**CS458**
**SOFTWARE VERIFICATION AND VALIDATION**
**2021-2022 SPRING**
**PROJECT # 3**
**INTRODUCTION TO TEST DRIVEN DEVELOPMENT (TDD)**

You are asked to do the following stuff as team:

1. With the help of Test Driven Development (TDD), develop a responsive page (that can run on laptops, mobile devices etc., no mobile native code required) for the following tasks:
    a) You enter your coordinates of your location and it shows your country (via Google Map API etc.).
    b) It gets GPS coordinates of your device automatically and it shows your distance to the Geographic North Pole (Terrestrial North Pole).
    c) You enter your coordinates or it gets GPS of your device, then it shows your distance to the Moon's core (You should also take the changing position of Moon into account).
    d) For each one (a,b, and c parts) write all the test cases.
    e) Write test code (red-line) and write the main code to pass just the red-line.
    f) Attach your code and UML diagrams (Activity, State, Use-case, Sequence, and Class)
2. Refactor your code and explain what you have done.
3. Evaluate your TDD experience in terms of development velocity and code quality.

**Instructions**

**A-Test cases and test automation**

- Try to come up with test cases of comprehensive nature. For instance if you say
    - Test case#1: Check how the code responds to valid/invalid name.
    - Test case#2: Check how the code responds to valid/invalid birthdate.
    - Test case#3: Check how the code responds to valid/invalid City. .. etc (you got the idea).

  Those will be considered as 1 test case, not 5. Try to be creative when designing your tests. Take your time when thinking about test cases; this is the most important part of your project. Please remember that, testing should focus on finding errors with minimal effort, then during test case writing, think like test expert not like a developer.
- When automating tests, think about executing thousands of tests in a few seconds. Do not write trivial test codes that require the simple interaction of the end-user (such as examining the behavior of the application during a test by displaying a pop-up message).

**B-Report preparation:**

The report should have the following properties:

- It should contain screenshots as well as UML diagrams of the actual application.
- It should contain important **excerpts** of the test code, explaining how those excerpts correspond to the automation of your test cases.

- It **should not** exceed 20 pages including the cover page.
- It **should** contain your code of the project not the code libraries. Auto-generated HTML or CSS codes should not be included. Your code should be submitted next to the report as discussed in the following.
- The report should be self-contained. That is, the reader shouldn't feel the need to refer to your code for understanding what you have done, s/he only needs to refer to the code if s/he feels curious about some implementation detail.
- Cheating is prohibited.
- Your own ideas and your own words are expected.
- Creativity is appreciated.

**C-Code submission:**
The code can be submitted either as a separate zipped file (attached or a link to online storage) or as a link to an online code repository. The code directory *__must*__ contain a README file with information about where to find the source as well as the test codes (or in case you send the whole project, the README file explains how the project is structured and instruction on how to run the code).

**D-Submission details:**
- Deadline: **25th of April, 23:59:59 PM.**
- Email subject: CS458 P#3 submission - [Name][Surname] (of the sender)
- Email body should contain: [Name] [Surname] - [Email address]-[Student ID] of all group members.
- Attach the report in (**pdf format only)** and send to *altunel@bilkent.edu.tr* and TA Cihan Erkan(*cihan.erkan@bilkent.edu.tr*) along with the code/ link to code.

Good luck