COS420/520 – Introduction to Software Engineering – Project Description – Spring 2023

Project Overview:

For the project, the students must develop a mobile (Android) or a web application. The high-level topics are given by the instructor; however, the students need to define the detail of their topic by themselves. The students need to identify 4 – 6 clients for their app from the university to elicit and validate the requirements (i.e., focus group) and for the usability study and acceptance testing.

The topics of the applications are:
- Health monitoring applications – Meditation; Healthy habits for kids; Fitness monitoring applications; Sleep assessment applications.
- Productivity applications – Planning the day; Time management.
- Shopping applications – Sell stuff online; Shop smarter; Home remodeling.
- Travel planner applications – Trip itinerary; Finding hotels or place to stay; Sightseeing.
- Education applications – Teaching computer science/code to girls; Kid's learning platforms.
- Financial application – Managing banking accounts and credits; Managing spending; Managing the stock market.
- Social networking – Photo/video sharing applications; Dating applications; University meetups or clubs.
- Gaming applications – Strategy games; Board or card games; Games for kids under 5 or 6-8; Coloring and painting.
- Others – Any other category of interest.

The teams should have **four to five** students. For the project, each team needs to identify the topic, and the category of their application, write a description of their application and provide a comparison between their application and the existing ones. Next, each team will gather requirements, including necessary security and privacy requirements, design the software with UML models, use architectural and design patterns to satisfy those requirements, implement a secure application and thoroughly validate the application through extensive testing. Each team has to follow software project management guidelines and must have a thorough configuration management plan.

Initial group selection will be done by the instructor. To ensure more balanced groups, the instructor will pick the group members based on the questionnaires given on the first day of the class and based on the stated preference of the students.

After the initial selection, the students will have **Two** full days to come up with arguments as to why they would like to be in another group. After the two days grace period, the groups will be counted as final.

Note that, to have diverse topics, the teams cannot have similar topics for their projects in the course. Each topic will be assigned to the teams based on a first come, first served.

Students need to submit their group's name, two preferred topics by January 22nd, 2023, 11:59 PM ET, via an email to sepideh.ghanavati@maine.edu.

The topic and the category of the application shall be approved by the instructor before the project is started.

The total grade for the project is calculated out of 100. However, it is worth 45% of the total final grade.

- Deliverable 0 – 5% – <u>Due date:</u> February 5th, 2023, 11:59PM Eastern Time
- Deliverable 1 – 15% – <u>Due date:</u> February 19th, 2023, 11:59PM Eastern Time
- Deliverable 2 – 15% – <u>Due date:</u> March 5th, 2023, 11:59PM Eastern Time
- Deliverable 3 – 15% – <u>Due date:</u> March 26th, 2023, 11:59PM Eastern Time
- Deliverable 4 – 15% – <u>Due date:</u> April 4th, 2023, 11:59PM Eastern Time
- Deliverable 5 – 20% – <u>Due date:</u> April 23rd, 2023, 11:59PM Eastern Time
- Proposal Presentation – 5% – February 28th, 2023, 11:59PM Eastern Time
- Final Presentation – 10% – April 26th, 2023, 11:59PM Eastern Time

The instructor periodically checks the GitHub repository and tracks each student's participation and contributions on GitHub. Note that GitHub shows the contributors, the distribution, and the percentage of the contributions. Each student will also be graded individually based on their contribution to GitHub. If the instructor decides that a student's participation in any of the deliverables is not satisfactory, the student will lose points.

All groups must meet with the instructor once a month to discuss their progress and challenges. The meeting times will be specified on Brightspace.

