

COS420 – Introduction to Software Engineering – Project Description – Fall 2022

Project Overview:

For the project, the students must develop a mobile (Android) or web application. High-level topic suggestions 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 or surrounding community for eliciting and validating the requirements (i.e. the focus group) and for the usability study and acceptance testing.

The topics of the applications are **computing education and/or sustainability, within Maine**.

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 user story about 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.

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 first come first served.

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

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

- Design Your Team - 5%
- Deliverable 0 – 5%
- Deliverable 1 – 10%
- Deliverable 2 – 10%
- Deliverable 3 – 10%
- Deliverable 4 – 15%
- Deliverable 5 – 15%
- Deliverable 6 – 15%
- Proposal Presentation – 5%
- Final Presentation – 10%

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 on 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 week to discuss their progress and challenges, starting after Design Your Team is due. The meeting times will be specified on [this spreadsheet](#). PMs, reach out to the instructor via Discord to coordinate this.

General Guidelines for All of the Deliverables:

- 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 as a link to a Google Document, shared with all of your group members and the instructor's @maine.edu email address with full editor privileges. **Do not share it using "Anyone at UMaine with the link can..."**. Note that any revisions/resubmission will also be done in those same Google Documents, to make it easier to understand what has changed. (Note you can also upload a Word document to Google Documents).
- Submit each Deliverable on Brightspace. Only Brightspace submissions are accepted for grading.
 - For example, for Deliverable 0, your team submission should look like:
Project Description: <web link>
User Story document: <web link>
 - 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 instructor for improvement or changes, the document **must be updated and be resubmitted in the next Deliverable** with a short changelog of what feedback was addressed and where.
 - Only documents that received no feedback or are graded as 100% do not require updating (unless a major change happened that makes it irrelevant - this should be noted in the changelog).
- The file name should have the following format:
GroupName_Deliverable_i_ [Name of the Document] for $i = 0 \dots 6$.
 - If the name does not follow this, the team will lose **50%** of the total grade of the Deliverable.
- Starting with Deliverable 1, the individually done peer-review reports must be submitted the day after each Deliverable's due date at 6:00PM EST. These are individual submissions.

Required Artifacts for Each Deliverable:

Deliverable 0 (5%)

- [Project Description](#) document (which includes Group Name, Team Members, App Name, App description and general overview of similar apps) – Two to Four Pages Minimum (50% of Deliverable 0)
Note that you may propose up to 2 apps if you would like feedback on helping choose between them
- [User Story](#) document – About two pages or at least 15 – 20 user stories (50% of Deliverable 0)

Deliverable 1 – Sprint 1 (Documents/Software) (10%)

- Daily/Weekly Scrum – 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 Fall 2022](#) Document.
- Update Project Description document based on the feedback given on Deliverable 0 (5% of Deliverable 1)
- Update User Story document based on the feedback given on Deliverable 0 (5% of Deliverable 1)
- Create [Product Backlog](#) document for Sprint 1 (10 % of Deliverable 1)
- Create [Sprint Backlog](#) document for Sprint 1 (10 % of Deliverable 1)
- Create the first version of your [Software Requirements Specification \(SRS\)](#) document ([template](#)) (40% of Deliverable 1)
 - You are expected to have about 15 – 30 functional requirements and 10 – 15 non-functional requirements. This will be under section 4 for functional requirements. NFR go in section 5.
 - 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.
 - You need to have some mock-ups of the UI. This is here so you get started. Put these in Section 3.1
 - You do not need to fill out anything about use cases in this document.
 - You do not need to fill out other parts of the document that are not listed here.
- Create [Sprint Review 1](#) document (10% of Deliverable 1).
 - Follow the [template](#). This document is generally 1-2 pages.
- Update [GitHub](#) with documents (5% of Deliverable 1). Make sure to download these documents and check them into your Github repo as well; this is also good practice in Industry as a backup.
- Start the software development and push to GitHub as you progress (10% of Deliverable 1).

You do not need to have a working application or make any significant progress on developing your application. Focus on setting up the build environment. Your README file should detail the installation and build process for your application. Make sure it works on each person's machine.

(separate individual submission due one day after the Deliverable's due date)

- Create your [Peer-Review Report](#) document and fill it out. You will use this same document throughout the semester, adding rows for each team member for each Deliverable. You will submit a link to this via Brightspace, for each deliverable, to signal you have completed it (and to remind you you need to complete it).

Deliverable 2 – Sprint 2 (Documents/Software) (10%)
(note there is [more detail in the Brightspace assignment and the Rubric](#))

This is a summary of the work for Deliverable 2, to help you understand Deliverable 2 overall. It is meant to be read in the beginning of the course as you look over all the deliverables and understand the team project as a whole. See the [grading rubric on the Deliverable 2 assignment in Brightspace](#) for key details and to help your team create and divide up tasks for Deliverable 2.

