To: Mr Teo Tse Tsong

From: Pow Rei Young

Date: 13/9/2016

Title: S.I.P. Report

**1. Introduction**

**1.1 Background**

The purpose of the attachment is to help the elderly with mental illnesses, like Alzheimer's disease or dementia. They may forget where they are when they are outside or they may forget what they were even doing in the first place, this leads to the elderly being lost or wandering off. So the project is meant to create an application that allows friends or loved ones to locate the elderly anywhere, letting their loved ones to find and help the elderly when they start getting lost or go missing for a period of time.

**1.2 Reporting Period**

The reporting period for the SIP is from 25th April 2016 to 16th September 2016.

**1.3 Objectives**

For the project, an application needs to be created that is able to track another phones location with Wi-Fi information, using methods like SMS or server. Wi-Fi information is needed in case the GPS is not on or the phone can’t detect the location, allowing the user to get a rough idea on the elderly location. Learning how to use Android Studio and Java is essential to create the application. Learning about SQL servers and how to code PHP files is also needed if the application needs to use servers, like OpenShift servers.

**2. Description of the DU Environment**

I needed to create a working Android application that can track another phone’s location and Wi-Fi information. It needs to have a reliable method of sending and receiving information and for the application to display the info on a map. Creating two applications was decided, one using SMS to relay information and the other using server to send and collect data. I assign a task to complete for me every day, ensuring that progress is smooth and being made. If I encounter an error or I get stuck, I will try to solve it by debugging. If I can’t solve it, I will go back to solve it at a later date.

I had to research on the types of GPS applications in the App Store to get a rough idea on how a GPS application works. I also had to learn how to use GitHub and SourceTree, allowing me to save my codes on the internet and being able to get any previous working codes if I encounter a major error. I had to learn about Android Studio and how to create a server, as I had no partners and had to write all of the codes myself. The server application needs to allow users to create their accounts and store their info in the server. The accounts also need to have their own friend list, letting them see the location and Wi-Fi information for people they have added.

Mr Teo Tse Tsong, my supervisor, was clear in what he wanted from me and he explained the objectives of the project clearly, allowing me to understand what to do for the project. Once my supervisor has made it clear on what he wants, he lets me to decide on how I want to code and create the application. He comes to check on my progress on the application every few weeks, trusting that I will do my work and complete it on time. Samuel, a senior, taught me the basics of Android Studio and how to create a server, allowing me to understand and create the codes needed for the application.

**3.** **Work Completed in the Reporting Period**

**3.1 Research**

For the first few weeks, I researched on some GPS applications in the app store and how to use GitHub with SourceTree. I tried to find out how the applications update the GPS information and how often. I also wanted to see different application ways for how each user gets another user’s information. I tested the applications by tracking multiple people with the application downloaded on their phone. I researched GitHub and SourceTree so I can upload my codes on the internet easily and my supervisor can view the latest codes at any time without asking for me to send him. It also allows me to back up my codes in case my files get deleted or my computer crashes. I can also return my codes to a previous state if I feel like I coded a wrong method that is not working or my codes have too many errors that can’t be solved.

**3.2 Android Studio and Google Maps**

I had to learn how to create a map in android studio for my GPS applications. So I watched some google map tutorials online. Samuel, my senior, also taught me how to interface android studio with my phone, letting me test the application and debug easily. Once I know the basics of creating a map application, I created an application that displays a map. The application shows the user’s location and is able to search an address and I added buttons to zoom in and out for the map. (See Figure 1a)

**3.3** **Collecting GPS and Wi-Fi information**

Then I had to make the app scan and store the location and Wi-Fi information. For the location info, I used a method to collect the coordinates and store it inside variables in the application. If the phone moves location, the coordinates inside the application will change, updating the location variables to the latest info. For the Wi-Fi info, I used a method to scan all the modems around the phone and storing the information in as string variables in the phone. (See Figure 1c) The application will only store the top three strongest modems near the phone. This is the foundation and the basics for both my applications, as I need to collect both these info in both my applications.