<u>General Guidelines for All of the Deliverables:</u>

- All documents should be uploaded and updated on GitHub by the person who either created the document or last updated the document.
- The documents should be submitted in Word or PDF format.
- Only Brightspace submissions are accepted for grading.
  - Submission via email, in person, or any other form, will be graded as **0**.
- If for any of the documents in each deliverable, the group did not get the full grade and/or received feedback from the TA or the instructor for improvement or changes, the document must be updated and resubmitted in the next deliverable.
  - Only the documents that received no feedback or are graded as 100% do not require updates (unless a change happens in your software development process).
- The file name should have the following format: **GroupName_Deliverable_i_ [Name of the Document]** for *i = 0 .. 6*.
  - If the name does not follow this format, the team will lose **50%** of the total grade of the deliverable.
- Peer-review documents must be submitted the day after each deliverable's due date at 6:00 PM. For the exact due date, refer to Brightspace.

Required Artifacts for Each Deliverable:

Deliverable 0 – Due date: February 5ᵗʰ, 2023, 11:59 PM ET (5%)

- <u>Project Description</u> document (which includes Group Name, Team Members, App Name, App description, and general overview of similar apps) – Two to Four Pages (40% of Deliverable 0)

- <u>User Story</u> document – About two pages or at least 15 – 20 user stories (40% of Deliverable 0)

- Create <u>Product Backlog</u> (10% of Deliverable 0)
  - Templates are posted on Brightspace.

- Set up a <u>GitHub</u> repository (Configuration Management Repository) and invite Dr. Ghanavati and the TA as contributors (5% of Deliverable 0)

- Set up a <u>Kanban board</u> on ZenHub or other similar Kanban board applications and invite Dr. Ghanavati and the TA as a contributor to it (5% of Deliverable 0)

Deliverable 1 – Sprint 1 (Documents/Software) – Due date: February 19ᵗʰ, 2023, 11:59 PM ET (15%)

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or other similar Kanban board applications.
  - Check the requirements for the daily scrum on the *"COS420_Projects' Artifacts Details _Spring 2022"* Document.

- Update <u>Project Description and User Story</u> documents based on the feedback given on deliverable 0 (5% + 5% of Deliverable 1)

- Update <u>Product Backlog</u> and create <u>Sprint Backlog</u> documents for Sprint 1 (10 + 10 % of Deliverable 1)
  - Templates are posted on Brightspace.

- Create the <u>Software Requirements Specification (SRS)</u> document (40% of Deliverable 1)
  - You are expected to have about 15 – 30 functional requirements and 10 – 15 non-functional requirements.
  - You need to have some mock-ups of the UI.
  - You do not need to fill out anything about use cases in this document.
  - Begin the requirements by selecting a few of the features that you would like to develop first and write the requirements for them. Continue with the next features if you still need to have more requirements.

- Create <u>Sprint Review 1</u> document (10% of Deliverable 1).
  - Follow the template posted on Brightspace – This document is generally about a page.

- Update <u>Kanban board</u> (5% of Deliverable 1)

- Update <u>GitHub</u> with documents (5% of Deliverable 1).

- Start the software development and push to GitHub as you progress (10% of Deliverable 1).
    - It is recommended that in this deliverable, you focus on setting up the environments and the necessary requirements and dependencies. A README file also needs to be created.

- Create <u>Peer-Review Report 1</u> document (This document is graded separately and is submitted a day after the deliverable's due date).

Deliverable 2 – Sprint 2 (Documents/Software) – Due date: March 5th, 2023, 11:59 PM ET (15%)

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or other similar Kanban board applications.
    - Check the requirements for the daily scrum on the *"COS420_Projects' Artifacts Details _Spring 2022"* Document.

- Update <u>Product Backlog</u> and <u>Sprint Backlog</u> documents for Sprint 2 (5 + 5 % of Deliverable 2)

- Update the <u>Software Requirements Specification (SRS)</u> document based on the feedback given on deliverable 1 (10% of Deliverable 2)

- Create the **Architecture Design** document with its description (15% of Deliverable 2).
    - You need to show the design of your software at the architecture level, explain component and their relationship and interaction with the other components and justify the selection of the architectural pattern.
    - This document is generally between two to four pages.

