National College of Ireland
BSc in Computing
2015/2016

Piotr Czerniejewski
12488942
piotr.czerniejewski@student.ncirl.ie

GeoMe - Post-it notes on the street

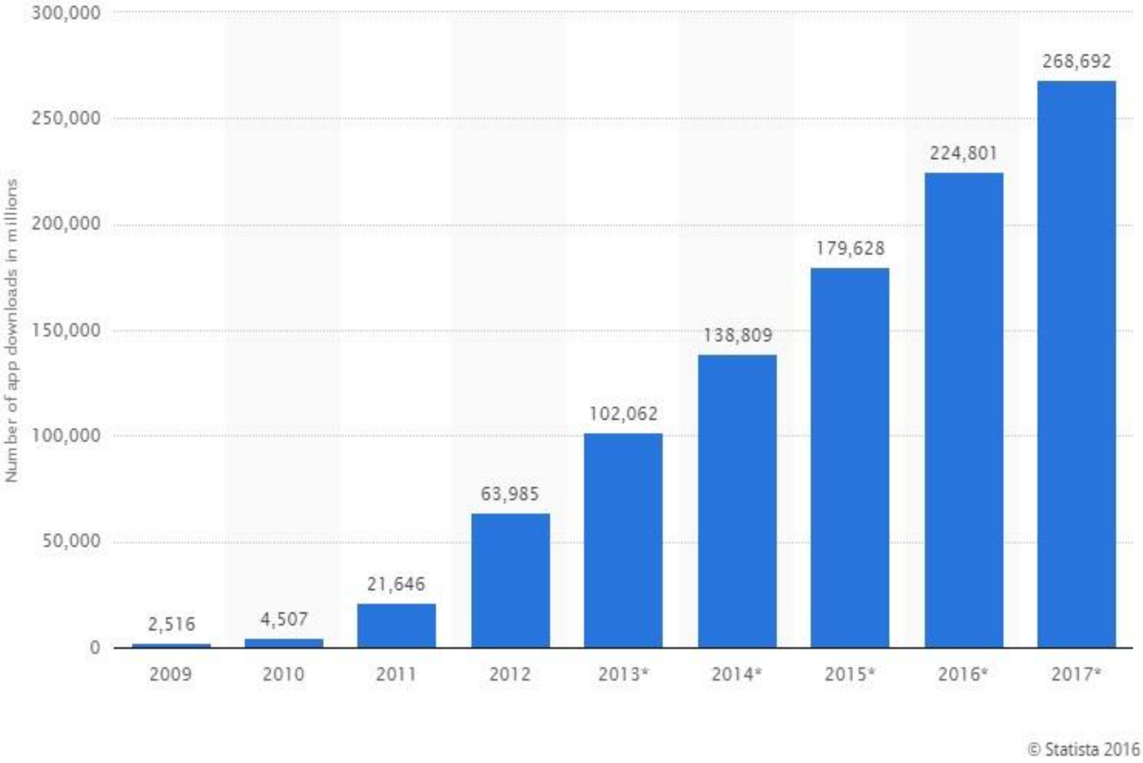
Final Presentation Report

Technical Report

Table of Contents

Executive Summary

With the mobile market growing, internet access in urban areas improving in Ireland and Worldwide, much more attractive mobile internet plans and the number of mobile apps downloaded each year increasing by 30%~ per year since 2012 (63,958 mil) to 2015 (179,628 mil) proves that people love their apps and use them daily. Combined with a recent study that showed that the average time spent on mobile phones by users worldwide is 90 minutes a day or a staggering 23 days a year, gave me an idea to use this statistics to provide an application for these users that would allow them to interact anonymously with each other based on their daily travel, mobile usage and places they visit by utilizing their mobile phones geo-location that would allow for them to leave messages for others to pick up with their devices as they pass the geo-location the message was left at.



© Statista 2016

These messages can range from geocaching for prizes and competitions, events happening later in the area that day, giving small reviews of local shops, cafes and restaurants. Share discount codes and promotions, advertise any personal belongings or rooms to rent, that any passer by would already be in the neighbourhood and could see the house in person and decide if it’s suitable for them. Or maybe inscriptions and poems for others to read and share. And with messages having a short-to-medium life span, would make each day a new experience as only the latest notes left by other users would be picked up the mobile device. Creating a sense of excitement and surprise whenever the user receives a notification of a new Geo-location based message they received.

With the popularity of other geo based application such as Swarm, Yik-Yak and Google places, I believe mobile app users would be very interested in this concept. And thus this gave birth to GeoMe, a native android application for anonymous social sharing of text, bound to geofences spread around cities and countries!

1 Introduction

1.1 Background

I feel that the excitement of using your smartphone is gone. We take them for granted, all the apps are copying each other, the top and profitable game is recreated by others. We watch Youtube and message friends and family constantly, our internet is always on waiting for these messages.

Then I had an idea to create an application that will give excitement of hearing the notification sound. By not knowing what other users will leave for you to pick up while in transit, hopefully it can create a sense of excitement. It would have to have minimal impact on the phone's performance as to not slow down or hinder day-to-day usage.

It would also test a lot of material I have covered over the years. Android development for the client application, PHP backend for API the client app will use to communicate with the MySQL database responsible for storage of these geo messages.

1.2 Aims

- Clean and user friendly UI
- Intuitive design for both the content creators as well as the content consumers
- Accurate geo-location system
- Minimal network and power usage with maximum uptime for receiving the messages

1.3 Technologies

1.3.1 Java in Android Studio

Java was chosen by me for the development of the client application. Being not only native for android development but also encourages object orientated programming, making easier to write modular code. This can allow for the application system to be split up into smaller parts that can be easier for testing, scaling, improving and reusable in multiple parts, saving time during the development schedule.

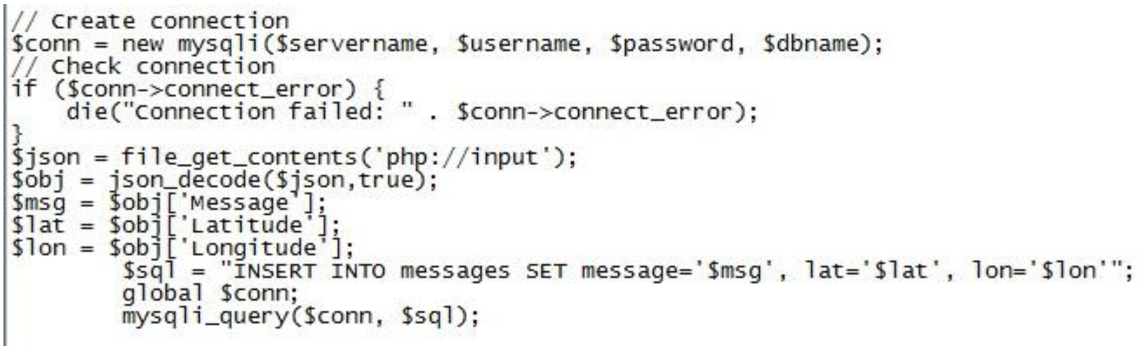
By being a mature and refined language there is a lot of tutorials, documents, plugins and libraries available to expand and enhance the system in a simple and quick way.

The IDE of choice is Android Studio. It is fully supported by google, the owners and developers of the android framework and the IDE comes in baked in with plugins, development tools, android packages to easily add functionality such as GPS, Maps, Advertising, touch screen functionality and internet access. It also comes in with a build in AVD, or the android virtual device manager that helps with local testing by emulating an android phone that allows the test application to be installed on it and test it’s functionality.

1.3.2 Server Side PHP

For communication between each android device where the application is running and the database where the messages are stored and pulled from into and from the device, a middleman is needed to receive and manipulate the data transmission to and from the server. PHP scripting was the choice. It is easy to learn but as equally powerful as any other server side language. It isn’t OS dependant meaning it can be run on any system or server. Scalability allows for easy expansion for more complex systems, this can be accomplished by adding more clusters and easy integration with database systems with its built in methods for selecting and inserting into databases.

It is also perfect for string manipulation with one line methods for changing the composition, length and format of any string being manipulated by the script. And finally by being a refined and older language there is many resources available for improving the code, with frameworks and documents and the PHP documentation on the official website.