**3.4 SMS Application**

I decided to create the SMS application first. The SMS application needs to use the contact list to send the SMS and it must be able to read and send SMS. So I created a layout for contact list and I used list view to generate and show the user’s contact list. (See Figure 1b) I allowed the user to select from the contact list and an SMS will be sent to the selected contact. I then used a method, SMSReceiver, allowing the application to detect any SMS that is sent to the user. I made the contact list to only send a request code SMS to the selected contact and I made the SMSReceiver to detect the request code from the SMS. The application will respond when it detects the SMS by sending back the Wi-Fi and location information with another info code in a SMS back to the sender of the request code. The application that requests will detect the info code and take the info from the SMS, splitting the information and displaying them on the app. Then I moved on to the server application.

**3.5 Server**

Now I had to create the server before I code the server application in Android Studio. I needed to learn how to make an OpenShift server and once I created one, I had to code PHP files to link the server with my application. I followed the OpenShift website instructions to create a server, so now I have to link the server with the server application. (See Figure 3) I used GitBash to upload the server files to the server, letting the application call the PHP files at any time with internet. I created PHP files for allowing users to store their accounts information inside. I also made XML files allowing the application to read and detect information from the server.

**3.6 Server Application**

I created a new application and inserted the map, location and Wi-Fi codes inside, allowing the map to be views and the information to be stored inside the variables. (See Figure 2a) I had to create a register, login, friend list and add friend layout for the server application. (See Figure 2b, 2d) Then I linked the application with the PHP files by adding in codes that connect and call the uploaded PHP URL on the internet. When the user is registering, the PHP add account file gets called, adding a new account in the database. I also made a unique code to be assigned to every account that registers, the code is meant for adding friends into their friend list. I also made the login layout to read and check if the information is correct by connecting to the XML files on the internet. Once the user logins, I made the application update the server with their location and Wi-Fi information for their account continuously, calling the uploaded PHP files constantly when the info changes. For the add friend layout, I made the user’s own account code be displayed, so the user can tell their friend the code, allowing the user to add them into their friend list. The user can input the code into the text box in the add friend layout and when the button is pressed, the info will be uploaded and saved inside the friend list server. (See Figure 2c) I also created a list view inside the friend list layout and I connected the list view to the server, letting the user see their own friends list. Then I made a delete button, allowing the user to delete any friend that they want, clearing the info from their friend list in the server. (See Figure 2e) I made the application collect the information from the server when they picked a friend from the friend list, displaying the friend’s location and Wi-Fi information on the map when selected. I also made the update continuous, so if the information in the server gets updated, the application will also update with the new information. (See Figure 2f)

**4. Analysis of the Impact of the DU Attachment\***

I did encounter a few problems, like how I could not return to the main map layer after leaving the map layer. If I returned to the map layer, the app will crash. I found out that whenever I return to the map layer, a new map fragment gets generated. So I deleted the map fragment every time I left the layer, creating a new map layer every time I go back to the map layer. I also had trouble getting the SMSReceiver method to work, as it can’t detect any messages. To solve this I added permission into the android manifest, allowing the SMSReceiver to work. I was struggling with coding at first as I did not know much about coding android applications and PHP. It was really hard, as I did not understand how the code works. But after a few weeks of coding, I managed to understand more of it and it got a lot easier to code. Luckily, my senior taught me the basics of Java and PHP, and I start to understand a lot after he taught me. Now I am able to code PHP and Java a lot better compared to when I was coding at the start of the SIP project. The online tutorials also taught me a lot, like how to build google maps in android and how to generate the contact list on the list view.

**5. Conclusion**

Overall, I feel that the work I did met my supervisor’s expectations and I achieved my objectives for the project. My objective was creating an application that can track the elderly anywhere with location and Wi-Fi, and I met the objectives with 2 different applications, both being able to track the elderly. The learning experience was good, as it taught me how to learn and research myself.

