Mobile Geo-Tagging in Social Networks with Android

GRADUATE PROJECT REPORT

Submitted to the Faculty of the
School of Engineering and Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Praneeth Koralla
Fall 2011

Committee Members

Dr. Scott A. King
Committee Chairperson

Dr. Dulal Kar
Committee Member

Dr. David Thomas
Committee Member
ABSTRACT

The main objective of this project is to develop a mobile app in android that gives a new perspective to social networks called event-oriented social networking. This application can build a social network and allow the users to create events at different geographical locations. The tagged events send notifications to all the user’s friends or followers when they are a mile distance away from that place. The events can be scaled from a single location to entire city. This application is enhanced using the geotagging feature where each event is added with the geographical metadata information. This application can be used as an alarm system during any natural disasters where, the effected location is tagged as an event and no one is allowed to enter the radial distance that is affected by that calamity. If any user unknowingly enters the radial distance, he is alerted with notification message.
# TABLE OF CONTENTS

Abstract .................................................................................................................................................. ii

Table of Contents ................................................................................................................................... iii

List of Figures ......................................................................................................................................... iv

1. Background and Rationale .................................................................................................................. 1
   1.1 Introduction ..................................................................................................................................... 1

2. Narrative ........................................................................................................................................... 5
   2.1 Implementation of a social network ............................................................................................. 5
   2.2 Geo-tagging operations ................................................................................................................ 7
   2.3 Integrating Geo-tagging with the social networks ...................................................................... 9

3. Requirements Definition .................................................................................................................... 10

4. System Design ..................................................................................................................................... 12

5. System Implementation ..................................................................................................................... 14

6. User Study ......................................................................................................................................... 20

7. Evaluation and Results ..................................................................................................................... 22

8. Conclusion and Future Work ........................................................................................................... 24

9. Bibliography and References ........................................................................................................ 25

10. Appendix A ...................................................................................................................................... 26
LIST OF FIGURES

Figure 2.1 Software architecture for geo-tagging application ........................................6
Figure 4.1 Geo-Tag Application System Design .............................................................12
Figure 5.1 Login and Registration .................................................................................14
Figure 5.2 Application home page ................................................................................15
Figure 5.3 Creating an event .........................................................................................16
Figure 5.4 Features to assist event followers .................................................................17
Figure 5.5 connecting to web server via internet service ...............................................18
Figure 5.6 Wrappers for MySQL database .....................................................................19
Figure 6.1 UI Modifications .........................................................................................20
Figure 7.1 Locating position and creating event ............................................................22
Figure 7.2 Viewing friends’ events ................................................................................23
Figure 7.3 Proximity alert and event assistance ..............................................................23
Figure 7.4 Relational table storing the events .................................................................24
Figure 9.1 Usecase diagram for the geo-tag application ...............................................28
Figure 9.2 Usecase diagram for managing the users’ network ......................................29
Figure 9.3 Usecase diagram for managing the user events ..........................................31
Figure 9.4 Usecase diagram for creating an event .........................................................32
Figure 9.5 Usecase diagram for listening and connecting to an event ..........................34
Figure 9.6 Usecase diagram for managing user profile ................................................34
Figure 9.7 Sequence diagram to search for a user on network ......................................35
Figure 9.8 Sequence diagram to search for creating an event .......................................35
Figure 9.9 Sequence diagram to listen and connect to any event ..................................36
Figure 9.10 Class diagram for the geo-tag application .................................................36
Figure 9.11 E-R Diagram for social network .................................................................38
Figure 9.12 Modified data model to integrate geo-tagging in social networks ..............38
1. BACKGROUND AND RATIONALE

1.1 INTRODUCTION

Mobile apps are the hottest software gadgets happening in the global market. Many smart phones are now offered by Apple, Samsung, and HTC etc. and the mobile networks are now offering 3G and 4G services with those smart phones. “Android is a hot, young, free and open source mobile platform making waves in the wireless world” [Shane, Lauren]

Along with the mobile phones, the numbers of social networking sites are expanding in terms of the number of users. Now, Facebook has 640 plus million, Twitter has 200 million, MySpace has 125 million and Orkut has 120 million users. A social network connects people irrespective of their geographical location [Social Population].