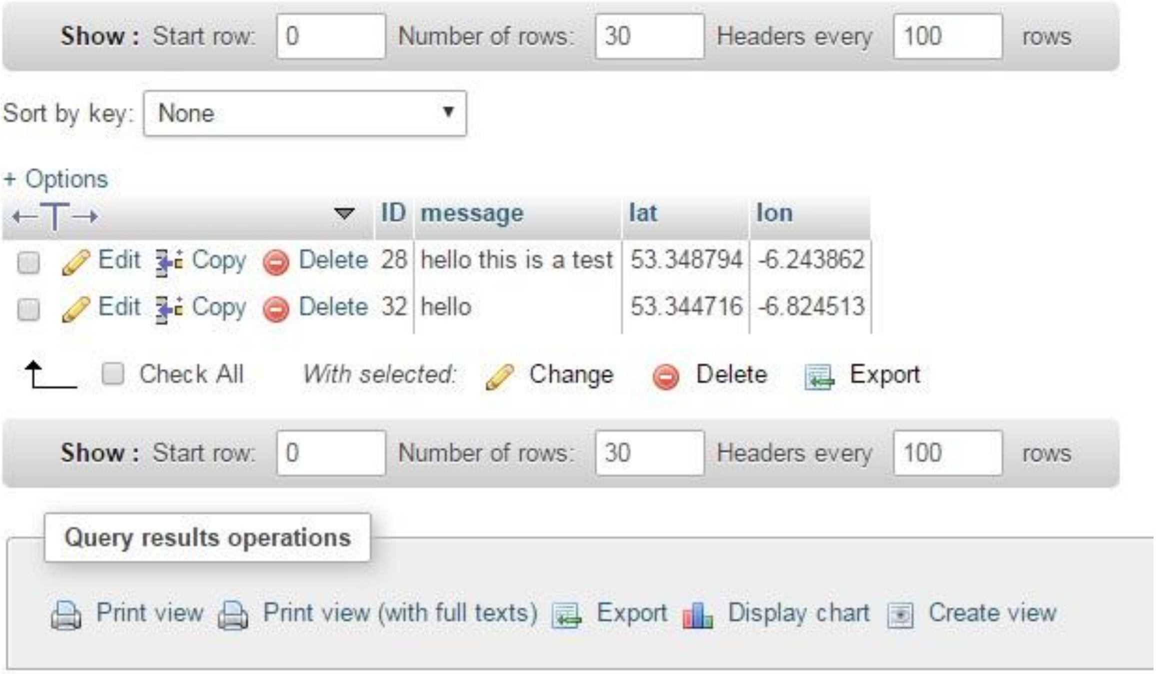
```php
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$json = file_get_contents('php://input');
$obj = json_decode($json,true);
$msg = $obj['Message'];
$lat = $obj['Latitude'];
$lon = $obj['Longitude'];
        $sql = "INSERT INTO messages SET message='$msg', lat='$lat', lon='$lon'";
        global $conn;
        mysqli_query($conn, $sql);
```

This is an example of how easy integration with a database and inserting data into a table can be using PHP.

1.3.3 MySQL

For messages storage and their attached GPS locations I’ve chosen MySQL for the current iteration of the project. With easy control of the database by either command line or the GUI provided by phpmyadmin I can control the type of data stored.

This is an easy to use GUI that helps with controlling the structure of the database storage and making sure everything is stored as it should be.



Show : Start row: 0 Number of rows: 30 Headers every 100 rows

Sort by key: None

ID	message	lat	lon
28	hello this is a test	53.348794	-6.243862
32	hello	53.344716	-6.824513

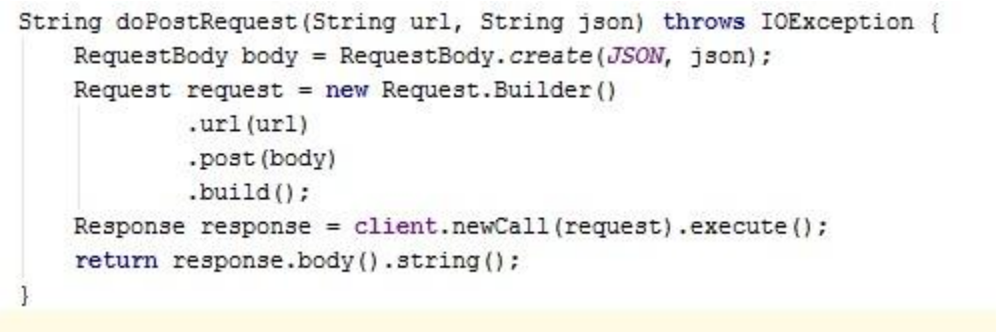
I’ve chosen it as it's easy to use, scalable in sense that with an increase in data flow I can create more tables and distribute the flow between them. As well as simple integration between my PHP scripts for handling the data.

1.3.4 OkHTTP

OkHTTP is an open source library for internet transmission between network devices, ranging from smartphone to smartphone, PC to PC ,to as is my case smartphone to server.

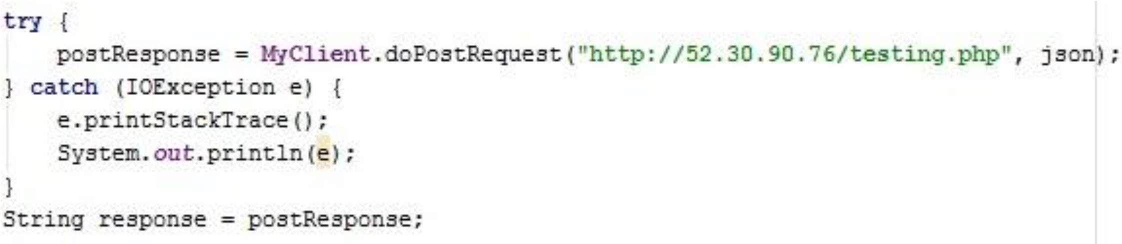
It uses the standard internet HTTP protocol with the option of HTTPS for increased security, which is important in my case. And the overall structure is based on REST. This allows to create a custom API for communication between the network devices that suits an individual's needs. These can range from CRUD operation to simple GET/POST methods for receiving and sending data.

The reason for choosing this library is that the standard Apache protocol is deprecated on the API version of Android I’m developing at. As I wanted to use the latest and up to date tools, this led me on a search for a newer and reliable library, which OkHTTP delivered.



```
String doPostRequest(String url, String json) throws IOException {
    RequestBody body = RequestBody.create(JSON, json);
    Request request = new Request.Builder()
            .url(url)
            .post(body)
            .build();
    Response response = client.newCall(request).execute();
    return response.body().string();
}
```

This is an example of a simple POST method to communicate with a server and send data in the POST message. This method then is called by creating an object of the class it is stored in and filling in the parameters that point to the internet location of the server, such as this:



```
try {
    postResponse = MyClient.doPostRequest("http://52.30.90.76/testing.php", json);
} catch (IOException e) {
    e.printStackTrace();
    System.out.println(e);
}
String response = postResponse;
```

1.3.5 Amazon Web Services

Amazon Web Services or AWS for short, is a cloud based platform for hosting, integrating and storing complex systems on a easy to use, scalable, reliable and well documented platform. As such, this is my choice for hosting the back end of my application as well as the database.

The part of AWS I will be using is the EC2 (Elastic Cloud), which creates a virtual server of your choice, be it a microsoft server or a linux server. In it, it gives full control of what type of systems and programs you want to utilize to make your system functional. From host servers, network performance tools, database systems and file storage. As each server is an individual instance, it can be easily scaled by adding more instances or upgrading the server to higher specifications to accommodate more data flow as well as more storage.

For my project, I’ve chosen a linux instance as it’s lightweight and secure. As well as the experience of using a linux system will be helpful in the future as from my research, I concluded most high tier server applications are stored on UNIX machines. On it I’ve installed a host server by the name of LAMP (Linux Apache MySql PHP), that gives me all the tools required for this system. Apache for communication over the internet, MySQL for data storage and PHP for the transmission and data manipulation. All in one single package installation. Furthermore, I can limit access to individual folders, making it a much more secure hosting solution, stopping any to be malicious users from accessing any sensitive data.

1.3.6 GPS

The core of the application functionality depends on the GPS system. GPS or the global positioning system allows to pinpoint a person's location in relation to where they are, as long as they have some kind of network connection. It is structured in two sets of numbers, one for latitude and the second set for longitude. By comparing the two in relation to the globe, where the two sets cross, that’s the determined position.

For accuracy, a certain amount of decimal places are required to more precisely pinpoint the location. For my needs, 5 decimal places are sufficient as it will give me a 1.1 meter accuracy range. This is standard in commercial application as it enough to distinguish trees from each other on a map. Also the minus or plus sign at the beginning tells the location to be either North or South in case of latitude and West or East for longitude.

But the GPS location of each user will be used to create a GeoFence. Rather than hoping they will pass or stop at the exact GPS value, a radius will be created around that GPS location, creating an invisible fence that once the user enters, it will trigger a specific function. In my case, a message will be received if a user enters that GeoFence and the message will disappear if the the user leaves the GeoFence. This will allow room for error in the location, give more freedom in implementing the system functionality as well as control for me as the system architect to determine the radius of the GeoFence, to determine when a message is send to the user, as this will be tweaked throughout the development to give best results and in the future control the performance.

2 System

2.1 Requirements

2.1.1 Functional requirements

2.1.1.1 Requirement 1: Sending a message

Description & Priority

This a vital requirement as the application is powered by the users and their ability to send messages to the database with their coordinates, allowing other users to pick those messages up. The user writes the message, phone registers the GPS of their location, appends it to the messages and sends it to the server to be stored in a database.

