Environment setup for Performing Lab 1 and Lab 2

1. Install Eclipse IDE for Java Developers: https://www.eclipse.org/downloads/packages/release/2018-09/r/eclipse-ide-java-developers;
2. Install Firefox Browser;
3. Add ChroPath add-on on Firefox: https://addons.mozilla.org/en-US/firefox/addon/chropath-for-firefox/ and restart Firefox once the add-on is installed;
4. Install Docker. Follow the instructions on this link: https://www.docker.com/products/docker-desktop for Windows and Mac. For Linux users: https://docs.docker.com/install/linux/docker-ce/ubuntu/#extra-steps-for-aufs; For Windows users not employing Windows 10 Professional or Enterprise, install Docker Toolbox: https://docs.docker.com/toolbox/overview/
    a. For Linux users install the *docker-compose* command by typing **sudo apt install docker-compose** on the terminal.
    b. For Windows and Mac the *docker-compose* command is installed together with Docker;
5. Download the test suite for the application (*appname* = {*addressbook,expresscart}*) from
   AddressBook: *testsuite-addressbook* in the repository (originally there was a Dropbox link)
   ExpressCart: *testsuite-expresscart* in the repository (originally there was a Dropbox link)
6. Import the java project *testsuite-appname* as *Maven* project in Eclipse by following *File -> Import -> Maven -> Existing Maven Projects*. In the resulting window add the path to the *testsuite-appname* folder in *Root Directory*. Select project *testsuite-appname* in *Projects* (pom.xml) and click *Finish*;
7. In Windows and Mac start Docker (click on Docker for Windows or Docker Quickstart Terminal); in Linux the *docker* daemon is always active once the software is installed; (for Docker Toolbox users take note of the IP: e.g., docker is configured to use the default machine with IP 192.168.99.100)
8. Open the terminal and head over to the project root (*testsuite-appname*) and run **docker-compose up** to install the *appname* application. The first time the command is executed, *docker* automatically downloads the application image; then it executes it by creating an instance (container).
9. On Eclipse run the *JUnit CheckTestCase* test. If the test cases passes (green color in the *JUnit* window in Eclipse) both the application and Eclipse are correctly set. The test case is in *testsuite-appname /src/main/java/tests/CheckTestCase.java*. In order to run it, it is sufficient to right-click on file *CheckTestCase.java* in Eclipse, and select *Run As -> JUnit Test*. (for Docker Toolbox change *localhost* in "BaseTest.java" with the IP found at step 7)
10. Install *Rabbit* time tracking Eclipse plugin:
    a. Download the executables (*jars*) from: *rabbit-jars* in the repository (originally there was a Dropbox link);
    b. Put the executables in folder *eclipse/dropins*. The *eclipse* folder is the one in which Eclipse is installed;
    c. Restart Eclipse and activate the plugin view with *Windows -> Show View -> Rabbit*.
    d. In Eclipse select the *CheckTestCase* file in the *testsuite-appname* project;
    e. On the Rabbit plugin window activated in step *c* go to *Resources* on the sidebar on the left-hand side. Nothing should be displayed on the main window. Click on the refresh button on the right-hand side of the plugin window; at this point the project *testsuite-appname* should be displayed on the main window.