Geo-tagging is the process of attaching some geographical identification information to content. The geographical identification information can be a latitude-longitude pair, a place like New York, New Delhi, a personal geo-tag such as our home or school or it can be a contextual geo-tag such as a birthday party or an official meeting. Geo-tagging is not regularly used as a separate application but it can enhance the content that is part of any application. Geo-tagging is nothing but creating some content related to a geographical location. We can use the geo-tagged data in two ways, either it can be produced on a mobile phone or it can be consumed from a mobile phone.

Geo-tagging can be physically achieved by acquiring geographical co-ordinates of a location through mobile phones. In general, a phone can tell where exactly we are on the map using the following [Android for Work]:

[Android for Work]
1. GPS (global positioning satellites): There are 27 GPS satellites orbiting the Earth. The GPS sensor inside the phone attempts to find the signal using at least three satellites and calculates the exact position.

2. Cell Towers: The location of a mobile phone can also be estimated by locating the nearest cell network tower.

3. WPS (wireless positioning system): The location of a mobile phone can also be estimated using the public Wi-Fi spots located nearby.

Geo-tagging when combined with a social network becomes a powerful information source. Social geo-tagging can make the people aware of the natural disasters like earthquakes, hurricanes. Social geo-tagging has many advantages that include:

1. Reducing advertising costs.
2. Personally exchange messages between people who are in a network or circle.
3. Filtering geographical information based on our personal interests.
4. Aiding disaster recovery teams.
5. Helping augmented reality applications.

Along with the advantages, geo-tagging brings many threats to the social world that include:

1. Exposes places of duty and home.
2. Identify the military population.
3. Increases the bandwidth usage through the embedded GPS.
4. Establishes patterns to allow strangers to track our paths.
5. Geographical location can be exposed from the tagged pictures.
This mobile application provides a way to easily create and manage tagged geo-points and a consolidated interface to a social network. The following example shows how geotagging makes life simple.

**Conventional event management:**

“Suppose you have decided to host a birthday event at your home and you have various group of friends including school, personal relations etc. You have to invite each of your friends who belongs to your network and keep track of their visit to your place and changes in your plan should be updated to your network”

**Event Management with geotag application:**

“Suppose you wanted to host a birthday event at your home and you have various group of friends including school, personal relations etc. All you have to do is to create an event (geo-tag) at your home location (assumed that all your friends directly follow you on this social network) and all your friends can view this on their mobile phones. This application will notify the people in your network upon any changes in the event. It also assists the friends while driving to the event by providing the navigation and event contact details”

The main purpose of this application is to achieve an event oriented social networking. The major concern is on the events that are happening in a social network rather than on the people who are in that network. The application provides the users with the following features,

1. Registering for a social network.
2. Searching for a friend on the network.
3. Connecting with a friend on the network.
4. Hosting events at various geographical locations.

5. Following friends’ events and navigate to the event locations.

These features enhance the communication mechanism between the people inside a social network and also reduce the effort incurred to host an event.
2. NARRATIVE

This project is divided into three different modules,

2.1 Implementation of a social network.

2.2 Geo-Tagging operations.

2.3 Integrating Geo-tagging with the social networks (Proximity alert and device communication).

2.1 Implementation of social network:

To implement a social network, we need to establish a network connection framework which includes the following components,

a. Data model.

b. Web server implementation

c. Server-side activities.

d. Client-side activities.

Data model: The data model includes the many to many relationships between the users, their profile information, their user account information, their messages etc. Figure 9.11 of Appendix A shows the data model for the social network.

Web server implementation: The overall software architecture for this project is shown in Figure 2.1 below,
The above architecture is based on a client-server model. The mobile application consists of three blocks: network, persistence and location. The location module acquires the user’s location and the persistence module stores the tags and locations in a persistent storage. The network module interacts with the web server over the 3G or Wifi. The web server is internally connected to a data base server. The web server and database server can be configured from computer or mobile phone browser.

