

# CIS 573 Software Engineering

## Group 22 (Ski Buddy) Iteration 1 Report

### 1. List of team members:

Tengyuan Ye, Yisi Xu, Wei Hu, Yue Yu

### 2. Code smells:

- 1) In MainActivity.java line 96 and 101, there are if-else blocks, where nothing has been done in the else block.

**Solution:**

In onItemSelected() method, if reading from parse fails or nothing was read from parse, the else blocks dealing with these situations are empty. So we added two dlgAlerts to deal with them in case sever shuts down or some accidents happen.

- 2) There is no way to get back from LoadActivity page to MainActivity page. Only a string "Hello world" is shown on LoadActivity page, Load Activity is redundant and should be removed.

**Solution:**

We merge LoadActivity.java into MainActivity.java, including deleting all intents in original LoadActivity.java as well as copy the parse.initialize method to MainActivity.onCreate(). Also, in AndroidManifest.xml file, move the <intent-filter> block from .loadActivity to .mainActivity block to let application starting from the main page. Finally, we deleted LoadActivity.java and <activity android.name=".loadActivity" .../> block in AndroidManifest.xml.

- 3) Method/ fields never used, e.g., TrailConditionActivity.java, line 145, "contains".

**Solution:**

In original code, it doesn't consider the situation when *slopeName* is not equal to the name got from the Json parser. So we fixed this bug. Here, *contains* means whether Json parser contains the *slopeName* or not. If it does not exist, we output a dlgAlert warning that the *slopeName* was not in parser.

- 4) Primitive obsession: In WaitTimeActivity.java line 69 and 70, there are two variables, toFill and toFillHelper which hold the information of slopes.

**Solution:**

we created a new class called Fill to hold these two variables and define methods that should be used. We fix all the codes that use these two variables.

- 5) Long method: In WaitTimeActivity.java line 91-103, there is a block for sorting the waiting time. This can be extracted as a separate sorting method.

**Solution:**

We extracted the sorting steps and defined a new method called *sortByWaitTime*.

- 6) Readability and Understandability: The original code lacks comments, which makes it hard to understand.

**Solution:**

We added comments to the code.

- 7) In *UserActivity.java*, line 178 and line 183, the catch block is empty.

**Solution:**

We modified the catch blocks.

- 8) In *UserActivity.java*, some fields such as *userExists* should not be a static field. It is only used in one method with *UsrActivity* class.

**Solution:**

We changed the static fields to non-static ones.

- 9) Unused variable: *friendList*. This variable is modified only once and never used.

**Solution:**

We removed it from *UserActivity.java*.

- 10) Unused variable: *friendList*. This variable is modified only once and never used. So we removed it from *UserActivity.java*.

**Solution:**

We extracted the *User.java* class which contains all the relevant fields.

In *User* class, the fields are:

*friends*: represent a set of current user's friends

*username*: a String to record the username of current user

*location*: a String to record the latitude and longitude of current user

*slopes*: an Array of String to record the favorite slopes of the user

Methods in *User* class:

*addFriends()* and *removeFriends()* are added to avoid message chain in *UserActivity.java*.

- 11) In *TrailConditionActivity.java*, the method *removeSlopeFromFavorites()* does not consider the situation when the user's favorite sloped list is empty.

**Solution:**

If the *JSONArray* *slopeArr* contains nothing, there will be a toast prompted to tell the user the slope cannot be removed from his/her favorite list.

- 12) When the user wishes to remove a friend but the user is not on his/her list, the user will not be notified.

**Solution:**

There will be a toast prompted, telling the user that the user he/she wants to remove is not currently on his/her friend list.

13) There are long methods In UserActivity.java, onCreate() method.

**Solution:**

We created two methods, setSlope() and setFriendLocations() to extract them from onCreate() method.



















### 3. Known bugs:

N/A

### 4. Issues to be resolved:

- 1) The log-in function is not a realistic one, it is effective only locally.
- 2) The function of showing friends' locations on the map is not realized.

### 5. Commit log:

	 Add .gitignore ...	bc325f3
	 add .gitignore files and refactor UserActivity.java Class	b3f4e3f
	 modify WaitTimeActivity.java	e54271a
	 modify TrailConditionActivity.java	3464939
	 modify MainActivity.java class	93304a0
	 AndroidManifest.xml	b737315
	 delete LoadActivity.java class	d9885de
	 fix bug when delete a trial that not exists	8574bed
	 extract long method onCreate in UserActivity.java	92117f4