LAS3007 Assignment

Lecturer: David Camilleri (dcami07@um.edu.mt)

INSTRUCTIONS

This assignment should be attempted individually.

Deadline: Thursday 14th January 2016 at 6pm. No extensions allowed.
Marking: This assignment will be marked out of 100.

- You will be required to hand in (80% of the study unit mark):
    1. all the code developed for this assignment (*upload the whole codebase folder as a compressed file to VLE*)
    2. a printed report (*along with the report document uploaded to VLE*) of not more than 20 pages discussing the following points:
        a. code structure
        b. test maintainability
        c. installation and configuration processes for Appium, Jenkins and Selenium Grid
        d. problems encountered during installations
        e. problems encountered during element locating
        f. further improvements to the framework

- You will be asked to present your work in a 10 minute presentation, which will be scheduled and students will be notified in advance (20% of the study unit mark).

OVERVIEW

You have been employed by the world renowned note-taking company **Evernote** as one of their test engineers. Evernote services are available on both desktop based website and mobile native applications.

Your main role will be to focus on developing stable and maintainable automated test scripts using a pre-existing automation framework written in Java and using Cucumber and Selenium WebDriver.

A further role will be to test selected features on their native mobile applications through automated test scripts using the same framework along with the mobile testing tool Appium. Prior to any investment in native mobile application testing, business requires that you create a Proof of Concept to show that this is feasible.

PREREQUISITES

- Set up **local** instances of:
    - Jenkins CI from where to execute your regression test suite
    - Selenium Grid to be used to allocate browser resources

WEBSITE TESTING (65%)

All website tests should be executed on your local instance of **Jenkins CI** and should test the site using your local instance of **Selenium Grid** on:

- Mozilla Firefox
- Google Chrome

Note: *Delete any notes created after each test.*

Features

1. **Login**
    a. Login with correct credentials. Ensure user is logged in.
    b. Login with no credentials. Ensure correct error message is displayed.
    c. Login with incorrect credentials. Ensure correct error message is displayed.

2. **Notes**
    a. Create a note - set note title and body. Make sure that the note just created is displayed in the Notes list.
    b. Create a note, log out, log in again and make sure that the note is still saved in the Notes list.
    c. Create a note, mark it as favourite (shortcut), then make sure that it is listed under Shortcuts.
    d. Create a note and create a 3x3 table inside the body. Ensure that the 3x3 table is created.
    e. Create 3 notes and ensure that sorting works well in the Notes list:
        i. Date Created (oldest first)
        ii. Date Created (newest first)
        iii. Title (ascending)
        iv. Title (descending)

3. **Search**
    a. Create 3 notes, then search for 1 note with the note title. Ensure correct note is shown.
    b. Create 3 notes, then search for 1 note with a part of the body text. Ensure correct note is shown.

4. **Notebooks**

   Create a new notebook, then create a note. Move the note to the newly created notebook. Make sure that it's present in the new notebook but not in the default one. Delete the newly created notebook at the end of the test.

5. **Trash Can**
   a. Create 3 notes. Delete them all, then navigate to the trash can. Make sure that all notes are in the trash can. Empty the trash can and make sure it's empty.
   b. Create a note and delete it. Navigate to trash can and restore the note. Make sure that the note appears back in the Notes list.

6. **Tags**

   Create 3 notes. Browse to 2 of the notes and assign them the same tag. Go to Tags section, click on the tag you assigned and make sure that only the correct 2 notes appear. Delete the newly created tag at the end of the test.

MOBILE APPLICATION TESTING (35%)

Evernote has asked you to create the Proof of Concept with tests against the preinstalled native Android *Contacts* application.

Application name: `Contacts`
Application appPackage: `com.android.contacts`
Application appActivity: `.DialtactsContactsEntryActivity`

You will need to include the Appium dependency in your Maven POM file and use AndroidDriver (as opposed to one of the Drivers used by Selenium WebDriver). In order to locate elements on the Android application (similar to browser developer tools) use UIAutomationViewer - <Android SDK installation path>\sdk\tools\uiautomationviewer.bat

Note: *Appium and Android installation requirements below.*
Note: *Delete any contacts created after each test.*

Features

1. **Create new contact**

   Create a contact with a name, mobile number, home number and email address. Upon saving, ensure that the info saved is the same as what you entered and that the contact appears under All Contacts.

