SYSC 4806 – Lab 5: Continuous Delivery of your AddressBook App

If you've made it this far, congratulations: you have turned your AddressBook program into a very simple web app! In this lab you are going to set up the continuous delivery pipeline for your app, so that it is tested and built on every commit, deployed on Heroku, and easy to update. Here are the steps to follow:

1- First, if you haven't already done so (should be second nature by now!) enable git version control for your project, and add a corresponding remote GitHub repo from IntelliJ (you may first need to create an empty repo on GitHub). Your GitHub repo may need to be public in order to take advantage of the free tier of Travis-CI and Heroku in the next steps. You should now be able to commit your changes and push them to GitHub using the VCS panel.

2- Let's write some tests. This tutorial shows you how you can run an integration test of your web app, and also a lighter way, using a mock, to run those same tests without having to start up a web browser every time. If your tests pass, move to the next step.

3- Now set up continuous integration on GitHub. For this we are going to use Travis CI. Follow these instructions (but don't just copy-paste what's in there: read carefully and only follow the instructions that apply to you!). Also note that the tutorial is fairly dated, and so the JDK version that you need to specify in the .travis.yml file should not be "oraclejdk8". It seems "oraclejdk13" or "openjdk8" work, but I just left out that field altogether (as per Travis-CI's own tutorial) and it worked for me. Now, if you push your code along with the tests you wrote on step 2 to GitHub, Travis CI should automatically run those tests and build your app if the tests pass.

4- Finally, let's set up Heroku. Heroku is a PaaS that can host your application. To make our continuous delivery pipeline as automated as possible, we are going to set up Travis CI so that every successful build is automatically pushed to Heroku. Again, I found the little blog post of step 3 very helpful.

5- Now try adding a simple feature to your address book, and test it. For example, add an address field to your BuddyInfo and write the corresponding test(s). Push your changes and see if your build passes and Heroku deploys the updated version

Success? Pretty exciting, no? Show your work to the TA. If you can't, upload a text file containing links to: your GitHub repo, your Travis CI, and your Heroku deploy. Or even better: just provide the link to your GitHub repo, and have a README.md on GitHub with links to your Travis CI and your Heroku deploy.

If your GitHub repo is private (but then I'm not sure if you can get a free Travis CI), please also invite your dedicated TA to be a member. The TAs' GitHub handles are listed on CuLearn in the top section.