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or other similar Kanban board applications. (5% of Deliverable 2)
 - Check the requirements for the daily scrum on the [“COS420 Projects' Artifacts Details Fall 2022”](#) Document.
- Create [Product Backlog](#) document for Sprint 2 (10 % of Deliverable 2)
- Create [Sprint Backlog](#) document for Sprint 2 (10 % of Deliverable 2)
- Update the [Software Requirements Specification \(SRS\)](#) document based on the feedback given on Deliverable 1 (15% 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 components and their relationships and interactions 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 Diagrams (you can use Miro or other diagramming software to make them).
 - 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 use case descriptions and diagrams for them.
 - One of your use case diagrams should contain most/all the use cases in your system.
- Create [Sprint Review 2](#) document (10% of Deliverable 2)
 - Follow the previous template.
- Update GitHub with documents (5% of Deliverable 2). Make sure to download these documents and check them into your Github repo as well; this is also good practice in Industry as a backup.

- Continue the software development and push to GitHub as you progress (10% of Deliverable 2).
Begin with the front-end UI.
Work on basic functionality most applications need, such as sign-up, login, etc; you should find open source libraries/packages for those features, instead of trying to write them from scratch.
There should be one command (e.g. npm) that runs all of the automated tests in your project.
- Create Peer-Review Report 2 document (This document is graded separately and is submitted individually a day after the Deliverable's due date on Brightspace).

Deliverable 3 – Sprint 3 (Documents and Software) – (15%)

This is a summary of the work for Deliverable 3, to help you understand Deliverable 3 overall. It is meant to be read in the beginning of the course as you look over all the deliverables and understand the team project as a whole. See the grading rubric on the Deliverable 3 assignment in Brightspace for key details and to help your team create and divide up tasks for Deliverable 3.

- 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 Fall 2022"*](#) Document.
- Update Product Backlog and Sprint Backlog documents for Sprint 3 (5 + 5 % of Deliverable 3)
- Update SRS document based on the feedback given on Deliverable 2 (if updated: 5% of Deliverable 3).
- Update the Architecture Design document with its description based on the feedback given on Deliverable 2 (if updated: 5% of Deliverable 3).
- Update Use Case Models and Descriptions 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 SRS did not need an update).
- Create Sequence Diagram document (35% of Deliverable 3).
 - You are expected to have about 7 – 15 analysis sequence diagrams and 3 – 7 design sequence diagrams.
 - Pick a few of the main features of your application and draw the sequence diagrams based on those. Prioritize drawing ones with the highest priority.
 - For each of the analysis sequence diagrams, start by writing a table describing the steps, with columns for subject, subject action (a verb), parameters, and object acted upon.

- Create Sprint Review 3 document (10% of Deliverable 3).
 - Follow the previous template.
- Update Kanban board (5% of Deliverable 3).
- Update GitHub with documents (5% of Deliverable 3). Make sure to download these documents and check them into your Github repo as well; this is also good practice in Industry as a backup.
- Push software development to GitHub as you progress. A new version of the application must be available on GitHub and it must run/compile. 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 major features of your application.
- Create Peer-Review Report 3 document (This document is graded separately and is submitted a day after the Deliverable's due date).

Deliverable 4 – Sprint 4 (Documents and Software) (15%)

This is a summary of the work for Deliverable 4, to help you understand Deliverable 4 overall. It is meant to be read in the beginning of the course as you look over all the deliverables and understand the team project as a whole. See the grading rubric on the Deliverable 4 assignment in Brightspace for key details and to help your team create and divide up tasks for Deliverable 4.

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or other similar Kanban board applications.
 - Update Product Backlog and Sprint Backlog documents for Sprint 4 (5 + 5 % of Deliverable 4).
- Update SRS document based on the feedback given on Deliverable 3 (if there were some mistakes in the previous version).
- Update the Architecture Design with its description based on the feedback given on Deliverable 3, (if there were some mistakes in the previous version).
- Update Use Case Models and Descriptions 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 + UCM + AD = 5% Maximum)
- Update Sequence Diagram document based on the feedback given on Deliverable 3 (10% of Deliverable 4 – If not update on SRS + UCM+AD then this will be 15%).