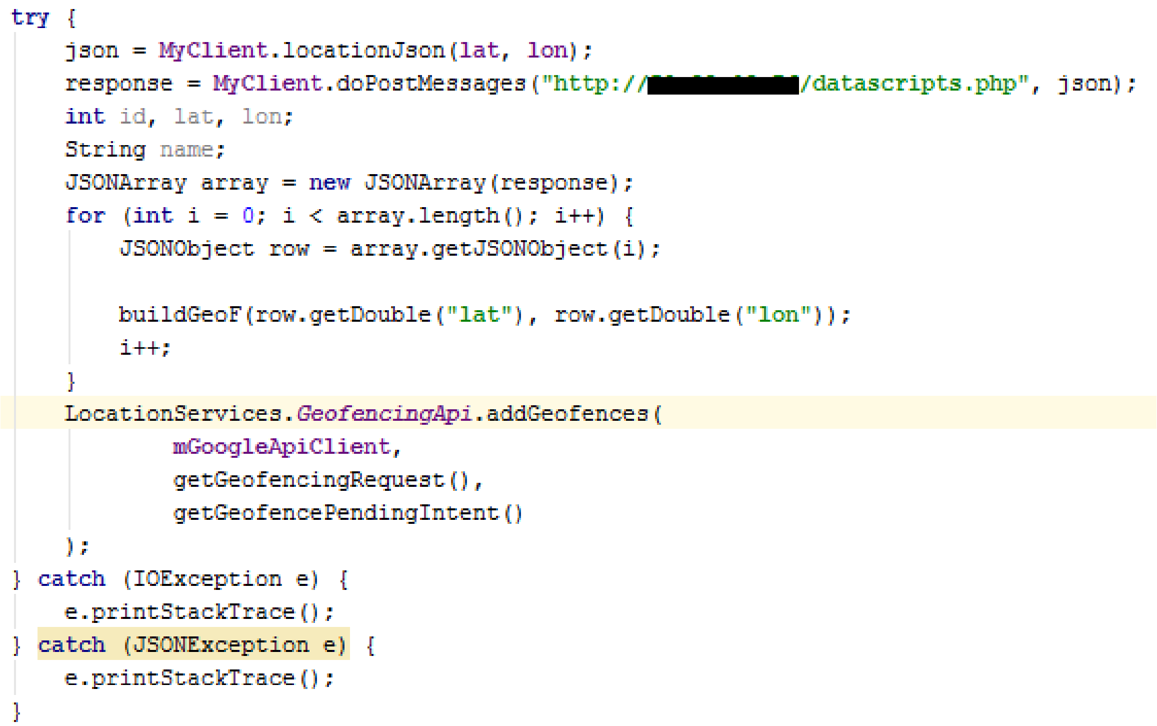
Assumption / Risk

A person might start to write their message at the location they want to leave their message at, but while they are writing they are on the move either walking or in the car / bus. Should the GPS co-ordinates be snapshotted while they are writing the message or should it be appended to message upon pressing the send button? Current evaluation from my focus groups led me to believe that the users will expect to the message to be ‘left’ at the location they click send but I feel like I should implement the snapshot feature and communicate that to the users of such functionality.

2.1.1.2 Requirement 2: Receiving Message

Description & Priority

The process of receiving a message from Geo fences. This is a top priority as it's at the core of the application. If the users can’t receive the messages left by other users, the the whole point of the application is moot. It has to be stable, fully functional and well tested as to work as its intended. The device constantly monitors the user's GPS location, communicates with the server to identify and messages close by, pull them from the server and create the geofences natively on the smartphone, and once the user enters the created GeoFence, the message will appear and notify the user. As well as, in the case that the user closes the application, it should work in the background, that’s why it should work as a service, showing up on the phone bar indicating that it’s running and will receive the messages even when the phone is locked.



```
try {
    json = MyClient.locationJson(lat, lon);
    response = MyClient.doPostMessages("http://[redacted]/datascripts.php", json);
    int id, lat, lon;
    String name;
    JSONArray array = new JSONArray(response);
    for (int i = 0; i < array.length(); i++) {
        JSONObject row = array.getJSONObject(i);

        buildGeoF(row.getDouble("lat"), row.getDouble("lon"));
        i++;
    }
    LocationServices.GeofencingApi.addGeofences(
            mGoogleApiClient,
            getGeofencingRequest(),
            getGeofencePendingIntent()
    );
} catch (IOException e) {
    e.printStackTrace();
} catch (JSONException e) {
    e.printStackTrace();
}
```

Here is a code snippet of the client application sending the user's location and receiving all the messages in their vicinity as a response, sending it to the next requirement of creating geofences.

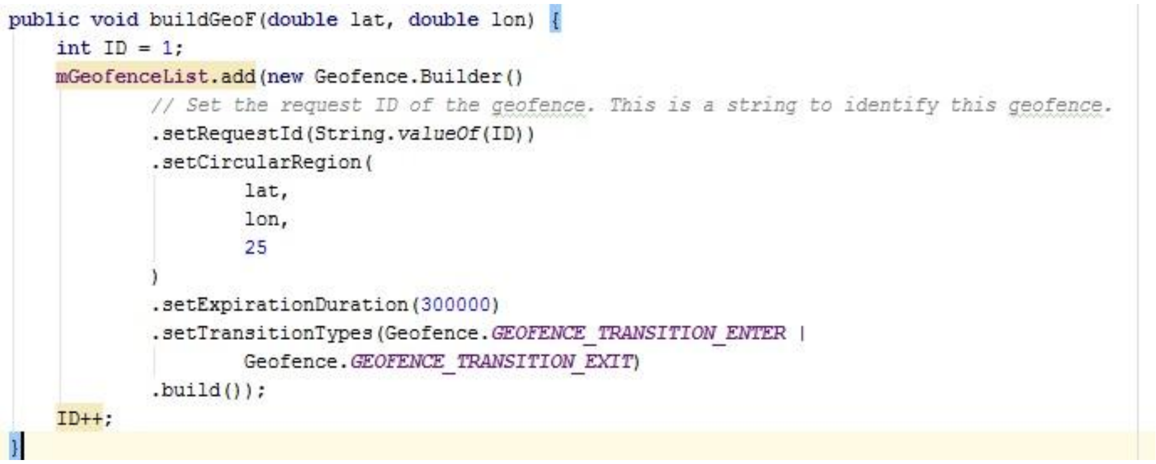
Assumption / Risk

A system will have to be in place to accommodate multiple messages overlapping their geofences, thus the GUI will have to scale to the amount of messages being displayed at one time.

2.1.1.3 Requirement 3: Creating Geofences

Description & Priority

This requirements is at the core of the application and is absolutely necessary for the whole system to work. Geofences are used to notify and display messages to the user. They are created by contacting the server and pulling all the messages from vicinity of the user’s current GPS location. By using the Google Maps API, each messages is bound to a geofence, which location is based on the coordinates of each message. These geofences have two transition parameters associated with them. Those are enter and exit. Upon entering the geofence, user will be notified and can read the message. And when the user exits the geofence, the message disappears and any notification still linked to that message, erased.



```
public void buildGeoF(double lat, double lon) {
    int ID = 1;
    mGeofenceList.add(new Geofence.Builder()
            // Set the request ID of the geofence. This is a string to identify this geofence.
            .setRequestId(String.valueOf(ID))
            .setCircularRegion(
                    lat,
                    lon,
                    25
            )
            .setExpirationDuration(300000)
            .setTransitionTypes(Geofence.GEOFENCE_TRANSITION_ENTER |
                    Geofence.GEOFENCE_TRANSITION_EXIT)
            .build());
    ID++;
}
```

This method is responsible for creating geofence for each message. By taking in the GPS coordinates, defining the radius, lifespan and transition types, it can then monitor any change to each geofence.

Assumption / Risk

It is dependant on the Google Maps API to fully function for it to work.

2.1.2 Data requirements

The data requirements were mostly research into the accuracy of GPS system and the amount of decimal places required for a desired precision range. There will also be a need to create a certificate to use HTTPS protocol correctly.

2.1.3 Resource utilization requirement

To maximize battery and network efficiency, the system will have to dynamically adjust the time period of updating and contacting the server with location updates. If a person is walking, the update time can be a bit slower but if the user is on a bike or car, the frequency of updates will have to increase. This will give maximum efficiency, response time, battery usage and network usage. As if the user is stationary, there won't be any location changed, thus no need to check for geo fences around the user.

