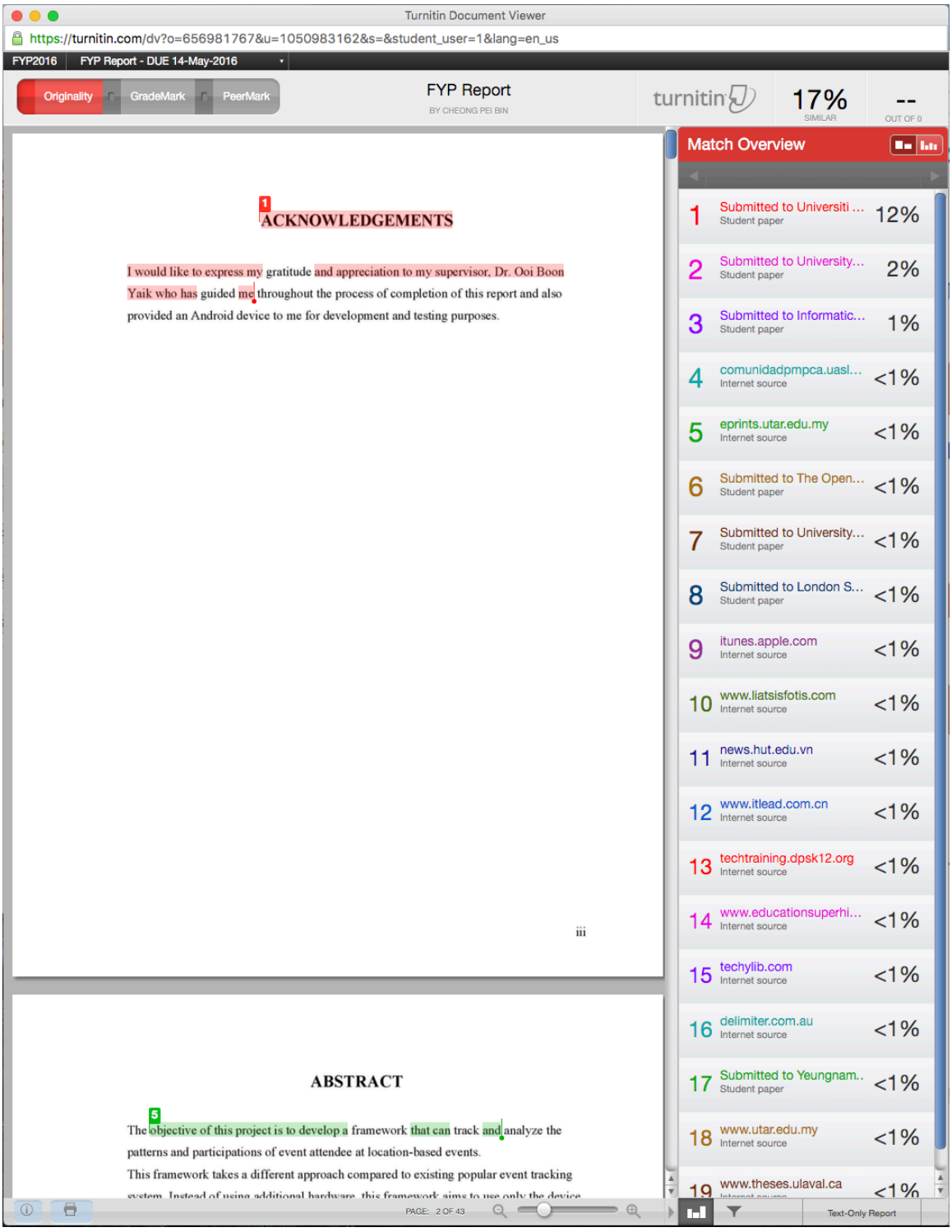
• WORK IN BACKGROUND •

NO APP LAUNCH

FICT·CS

JAN 2016

Poster



Turnitin Results

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF _____

Full Name(s) of Candidate(s)	
ID Number(s)	
Programme / Course	
Title of Final Year Project	

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: _____ % **Similarity by source** Internet Sources: _____ % Publications: _____ % Student Papers: _____ %	
Number of individual sources listed of more than 3% similarity: _____	

Parameters of originality required and limits approved by UTAR are as follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

_____
Signature of Supervisor
Name: _____

Date: _____

_____
Signature of Co-Supervisor
Name: _____

Date: _____