**Server side activities:** All the server side activities are directly mapped to the data model shown in Figure 2.1. Most of activities in the social network are grouped as the server side activities which include,

a. **Social Users Activity:** This activity lists all the connections related to a specific user and is directly mapped to the Connected To entity. It is also mapped to the SocialUserProfile entity to retrieve the profile information of the connected user.

b. **Search Activity:** This activity searches for the users based on their first name or last name or email.
c. **Social Message Activity:** This activity involves the connection request message, connection accept/reject messages and general messages between connected users.

**Client-side activities:** Similar to the server-side, client-side activities are also directly mapped to the data model presented in Figure 2.1. The client-side activities mainly includes,

a. **User Registration and Login Activity:** Every user needs authentication to login to the social network. Registration is the process to enter the social network before getting connected to the network of users.

b. **View Notification Activity:** The user should view his current and past notifications to keep track of the requests and responses of the users connected to him.

c. **Browse Users Activity:** The users should be able to browse the profiles of the connected users.

d. **Post/Reply Activity:** The users can either post messages or reply to a specific message posted by another user.

e. **Disconnect Users Activity:** The users can disconnect any user at any time from his network.

2.1 **Geo-tagging**

Geo-tagging is method of tagging a geographical location based on its latitude, longitude and altitude. To tag a given location, the mobile phone sensor must be able to retrieve the co-ordinates of a specific location. The geo-tagging can be achieved using the following mechanism,
a. Read the co-ordinates of the current location from the location manager. The location manager uses the location service provided by the mobile device. The location manager then contacts the GPS provider via location service to retrieve the current location.

b. Map the co-ordinates to the nearby address using the geocoder. The geocoder takes the co-ordinates as inputs and returns the textual address of that particular location.

c. Choose the extent of tagged area and fix the exact location. The GPS provider gives the exact co-ordinates of the current location but the area of interest may extend to a wide range and hence the tag length needs to be flexible in size.

**Activities in geo-tagging:** We need few activities to tag a geographical location they are,

1. **Read Location Activity:** This activity helps to read the current location of the mobile device as the co-ordinates. A GPS provider can fetch the co-ordinates of the current location. These co-ordinates are mapped to the address by using a geocoder and all the locations near to the current locations are retrieved and shown to the user.

2. **Scale Location Activity:** This activity gives the user a freedom to scale his tag. The tag can be scaled from a single location point to a wide geographical area like a city, state or a country.

2.3 **Integrating geo-tagging with the social networks**

The main objective of this project is to integrate a geo-tagging system with a social network. The integration can be achieved by re-modeling the social network data model shown in Figure 2.1. There are some new tables that are to be added and also there are
some changes that are to be made to the existing tables to support the geo tagging
mechanism. The changes are shown Figure 9.12 of Appendix A.

In order for the social network to adapt to the geo-tagging, a new table Social User
tag is introduced. This table gives the information about the tags that are placed by
different users on the network. The columns Current Location and Proximity Alert are
introduced into the Social User table. The Current Location keeps tracks of the co-
ordinates of a user in the social network. The Proximity Alert is a user preference to
either receive the proximity alerts or not.

**Activities evolved due to integration:**

a. **Proximity Alert Preferences Activity:** This activity can configure the minimum
frequency to update location (to set a proximity alert), the minimum distance to
update location (to set a proximity alert), and the proximity radius required to
trigger a proximity alert.

b. **Set Proximity Alert Activity:** This activity helps user to set the alerts at a given
location and gives an expiry date for it.

c. **View Alert Activity:** This activity shows all the alerts that are within the tagged
area and the user can choose any one of it to move forward.
3. REQUIREMENT DEFINITION

The following section provides the user requirements in detail and also gives the structural view of those requirements.

3.1 Plain English Description

1. The users must be able to register with the application.
2. The users must be able search for any other user who is already registered on the network.
3. The registered users must be able to follow or un-follow any other registered user.
4. The registered users must be able to create any event which includes the event name, event description, event contact details (phone and email), event time and event location.
5. The event creator must be able to locate his current position to create a tag at that location.
6. The registered users must be able listen to the events that are active on his network.
7. The event follower must be able to receive the event details as a notification if he/she is in a 1000 meter range of the event location.
8. The event follower must be able to contact (phone call or email) the event directly from the notification received.
9. The event follower must be able to navigate to the location where the event is hosted.