2.1.4 Security requirement

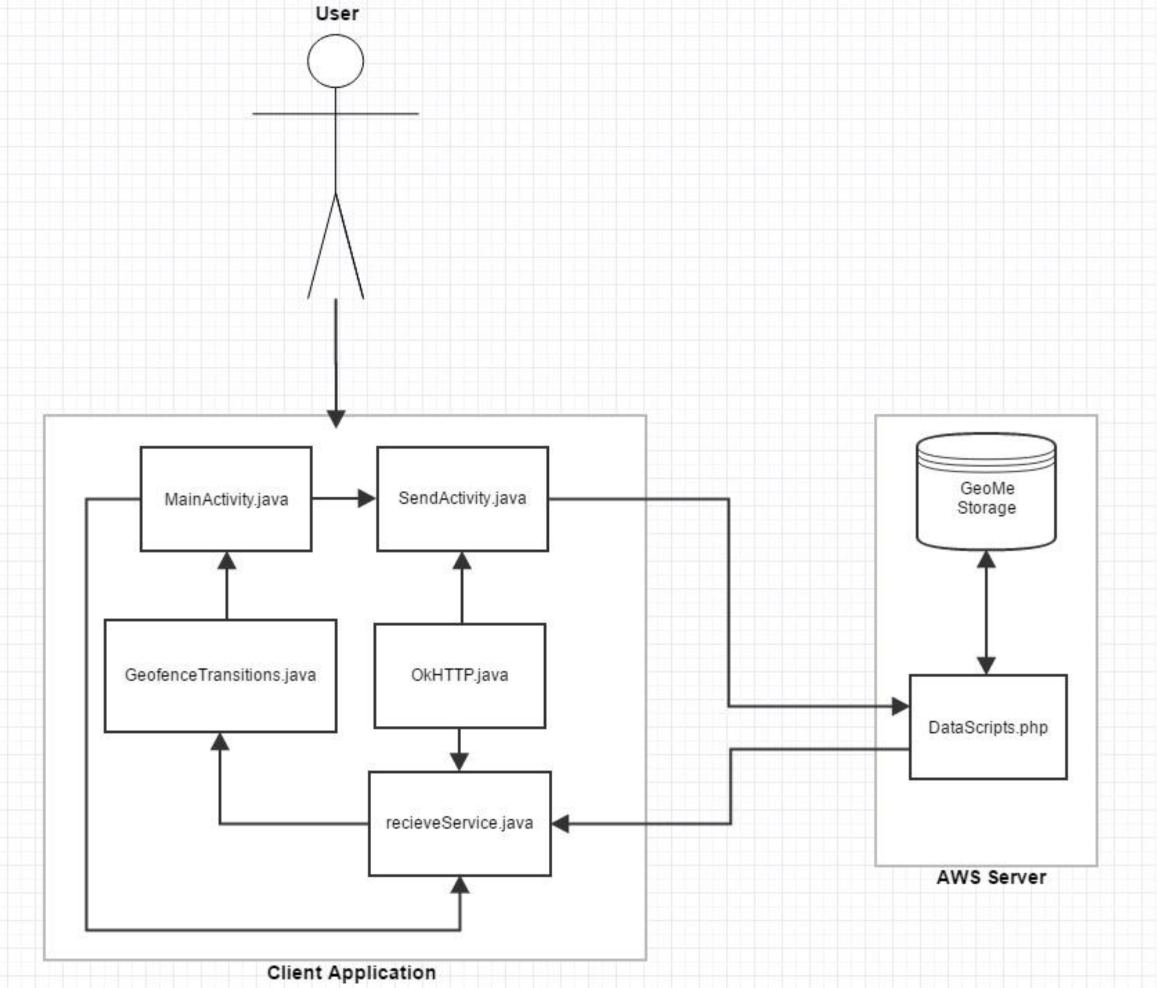
As the GPS co ordinates may be considered sensitive information, that could in be used in a malicious way of tracking the user's location, it will require a secure mode of transport from and to the server. Thankfully the latest version of OkHTTP has allowed the implementation of HTTPS protocol for secure transmission. Only requirement is the creation of a certificate to validate the connection.

With messages being stored in a SQL database, I had to be careful of SQL injections and any potential harm that may come from it. A simple solution was inherited into php that allows to strip a string of any special characters. Stopping from any malicious user from extracting data or breaking into the database itself.

2.1.5 Usability requirements

The user will only interact with the Android application, using their smartphone. Thus the main input will be the touch screen. It also requires a proper internet connection to fully function as well as the user has to give permissions to the app, to access their current location.

2.2 Design and Architecture



The system is split into two distinct systems, the android application and the web service. The Actor, being the user initializes the application and the MainActivity starts. The activity initializes and starts the recieveService responsible for contacting the server, using the OkHTTP class which includes API calls to the server. This allows for the client to receive messages from the database using the DataScripts script hosted on the server side. This service is responsible for creating and monitoring geofences, and any transition changes within a geofence is parsed by the GeoFenceTransitions class.

The GeoFenceTransitions class is responsible for identifying what transition has been triggered, be it either a user has entered or left a geofence. If entered, it will notify the user and display the message associated with that geofence. If left the geofence, it will remove the message and delete the notification invoked on the initial transition.

Functionality of leaving messages at a location of the user is handled by the sendingActivity. It will compose of a simple GUI to allow for the message to be written and upon sending it, the user's location will be captured, appended to the message and using the OkHTTP class it is send to the web service to be handled by the DataScripts file and stored in the database.

The Web Services system is responsible for the storage of messages as well as selecting messages to send to each individual user based on their location. The DataScripts is the end point for the data transmitted from the android application. It will validate the data, split it into individual parts and insert them into the database. It is also responsible for extracting and passing on messages from the database back to the android application to be displayed to the user.

The database, being a MySQL based, for the time being will consist of a single table that will store the messages and their associated GPS coordinates. As well as having a parameter table that will have values for the DataScript to use, in relation to the amount of messages to send back to the user and the radius for the GeoFences. This way, no parameters are hardcoded into the application and will allow for tweaking whenever any issues arise.

2.3 Customer testing

I have conducted a focus group to assess the viability of the project as well as decide on the direction of the project. The group consisted of 10 people from the ages of 19 to 49, with a Male to Female ratio of 7:3.

They gave valuable insight into the different aspects and directions I can take with the project. Such that the idea of a dedicated geocaching (treasure hunting) was good, it was too much of a niche and novel idea for major success. Another path I considered was replacing street signs with advertising and posters, with an application that would replace them and show the advert while passing the shop/business but no one from the group would download a smartphone app to be advertised to.

This led to the final idea of a general social application, for anonymous message sharing and communication using GPS locations. Furthermore I asked who would write messages themselves for others to read and it was a split right down the middle. This made me realize there are two types of users. Content creators and content consumers. This will require ease of usability, intuitive design and clean interface for the GUI to make both groups of users happy.

2.4 Functional Testing

Functional testing was conducted with a practical approach and constant field testing. In the early stages I have placed geo messages along the path I commuted to college and try to pick all them up along the way. At this point I found a bug where the inaccuracy of GPS made me not enter some of the geofences, which had a radius of 5 meters. This made me increase it to 25 meters, the width from one side of the street to another and it worked perfectly. This had the positive effect of increasing the potential range of messages to be received and gave room for error.

Next stage was a beta test for a group of individuals I could count on. I planned an online recruitment for my beta but was afraid of people not sending back feedback. This made me turn to my friends. I picked 6 of them, asked them to install my application and try to use it as much as they could. I went into my database and inserted some default messages and let them pick them up, as well as write their own. Every few days they wrote to me their feedback on what they received, how often and their thoughts.

The results were as follows: the mean number of geo messages received was 5 a day. Mean number of messages left was 3. Battery usage was 10% over a period of 5 hours run time. Another issues that arose from the beta test was that emojis were not properly displayed, character limit was too big as it broke the flow of the GUI and one participant couldn’t use it at all as his location service settings were not at an appropriate level required.

After resolving these issues I focused on edge case scenarios and tried to break my application. And at each step added error handling. Such as trying to leave a message with internet off, in which case the user would be asked to turn it back on. Leaving an empty message or with location turned off. For each error I could find, I added a form of error handling.

3 Conclusions

One of the major limits for this project to fully start will be the lack of messages left for people to pick up in the early lifespan of the application. There would have to be a synthetic injections of generic, pre generated messages inserted into the database. That way new users will have some incentive to start using it, get accustomed to the UI and the idea, while more users join in and an organic system of fresh messages will be generated by the users.

But a main advantage is that there is nothing like it out there on the market. There are similar concepts and multitude of GPS based applications, but no exact competition with this project. It may or may not be a start of a new trend, but if it will, this project will be at the top as the first one to be released.

4 Further Research and Development

Over time with the database growing over time with more and more messages, a more robust and performance driven system may be required. I heard good things about mongoDB but due to time constraints and no knowledge I had to put it off the work list.

More functionality is also a major consideration. Such as video and picture capabilities. Able to leave short duration videos at locations. But this brings a larger need for moderation as well as storage capabilities.

As the application at the moment is solely available for Android, in the near future I plan on researching Apple development to bring my application to a wider audience. By porting it to iPhone devices, my market share would greatly increase.

5 References

Matt Silverman. (2016). *Number of mobile app downloads worldwide from 2009 to 2017 (in millions).* Available: http://www.statista.com/statistics/266488/forecast-of-mobile-app-downloads/. Last accessed 2nd Feb 2015.

