**UNIVERSITY OF GHANA**
DEPARTMENT OF COMPUTER SCIENCE
BSc. COMPUTER SCIENCE

## CSCD 306 - Software Engineering Course Project

- Objective: design and implement a complex software system utilising the skills acquired in this course

## Project phases, deliverables, and due dates

This project must be completed by teams of seven students using an agile process model.

Each deliverable must be submitted on or before the indicated due date. Teams will not be allowed to proceed to a subsequent project phase without providing a satisfactory deliverable for the previous stage.

Each deliverable document must have a title page indicating the project title, deliverable name, team name and membership, and submission date.

The following will always be considered when grading your work in this course:

- **Presentation**: the report is well-formatted, easy to read, and easy to navigate;
- **Quality of writing**: language, grammar, clarity, professionalism;
- **Consistency**: all required points are addressed in the same order as listed in each section below.

## Project team info sheet (10 pts, due 14/2/20)

Get to know your team and make decisions about your own rules. Submit a PDF document containing the following:

- Introduce your team: give your team a unique name and provide a group picture of the entire team. What are the strengths of your team? ***(1 pt)***
- Introduce each team member: include a picture and a 100-word biography of each team member. A biography should be professional as if you were introducing yourself to a prospective employer. Identify one team member who will serve as the primary contact for the instructor. ***(3 pts)***
- Hold your first meeting: include a picture of all team members sharing a meal (breakfast/lunch/dinner/coffee/whatever--the choice is entirely up to you) ***(1 pt)***
- Discuss every aspect of the team agreement in detail and describe all your decisions in the writeup. ***(5 pts)***
  Your team agreement must include the guidelines for the following (but feel free to add any other items you believe are essential):
  - methods of communication (email, phone, messenger, text...)
  - communication response times (email, phone, messenger, text...)
  - meeting attendance (when to meet, whether all meetings are mandatory, ...)
  - running meetings (when, where, face-to-face vs online, who takes minutes/notes, ...)
  - meeting preparation (whether the preparation is needed, what to prepare, ...)
  - version control (what to/not to commit, the content of log messages, ...)
  - division of work (how to divide work, who will decide who does what, ...)
  - submitting assignments (when to submit, who will submit, who will review the submission, ...)
  - contingency planning (what if a team member drops out, what if a team member consistently misses meetings, what if a team member is academically dishonest, ...)
- Each team member needs to create a GitHub account. Include each member's GitHub account name in the writeup. ***(0 pts)***

## Project proposal (10 pts, due 1/3/20)

Submit an informal essay *(PDF, 2 pages max)* describing the following:

- Intended use of the system: who and how will use the system. ***(4 pts)***
- Its overall functionality: what will the system do, how will the system help its users accomplish their tasks. ***(3 pts)***
- Main components of the system: break down the system into logical or architectural components and provide the rationale for this breakdown. ***(3 pts)***

## System requirements (10 pts, due 8/3/20)

Submit a system requirements document *(PDF with a UML use case diagram, no page limit)* as described below:

- Begin with a narrative to give a general overview of the system's functional requirements. Provide one big use case diagram illustrating the overall functionality of the system. Describe each use case in an easy to understand natural language. ***(2 pts)***
- Using the use case diagram as a starting point, convert each use case into a user story. Organise your user stories as a numbered list. ***(2 pts)***

  A typical format of the user story is as follows: As a *user role*, I want a goal so that *reason*:
  - The user role represents an actor or a developer who benefits from this story;
  - The goal of the story is a feature or function in the system, a tool, or a part of a production pipeline;
  - The reason describes the benefit to the customer or user when this feature or function is used.
- For each user story, provide a set of pre- and post-conditions (refer to each corresponding user story by their numbers). **Note**: list pre- and post-conditions under the corresponding user story. **Note**: there should be a single list of user stories in your document. ***(2 pts)***
- Are there any user stories that are too big or complex? Can some of them be decomposed into smaller user stories? For each user story, mention whether or not it may need to be broken down *(this will be done later)*. ***(1 pt)***
- Provide a separate list of any relevant nonfunctional requirements. ***(1 pt)***
- Include a glossary that defines all relevant terms that may have a special meaning in the context of the system. ***(2 pts)***

## Product backlog (10 pts, due 29/3/20)

Submit a detailed, prioritised product backlog document *(PDF, no page limit)* as described below:

- Refine your user stories taking into account the instructor's feedback. Break down previously identified large user stories. ***(2 pts)***
- Estimate the size of your user stories. Use Fibonacci numbers to represent a relative size of each user story. Note the cumulative size of all user stories in your product backlog. ***(2 pts)***
- Provide an updated numbered list of all user stories; indicate pre- and post-conditions. ***(1 pt)***
- Taking into account the pre- and post-conditions, identify a subset of user stories to be implemented during the first iteration *(there will be a total of three iterations)*. Be sure that the cumulative size of the selected user stories is about 1/3 of the size of the full backlog. Describe the functionality that your partially implemented system will have at the end of this iteration. ***(3 pts)***
- Design key features of the user interface; provide sketches of your designs. ***(2 pts)***

## Iteration reports (3 @ 10 pts each, due 3/4/20, 17/4/20, 24/4/20)

At the end of each iteration, submit an iteration report *(PDF, no page limit)* containing the following:

- What functionality does the system have at the end of this iteration? ***(2 pts)***
- List the user stories that you have implemented at this iteration. ***(1 pt)***
- Did you end up making any changes to any of these user stories? Did you break down further any the user stories? Did you identify any new user stories during this iteration and, if so, did you add them to the product backlog or decide to implement them right away? Explain. ***(2 pts)***
- What are the "lessons learned" at the end of this iteration? What would you do differently next time? Explain. ***(2 pt)***
- Provide an updated numbered list of all user stories yet to be implemented; indicate pre- and post-conditions. ***(1 pt)***
- ***(2 pts)***
  - If this is not the last iteration:
  - Given the current functionality of the system and taking into account the pre- and post-conditions, identify a subset of user stories to be implemented during the next iteration. Be sure that the cumulative size of the selected user stories is about 1/3 of the size of the full backlog. Describe the functionality that your (partially implemented) system will have at the end of this iteration.
  - If this is the last iteration:
  - Are there any user stories left unimplemented in the backlog? Are there any new user stories that you would consider adding to the backlog. List these user stories and explain them.

## User manual (10 pts, due 29/4/20)

Submit the user manual document *(PDF, 2 pages minimum not including the screenshots)* as described below:

- Detail all necessary steps needed to deploy/install your system. Provide all necessary technical specifications. ***(2 pts)***
- Explain the main features of the system to a potential user who may not be familiar with it. ***(3 pts)***
- Provide a walkthrough for the main scenario of using your system; include screenshots as necessary. ***(3 pts)***
- Provide walkthroughs for at least two additional scenarios with additional/alternative functionality; include screenshots as necessary. ***(2 pts)***

## Project demonstration (20 pts, due 12/4/18, 12/6/18)

All software developed within this project must be successfully demonstrated in class. Each demonstration must be accompanied by a brief presentation explaining the nature/specifics of the project.

## Grading and final survey

All deliverables will be graded as a result of the work of the entire team. However, individual students may receive different grades based on the degree and quality of their involvement in the project. To facilitate the objectivity in grading, each student will be required to complete a confidential survey about the participation of other members of his or her team in the project. These surveys will be strictly confidential. Students who fail to complete this survey will receive a grade of 0 for the entire course project.