2. **Search for contact**

   Create a contact and make sure it appears under All Contacts. Select it and edit the contact's name and mobile number. Make sure that the new info is saved.

3. **Favourite contact**

   Make sure that Favourites are empty. Create a contact and favourite it. Go to Favourites and make sure that newly created contact is present.

Installations (Appium and Android Emulator)

1. Download Appium - http://appium.io/

2. Download and install Android SDK (SDK Tools only are enough. No need to download the Android Studio) - https://developer.android.com/sdk/index.html

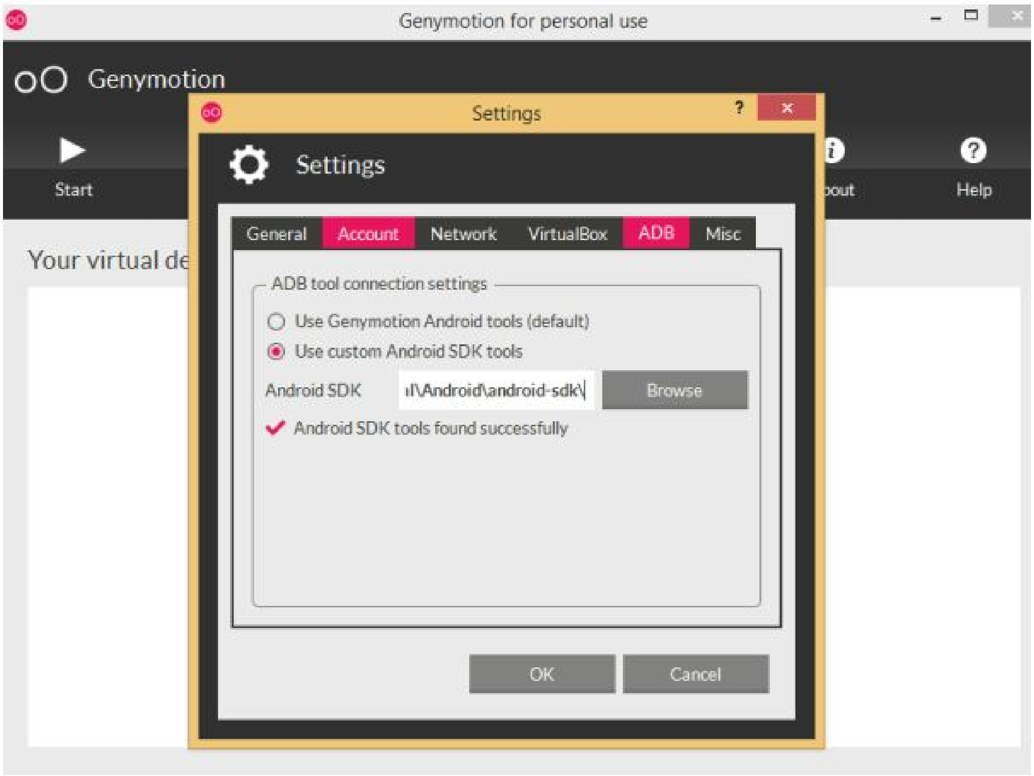
3. Once the Android SDK is installed, proceed in launching the SDK Manager.
   a. Under Tools>Options, tick Force https://…
   b. Install these required packages:
      i. Android SDK Tools 24 or higher
      ii. Android SDK Build Tools (highest version available)
      iii. Android SDK Platform Tools (highest version available)
      iv. SDK Platform of your installed API

4. Set the System Environment Variables:
   a. set ANDROID_HOME=<Android SDK installation path>\android-sdk-windows
   b. set PATH=%PATH%;%ANDROID_HOME%\tools;%ANDROID_HOME%\platform-tools
   c. Open Command Prompt and type '`adb`' from default directly to ensure that the above was set up correctly.

