Vayu Kishore

Jordan Kodner

Xiao Li

Jose Varela

CIS 350

Netter Center Attendance App Final Report

User Stories:

As a user, I want to save information from my phone to a database (3 points)

As a user, I want to be able to view database information on a Google Docs spreadsheet (3 points)

As a user, I want to be able to sync my phone with an outside source (3 points)

As a user, I want to be able to view student info on the phone (revised, 1 point)

As a user, I want to be able to leave a comment on a student (1 point)

As a user, I want to be able to log in and out, create new user (1 point)

As a user, I want to be able to sync back to the phone from Google Docs (3 points, pending)

Project Velocity:

12 for this iteration. That is 3 per person. That is good.

Bugs:

We have no known bugs at this time.

Untested Code:

We were able to write tests for all of our code, but not always to a very thorough extent. The user interface was difficult to write tests for and not always completely worth it. Tests that made sure that buttons actually causes the backend to behave properly were more useful than ones that, for example, checked in pressing a list item selected it. Those can be done in person.

As of now, we have tests for our local database, but we don’t know how to write tests to make sure syncing with Google Docs is working correctly. We can confirm it is manually by looking at output, but we haven’t written tests.

Additional Features:

The customer would have liked automatic syncing. We could have done this more easily if we had used Parse like the other group. According to them though, it isn’t that important. They are happy with manual syncing, and they liked that we chose to use Google Docs directly instead of Parse.

Our code allows syncing the phone to the database (primary purpose), but not back from the database too. This is a problem. We ran into a lot of difficulty on this part of the project. Fortunately, it is not nearly as important as syncing to the database. Even if this is not implemented for Friday, it will be done soon and redelivered to the client. As a description for future workers: a sync back returns atom, and we need to be able to convert it to JSON.

If We Could Do it Differently:

If we had written more tests early on, things would have been easier. We had close to zero ui tests early on because we couldn’t figure out how to write them. Near the end, someone told us about Robotium, which made testing much easier. It would have been nice to know about that earlier.

Our Client:

We found our client to be reasonable and easy to work with. They were understanding of Android’s technical limitations and of the limitations imposed by our time constraints and group size. They did a good job at communicating what they wanted. We would recommend them as a CIS 350 client in the future.

Copyright:

We are happy to give the Netter Center our project. They can use it for Netter Center-related initiatives however they wish. They cannot give it to outside organizations.

Code Inspection:

The following is the result of our code inspection. The description column is the original problems. The Action column is our fixes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Line #(s) | Description | **Action** | New line #(s) |
| doHttp | 159-175 | Duplicate code; create new instance, open database, exportJson(), close database for multiple types of data -> can be combined into one method | extracted method into getJSONFromDataSource | 159-164 |
| doHttp | 177-191 | Method chain; "allStrings.append…append…append…append…append", refactor to "clean up" code | chose to leave this as it is. .append functions like concatenation for stringbuilders, and its easier to read a concatenated string on one line, rather than split up per concatenation |  |
| doHttp | 155-157 | Readability, Style --> Would be more readable if all lines are kept to 80 characters max | Ensured all lines of code in the method are under this limit. The long lines that were previously 155-157 are a comment that explains what the method does and was put in for the purpose of the code review, thus these lines were unchanged. |  |
| doHttp | 159 | method name doesn't reflect what method does that well, a little vague | renamed method to synchronize | 171 |
| loadData | 475 | can clean up these unneeded, informal comments | comments removed | 520 (removed) |
| populateFields | 96-97 | get rid of commented code | removed |  |
| onAddFrequently | 266 | method name is a little confusing | Changed method name to “onAddActivityToFrequentlyUsedList” |  |
| checkInOutStudents | 243-271 | logic is hard to follow; especially with all the if-statements | Encapsulated conditions into methods of Checkin class; used enum instead of boolean to make purpose of parameter clearer. (note that the method was renamed markStudents, and some functionality was also moved to doMarkStudent ) | 239, 285-290 |
|  |  | Void methods might be hard to unit test, but I don't know how you are choosing to handle that. | For many cases, an exception should be thrown if something goes wrong. We did change/create some methods from to return a variable instead of void: in StudentSelectionActivity, markStudents returns an int; loadStudentListData returns ArrayList<Student>; loadSchoolActivityData returns SchoolActivity. | n/a |
| checkInOutStudents | 272 | makeText function call is long and difficult to read; consider creating a string variable for increased readability and changeability | This is actually considered standard practice, as discussed in meeting. | n/a |
| loadData | 467 | Ambiguous method name; had to read comments / analyze code to understand what the function does. A more descriptive name would also promote abstraction | Since this method loads all the necessary data from the database, it was not clear what a better name would be. However, its functionality was split into to two two methods that it calls were renamed to loadStudentListData and loadSchoolActivityData to be clearer. | 517-518 |
| populateFields | 72 | Visibility - This method is called only within the StudentDataActivity class, therefore consider non-public visibility | made method private | 58 |
| loadData | 475 | Seems as if this method is important in terms of overall functionality.Could have return statements to allow for better testing | Functionality was moved to two methods, loadStudentList and loadSchoolActivity that return ArrayList<Student> and SchoolActivity, respectively. These should be more easily testable. LoadData calls both of these methods. | 517-518, 522-537 |
| onAddFrequently | 266 | Could possibly declare the Alert variable at the beginning of the code just to follow convention and then set the text inside "if statements" | None |  |
| onAddFrequently | 275-279 | The method chain on the alert builder is long and called as the alert is instantiated. Consider restructuring how you show the Alert, | Extracted alert title and message as variables to increase mantainability. |  |