-. (2014). *23 DAYS A YEAR SPENT ON YOUR PHONE.* Available: http://www.mobilestatistics.com/mobile-news/23-days-a-year-spent-on-your-phone.aspx. Last accessed 2nd Feb 2016.

developer.android.com

6 Appendix

6.1 Requirements Specification

Table of Contents

1 Introduction

Purpose

The purpose of this document is to set out the requirements for the development of GeoMe, a location based android application that allows users to leave and receive anonymous social messages based on their phones GPS location. Main aspect of the application is the use of geofences to alert users of a message allowed to be read inside that geofence, written by another user at that location.

The intended customers are geocachers creating treasure hunts, radio stations creating prize events to find their messages left for their listeners, energetic young people who love adventures, event marketers to engage their demographic in a different and fun way, social groups sharing their events or information and tourist boards allowing to leave information outside points of interest.

Project Scope

The scope of the project is to develop an application that utilizes the google maps API and it’s geofencing library. The client will be a native smartphone application that fully utilizes the phones GPS functionality, creation and listening of geofences and push notifications. All of the geofencing and server connectivity will be handled by a background service, listening on location changes and building/listening of the geofence transitions. This application will also need a back end server for handling user submitted messages, calculating distance to retrieve closest geo messages and connectivity to the database. The database will be necessary for all the storage, as well as an automated DB process that will keep it updated and fresh, deleting any old messages based on a set time period.

The client site will be build for Android due to my experience with Java and android development as well as the ease of testing and emulating due to free nature of Google's development tools. The back end server side will be written in PHP as the documentation, libraries and tools will allow me to quickly create a manageable and scalable REST web service that will be a mid point handler between the client and database. While the database will be MySQL based due to time constraints I want to use something I’m familiar with, but I might use MongoDB due to performance increases if the development time will allow it or the near future.

The whole back end; including the database, will be hosted on the AWS EC2 system. Mainly due to their reliability as in 2014, their uptime was 99.9997%, being down only for 2 hours in the whole year. This is an important factor to me. The EC2 is linux based, specifically Ubuntu, which shouldn’t be a problem for me due to my experience with linux.

Definitions, Acronyms, and Abbreviations

EC2	Elastic Cloud
AWS	Amazon Web Service
PHP	PHP:Hypertext preprocessor
GPS	Global Positioning System
REST	Representational State Transfer

2 User Requirements Definition

The users and clients of this application will want a simple and clean user interface, that they navigate through. Battery usage will be a big issues as well, as they'll want to run this application throughout the whole day, which can put a stress on the battery life. They wouldn't want their name to be attached to the message, so it will be fully anonymous, unless otherwise specified.

Due to the anonymous nature of messages, I may risk myself as the sole reason for legal issues in cases of harassment or cyberbullying. This will require me of someway to track what user wrote which messages. For the time being each message written by a user is associated with that device's unique Android ID but in the future I plan on a more elaborate login system for administrative purposes.

3 Requirements Specification

The requirements for this application will be most importantly, access to an internet connection, without it the application will not function as it is intended. Build in GPS functionality as well as privileges to use it will be necessary to send and receive messages. Lastly as it will be an Android based application, an android smartphone will be required with a minimum android version of 4.1, as certain google play services are required to be running on the device.

Functional requirements

The application will have a GUI (Graphical User Interface) that will be the central point of the whole system. This is where users will be able to the send and receive messages. This is the highest priority as there is a need for a way to interact with the whole system.

3.1.1 Use Case Diagram
3.1.2 Requirement 1: Sending a message

3.1.2.1 Description & Priority

This a vital requirement as the application is powered by the users and their ability to send messages to the database with their GPS, allowing other users to pick those messages up. The user writes the message, phone registers the GPS of their location, appends it to the messages and sends it to the server to be stored in a database.

3.1.2.2 Use Case

Scope

The scope of this use case is to allow the user to send their message to the server

Description

This use case describes the process a user takes to write and send their message

Flow Description

Precondition

The system is in initialisation mode once the user clicks on the option to create a new message and a valid HTTPS connection is set with the server.

Activation

This use case starts when an <Actor> starts to write their message.

Main flow

1. <Actor> writes in the textbox their message
2. <Actor> clicks the button to send (See A1)
3. The system gathers their GPS position
4. The system bundles the message and GPS position
5. The system sends the bundle to the server(See E1)
6. The system receives response from server

Alternate flow

A1 : Empty Field

1. The system creates a toast notification
2. The system uses the notification to inform the <Actor> that the field is empty
3. <Actor> fills the text field with a message
4. The use case continues from point 3

Exceptional flow

E1 : No internet connection

5. The system creates a toast notification
6. The system uses the notification to inform the <Actor> that an internet connection is necessary
7. <Actor> turns on internet connection
8. Use case continues from point 3

Termination

The system presents the <Actor> of the result, be it successful or unsuccessful with reason, of their message being send.

Post condition

The system goes into the main menu

3.1.3 Requirement 2: Creating Geofences

3.1.3.1 Description & Priority

The process of building geofences around the user. This is a top priority as it's at the core of the application

3.1.3.2 Use Case

Scope

The scope of this use case is to allow the system to build and listen the geofences for any transition changes

Description

This use case describes the applications process of building and initializing the geofences.

Flow Description

Precondition

The system is in initialisation mode once the application is turned on with a valid internet connection and the location service has been initialized and running in the background.

Activation

This use case starts when the System detects a change in the <Actors> current GPS location.

Main flow

1. The system registers the <Actors> current GPS location

2. The system sends a GET request with GPS to receive messages in the closest vicinity(See E1)
3. The system builds geofences with the messages
4. The system starts to listen in for <Actors> transition into the geofences

Exceptional flow

E1 : No messages within the specified Radius

1. The system interrupts the method
2. The system returns back to wait state

Termination

The system closes the GUI and goes back to listening for geo fences

Post condition

The system goes into a wait state

3.1.4 Requirement 2: Displaying Message in the geofence

3.1.4.1 Description & Priority

The process of receiving and displaying a message using geofences. This is a top priority as it's at the core of the application

3.1.4.2 Use Case

Scope

The scope of this use case is to allow the <Actor> to receive messages from other users in the geofences in the vicinity of the <Actor>

Description

This use case describes the applications process of receiving and displaying messages from the server.

Flow Description

Precondition

The location service is running in the background and the geofence listeners is scanning for any transition into the build geofences inside the system.

Activation

This use case starts when the System detects a transition into a geofence by the <Actor>.

Main flow

1. The system registers the ID of the geofence transition trigger
2. The system identifies if the transition is ENTER or EXIT
3. On ENTER transition the system builds a status bar notification (See A1)
4. The notification is send to the <Actor>
5. <Actor> opens the notification
6. Intent is started to display the message associated with that geofence ID

Alternate flow

A1 : Transition is EXIT

1. The system identifies the geofence transition
2. The system checks if the status bar notification is still displayed
3. If still displayed the system removes the notification
4. Returns to back to wait state and awaits activation

Termination

The system closes the GUI and goes back to listening for geo fences

Post condition

The system goes into a wait state

Non-Functional Requirements

3.1.5 Performance/Response time requirement

As the geofences have to be build by the client application I have decided on a bulk workload for the system. By gathering a certain amount of messages within a specified vicinity of the user, there will be multiple geofences ready to be activated by the users location change. This will decrease the frequency required to communicate with the server and download the messages left by other users.

3.1.6 Security requirement

As the GPS co ordinates may be considered sensitive information, that could in be used in a malicious way of tracking the user's location, it will require a secure mode of transport from and to the server. I have decided that HTTPS will be the minimal requirement to establish a secure connection with the server. Thankfully the library I use for connecting with my server, supports HTTPS protocol.