- Create the **Use Case Models and Descriptions** document (30% of Deliverable 2)
    - You are expected to have about 10 – 20 Use Case Descriptions and 2 – 4 Use Case Models.
    - Ideally, 3 – 5 of your use case descriptions are merged into one use case model.
    - Pick a few of the main features of your application that you would like to begin developing first, and based on their FR/NFR, create the use case models and descriptions.

- Perform and create a **Focus Group** document (10% of Deliverable 2).
    - Follow the guidelines for the focus group in *"COS420_Projects' Artifacts Details _Spring 2022".*

- Create <u>Sprint Review 2</u> document (5% of Deliverable 2)
    - Follow the template posted on Brightspace – This document is generally about a page.

- Update <u>Kanban board</u> (5% of Deliverable 2)

- Update <u>GitHub</u> with documents (5% of Deliverable 2).

- Continue the software development and push to GitHub as you progress, create the README file (10% of Deliverable 2).
  - It is recommended to begin with the UI and some classes, such as sign-up, login, etc.

- Create <u>Peer-Review Report 2</u> document (This document is graded separately and is submitted a day after the deliverable's due date).

Deliverable 3 – Sprint 3 (Documents and Software) – Due date: March 26th, 2023, 11:59 PM ET (15%)

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or other similar Kanban board applications.
  - Check the requirements for the daily scrum on the *"COS420_Projects' Artifacts Details _Spring 2022"* Document.

- Update <u>Product Backlog</u> and <u>Sprint Backlog</u> documents for Sprint 3 (2.5 + 2.5 % of Deliverable 3)

- Update the <u>SRS</u> document based on the feedback given on deliverable 2 (if updated: 5% of Deliverable 3).

- Update the <u>Architecture Design</u> document with its description based on the feedback given on deliverable 2 (5% of Deliverable 3).

- Update <u>Use Case Models and Descriptions</u> document with potential new use cases and based on the feedback given on deliverable 2 (10% of Deliverable 3 or 15% of Deliverable 3 if the SRS did not need an update).

- Create a **Sequence Diagram** document (30% of Deliverable 3).
  - You are expected to have about 7 – 15 analysis sequence diagrams and 3 – 7 design sequence diagrams.
  - For each of the analysis sequence diagrams, you need to write the descriptions of the steps.
  - Pick a few of the main features of your application that you began developing and draw the sequence diagrams based on those.

- Create the **Domain Model** document (20% of Deliverable 3).
  - You are expected to have the model of the domain with its links and relationships. The expectation is that you have at least 10 classes in your diagram.
  - If the model is too large, break it down into 2 – 4 smaller models.
  - You also need to have a description of the domain model as a whole and how each smaller model is related to the rest

- Create <u>Sprint Review 3</u> document (5% of Deliverable 3).

- Follow the template posted on Brightspace – This document is generally about a page.

- Update <u>Kanban board</u> (2.5% of Deliverable 3).

- Update <u>GitHub</u> with documents (2.5% of Deliverable 3).

- Push software development to GitHub as you progress. A new version of the application should be available on GitHub. The software should include all the dependencies and be downloadable. Update the README file (15% of Deliverable 3).
  - It is recommended to develop 1 to 2 of the features of your application.

- Create <u>Peer-Review Report 3</u> document (This document is graded separately and is submitted a day after the deliverable's due date).

Deliverable 4 – Sprint 4 (Documents and Software) – Due date: April 9th, 2023, 11:59 PM ET (15%)

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or other similar Kanban board applications.

- Update <u>Product Backlog</u> and <u>Sprint Backlog</u> documents for Sprint 4 (2.5 + 2.5 % of Deliverable 4).

- Update the <u>SRS</u> document based on the feedback given on deliverable 3 (if there were some mistakes in the previous version).

- Update the <u>Architecture Design (AD)</u> with its description based on the feedback given on deliverable 3, (if there were some mistakes in the previous version).

- Update <u>Use Case Models and Descriptions</u> document with potential new use cases and based on the feedback given on deliverable 3 (if there were some mistakes in the previous version) – (SRS + AD + UCM = 5% Maximum)

- Update the <u>Sequence Diagram</u> document based on the feedback given on deliverable 3 (10% of Deliverable 4).

