CS 560 – Software Engineering

BU-tools

1. Contribution of each member of your group

Full name	Contribution
Yi Ren (002269013)	1. Requirement engineering
	2. Android app implementation
	3. Testing and maintenance
Wentao Lu (002276355)	1. Requirement engineering
	2. Web server and database implementation
	3. Testing and maintenance

2. Requirement engineering

- Give a description of your project;

  Our project 'BU-tools' is a mobile application for the BU website, however, it is more than a website. The main functions are as below:

  (1) Map and Location: Select a certain building in the list and the map will find the way to it, you'll never get lost in our campus!

  (2) Check the schedule of the gym: Get the schedule of our gym activities, such as basketball training, open swim and SRC soccer.

  (3) Register & Login.

  (4) Other useful tools, such as moodle, mybu and our webmail.

  Refer to Github for more details.

  https://github.com/Grindewald1900/BU-tools

- Explain the different stages to realize your project.

  Considering that we only have a small team with limited time to complete all the projects, we decided to build our system with rules of Extreme Programming, such as small release, simple design, and pair programming. However, the project is mainly implemented according to the waterfall model, because we don't have enough time for project management, and waterfall model is a proper choice to reduce the overhead of our project.

  Time consumption for each stage:

Stage	Time-consumption
Requirement engineering	about 25 man hours
Design	about 15 man hours
Implementation	about 60 man hours
Testing	about 30 man hours
Maintenance	N/A

(1). Requirement engineering :

The prospective users for our system are students of BU, especially the new students. In this way, we not only play a role as the developer, but also the user.

We came up with the idea to develop a tool for BU students since we found it hard to find our way on campus. As new students, we spent so much on way-finding. So we decided to build a location system for the mobile phone.

Also, considering that our users are students, we hope to integrate some useful tools related to our campus, that's why we add Moodle, Mybu, and Webmail as new features.

(2). Design :

In the designing phase, we decomposed the whole system into two parts, that is Client and Server (C/S Architecture).

For the Client part, the Android application is our first choice. Our app should include some components as follows:

Welcome page, which shows up at first installation, shows our users the main function of the app.

Home page, which embeds a browser kit, to show the main page of our university website.

Location select page, which contains a few list sheets, to show every building on our campus, so that users could choose one of them for campus direction.

Map page, which integrates Google Map API and Direction API, shows the location and route on the map.

Gym schedule page, which gets data from our gym web page, shows the schedule for all the facilities.

Tools page, which includes a set of useful tools, when clicked, will go to the web view for display.

Http request module, which connects with Server with HTTP GET and POST method.

Handler, which deals with the result from Server.

For the Server part, a web server and database are compulsory. The web server should include the components as below:

Register Servlet, which deals with the registration post request, add new tuple to database .

Login Servlet, which deals with the Login post request, check the validity of user information received.

Location Servlet, which will send the location data back to the Client.

Validation model, which will check the registration information from the Client.

User information table, which keeps data for user information, such as name, password, email, validation, portrait image URL, personal introduction and so on.

Location table, which keeps the latitude and longitude for every building on our campus.

Direction table, which keeps the information of the direction record for users. These records could be used for further behavior analysis.

(3). Implementation

In this phase, we worked together as pair programming. One of us works on Android App development, the other will do the server and database stuff.

We develop our project based on the test result, so we did several unit tests and released more than 10 versions in one week. For each version, we only focus on a certain function.

(4). Testing

In this phase, unit testing and integration testing are the main methods we used.

We completed some error-based testing and black-box testing at the implementation phase. After the completion of both Client and Server, we did some integration testing to ensure the feasibility of all the features.

(5). Maintenance

With the limited time, we didn't deploy our server on the remote server. In the next few months, we'll deploy the server on AWS and add some new features.

3. Modeling with UML

Click on the hyperlinks below for full-size diagrams. If there's any problem to open these diagrams, please contact us.

a. Use cases

b. Sequence diagrams

c. Class diagram

d. Transitions diagrams or state diagrams

State diagram-1: user login state diagram

State diagram-2: map module state diagram

Source project of UML:

Use cases:

https://www.processon.com/view/link/5e87f468e4b064d902cf0c0a

Sequence Diagram:

https://www.processon.com/view/link/5e87f940e4b03231c714ced2

Class Diagram:

https://www.processon.com/view/link/5e87fb05e4b03231c714d1dd

State diagram:

https://www.processon.com/view/link/5e8a432ae4b09396a4a0ca55

4. Implementation

Developing tools:

(1) JDK: Java development kit, which is released for Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.

(2) Android studio: an Android IDE released by JetBrains, it supports both Java and Kotlin as developing language.

(3) Google map platform: a platform that provides a set of API about the map, such as Maps API, Location API, and Direction API.

(4) IntelliJ IDEA: also released by JetBrains, which is a capable IDE for Java.

(5) Apache Tomcat: an open-source implementation of the Java Servlet, which is used to implement our web server.

(6) MySQL : a Relational Database Management System, which is used to implement our database.

(7) ASW EC2: a remote server provided by Amazon, which is used to deploy our server.

(8) Git: a version control tool widely used by developers, which is used to manage our source code and version evolution.

Connections:

(1) HTTP: HyperText Transfer Protocol, which is an application layer protocol, used for connection between client and server.

(2) HTTPS: HyperText Transfer Protocol over Secure Socket, which is based on HTTP, used for some secure connection with BU's server.

(3) JDBC: Java Database Connectivity, provided by Oracle, which defines how a client may access a database.

5. Testing scheme

For the testing phase, we completed a unit test while component developing, and also integration test after all the modules completed.

(1) Error-based testing: it focuses on error-prone points, based on knowledge of the typical errors that people make. We tested some possible errors, such as network errors, invalid input, and invalid location.

(2) Black-box testing: in black-box testing, test cases are derived from the specification of the software. We mainly consider the function of each module, such as a map module and tools module.

(3) Integration testing: it focuses on the overall function of the system with all components integrated. We tested the connection of client and server at localhost, and also the connection between servlet and database.

6. Possibility for maintenance

(1) Deploy our server on AWS: actually, we are now working on this, we have already set up an EC2 server on AWS, which has 1GB RAM and 30GB SSD. We believe that's enough to support hundreds of potential users in our university.

(2) Add more details to the map, which means we need to add more locations to our location table.

(3) Add Street View to our map, which will help students to recognize the buildings.

(4) Design and add a simple social module based on location, which supports message sending and receiving in a certain range.

7. Conclusions and future works

Highlight:

we believe it's a great idea to develop something that can help our friends and schoolmates. Also, we designed our requirements specifically because we are target users ourselves. Then we did an efficient work considering all the designing, implementation and testing and documentation stuff are completed within 10 days.

Problems:

Time and money are always the problems faced by developers, with limited time given (actually more than 1 month, but we had to do a lot of other assessments), we found it hard to complete our initial designed requirements. So we redefined our requirement, which means we moved some of the features to the maintenance phase, to complete the whole project ahead of schedule. Then about the money matters, we applied for Google API and AWS server for free, which means limited performance.

Also, we could have done better in requirement engineering, the fact is that we designed more functions than we could complete. After a brainstorm, quite a few ideas came into our mind, we overestimated ourselves and that's why time needs to be spent on re-designing the requirement.

Plans:

In our future projects, we should put more attention and effort on requirement engineering, a feasible requirement specification is of great importance for a project.

Also, we should be aware of the functions of software are designed for its target users. It's improper to design something ‘cool’ just for developers, so the first goal of designing is to satisfy the users' demands.