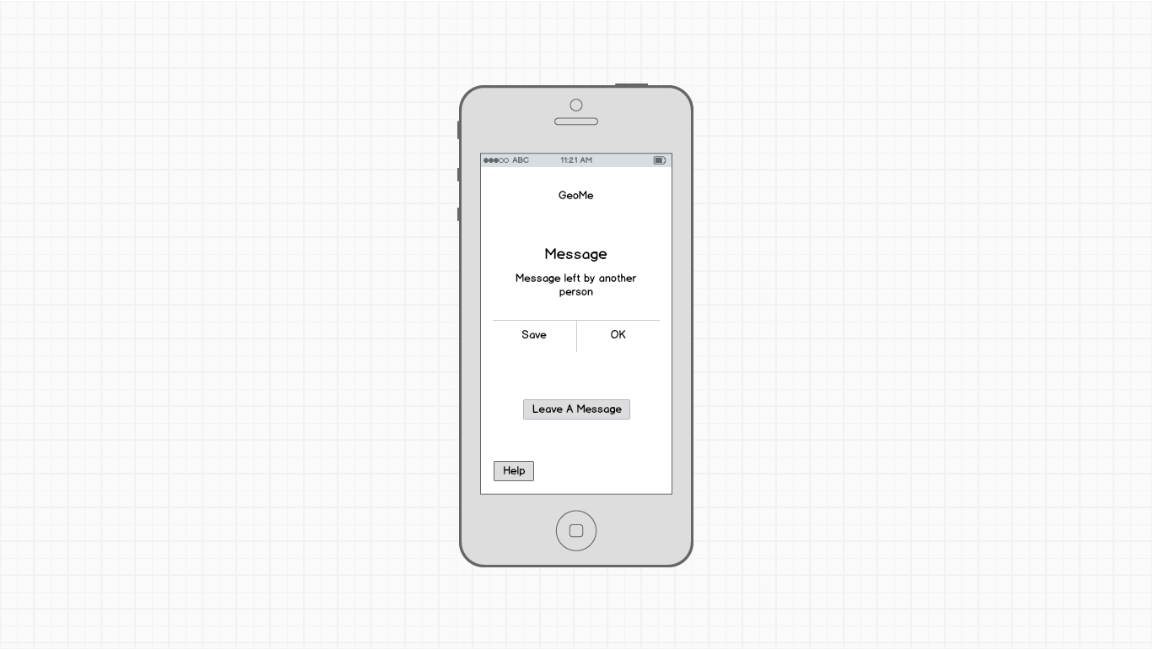
3.1.7 Resource utilization requirement

To maximize battery and network efficiency, the system will have to dynamically adjust the time period of requesting location changes on the user's device. This can be achieved by calculating the rate of change between the last and the most recent GPS location and changing the time frequency based on that. If a person is walking the frequency will be lower but if the users is in public transport or another faster mode of transportation, the frequency will increase.

4 Interface requirements

The application will only have an interface for the client side smartphone application. It will be navigated using touch on the smartphone itself. This client will have a custom REST API that will connect to a server that is connected to the the database, which will be MySQL. There is also a need to connect the Google Play Services API to the client to utilize its methods for GPS locations as well as Google Maps API for building the geofences.

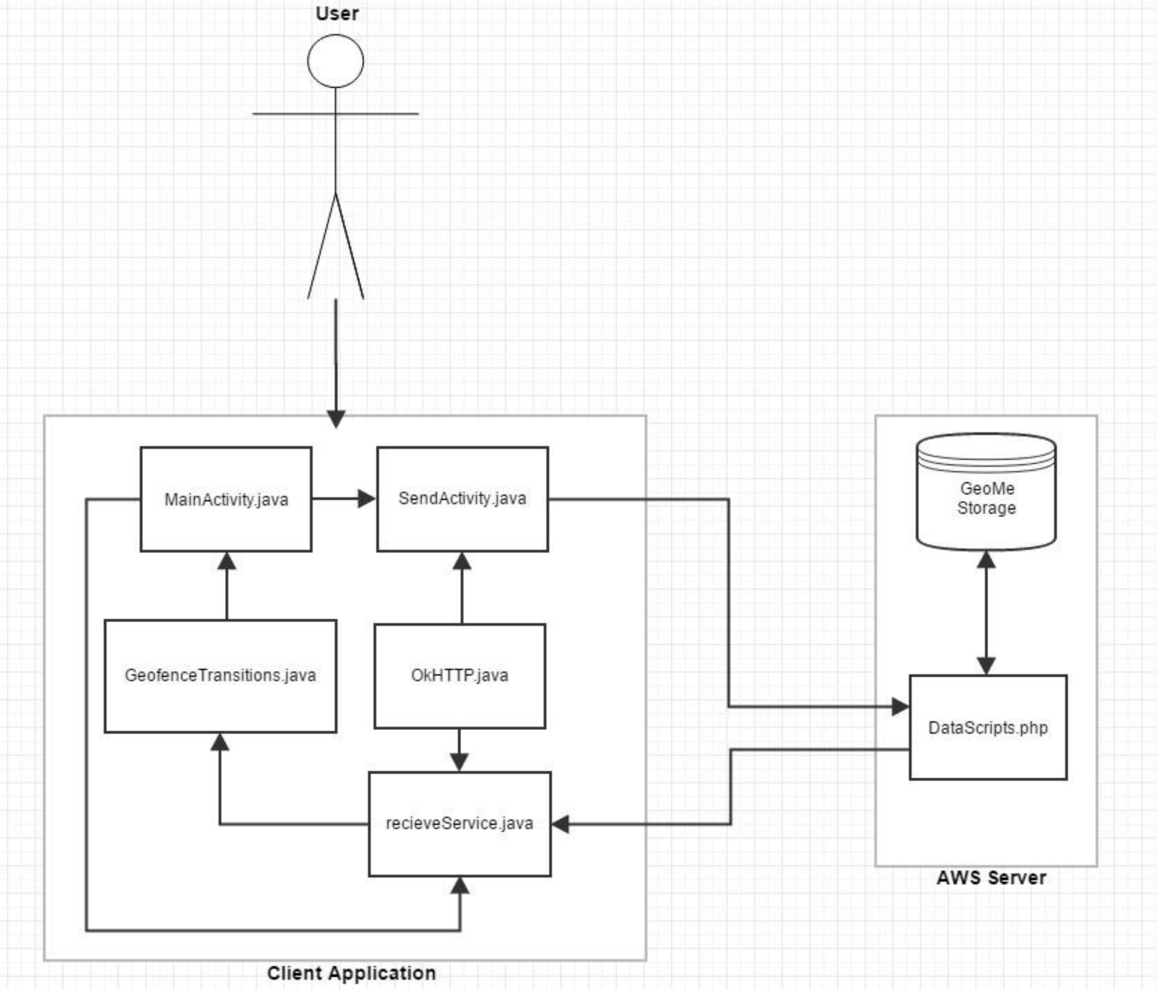
GUI



Application Programming Interfaces (API)

The system will use Google Play's Services for getting the user's GPS coordinates as well Google Maps API for it's geofencing functionality. By contacting the API, I will be able to make calls to their service and track the user's location using either the cell towers, wifi or GPS, but for precision I will have to use GPS. I will also write my own API for the RESTFUL service to send and receive user submitted messages to the server's database.

5 System Architecture



The system is split into two distinct systems, the android application and the web service. The Actor, being the user initializes the application and the MainActivity starts. The activity initializes and starts the recieveService responsible for contacting the server, using the OkHTTP class which includes API calls to the server. This allows for the client to receive messages from the database using the DataScripts script hosted on the server side. This service is responsible for creating and monitoring geofences, and any transition changes within a geofence is parsed by the GeoFenceTransitions class.

The GeoFenceTransitions class is responsible for identifying what transition has been triggered, be it either a user has entered or left a geofence. If entered, it will notify the user and display the message associated with that geofence. If left the geofence, it will remove the message and delete the notification invoked on the initial transition.

Functionality of leaving messages at a location of the user is handled by the sendingActivity. It will compose of a simple GUI to allow for the message to be written and upon sending it, the user's location will be captured, appended to the message and using the OkHTTP class it is send to the web service to be handled by the DataScripts file and stored in the database.

The Web Services system is responsible for the storage of messages as well as selecting messages to send to each individual user based on their location. The DataScripts is the end point for the data transmitted from the android application. It will validate the data, split it into individual parts and insert them into the database. It is also responsible for extracting and passing on messages from the database back to the android application to be displayed to the user.

The database, being a MySQL based, for the time being will consist of a single table that will store the messages and their associated GPS coordinates. As well as having a parameter table that will have values for the DataScript to use, in relation to the amount of messages to send back to the user and the radius for the GeoFences. This way, no parameters are hardcoded into the application and will allow for tweaking whenever any issues arise.

6 System Evolution

The system could evolve over time by adding more functionality and options to the message sharing. Such as; able to upload pictures and videos, links, captions check ins and maybe integrate a map that will show location of these messages for people to scout their area to pick up these messages.

The API and core functionality could be used for other applications as well, such as geocaching that could act as a treasure hunt.

A sign in option and switching from being anonymous to user based could be another option. This wa

6.2 Analysis and Design

GEOME

PRODUCT DESIGN SPECIFICATION

Version <*1.1*>
<*08/5/2016*>

VERSION HISTORY

Version #	**Implemented By**	**Revision Date**	**Approved By**	**Approval Date**	**Reason**
1.0	*Piotr Czerniejewski*	*01/10/16*	Piotr Cz	01/10/16	Initial Design Definition draft
1.1	*Piotr Czerniejewski*	04/04/16	Piotr Cz	08/05/16	Revision and clean up

UP Template Version: 12/31/07

TABLE OF CONTENTS

1. Introduction

1.1 Purpose of The Product Design Specification Document

The Product Design Specification document documents and tracks the necessary information required to effectively define architecture and system design in order to give the project developer the outline of system design and architecture during testing and developing.

2 General Overview and Design Guidelines/Approach

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

2.1 Assumptions / Constraints / Standards

The applications design should be simple to implement but there is a lot of small details that have to work perfectly as to not increase downtime between the content on the server and the user's application. The biggest constraint would be time for testing as with location based services, there are a lot of factors that depend on the smooth operation of this system. Such as accuracy, internet availability and power consumption.

