CS-UH 2012: Software Engineering
Fall 2019

Group Project (20 points)

Step 1 (Due Date: Nov 13th):

- Each team will create their own private GitHub repository for the project.
- Share the GitHub repository representing the team via email once created. Your commits will be regularly checked to monitor your progress.
- Upload the project proposal (System request).
- Give access to Dena and me as a collaborator:
  - Mai-Oudah
  - daa4-nyuad
- Upload all the deliverables with respect to their due date, if any.

Step 2:

- Produce the Software Requirement Specifications (SRS) document. In this document, a detailed description of a software system to be developed with its functional and non-functional requirements is provided.
  - In addition, provide a report on how the requirements were gathered and elicited.
- Produce a feasibility study from a technical and an economic point of views. Show clearly the risks taken in the project.
- Estimate the size of the project using the function points analysis (FPA).
- Each team should draw a Project Development workplan (via Gantt Chart).
- Each team will determine the Software Process Model (The Waterfall Model, The V-Model, The Incremental Model, Evolutionary Models, etc.) which is the most suitable for its project. Please, justify it!

Step 3:

- Produce a detailed use case model including:
  - A use case diagram
  - A textual description for each key use case
- Produce a detailed activity diagram for two key use cases
- Produce a detailed class diagram
- Produce a detailed sequence diagram for two key use cases

Step 4 (Final Project Due Date: Dec 12th):

- Each team will determine which programming language, integrated development environment (IDE) and/or other tools will be used for implementation.
- Carry out the implementation/construction as planned in the project workplan.
- A Test Plan is created considering use cases derived from the SRS document.
    - This plan consists of acceptance/validation tests for all use cases you indicated in the SRS document.
    - The test plan is supposed to describe the scope, approach, objectives, resources, and schedule of a software testing effort. It identifies the use cases to be tested, who will carry out the testing, who will take the test, what will be the pass/fail criteria, the testing schedule etc.
- After implementation, the system should be tested it so as to rest assured whether it is working as the customer/stakeholder is expecting.
- Design Test cases for each use case, and cover both success and alternate scenarios. Test your system following the Test Plan.
- If unexpected outcome is encountered during tests; this means the system has bug(s)/defect(s). In such a case, the root causes need to be traced back to and be resolved.
- Each team will submit a Test Results document summarizing the encountered bugs, which test cases uncovered these bugs, root causes of each and how they were resolved.

Additional Instructions:

- Use the templates provided in NYUclasses as needed.
- The code for implementation should be clearly documented.
- Creativity and going beyond what's expected, will be rewarded.
- Plagiarism will not be tolerated and will result in failing the project.
- Late submissions will be penalized: -20% for each late day.