**Appendix**

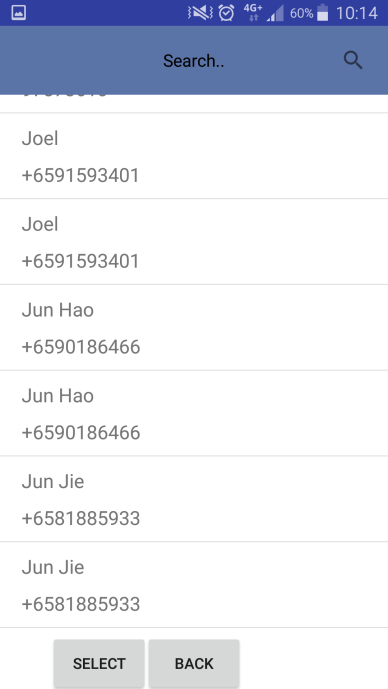
  

Figure 1c

App Wi-Fi Page

Figure 1b

SMS App Contact Page

Figure 1a

SMS App Main Page

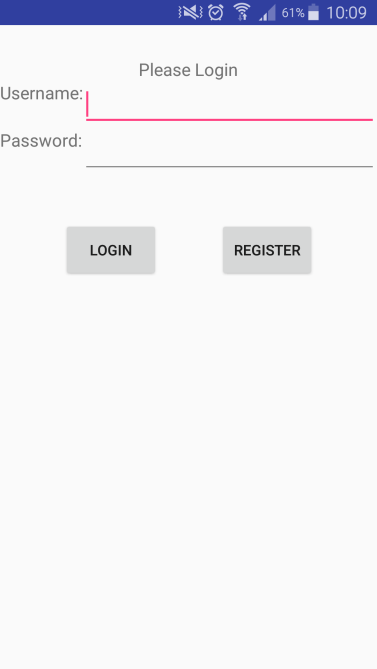
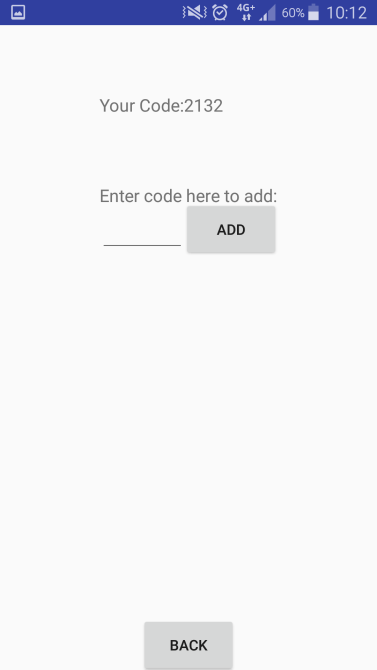
  

Figure 2c

Server App Add Friend Page

Figure 2b

Server App Login Page

Figure 2a

Server App Map Page

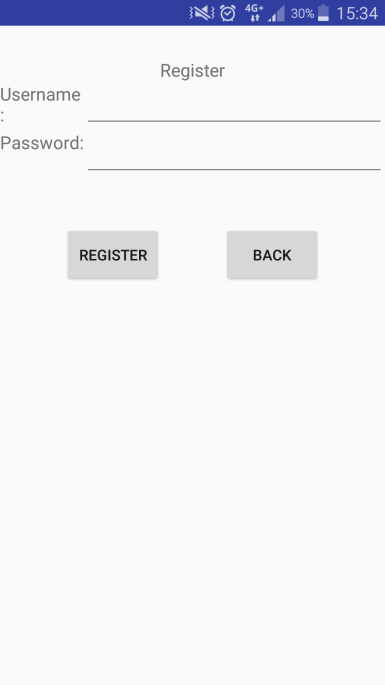
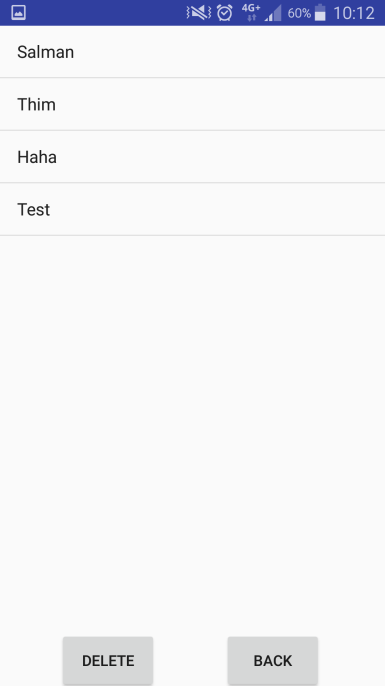
  