3.2 Structured Requirements

1. **Manage network:** An application user can manage his/her social network
   1.1 **Search user:** A user can search for his friend of interest on the application.
   1.2 **Follow user:** A user can follow another registered user of his interest.
   1.3 **Un-follow user:** A user can remove a user from his network any time.
2. **Manage Event**: As the application is event oriented, the registered users must be able to manage their events on their network.

   2.1 **Create event**: A registered user can create an event of interest at any geographical location.

   2.1.1 **Acquire location**: The current location must be acquired from using the GPS sensor inside the mobile phone.

   2.1.2 **Upload Event**: The details event details (email, phone number, date-time, geographical location and event description) are uploaded to the server.

   2.2 **Edit event**: The event host can edit his created event and re-upload to the server.

   2.3 **Delete event**: The event can be deleted by the host at any time.

   2.4 **Listen and connect to event**: All the guests users can listen to the events created by his friends whom he/she is following.

   2.4.1 **Update current location**: The event guest who is travelling to the event must update his current location using the GPS sensor located inside the guest’s mobile phone.

   2.4.2 **Measure radius**: The guest who is following the event must keep track of the radius of the hosted event.

   2.4.3 **Process notification**: As soon as the user enters the visibility of any event, he/she receives the notification which can direct him to the event.

3. **Manage profile**: A registered user can manage his/her profile information.

   3.1 **Edit profile**: A registered user can modify his/her profile any time.

   3.2 **Delete account**: A registered user can withdraw from social network at any point of time.
4. SYSTEM DESIGN

This section elaborates the design issues of the mobile application which includes the OOAD (Object Oriented Analysis and Design) and the Database design.

4.1 Design Rationale

![Geo-Tag Application System Design](image)

Figure 4.1: Geo-Tag Application System Design

The system is designed as a client server model as shown in the Figure 4.1. Each request the user makes, is served from a remote server which has both web and database server. Any smart phone user can download the application from the server and install it on his/her phone. Any android application cannot directly connect to MySQL server that is hosted on internet that is why, all the requests to the server are made via the PHP pages. The key parameters to the MySQL database are passed as post parameters to the
PHP page and PHP page submits the query to the database and the retrieved results are printed on the html page in the JSON (java script object notation) and these results are sent back as response to the mobile application.

All the user network information is maintained on the MySQL database server that is hosted at the remote location on internet. All the requests to the database are made indirectly via PHP web server.

4.2 Object Oriented Analysis and Design (UML)

This application followed all UML design guidelines in the software development process. After the requirements definition, the use case diagrams are designed as shown in chapter 1 of Appendix A. The use cases are analyzed and the behavior of the system is designed using the sequence diagram as shown in chapter 2 of Appendix A. The structural view of the application is presented using the class diagram as shown in chapter 3 of Appendix A. The classes are also populated with their attributes and operations and the relationship between the classes are also marked in the diagram.

4.3 Database Design

This application uses MySQL database to hold the user information, geotagging information and social network information. An E-R diagram is shown in Figure 9.11 of Appendix A which contains all the tables required to hold social network information and their mapping information. A new table shown in Figure 9.12 is introduced into the social network model to achieve the geotagging operations.
5. SYSTEM IMPLEMENTATION

The mobile client was implemented in Android in approximately 1500 lines of code using Eclipse IDE, the web service was implemented in PHP and the user data is stored in the MySQL database in the backend. Here are some of the implementation details.

1. Mobile Client

The mobile client was developed using the Android SDK integrated into Eclipse IDE in Windows XP. The application requires access to the location services through the location API for Android and also to the networking services for communication with the web service. The Android platform abstracts many of the lower level implantation details that are involved in mobile communication that occurs internally or externally.

