# Android application

Every time bugs were fixed or new features were added, we go through a release pipeline. The first phase is the integration. We have our android project folder in github, just like with the other 2 applications. Every time someone commits to the project he describes what he added and github is essential because we can see what exactly was changed at each commit and can easily go back to the last successful build.

The second phase of the release pipeline is the build process. We will execute a build after each integration because it helps detecting integration errors very quickly and easily. We will use Gradle for our android app instead of Ant because it has Ant’s power and flexibility and it is also easy to use with android studio. It is the default build tool for the android OS. When we fix a bug for example, we don’t want to recompile files that we did not modify, but we do need to recompile files related to those that have been modified. Those are called dependencies and Gradle will build all of a file’s dependencies first before building that file.

After the project is build, the deployment stage of the release pipeline is conducted. Android studio has its own tools that we can use for deployment. It can either deploy to the emulator with an android virtual device or it can deploy to a physical device. We can then test the new features and validate the bug features. Then we write more code and go back to the integration phase.

The code flows through this release pipeline in the following manner. Programmers are constantly making changes and improvements to the code. These changes are pushed to the Github repository frequently, with specific names to keep track of each commit. When a change is committed, Travis CI will begin automatically begin building this new commit. The main file commanding Travis dictates that the build for the android program is to be produced using Gradle. During this process, the tests written for the program are run ensuring that the latest commit does not hinder the program from functioning. If any of the tests fail, the build is unsuccessful and it is advisable to return to the previous commit. If all of the tests pass, then the deployment stage is performed. As the build file has been created, it is possible to run this file to view the full program running as it would for the client. This stage involves the programmers using the build-in android emulator to see the latest changes to the app physically working in the full program.

Each of the phase specifications was designed in a simple manner. The functionalities and tools used to produce the release pipeline were relatively new to the programmers. Thus, it was of interest to keep the process simple and operational. Travis CI is a very simple to use resource with large amounts of introductory documentation. This application solely required a link to the Github account and a Travis command file to run, and efficiently reproduced builds for each new commit. Gradle has been used for all android functionalities in the past deliverables and was a comfortable choice for the build method. It also has strong ties with the android studio program used to produce the android code. Finally, Github is a widely used cloud service that is ideal for groups of programmers working on a project. It is easy to link with the Travis CI web service, and has been used for the duration of the project already. Github allows the programmers to make and track frequent code commits and allowing all group members to instantly be informed of these changes. Github is a wise choice for the implementation phase, Travis and Gradle were effective choices for the build phase, and the built-in android emulator was excellent for the release phase.