**WithU - Iteration 2 Review**

1. **What were the main difficulties so far?**

For iteration 2, the main goal of the iteration was to implement the main walk feature. This included everything from the pairing of walkers together/assigning pairs to the walk, to the main request and navigation features of the walk itself. Working with all of these moving parts was one of the main difficulties we encountered. Most of the processes were dependent on something else being done first, which made the testing/integration more difficult than iteration 1, where a lot of what we did was separate from others which made testing easy. For example, in order for a walk request to be sent to a pair of walkers, there first has to be a functioning pairing service. Then, for there to be a functioning pairing service, there needs to be two individual walkers shown active within the system. As it is clear to see, there is an important reliance on several components to make the overall application function properly.

Another difficulty presented in iteration 2 was the working with Google Maps to get directions. In iteration 1, we implemented the basic functions of Google Maps API. However, in this iteration, we had to work with Google Maps and the Location class in Android Library in order to manipulate location data points to create the navigation that would be given to the walkers in order to get to and from where they needed to be. Google Maps does not allow for this navigation route to be given directly, so instead we had to look around online for help with the algorithms for taking in all of the various data points to then draw the correct navigation route on the screen. This was an added difficulty that we didn't expect to encounter, which took some time to overcome.

1. **Were there any features you did not implement as planned, and why? Are you pushing some features to later iterations, and if so, why?**

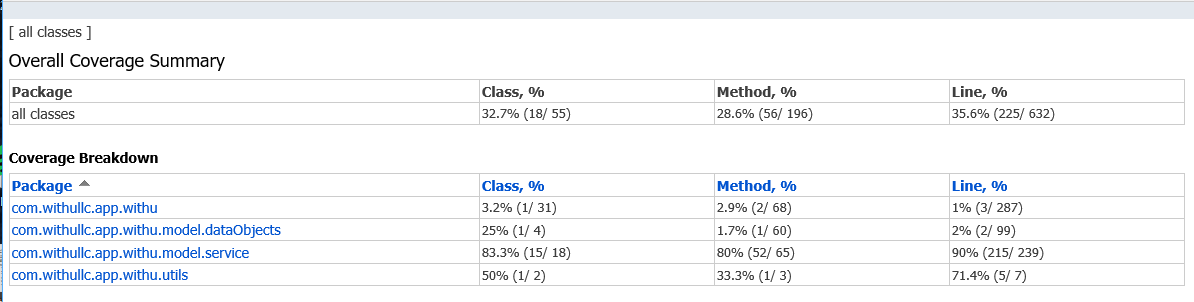
One of the main features not implemented is the Rating feature. If we finished the walk process quicker than planned we would’ve tried to implement it, but with the navigation difficulties mentioned above we decided to just leave that for later. We wanted a way for both the user to rate the walkers and the walkers to rate the users to ensure that all walkers were receiving good reviews or they wouldn’t be allowed to be walkers, and vice versa to ensure that a user was not consistently giving walkers a hard time. While this would be a cool feature to still try to implement, given that it has little impact on the overall functionality of the application and would require us to try and determine punishments for when ratings fall below a certain threshold, we decided it's best to leave that aside for now. Another feature we discussed earlier was to let walkers decide who they want to be paired with no matter how far apart they were, but for now it was easier to just have them automatically be assigned the closest person of the opposite gender in order to simplify the pairing process. The last thing that has been in the works but not implemented as part of the code we turn in is the backend work for Walker pairing and assign walkers to the walk. We were hoping to have it fully working before the end of the iteration, but decided it was best to leave it off of this iteration so we don’t mess up any of the working code we already have. We are close to having this part fully functional, but with all of the moving parts on the backend in regards to location we have had a harder time getting everything synched to the database correctly.

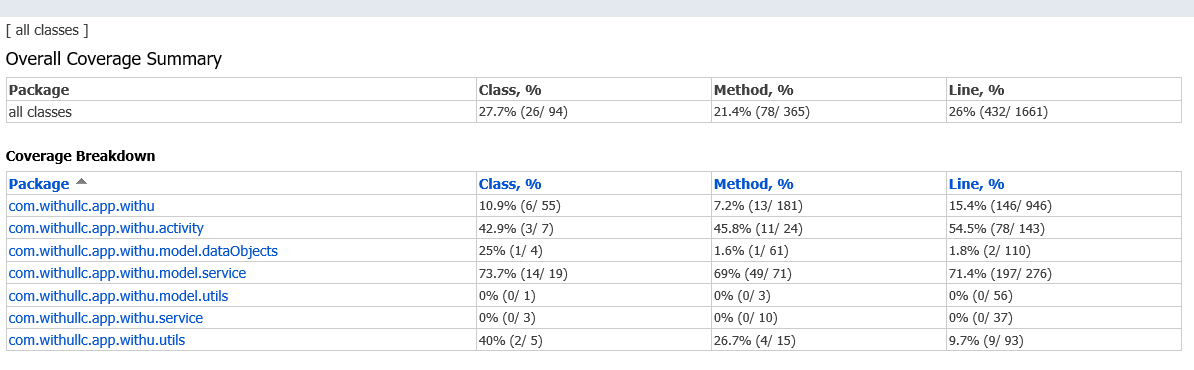
1. **What tests did you prepare for this iteration, and what are they covering? Did the tests you wrote deviate from your plan? What features are you not testing yet? Did you use any test frameworks, such as JUnit, the Android Monkey, Selenium, etc.?**

For this iteration we focused a lot more on the UI testing as opposed to the Unit and Integration tests. We still used the same Espresso testing framework and AndroidJUnitRunner test runner for these UI tests. Iteration 1 featured more of the true test driven development compared to iteration 2, as the majority of what we did was add logic to our application rather than tests.

1. **Give a sample of code coverage tool output both before and after adding to your test suite. What did you learn from the code coverage data? How did you use this information to expand your test suite and improve coverage?**

Code coverage at end of iteration 1:



Code Coverage at end of iteration 2:

As stated earlier our code coverage did in fact go down in this iteration as the code implemented in this iteration lended itself more to logic/reasoning than true testing like what was needed in iteration 1. The coverage also went down due to the fact we emphasized testing way ahead of schedule (early on in iteration 1) so a majority of the tests were written then, before a bulk of the code for iteration 2 was added. That being said, the code coverage tool has allowed us to identify fragments that weren’t covered, as well as allow us to identify key vulnerability points in our flow of logic in iteration 2. Certain activities like login and home have redundant layers of UI tests for fast and accurate feedback, while less frequented interactions like report button we chose would just be unit tested. Forgot email was a great example of where code coverage came in handy, as it would require way more effort to try and setup and monitor an email server for automated testing of this functionality. So instead, we were able to cover this user case manually, once per iteration, to ensure that it hasn’t been broken at any point.

1. **Optionally give a URL and instructions for using your application in the current stage. This makes sense for purely web-based projects, but it may be impractical for projects that must be installed on a client device.**

See ***README.md*** in GitHub repository. We have sent email invitation to give two TAs and Professor Tracy Lewis-Williams access to our GitHub repository ***spatterson4/WithU***. If you have problems accessing the repository, please contact Sam([spatterson4@wisc.edu](mailto:spatterson4@wisc.edu)).

We also submit our updated Design and Requirement Documents with this report and all changes have been highlighted with different colors.