![Figure 5.1: Login and Registration](image)

Figure 5.1: Login and Registration
The application is designed with a simple user interface with the purpose of providing great usability to users. For the first time when a user downloads the application, the user has to register with the application using his identification information (username, password and email) as shown in Figure 5.1. Once the user is registered, the user is allowed to login into the application and the user can perform some social network activities such as following a friend, view friends’ status, view people on your network, update your current status etc. Each of the functionality is achieved using the “Activities” in android platform. The GUI for the mobile client consists of standard UI controls such as text fields, labels, buttons, list views etc. Eclipse IDE provides a GUI tool to draw all the user interface screens. Figure 5.2 shows all the features of the applications.

![Application home page](image.jpg)

Figure 5.2: Application home page
Along with the social network activities, this application introduces various features to achieve geo-tagging in social networks. The Events tab and Geotag tab are the two features that help in achieving geotagging functionality. The Events tab only shows the list events that are hosted in your network. Figure 5.3 shows the process of creating an event that is associated with a geotag.

![Figure 5.3: Creating an event](image)

Every event has geographical location information, contact information and other event relevant information (date-time, description and message) associated to the event. The feature “Locate your position” helps the user to find the current location using the GPS sensor. This application also prompts the user to turn on the location based services to get the location information.

Figure 5.4 shows the features that assist a guest to attend an event. The user can view all the events from his network and he chooses to listen to the events nearby, he needs to hit the “Listen To Your Events” button shown in Figure 5.3 and then the local mobile client
acquires the current location from the GPS sensors. Then the application acquires the geographical location of the hosted event and it filters the events based on the radial distance of the events. If an event is within the radial distance, the Broadcast-Receiver in Android platform creates a notification on the listener’s device as shown in Figure 5.4 and each notification is associated with a phone number, email and navigation that helps the listener to contact the event and also to drive to the event. Android has inbuilt activities such as ACTION_CALL for calling and ACTION_SEND for email. This application also provides with the navigation using the Google Map API in Android.

Figure 5.4: Features to assist event followers

A new notification is pushed for every 10 meter change in the distance and the maps are updated accordingly. Once the event expires, it is no longer visible to any followers in the network.
2.1 Web Server

Any android mobile client cannot directly connect to a database server on the cloud and hence a this web application is implemented around the new concept of Cloud Computing. The cloud infrastructure is provided by a private vendor (GoDaddy). This cloud infrastructure lets you run your applications using PHP programming language all that is to be done as a mobile client is to submit a HTTP Post request to the web server as shown in Figure 5.5 below. A HTTP Post request consists of a URL and Post parameters. The URL decides which page is to be chosen and the parameters are associated with the SQL query that is to be submitted to the data base.

```java
public static String executeHttpPost(String url, 
        ArrayList<NameValuePair> postParameters) throws Exception {
    BufferedReader in = null;
    try {
        HttpClient client = getHttpClient();
        HttpPost request = new HttpPost(url);
        UrlEncodedFormEntity formEntity = new UrlEncodedFormEntity(
                postParameters);
        request.setEntity(formEntity);
        HttpResponse response = client.execute(request);
        in = new BufferedReader(new InputStreamReader(response.getEntity().
                getContent()));
        StringBuffer sb = new StringBuffer("");
        String line = "";
        String NL = System.getProperty("line.separator");
        while ((line = in.readLine()) != null) {
            sb.append(line + NL);
        }
        in.close();
        String result = sb.toString();
        return result;
    } finally {
        if (in != null) {
            try {
                in.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Figure 5.5: Connecting to web server via internet service

18
A PHP web server reads the Post parameters that are passed from the mobile client as shown in Figure 5.6 and submits the appropriate SQL query to the MySQL database server and all the results are printed on the results page using the JSON and the results when sent back to the client can be read using the JSON API in java.

