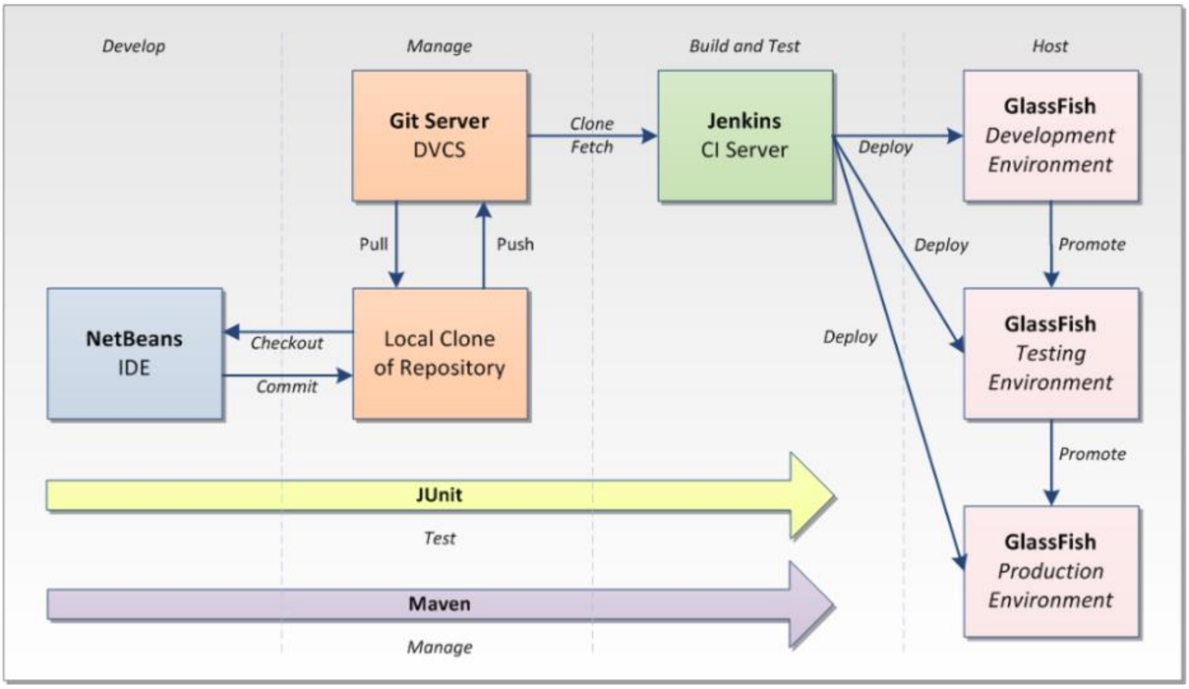
CT417 - Assignment 1

– Student Registration System –

Assignment Background

Your final year project will most likely contain a major software development component written in Java, C or C++. Therefore the aim of this assignment is to reinforce your understanding of some modern software development frameworks including Maven, JUnit, Git(Hub), and NetBeans, thereby emulating a continuous integration / deployment pipeline similar to the one outlined in the figure below. While you may not use all these technologies in your final year project, you are encouraged to pick-and-choose individual components that make your development process more robust, like for example the use of a software version management system (i.e. Git, via Git Bash or via IDE integration) or a unit testing tool.



Technology Frameworks used

For this assignment you should download / configure the following tools on your own computer:

1. A Java IDE with Maven and JUnit support. For this assignment I used Apache NetBeans 12.0, which by the way has a very handy Git integration (i.e. no Git Bash required); see https://netbeans.org/
2. If needed the Git version control management system; see https://git-scm.com/
3. A (free) account on GitHub; see https://github.com/
4. Jitpack access to your GitHub repository; see https://jitpack.io/
5. A free account with Shippable and access to your GitHub repository; see www.shippable.com
6. You will also access the Maven repository (to get the Joda time classes); see https://mvnrepository.com/

Assignment Outline

You have been asked to build a student registration system as follows:

a) **[7 marks]**
Create a Java project using Maven. The project should contain three classes and each class should have suitable accessor and mutator methods:
   a. A student class should contain some attributes such as (Name, Age, DOB, ID, username, courses and modules registered for, etc.).
   A specific method (getUsername()) will generate the student's username by concatenating their name and age.
   b. A module class, which should contain information such as module name, id (e.g. CT417), list of students, courses it is associated with).
   c. A course programme class containing course name (CS & IT or ECE, etc.), list of modules, list of students that are enrolled, academic start date and end date. Start and end dates should use **Joda Time classes (i.e. DateTime)**, which must be added as a project dependency.

Write a simple unit test using JUnit to make sure that all the above works (e.g. get getUsername()) as expected.

Also, use Git and Github to setup a local / global repository (the latter is required for part b).

Important: For this assignment you must use Java 11, rather than a later version (e.g. don't use Java 14 as supported by NetBeans 12), as part b) won't work otherwise (Jitpack issue). You can do this by changing the following 2 lines at the bottom of the pom.xml file as follows:

<maven.compiler.source>**11**</maven.compiler.source>
<maven.compiler.target>**11**</maven.compiler.target>

Further on, you should create a release of your repository on GitHub, before moving to part b). This makes it easier for Jitpack to find the program code.

b) **[6 marks]**
A second Maven project adds the project release in part a) as a dependency using Jitpack, i.e. pulling it from the GitHub repository on demand and providing you with the artefacts on demand.
Part b) acts as a driver project, creating a handful of fictional students with mock data, adding the students to a number of modules and adding a number of modules to a course. Further on, the program should print out a list of all the courses, their respective modules, and all students, their usernames, assigned modules and the course(s) they are registered for to the console. Again, use Git and Github to setup a local / global repository.

c) **[2 marks]**
Finally, you must create an account using Shippable which will act as the continuous integration server for part a). You will need to specify a .yml file to automate the build.

Assignment Deliverables

- Links to your Github repositories which contain the source code for part a) and b).
- Log files /screenshots from your local build (of part b)) / test run (of part a)), as well as a screenshot of the console output of part b).
- Screenshots of your local revision histories, i.e. your Git commits, for part a) and part b).
- Screenshots from Shippable’s Jobs and Insights pane to see the build history of the automatic build of part a).

Submission Deadline

Please see Blackboard.

Technical Support

There will be a discussion forum on Blackboard where you can post questions.