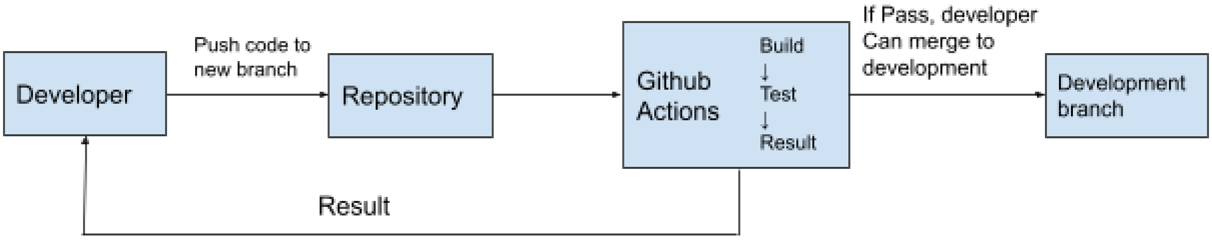
Continuous Integration

Team 6
Sanjel Sadikaj
Adam Hagan
Tudor Ciobanu
Na Tang
Armintas Zadeika
Liu Zhang

At the beginning of implementing continuous integration to our project we decided to use Travis CI as it was widely available and a popular choice for Continuous Integration. However, we soon ran into a couple of issues. We kept getting errors about our build.gradle file, however it was working fine when built locally. Because of this issue, we decided that Github Actions will be easier to set up and use, as well as that it has better documentation, if any errors occurred.

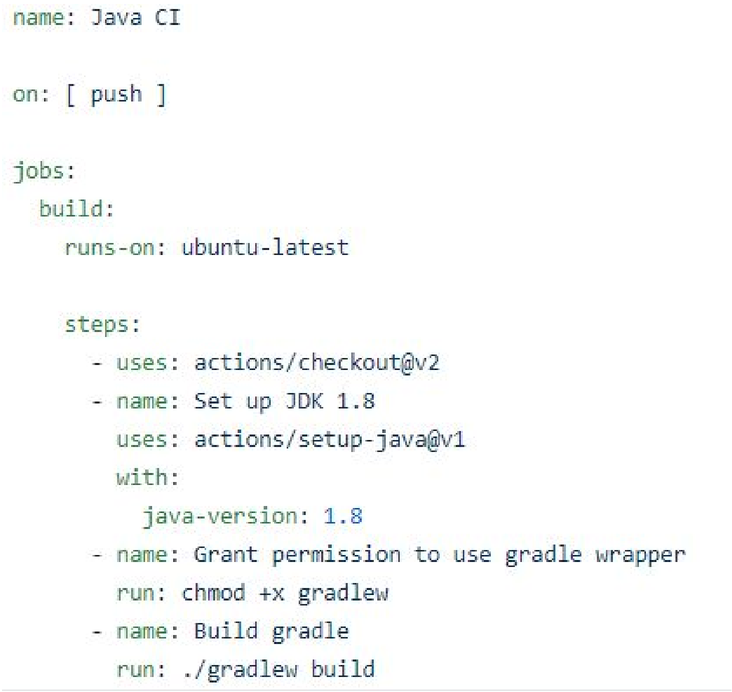
We have set up our CI to work with VSC (which we are using Github), so when anyone pushes code into a branch and makes a pull request, the continuous integration checks if there is a file with an extension .yml in .github/workflows folder, which is the configuration file for CI. If it exists, it runs the file, and after everything is done the person will be notified of the result (whether everything passed or failed). In the case of everything passing, the branch then can be merged to development.

The continuous integration infrastructure is depicted by the diagram below



After the changes are pushed to the development branch, they are tested by other developers, if everything is working as expected, then the development branch is merged to the production branch, which is the stable branch for use.

Github actions runs a file called gradle.yml, as seen below:



```
name: Java CI

on: [ push ]

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2
      - name: Set up JDK 1.8
        uses: actions/setup-java@v1
        with:
          java-version: 1.8
      - name: Grant permission to use gradle wrapper
        run: chmod +x gradlew
      - name: Build gradle
        run: ./gradlew build
```

It creates a virtual machine that is running the latest version of ubuntu and executes the given steps.
First it downloads the repository, configures Java 1.8 JDK, grants permission to use gradle wrapper command gradlew and then runs ./gradlew build which builds gradle.
The build command checks all of the code as well as runs all of the tests. Once everything is completed, it produces a result.

Github Actions allows us to check what the CI is executing and if there is an issue; where the issue is located. This helps track issues with both CI (whether mistakes were made writing the config file for CI) as well as any other issues regarding tests failing and code not being built.