```php
<?php

include 'db.php';
$um=$_POST['userid'];
$search=$_POST['search'];

//run the query to search for the usernames that match
$query = "SELECT su2.user_id, su2.username
FROM SocialUser su2
WHERE su2.username LIKE '%$search%' AND su2.user_id!=$um AND su2.user_id NOT IN
(SELECT ct.dest_id FROM ConnectTo ct where ct.source_id=$um |");

$result = mysql_query($query) or die("Unable to verify user because : " . mysql_error());
//this is where the actual verification happens
$un=mysql_num_rows($result);

while($e=mysql_fetch_assoc($result))
    $output[]=$e;

print(json_encode($output));

mysql_close();
?>
```

Figure 5.6: Wrappers for MySQL database

Every request to the web server involves the same mechanism except that the mobile client activities and the PHP pages keep changing depending on the feature you use.
6. USER STUDY

A user study was performed on the design and usage of this application and there are some useful feedback from the users that helped in modifying the design and implementation of the application.

1. User Interface Study:

The User Interface is simple and intuitive and looks similar to any other social networking application. The UI elements included the EditText, Buttons and Labels. In this study, the users are given with the mobile application installed on an Android phone and asked to work with the interface of the application. This user study gave the following feedback from users regarding the user.

1.1 The Absolute layout used in case of an event creation interface tightly packs all the elements of that screen and also the form is not readable and hence, the interface is changed to a scrollable layout as shown in Figure 6.1.

![Figure 6.1: UI Feature Modifications after User Feedback](image-url)
1.2 The date and time for any event are manually entered in a text field but that looks like a fuzzy to the user and later, the time picker and date picker UI elements are used as shown in above Figure.

2. Application features study:

The application features meet all the features mentioned in the requirement definition. The features include all the social network functions and the geotagging operations. In this study, users are given with the applications installed on an Android phone asked to work with features of the application. This user study gave the following feedback.

2.1 For every geotag operation, the user has to go away from the application to change the location settings which makes user to lose track of the current state and hence, the application is modified to detect the current location settings and prompt user at the current page.

2.2 To create an event, the user has to find the location details on internet and upload the event to the remote server. This process kills the time of the user if he is at the event location. Hence, the application is modified to populate the location parameters automatically if the user wants to tag the current location.
7. EVALUATION AND RESULTS

To evaluate this application, a small user networks are created and each network is populated with a set of events and application successfully kept track of the events and directed the event followers to the event locations.

An event is created by a user in a network as sown in Figure 7.1. The user located his current location using the application and then uploaded the event to the remote server.

![Locating position and creating event](image)

Figure 7.1: Locating position and creating event

The user who follows the above user who created the event will be able to see the event in his application as shown in Figure 7.2. Once the follower is within the radial distance of the event he receives an event alert and from there he can call to the event or email to the event or navigate to the event as shown in Figure 7.3
The tagged location is visible to the users who are directly connected in the social network. Any user can create a tag at any location by adding some greeting message to it and he/she can also edit his time at any time before the tag expires. The users who have the app installed on their mobile devices will find the tagged places if they turn on their proximity alerts.
Each feature in the application is associated with a PHP page and an SQL query inside the PHP code. All the inputs from the mobile clients are stored in the database through the PHP pages as shown in Figure 7.4. All the inputs are the insert queries in PHP page and all the filters are the select queries inside those pages.

![Table Image]

**Figure 7.4: Relational table storing the events**

The event created in Figure 7.1 is directly stored into the database table as shown in above Figure. These data are submitted using a saveevent.php page which contains an insert query shown below.