Figure 2f

Server App Location Page

Figure 2e

Server App Friend List Page

Figure 2d

Server App Register Page

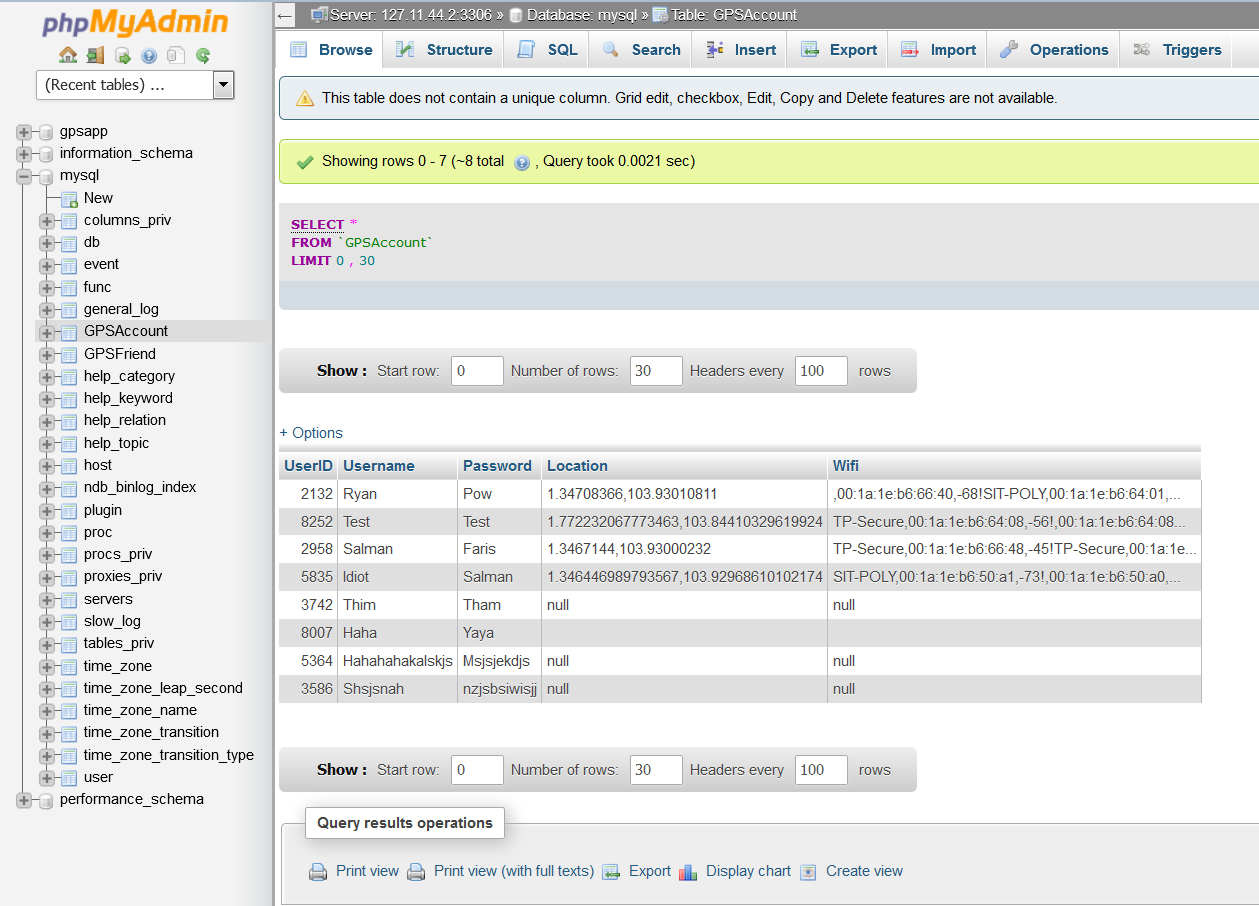


Figure 3

OpenShift Server Page