5. Download GenyMotion (with Oracle VirtualBox), which is a free Virtual Machine to run Android on - https://www.genymotion.com/#!/download

6. Download an Android Virtual Device and import it through Oracle VirtualBox (this will then automatically appear in GenyMotion).
   a. Link to an Android Virtual Device that is compatible with GenyMotion: http://files2.genymotion.com/dists/5.1.0/ova/genymotion_vbox86p_5.1_150409_102009.ova

7. Open Genymotion and click on Settings - enter the location of your Android SDK installation as shown below:



ADB tool connection settings

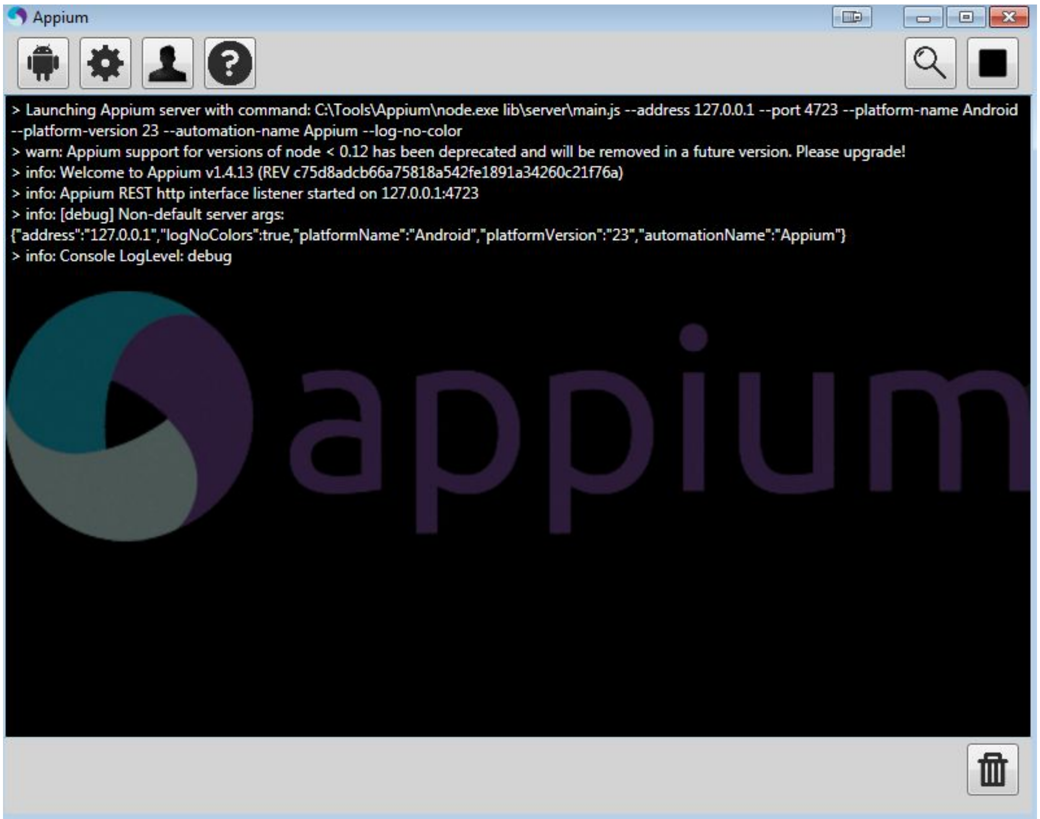
○ Use Genymotion Android tools (default)

● Use custom Android SDK tools

Android SDK ıl\Android\android-sdk\

✓ Android SDK tools found successfully

8. Before running a test, launch Appium and start it. Make sure that the server is running. It should look something like this:



```
> Launching Appium server with command: C:\Tools\Appium\node.exe lib\server\main.js --address 127.0.0.1 --port 4723 --platform-name Android
--platform-version 23 --automation-name Appium --log-no-color
> warn: Appium support for versions of node < 0.12 has been deprecated and will be removed in a future version. Please upgrade!
> info: Welcome to Appium v1.4.13 (REV c75d8adcb66a75818a542fe1891a34260c21f76a)
> info: Appium REST http interface listener started on 127.0.0.1:4723
> info: [debug] Non-default server args:
{"address":"127.0.0.1","logNoColors":true,"platformName":"Android","platformVersion":"23","automationName":"Appium"}
> info: Console LogLevel: debug
```