WIFI LOCATOR FOR THE SOFTWARE ENGINEERING CLASS AT UNIVERSITY OF ENGINEERING AND TECHNOLOGY

By

Tran Duc Muoi

A Project Report

Submitted to

Professor Randy Ribler

Mr. Truong Anh Hoang

As a final report of group project

In Software Engineering class

Hanoi, Vietnam

May 2013

WIFI LOCATOR FOR THE SOFTWARE ENGINEERING CLASS AT UNIVERSITY OF ENGINEERING AND TECHNOLOGY

By

Tran Duc Muoi

Approved:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tran Duc Muoi

Student ID: 11020204

(Group Leader)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dinh Trung Anh

Student ID: 11020004

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nguyen Truong Thinh

Student ID: 11020313

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Bui Van Vuong

Student ID: 11020408

ACKNOWLEDGEMENTS

I would like to thank Professor Randy Ribler and Mr. Truong Anh Hoang, the instructors of the Software Engineering class that I was enrolled in, for the comments and suggestions that helped me decide to pursue this project. I would also like to thank the others group who also follow Wifi Locator Project for their helpful comments. Finally, I would like to thank my Software Engineering team members for helping inspire this project.

TABLE OF CONTENTS

CHAPTER

I. INTRODUCTION …………………………………………………………………………… 5

II. SYSTEM DESIGN AND IMPLEMENTATION

2.1 Functional Requirements ………………………………………………………. 6

2.2 User Interface Design …………………………………………………………… 6

2.3 Implementation Process ……………………………………………………...… 7

III. SYSTEM ENVIRONMENT …………………………………………………………… 8

IV. SYSTEM TESTING ……………………………………………………………...……… 9

V. PROJECT CONCLUSIONS …………………………………...…………….……… 10

VI. PROJECT TEAMWORKING

6.1 Working Process ……………………………………………………………………… 11

6.2 Working Evaluation ………………………………………………………...……… 13

CHAPTER 1

INTRODUCTION

Wifi Locator is a project following user’s wishes (user story). There are several elements of this software. First of all, Wifi Locator uses wifi system, which contains a number of different wireless routers of a building. Wifi Locator has released 5 versions during the course, provided step by step from simple feature to more complicated feature. After the first 2 weeks, the users can be able to know which wifi access point is available and some other information of that wifi when using the software. For example, when “Scan” button is clicked, SSID, BSSID, level, frequency,… of the available wifi access point are shown on the screen. The next release allowed users to “Scan automatically”, if he/she witnesses then it means he or she is moving and the his/her location has changed. Just tick on “Scan Automatically” button, walk and see the information change automatically on the screen. Next 2 weeks, Wifi Locator version 3.0 (release 3) has gone a big step with the most important feature. When the users come into a new place, they can assign a new label to the current place where they stand. The label information will be sent to database on a server instead of saving in the device and this feature was demo successfully in this release. The next release bases on the success of previous releases and provides features that show users their location if they are in a labeled place. Finally, the last release fix some bugs, exceptions in the previous releases, can be able to show the user’s location continuously as the users is moving, change the background into more friendly interface. The final version of Wifi Location was accepted and recognized by the user, Prof. Randy Ribler.

CHAPTER II

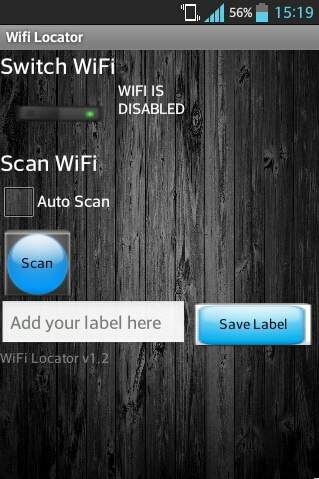
SYSTEM DESIGN AND IMPLEMENTATION

**2.1 Functional Requirements**

In Wifi Locator project, the following functional requirements were identified and then implemented:

1. The users can open software on their Android device
2. The users can change wifi status of the Android device (On/Off) by using software
3. The users can scan Wifi access points manually by Scan button.
4. The users can scan wifi access points automatically by ticking Scan automatically.
5. The users can save a label for a new place.
6. The users can be shown the saved label for a known place.
7. The users can exit the software.