Also, as GPS co ordinates can be classified as sensitive information, it will have to be stored and transmitted securely over the network as it travels between the two main components of the system; the application and the server/database.

Due to a lot of components being dependant on outside API’s to function, a certain level of maintenance, upkeep and research will be required to uphold the application fully functional and up to set out industry standards. These include the Google Map API as well as the Google Play Services API. And with the prospect of Google switching programming languages from Java to Swift due to legal battle between Google and Oracle, an complete overhaul may be required.

3 Architecture Design

This section outlines the system and hardware architecture design of the system that is being built.

3.1 Hardware Architecture

There is no elaborate hardware architecture needs. Due to working with the android system, everything is contained within the API during development and with the recent standardization, UI scalability of applications and performance similarity between multiple devices, the application should work on all the devices that can run the android version that this system is designed for.

The user's input will be taken in on the phones touch screen, and the messages will be displayed on the screen as well.

3.2 Software Architecture

The main user end application will consist of the android application that will allow to interact with the overall system, by receiving and sending messages with their GPS location attached. It will be dependant on internet access as well as allowing permission to access their current location. The Android application will consist of a main GUI which displays the geo messages. Another GUI for writing these messages by users as well as a background service that handles the geo fencing and communication with the server to store and retrieve messages from a database.

The location functionality will be handled by the Google Play Services API. It allows for tracking the user's location, request location updates and invoke specified functionality based on conditions. Such as location changed, GPS turned off or on.

Geofencing will be build using Google’s Map API. It provides basic functions to build geofences, define transitions and intents to start activities or services. This is a key factor for the software as the critical key components are dependant on it working 100%.

While the server consists of a PHP script responsible for data manipulation and insertion to the server as well as compiling the messages to b send to the user's app. And finally a server that will store all the messages created by the users in a MySQL database.

3.3 Security Architecture

The major security architecture design will be within the transmission of information between the user's application and the server. When the user sends their message with their appended GPS location, it has to travel securely, preferably encrypted using an algorithm such as the AES encryption. Tracking each user while they send their GPS co ordinates can be vulnerable to abuse by outside listeners, scammers, hackers and governments.

Thus the encryption will have to happen on both the users end as well as the back end. AES will be sufficient although if a HTTPS certificate can be created and added to the system, it will allow for secure data transmission, making the local encryption redundant.

3.4 Communication Architecture

The communication will happen over the internet using the standard HTTP protocol, although HTTPS will be much better, first will have to create a certificate that will allow the usage of it. After the initial testing of functionality, this will be top priority to implement. It will resolve the issue of safe transmission.

Also, there is no plan for communication between the users directly using the android application, removing the need for extra communication functionality. The messages left will be for the time being anonymous as a name or mobile number attached to it, for the needs won’t add anything valuable to the content of the message.

3.5 Performance

Performance is a big issues due to the application usage being over long periods of time. It has to be lightweight as to not slow down the network and the user's smartphone that could hinder any day to day usage of the device. This will require smart algorithms that monitor and influence the resource usage based on their movement, location and time. If the users is not on moving, there is no need to contact the server, send their location or initialize listeners as there is no need while stationary. This will increase battery life and decrease network usage.

The bulk of it will be done by building geofences based on the user's proximity to them. Rather than individually querying the server each time, asking for messages, a bulk will be downloaded. Then once the user enters the geofence, a notification will be send to the user, notifying them of an incoming geo message. Having a timed life span of messages, synced with querying the server for the next bulk of geofences, will make sure only up to date and relevant messages will be tracked by the device.

4 System Design

4.1 Use-Cases

The use cases can be found in the project requirements specification document.

4.2 Database Design

The database will consist of a single table that will handle the storage of user submitted messages. Each row of data in the database will consist of a unique ID to identify each message uniquely, column for the written actual message, which will be a String, with currently planned limit of 250 characters and finally two

columns for the latitude and longitude.
For early development and testing, one table will be enough. But future deployment and expansion, will require separate table for each region to speed the process of transmission to each user within their location.

4.3 Data Conversions

Due to time constraints the database is a SQL based, using MySQL for storage. This has to be converted into a format that the user client can extract messages and their associated GPS coordinates. This is done with PHP's built in function to encode an array bulld using the SQL commands into a JSON file.

This JSON file is then send to the client which is broken down by the built in java JSONObject class and fed into the method responsible for building geofences.

4.4 Application Program Interfaces

The system will incorporate three API's. Google Maps, Google Play Services and a server API. Each playing a vital role within the system. Google Play Services API responsible for requesting user's given location, and handling GPS coordinates as well as error handling such as network connection or location services being disabled by the user.

Google Maps API will be used for creating geofences as well as monitoring their status and any transitions invoked by the change in the user's location. This API provides builders and handlers for the functionality of geofencing.

Lastly an API will be required for handling the transmission of data between the client and server. It needs API calls to send user created messages as well as receive messages in their proximity to allow the Google Maps API to build geofences based on their attached GPS coordinate.

4.5 User Interface Design

There will be a need for three different components for the client UI. One for each of the main softwares functions. A main activity greeting the user, telling it that the application is running and will be also used to display any geo messages that the devices has received.

Secondly, another UI component will be required to allow the users the functionality to write and send geo messages. This will be composed of a simple text field that can be filled with the message.

And lastly status bar notifications. These will alert the user upon entering a geofence and by clicking it, will bring to the main activity and shall display the message to the user.

4.6 Server Design

Server which will be hosted on the AWS will consist of 2 key components. The API and the database.

The API will be responsible for sending and receiving information from client application. The geo messages written by users. These then will be stored inside a MySQL database. Database consisting of a single table with rows for the GPS coordinates and the message written at that location.

The whole server running on a LAMP stack for easy control and manipulation and scalability.

4.7 Performance

Based on initial testing, with a frequency rate of 5 minutes between contacting the server to update the geofences in the user's proximity, the client application is using 3% of battery per hour. This is a good result as it was expected for GPS on a device to use a lot of power but this is within the nominal range that shouldn't impact the user in a significant way. Based on an approximate running time of 6 hours, the battery usage is 18% (tested on an Xperia Z3C with a lifespan of 1 year). There could be further improvements to be made but for the time being it's sufficient.

Data usage on the other hand, is really negligible. The client only downloads bits of characters and the location updates are merely pings to the network provider returning double values. Further testing is required but the estimate is not in any way harmful to the user's data plans.

Appendix A: References

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
Project Requirement Specification	*Requirements for the project including functional and nonfunctional requirements*	*Requirements*

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

Term	Definition
GPS	*Global Positioning System*
API	*Application interface Program*
HTTPS	*Hyper Text Transfer Protocol with added SSL encryption*

6.3 Project Proposal

Project Proposal

GeoMe

Piotr Czerniejewski, x12488942, x12488942@student.ncirl.ie

BSc (Hons) in Computing

Networking and Mobile Technologies

28/09/15

1.Objectives

The primary objective of this project is to create a mobile application that allows users from around the world to communicate anonymously using their current location. By utilizing their current GPS location, they can leave messages for other users to pick up once in the proximity of the same GPS location of the message being left.

This in turn allows mobile phone users to leave messages attached to a specific location, sharing ideas, places to visit, warning of danger in the particular area, advertise their business whenever someone walks by their door or meet people with similar interests. In essence, leaving invisible floating messages on the street for other users to find.

The secondary objective is to create an online server that stores and manages these messages being shared. By having a database that stores and transmits these messages to the users, easing the load on each individual devices by needing to store all the messages in the area only displaying once needed, saving bandwidth and power useage of each individual users running the mobile application.

Third and final objective is to implement a user engagement system that will keep them using this application. By rewarding the users with most found messages that will be scattered in the area or the by creating the most messages in a particular city or country, give them a reward or feedback. By naming the 'King of City' or giving a quick glimpse of the local map to see where to find more messages, this should encourage the user to keep using the application for a longer time, increasing user retention.

2.Background

With the recent increase and improvements to the mobile networks, with increased data caps, 4G becoming a standard giving high speed internet access to anyone on the move, I wanted to utilize that increased inter connectivity between mobile users to engage them in a fun and interesting way. Information is key in today's times, messaging using only the internet, having maps and directions to any place in the city and streaming music online, while all of these make great use of the mobile data, there is nothing that interesting about it, it became a norm. I want to capitalize on that lack of a novelty app that makes the users excited again by getting a notification on their phone. A message you randomly get by walking on the street, never knowing what you get.

3.Technical Approach

My main target audience of this app is young, adventurous and social people who know their city, what's going on where and when and constantly connected.

So for research I will ask social clubs, college societies and individuals if not only would they like to use it but also what features would they like to see.

It will also require:

-Smartphone, preferably Android as this will be primarily an Android app

-Internet connection to send GPS location, messages and to receive them

-Authorization to request the user's location

I will also have to research on how to get the person's location, how many requests per minute, to either use GPS or Cell Towers or WiFi or all and how will it impact the battery usage.