- Update the <u>Domain Model</u> document based on the feedback given on deliverable 3 (10% of Deliverable 4).

- Create the **Detailed Design** document, with <u>design patterns</u> and design class diagram (35% of Deliverable 4 or if there is no update on the SRS + AD + UCM, then this will be 40%).
  - You need to pick 1 – 2 design patterns related to your application development, show how they are implemented with class diagrams, and explain and justify your decision.
  - You will also show an overview design of your application with a class diagram. Your DCD needs to match your development and shows how design patterns are used.
  - You may break your DCD into more than two diagrams if the diagram is too large.

- Create the **Configuration Management Plan** document (10% of Deliverable 4).
  - Follow the template posted on Brightspace.

- Create Sprint Review 4 document (5% of Deliverable 4).

- Update GitHub with documents (2.5% of Deliverable 4).

- Update Kanban board (2.5% of Deliverable 4).

- Push software development to GitHub as you progress. A new version of the application should be available on GitHub. The software should include all the dependencies and should be downloadable. Update the README file if needed (15% of Deliverable 4).
  - It is recommended to develop 1 to 2 new features of your application.

- Create Peer-Review Report 4 document (This document is graded separately and is submitted a day after the deliverable's due date).

Deliverable 5 – Sprint 5 (Documents and Software) – Due date: April 23rd, 2023, 11:59 PM ET (20%)

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or other similar Kanban board applications.

- Finalize Product Backlog and Sprint Backlog documents for Sprint 5 (2.5% + 2.5 % of Deliverable 5).

- Finalize the SRS document based on all the feedback and the changes made (if needed).

- Finalize the Architecture Design document based on all the feedback and the changes made (if needed).

- Finalize the Use Case Models and Descriptions document based on all the feedback and the changes made (if needed).

- Finalize the Sequence Diagram document based on all the feedback and the changes made (if needed).

- Finalize the Domain Model document based on all the feedback and the changes made (if needed) (SRS + AD+ UCM + SD + DM = 5% of Deliverable 5)

- Finalize the Configuration Management Plan document based on all the feedback and the changes made (5% of Deliverable 5)

- Finalize the <u>Detailed Design</u> document, with design patterns and design class diagram models based on the feedback given on deliverable 4 (10% of Deliverable 5 or 15% if the above documents do not need any update).

- Create and complete the **Test Plan** document: Unit testing (screenshots), use case testing, and acceptance testing (40% of Deliverable 5).
  - You will have about 5 – 10 use case testing.
  - You need to have 10 – 20 unit test cases for different methods, and classes in your program.
  - You should have screenshots of the tests, their results, and descriptions of the tests.

- Create a document with the <u>remaining tasks and the future implementation plan</u> (5% of Deliverable 5).

- Create <u>Sprint Review 6</u> document (5% of Deliverable 5).

- **(Optional – Extra Credit – 20% of Deliverable 5)** Perform a <u>Usability Study</u> and create relevant documents for it.
  - Follow the guidelines for the usability study from the *"COS420_Projects' Artifacts Details_Spring 2022"* Document.

- Finalize <u>Kanban board</u> (2.5% of Deliverable 5).

- Finalize <u>GitHub</u> with documents (2.5% of Deliverable 5).

- The final version of the application and its test cases should be available on GitHub. The software should include all the dependencies and should be downloadable. It also needs to include a README file (20% of Deliverable 5).
  - Besides finalizing your application, you need to have at least five of the unit tests implemented.

- Create <u>Peer-Review Report 5</u> document (This document is graded separately and is submitted a day after the deliverable's due date).

Deliverables: Detail of the deliverable's artifacts are given in the *"COS420_Projects' Artifacts Details_Spring 2022"* Document.

Presentations (15%): Refer to the syllabus for the dates and the *"Presentations' Details"* Document for details.