**2.2 User Interface Design**



The system will be Android application and will use a Android device to view the user interface. Therefore, the interface has the simplest designs, just enough for functional requirements. It has identifying information as a header, a switch button, a text box, a chosen button and 2 submit button: Scan and Save Label. When the users click Scan button then he/she will see a list of wifi. This design is compatible with all Android device resolution.

**2.3 Implementation Process**

<https://github.com/anhtrung93/Wifi-Locator/issues/milestones?state=closed>

The development of Wifi Locator software is shown as the link above and as below:

*Release 1 (14 Mar)*: The following functions have done:

1. The users can open software on their Android device
2. The users can change wifi status of the Android device (On/Off) by using software
3. The users can scan Wifi access points manually by Scan button.

The software activity was tested and passed all tests.

*Release 2 (28 Mar):* The following function have done:

1. The users can scan wifi access points automatically by ticking Scan automatically.

*Release 3 (11 Apr):* The following functions have done:

1. The users can save a label for a new place.

Also in this release, the code was refactor. A new activity which allow user to assign a label was written. Success on testing linking from the device to server’s database.

*Release 4 (25 Apr):* The following functions have done:

1. The users can be shown the saved label for a known place and the locator change continuously if the users tick on “Scan automatically”.

The change log of this software version: Fix linking from the device to the server. User interface was updated. Write an algorithm to detect user’s location.

*Release 5 (9 May):* Release the final version. Fix some minor bad codes.

CHAPTER III

SYSTEM ENVIROMENT

The Wifi Locator was implemented using Java Android as the language and Eclipse as the IDE. Java is a freely available language that can be installed on many platforms. This language was chosen because it is the most suitable for project Wifi Locator, which will be definitely depend on wifi system - available in most of android device. Become an android application is the best way WifiLocator can be use in real life. Why Eclipse? Powerful, friendly, fast and reliable. If the programmer does not have any android device, then it is a great challenge to finish the project. Eclipse supports AVD – Android virtual device will be helpful is this situation. The android was used to implement the code is in Android API 16, aka Android 4.1.

Finally, Wifi Locator is available as independence android application, can be installed and run normally in any android device which support Android 4.1 at least.

CHAPTER IV

SYSTEM TESTING

Wifi Locator has been tested from the first release till the final version released. This software was tested in two ways: by Junit test and by Android application test. This functional test verified the correct functionality was achieved.

To test each function of each class, we decided to use Junit, which is the simplest system testing. To test the activity of the software, of course we uses Android application test. The test will be run on the device or in AVD, then give the result back. Just implement android application test in case if necessary because of its slowness.

If errors were found, they were corrected immediately as to not disrupt the operation of other functionality. Also, during the functional tests, regression testing was done. This ensured that all functionality that had previously been implemented still produced correct output. After all the desired functionality was implemented, system testing was performed. This involved the developer carrying out example scenarios of operation.

CHAPTER V

PROJECT CONCLUSIONS

The Wifi Locator provides any user with an Android application for determining the current location by using the system of available wifi access points. The application allows users to save the history of where they gone, which room did they get in. By click and tick on simple buttons, the users can be able to know everything they want about a wifi access point, is it strong or weak (means is it far or near the current place). They can also see the information change on the screen when they’re moving slowly. To conclude, Wifi Locator meets all the goals of this project, the developer team has done work well.

CHAPTER VI

PROJECT TEAMWORKING

*This is a piece of report that written by the team leader about teamwork during project process. I will use “We” instead of “Our team”.*

**6.1 Working Process**

After regroup to be a team with 4 students, we decided to pursue a project that is the most difficult one in our opinion, Wifi Locator. At the beginning of the project, there was no idea about what would be done, what should we do about the project. For the newbie, choosing Java Android to be the system implementation was a big step, because Java a friendly language that we already use a lots. We also try Github as a tool to store and share code over Internet, manage documents, tasks. A facebook group is also created for exchange information and discussion about the project. The first time using Github, it is not easy as we think, some troubles, some losing code appear. But after 2 weeks, Github system is under-controlling. We start with writing “user story” aka the customer’s wishes, divide it into some points and the Wifi Locator project officially start!