- Create the Domain Model document (25% of Deliverable 4).
 - You are expected to have the model of the domain with their 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 in to 2 – 4 smaller models.
 - You also need to have a description for the domain model as a whole and how each smaller model is related to the rest.
- Perform and create Focus Group document (10% of Deliverable 4).
Follow the guidelines for focus group in [*"COS420 Projects' Artifacts Details Fall 2022"*](#).
- Create Sprint Review 4 document (5% of Deliverable 4).
- Update GitHub with documents (5% of Deliverable 4). Make sure to download these documents and check them into your Github repo as well; this is also good practice in Industry as a backup.
- Update Kanban board (5% of Deliverable 4).
- Push software development to GitHub as you progress. A new version of the application should be available on GitHub and it must run/compile. 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) (15%)

This is a summary of the work for Deliverable 5, to help you understand Deliverable 5 overall. It is meant to be read in the beginning of the course as you look over all the deliverables and understand the team project as a whole. See the grading rubric on the Deliverable 5 assignment in Brightspace for key details and to help your team create and divide up tasks for Deliverable 5.

- Daily/Weekly Scrum – The progress should be reported on GitHub and ZenHub or etc.
- Update Product Backlog and Sprint Backlog documents for Sprint 5 (5 + 5 % of Deliverable 5).
- Update SRS document based on the feedback of Deliverable 4 (if there were some mistakes).

- Update Architecture Design based on the feedback of Deliverable 4 (if there were some mistakes).
- Update Use Case Models and Descriptions document with potential new use cases and based on the feedback given on Deliverable 4 (if there were some mistakes in the previous version).
- Update Sequence Diagram document based on the feedback given on Deliverable 4 (if there were some mistakes in the previous version). (SRS + UCM + AD+ SD = 5% Maximum)
- Update the Domain Model document based on the feedback given on Deliverable 4 (10% of Deliverable 5 or 15% if the other four documents mentioned above did not need an update).
- Create the Detailed Design document, with design patterns and design class diagram (35% of Deliverable 5).
 - 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 need to match your development and shows how design patterns are used.
 - You may break your DCD into more than two diagrams if it is too large.
- Create Sprint Review 5 document (5% of Deliverable 5).
- Update Kanban board (5% of Deliverable 5).
- Update GitHub with documents (5% of Deliverable 5). Make sure to download these documents and check them into your Github repo as well; this is also good practice in Industry as a backup.
- Push new version of software to GitHub as you progress. The software should include all the dependencies and be downloadable. Update the README file if needed (20% of Deliverable 5).
 - It is recommended to develop 1 to 2 new features and/or refactor some of your development to match your design patterns.
- Create Peer-Review Report 5 document (This document is graded separately and is submitted a day after the Deliverable's due date).

Deliverable 6 – Sprint 6 (Documents and Software) (15%)

This is a summary of the work for Deliverable 6, to help you understand Deliverable 6 overall. It is meant to be read in the beginning of the course as you look over all the deliverables and understand the team project as a whole. See the grading rubric on the Deliverable 6 assignment in Brightspace for key details and to help your team create and divide up tasks for Deliverable 6.

- 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 6 (5 + 5 % of Deliverable 6). •

Finalize 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 Use Case Models and Descriptions based on all the feedback and the changes made (if needed).
- Finalize 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).
- Finalize the Configuration Management Plan document based on all the feedback and the changes made (if needed) (SRS + UCM + AD+ SD + DM and CM = 5% Maximum)
- Finalize the Detailed Design document, with design patterns and design class diagram models based on the feedback given on Deliverable 5 (10% of Deliverable 6 or 15% if the above documents do not need any update).
- Create and complete Test Plan document: Unit testing (screenshots), use case testing, acceptance testing (35% of Deliverable 6).
 - You will have about 5 – 10 use case testing.
 - You need to have 10 – 20 unit test cases for different methods, classes in your program.
 - You should have screenshots of the tests, their results and descriptions of the tests.
- Create a document with remaining tasks and the future implementation plan (5% of Deliverable 6).

- Create Sprint Review 6 document (5% of Deliverable 6).

• **(Optional – Extra Credit – 20% of Deliverable 5)**

Perform Usability Study and create relevant documents for it.

- Follow the guidelines for the usability study from the [“COS420 Projects' Artifacts Details Fall 2022”](#) Document.

- Finalize Kanban board (5% of Deliverable 6).

- Finalize GitHub with documents (5% of Deliverable 6). Make sure to download these documents and check them into your Github repo; this is also good practice in Industry as a backup.
- 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 README file (20% of Deliverable 6).
 - Beside finalizing your application, you need to have at least five of the unit tests implemented.
- Create Peer-Review Report 6 document (This document is graded separately and is submitted a day after the Deliverable's due date).

Deliverables: Detail of each of the Deliverable's artifacts is given in the "*COS420 Project Artifacts Details Fall 2022*" Document.

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

Acknowledgements:

The project assignments are based on Professor Sepideh Ghanavati's work in prior versions of this course.