```
"INSERT INTO UserEvents (User_Id, Event_Name, Event_Desc, Event_Date, Event_Time, Lat, Longtd, event_phone, event_email) VALUES ('$uid',
'sename', '$edesc', '$edate', '$etime', '$lat', '$longtd', $phone', $email')"
```

Along with the saveevent.php page, friends.php page to list the friend, location.php page to get the event location, register.php page to register the user with this application, searchuser.php to search for a specific user and status.php to list the status of the friends. Each of those pages is associated with the input/output parameters that are useful in exchanging the information between the mobile client and the database server.

All the parameters are submitted as post parameters to enhance the security of the application. No intruders can pass their own parameters via URL.
8. CONCLUSION AND FUTURE WORK

This application provides event oriented social networking using the geotagging functionality. This mobile app strengthens the social network and also markets the geographical locations. This app may also drag attention of the companies which need to communicate to their users by some tags and messages while on road.

This application is done by following the Software Engineering practice which included the requirements gathering, analyzing the gathered requirements, designing the system, implementation and testing. This application highly emphasizes on the geotagging functionality rather than on the social networking.

Some of the technical challenges involved in this project include the connectivity of the mobile client with relational database. This is achieved using the PHP pages and the JSON objects in PHP and Java. There is much more work included to design a seamless and easy to use application for small and limited devices such as smart phones.

The manufacturing rate of smart phones is increasing rapidly and this type of mobile applications is required for the users to find their useful applications that take the advantages of the smart phones. This application is one such attempt that encourages users to develop a healthy and interactive social network.

Future Work:

As the part of feature work the application should provide all the major functionalities that are provided by most of the social network applications (Facebook, Twitter, MySpace etc.) such as photo uploads related to the events, photo tagging, post comments on friends activities on the network etc.
To enhance the geotagging features, the user should be allowed to upload a picture of interest to the tagged location to customize the destination pictures seen by the followers. The navigation in the current application could only show the map to direct to the event. The map must be embedded with the driving directions to enhance the communication with the event.

This application uses the location services for all the geotagging operations. In case of any GPS failure, the application must use alternative sources for acquiring the approximate location details. Public Wi-Fi hotspots or cellular network towers can be used to approximate the locations.
BIBLIOGRAPHY AND REFERENCES


APPENDIX A

1 Use-case Diagrams

1.1 Geo-Tag application system diagram:

![Use-case diagram for the geo-tag application]

Figure 9.1: Use case diagram for the geo-tag application

Usecase description:

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case allows users to register into Geo-Tag Application</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application Users</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Users must navigate to their respective registration pages</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
<tr>
<td>Main Successful Scenario:</td>
<td>-Users must enter the desired username and password that can be used in future to log into the Geo-Tag application.</td>
</tr>
<tr>
<td></td>
<td>-The users must enter their personal details</td>
</tr>
<tr>
<td></td>
<td>-If the username doesn’t exist already then, a new user is created and added to the database</td>
</tr>
<tr>
<td>Alternative Scenarios:</td>
<td>Existing Username</td>
</tr>
<tr>
<td></td>
<td>-If the username already exist, then the users have to choose</td>
</tr>
</tbody>
</table>
1.2 Manage network usecase:

**Usecase description:**

<table>
<thead>
<tr>
<th><strong>TITLE:</strong></th>
<th><strong>Search User</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>This use case allows users to search other registered users</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Application Users</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>Users must login to the application</td>
</tr>
<tr>
<td><strong>Flow of Events:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| **Main Successful Scenario:** | -Users must enter the username to search.  
                              -If the username exists, matched usernames are returned |
| **Alternative Scenarios:** | **Non-Existing Username**  
                              -If the username doesn’t exist, no results are returned.  
                              -Try different username |
<table>
<thead>
<tr>
<th><strong>TITLE:</strong></th>
<th>Follow User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>This use case allows users follow other registered users</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Application Users</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>Users must login to the application and search for a specific user</td>
</tr>
<tr>
<td><strong>Flow of Events:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| **Main Successful Scenario:** | - Users must enter the desired username and password that can be used in future to log into the Geo-Tag application.  
- The users must enter their personal details  
- If the username doesn’t exist already then, a new user is created and added to the database |
| **Alternative Scenarios:** | Existing Username  
- If the username already exist, then the users have to choose a new username.  