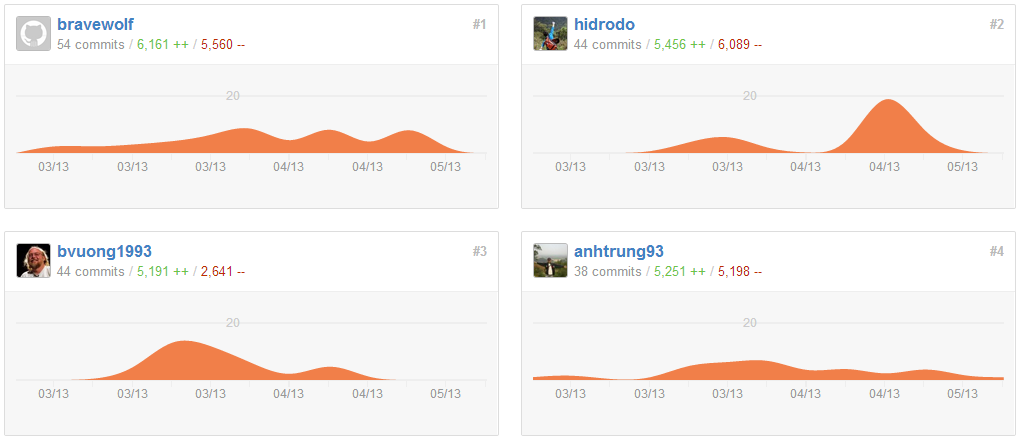
For the method, we use agile development method in this project. We release a version with some new features every two weeks. Our goal is finish a “user story” per version. This is the reason why we have to divide project into some points depends on the size of work and difficulty in implementation. Following up the user story is the tasks. Task was given for each member in the group, and the user story could be done if and only if all tasks have done. The team leader has to update task, create new issue and assign for the others in Github frequently. The working is planned carefully and suitable, so the processing is pretty good.

It does not have much thing to say about the first project except our happiness for the first software was run successfully on the Android device. The first release, the WifiLocator ver. 1.1 does not have many features, but provide some basic function for the user. Our customer, Prof. Ribler is satisfied about the first “user story” was done, he add some comments and hope our application will be better next time. The next releases, we follow the planned process with the XP agile development method, so everything comes easier. We finish the code on time, release and demo the new features every version to the customer. After the final version has released, we cannot figure out that we have created an application available on Android device in a simple way like that.

One of the most important things in our project is testing system. Test was written before code, following the planned project design. If a member in team is providing a class of code, then the tester has to be next to him, discuss about the mistake may come. If the code has passed all tests, nothing to say, but if not, the tester has to make sure that the programmer understands the test, and both of them will correct the mistakes together. This method is called pair programming. Although we do not have much time to meet each other, we still use the test and pair programming well by using Github and utilizing class time as much as possible.

Another session is refactoring. Frankly, we did not spend much time on this because since the project started, all member in the group has consistence only one coding standards was chosen to implement code. We just add some comments, fix some meaningless variables name, and then we receive great feedback from code reviewers from other group.

**6.2 Working Evaluation**

Let talk about the good things first. There are 4 students in our team. We have great teamwork together by pair programming. The work has divided equally to 4 members and the project has done by all the members instead of someone. Another thing I suppose that we have done very well that is the project planning. With a good plan and following agile way, we do not meet any big problem when process project.

And the bad thing is because each member receives the divided work: Thinh: the user interfaces, the activity; Trung Anh: Algorithm; Vuong: Server and Linking; Muoi: Testing System, then sometimes we are misunderstanding about the other’s works. Luckily, we use pair programming effectively, so at the end, the application which is compounded from 4 pieces of codes can run normally.