

ECSE 321 DELIVERABLE 4

TEAM 12

19th November 2016

1 Release Pipeline Plan

The integration phase of the release pipeline is worked on by all members. Each team member has been given the role as a major contributor for one platform and the other members work as reviewers or minor contributors. This has been done to reduce the possibility of bugs and faulty code. Once a team member has finished writing a new iteration of the code or once they finish adding new functionality, the commit and push the code to GitHub. Changes to the code are done using eclipse and tested using the test functions. The android, Java and PHP code is the automatically tested through Travis by unit tests to make sure that nothing was broken. The automatic tests act as a sanity check. The tests are made in the same manner that was specified in Deliverable 3. If those tests are passed, team members will then do a code review to make sure that the changes were implemented properly and to maintain a constant flow in the code structure and architecture. Once the reviews are passed the code will be deployed in different forms based on the platform. The java files are deployed using a jar file, the android application is packaged into an APK file and the PHP files are updated and integrated on the server by updating the PHP files. The files to be deployed are all compiled and created by Travis to reduce the amount of time that it takes to make a release.

If an error is found during the automatic tests, then the developer is alerted of the failed tests and they will then work on a solution and then submit the code again for the automatic tests. False positives can happen during the automatic testing process and this will be addressed once the developer sees that a test failed when it should have passed. Once the automatic tests are passed, code reviews are done to avoid the possibility of false negatives. Not all false negatives can be found by code reviews but sometimes a review can help determine why a test passed when it should have failed. This thorough automatic testing process should reduce the possibility of deploying faulty code.

The rational behind having both automatic tests and code reviews is simple. The automatic tests done by Travis through unit tests is used to make sure that nothing was broken during the coding process. The code reviews are useful because they allow the team to look at some parts of the code that are not complete yet and therefore cannot be unit tested and it also allows the team to judge completed code to make sure that it matches up with the overall

vision of the software.