Software Engineering

Lab3 – Testing

Lab Overview:

In this lab, we will introduce the basics of testing and automation of both end-to-end(E2E) testing and unit testing. You will be required to write automated tests for an implemented code similar to the examples given to you.

Objectives:

After this lab, you are supposed to have good basic knowledge about:

- Difference between unit testing and E2E testing
- Concept of mocking
- Writing E2E testing scripts
- Writing unit testing scripts

Requirements

Task 1: Write a unit test for credit function in Bank class given that the function should:

1. Add to the customer account in the normal case
2. Raise a TypeError if the amount is equal to None
3. Return -1 if the amount entered to the function is negative

Task 2: Write a test that logs in then logs out of the app implemented by the code in flask folder

Folder Hierarchy:

- Testing Slides.pdf
- This document.
- test_google.py (E2E testing example)
- Unit Testing(unit testing examples)
  - student.py
  - test_student.py
  - test_fn_exa
- Mocking(mocking example)
  - student.py
  - test_student_by_mocking.py

- “Requirement files” folder:
    - bank.py (code to be tested in task 1)
    - “flask” folder similar to lab 2 example (code to be tested for task 2)

Running Files Notes:

- Make sure that you can run the flask application in “flask” folder as per instructed in the previous lab
- Unittest should be installed by default. If python cannot import unittest install it using “pip install unittest”
- To run the test you can open your terminal cd to where the test file is run(python -m unittest discover -p "test_student.py") or you can use the unit test discovery feature in your IDE
- Install selenium using “pip install selenium” or in PyCharm settings go to project interpreter and install the “selenium” package to the interpreter you will use
- Install Selenium driver for your browser(check that the driver is compatible with your browser version). Refer to this page for links to download and setup the browser driver https://www.selenium.dev/documentation/en/webdriver/driver_requirements/

- Run test_google.py after changing webdriver.Chrome() to the driver that suites your browser and make sure that it opens google.com and makes a search