4.Special resources required

-Android smartphone

-Server stack for database storage

5.Technical Details

As this will end up being an Android app, I will most likely develop it using Java as I've been learning it for almost 4 years now. As for the database I will be using MySQL as I am most comfortable with it, and for the back I end I will use PHP5 to link the database to the android application.

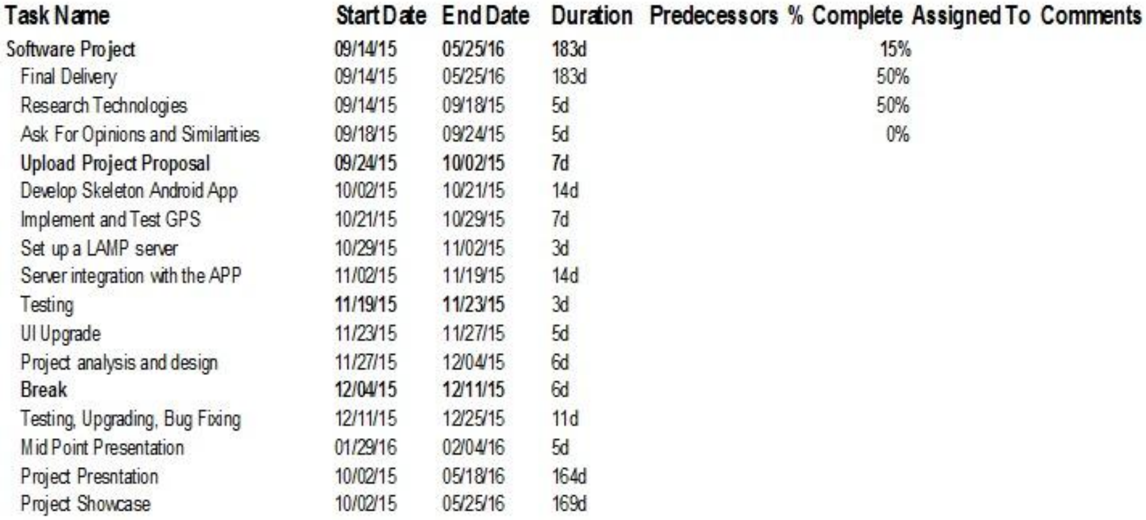
6.Evaluation

I can evaluate my back end and databases as I go by inputting test data to be send to the application to make sure it is displayed correctly. For the android app side, I can run each build anytime I want by connecting my smartphone to the IDE and running the build on it, rather than emulating it in, from my experience a slow android emulator

Piotr Czerniejewski 2/10/15

Signature of student and date

6.4 Project Plan



Task Name	Start Date	End Date	Duration	Predecessors	% Complete	Assigned To	Comments
Software Project	09/14/15	05/25/16	183d		15%		
Final Delivery	09/14/15	05/25/16	183d		50%		
Research Technologies	09/14/15	09/18/15	5d		50%		
Ask For Opinions and Similarities	09/18/15	09/24/15	5d		0%		
Upload Project Proposal	**09/24/15**	**10/02/15**	**7d**				
Develop Skeleton Android App	10/02/15	10/21/15	14d				
Implement and Test GPS	10/21/15	10/29/15	7d				
Set up a LAMP server	10/29/15	11/02/15	3d				
Server integration with the APP	11/02/15	11/19/15	14d				
Testing	**11/19/15**	**11/23/15**	3d				
UI Upgrade	11/23/15	11/27/15	5d				
Project analysis and design	11/27/15	12/04/15	6d				
Break	**12/04/15**	**12/11/15**	6d				
Testing, Upgrading, Bug Fixing	12/11/15	12/25/15	11d				
Mid Point Presentation	01/29/16	02/04/16	5d				
Project Presntation	10/02/15	05/18/16	164d				
Project Showcase	10/02/15	05/25/16	169d				

6.5 Monthly Journals

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHC Computing

Month: September

My Achievements

This month I was able to finish my project proposal and uploaded to be assigned a project supervisor. I'm pretty happy with it and I still think it's a good idea, although I'm still thinking if I should with the social aspect or marketing one. As I am a person that likes to get straight into work, I've already started to code my prototype. I've setup an AWS instance for my LAMP server stack, and a skeleton android app with already integrated google play services to receive gps location updates. Now with everything in place, I just have to expand, document, test and I should be done with the main bulk of work before December.

My Reflection

I'm really happy with how the google play service works, but the location updates are a bit slow. I have to fix permission access for the app as well as optimize the times that it connects to the service for the location update. I'm also happy with how smoothly the AWS set up went, already build a testing MySQL database that can store the messages and location. But I will have to think about encrypting the information as it is sensitive information, people's location that is.

Intended Changes

Seeing as I already have a good start in coding and everything I will focus more on the design and planning aspect. The database, networking and system architecture. That will give me a good idea of how to code my project.

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHCNMT4

Month: October

This month I finally finished my skeleton app for the client side of the application. It has a working GPS functionality, I'm able to send a message with the coordinates appended to it. The server reads it and displays it correctly. All thats left is to connect the database to the back end and store the messages. I also met up with my supervisor, we talked about my idea, what I should do such as a focus group, how to focus on my other studies and what I could do to polish my idea.

It was a very productive month for me. But thankfully I had a rest during the reading week and caught up on other assignments.

My Reflection

I am happy with my current progress . I am well ahead than my peers and don't feel too much pressure. I can take my time for documentation as I am confident in my abilities to code the actual application

Supervisor Meetings

Date of Meeting: 22nd October

Items discussed: Project Proposal, Requirement Spec, Use cases

Action Items: N/A

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHCNMT4

Month: November

This was a very stressful month with many assignments needed for me to complete on time such as CA's for my NPDS module and work on presentation for Business network security module. I've done my best in them and now I still have to plan my other projects, especially the Change Management paper.

For my project I've done a lot more of theory research and planning rather than practical work. I know how to structure my database, now that is left is to come up with a solution to send user generated messages that have been uploaded to the database back to the users phones who cross the GPS coordinate of the message and the implementation of geo fences.

My Reflection

I wish I could do more work on my project but due to wanting a high overall grade for the course, I couldn't neglect my other CA's. But over all I'm happy with my college work.

Supervisor Meetings

Date of Meeting: N/A

Items discussed: N/A

Action Items: N/A

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHCNMT4

Month: December

This month was the final touches and finishing my Business Security written CA for our topic of social network security as well as our main NPDS project where we created a CDN service. Again, due to these projects taking up most of my time, could not do major work on my project. But in my spare time I've read more of Android docs and specifications for geo fencing and networking between device. This will give me a good start once I can start to code my implementation of these solutions to my project.

I also done a lot of study for my exams that are coming up in the next upcoming weeks.

My Reflection

I am very happy with both CA for the other modules this semester, hopefully it will give me a big mark and help in achieving a high grade. Also feel prepared for the exams to come.

Supervisor Meetings

Date of Meeting: N/A

Items discussed: N/A

Action Items: N/A

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHCNMT4

Month: January

The month of exams was relatively easy due to studying hard before hand. Felt good walking in and walking out from them. Hopefully I will get good grades to give me at least a solid 2:1 start for my degree. Other than that I was able to catch up on my documentation for the project, such as doing more analysis and design from the technical point of view, as well as slowly preparing for my midterm presentation. I also met up with Simon my supervisor just to update him on my progress and let him know what I will be showing during the presentation as well as asking for some tips.

My Reflection

It didn’t hit me yet that’s it's my final semester of college and soon enough I will be finished and getting my degree. I hope I can do a good presentation as I could really use the marks to go towards the end grade.

Supervisor Meetings

Date of Meeting: 2/2/16

Items discussed: Midterm presentation

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHCNMT4

Month: February

This month I had my mid term presentation. I gave it my best but unfortunately I didn’t do as good as I hoped. I wasn’t sure if the presentation was backed up by my documentation or vice versa. But what’s done is done, have to move forward and continue with my project.

My Reflection

I know that I have to improve my presentation skills moving forward for my final presentation in May. But i know that the technical bits I have nailed it.

Supervisor Meetings

Date of Meeting: N/A

Items discussed:N/A

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHCNMT4

Month: March

After troubles reaching out for my supervisor and trying to get my mid term presentation mark I finally did. 60%. A 2:1 grade which I’m happy with but disappointed at the same time as I wanted a lot more.

My Reflection

After talking with my supervisor I decided to treat this presentation as a mock. Now I know what to expect for my final I can now focus on improving my skills. I have also done a major accomplishment by successfully implementing geofences for my application, taking me one step closer to the finish goal.

Supervisor Meetings

Date of Meeting: 16/03

Items discussed:Midterm and Final Presentation

Reflective Journal

Student name: Piotr Czerniejewski

Programme (e.g., BSc in Computing): BSHCNMT4

Month: April

Final stretch, done a lot of project work, finished any assignments I had left for other modules and now can focus on studying for my exams. Getting high grades in my assignments I am not scared of the exams. The results will be icing on the cake for me.

My Reflection

Now that everything is wrapped up I can now focus on polishing my documentation, finishing any functionality and start working on my poster.