- Then submit the request again |

<table>
<thead>
<tr>
<th><strong>TITLE:</strong></th>
<th>Un-Follow User</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>This use case allows users to register into Geo-Tag Application</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Application Users</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>Users must navigate to their respective registration pages</td>
</tr>
<tr>
<td><strong>Flow of Events:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| **Main Successful Scenario:** | - Users must enter the desired username and password that can be used in future to log into the Geo-Tag application.  
- The users must enter their personal details  
- If the username doesn’t exist already then, a new user is created and added to the database |
Alternative Scenarios:

**Existing Username**
- If the username already exists, then the users have to choose a new username.
- Then submit the request again

1.3. Manage event usecase:

![Usecase diagram for managing the user events](image)

**Figure 9.3: Use case diagram for managing the user events**

**Usecase description:**

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Edit Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case allows users to register into Geo-Tag Application</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application Users</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Users must navigate to their respective registration pages</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
</tbody>
</table>
| **Main Successful Scenario:** | - Users must enter the desired username and password that can be used in future to log into the Geo-Tag application.  
  - The users must enter their personal details  
  - If the username doesn’t exist already then, a new user is created and added to the database |
| **Alternative Scenarios:** | **Existing Username**  
  - If the username already exist, then the users have to choose a new username.  
  - Then submit the request again |

<table>
<thead>
<tr>
<th><strong>TITLE:</strong></th>
<th><strong>Delete Event</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Summary:</strong></td>
<td>This use case allows users to register into Geo-Tag Application</td>
</tr>
<tr>
<td><strong>Actors:</strong></td>
<td>Application Users</td>
</tr>
<tr>
<td><strong>Preconditions:</strong></td>
<td>Users must navigate to their respective registration pages</td>
</tr>
<tr>
<td><strong>Flow of Events:</strong></td>
<td></td>
</tr>
</tbody>
</table>
| **Main Successful Scenario:** | - Users must enter the desired username and password that can be used in future to log into the Geo-Tag application.  
  - The users must enter their personal details  
  - If the username doesn’t exist already then, a new user is created and added to the database |
| **Alternative Scenarios:** | **Existing Username**  
  - If the username already exist, then the users have to choose a new username.  
  - Then submit the request again |
1.4. Create event usecase:

**Figure 9.4: Usecase diagram for creating an event**

**Usecase description:**

<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Acquire Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case allows users to register into Geo-Tag Application</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application Users</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Users must navigate to their respective registration pages</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
</tbody>
</table>

**Main Successful Scenario:**
- Users must enter the desired username and password that can be used in future to log into the Geo-Tag application.
- The users must enter their personal details
- If the username doesn’t exist already then, a new user is created and added to the database

**Alternative Scenarios:**
- Existing Username
  - If the username already exist, then the users have to choose a new username.
  - Then submit the request again
<table>
<thead>
<tr>
<th>TITLE:</th>
<th>Upload Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary:</td>
<td>This use case allows users to register into Geo-Tag Application</td>
</tr>
<tr>
<td>Actors:</td>
<td>Application Users</td>
</tr>
<tr>
<td>Preconditions:</td>
<td>Users must navigate to their respective registration pages</td>
</tr>
<tr>
<td>Flow of Events:</td>
<td></td>
</tr>
<tr>
<td>Main Successful Scenario:</td>
<td>- Users must enter the desired username and password that can be used in future to log into the Geo-Tag application.</td>
</tr>
<tr>
<td></td>
<td>- The users must enter their personal details</td>
</tr>
<tr>
<td></td>
<td>- If the username doesn’t exist already then, a new user is created and added to the database</td>
</tr>
<tr>
<td>Alternative Scenarios:</td>
<td>Existing Username</td>
</tr>
<tr>
<td></td>
<td>- If the username already exist, then the users have to choose a new username.</td>
</tr>
<tr>
<td></td>
<td>- Then submit the request again</td>
</tr>
</tbody>
</table>

5. Listen and Connect to event usecase:

![Use case diagram](image.png)

Figure 9.5: Use case diagram for listening and connecting to an event
6. Manage user usecase:

![Diagram](image)

Figure 9.6: Usecase diagram for managing user profile

2 Sequence Diagram

2.1 Follow user sequence:

![Diagram](image)

Figure 9.7: Sequence diagram to search for a user on network
2.2 Create event sequence:

Figure 9.8: Sequence diagram to search for creating an event

3. Listen and Connect to event sequence:

Figure 9.9: Sequence diagram to listen and connect to any event
4.4 Class Diagram

Figure 9.10: Class diagram for the geo-tag application
4.4 Database Design (E-R Diagrams)

1. Social Network

![E-R Diagram for social network.](image1)

Figure 9.11: E-R Diagram for social network.

2. Event Tag Database

![Modified data model to integrate geo-tagging in social networks.](image2)

Figure 9.12: Modified data model to integrate geo-tagging in social networks