Submit each file as a link to a Google Document and share it with gregory.nelson@maine.edu with full editor privileges and all group members **(do not share it using "Anyone at UMaine with the link can...")**.

For example, your text submission should look like this:

* Project Description: <web link>
* User Story document: <web link>
* Product Backlog: <web link>
* Sprint Backlog: <web link>
* Software Requirements Specification (SRS): <web link>
* Sprint Review: <web link>
* Github Repo: <web link>
* Issue/Task Management Software: <web link> (for example, Github Issues; make sure my gregory.nelson@maine.edu Google account can access the software with full privileges)

| **Grading Rubric:**  ✅ Updated Project Description Document ( /5)   * Based on the comments from deliverable 0. * If the comments were not addressed, the document will be marked as 0. * If the comments are partially addressed, the document will be marked as 2.5.   Updated User story document ( /5)   * Based on the comments from deliverable 0. * If the comments were not addressed, the document will be marked as 0. * If the comments are partially addressed, the document will be marked as 2.5.   Product Backlog ( /10)   * Use [this template](https://docs.google.com/spreadsheets/d/1422Rr_2YRKFydVoz7m8bNor666bmfIwj/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true). * Product backlog and sprint backlog should match i.e. product backlog items in the sprint backlog need to have the same IDs etc. If they do not match deduct -2 points. * Make sure that all of your user stories are in the product backlog. For any missing user story deduct -1 point. * All columns based on the projects’ details document should be in the product backlog and must be filled out. For any missing column or missing information deduct -2 points.   Sprint Backlog 1 (/10)   * Use [this template](https://docs.google.com/spreadsheets/d/18AXZZxBCle6GE3ufvEgtNAKRVHvWBjZV/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true) * Should reflect the product backlog. That is, all the user stories that are assigned to Sprint 1 in the product backlog, should be shown in the Sprint Backlog 1.   + For any missing user story, deduct -2 points.   + If the Sprint Backlog 1 does not reflect the product backlog, deduct -5 points. * You should have other technical tasks in your backlog * You should also have other tasks in the backlog, like documentation, requirements, UI mockups, etc. As covered in class sessions, these are called non-technical stories. * All columns based on the projects’ details document should be in the product backlog and must be filled out. For any missing column or missing information deduct -2 points.   Github (/5)   * Contributions should be consistent and all students contribute. * If a student does not contribute to Github, the student will be graded as 0.   Kanban (/5)   * Kanban board should show progress. * If no progress is shown, the grade for this part will be 0.   Sprint Review (/10)   * Should follow the [template](https://docs.google.com/document/d/1QkzW3zqLNUdav9vv93DhmefqulpcLdep/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true). * This document is generally 1-2 pages. * Should be written for this past sprint / week (i.e. for Deliverable 1, Deliverable 1 and Revisions for Deliverable 0) * Use the template as an agenda for the sprint review meeting. Step through each point as a team, have a discussion and have someone take notes. The team should review the notes during the meeting in a shared Google Doc to make sure everyone is heard, and anyone can edit the notes. * For any missing item from the template, deduct 2 points. * Document should contain all prior Sprint Reviews as well, with the most recent at the top (the first sprint review happens in Deliverable 1)   SRS ( /40)   * Has about 15 – 30 functional requirements   + If the number of FR is below 15, -5 points   + Put them in Section 4 * Has about 10 – 15 non-functional requirements.   + If the number of NFR is below 10, -5 points   + Put them 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. * For each wrong requirement deduct -2 points. * For each conflicting/contradictory requirement, deduct -2 points. * If there are no UI mockups, deduct -20 points.   + This is here so you get started. Put these in Section 3.1 * Use cases are not required. * Other sections will be filled in in the next deliverable. Here’s an example for the eventual **Deliverable 2** [level of detail](https://docs.google.com/document/d/1WkqFkOZBiUtC19x8w7Zt7qIg8MJishZp/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true).   Software (/10)   * The implementation should have been started.   + If there is no code/HTML/scripts in Github, this part will be graded as 0. * Setting up the build environments and the necessary dependencies.   + You do not need to have any significant progress on developing your application. Focus on setting up the build environment.You can use create-react-app or otherwise create your project structure. * Your README file should detail the installation and build process for your application. Make sure it works on each person’s machine.   + Submit screenshot(s) or a single video showing the app running on each person’s machine   + If there is nothing, deduct -5 points.     Total = /100 | |
| --- | --- |
|  |  |
| **Deliverable 1 – Sprint 1 (Documents/Software) (10%)**   * [ ] Daily/Weekly Scrum – Progress should be reported on GitHub and ZenHub or other similar Kanban board applications.   o Check the requirements for the daily scrum on the [*COS420\_Projects' Artifacts Details \_Fall 2022*](https://docs.google.com/document/d/1qB1jKnywW-7qo_OnDW53GgONvhFmXruwztN4wGEjjHE/edit?usp=sharing)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](https://docs.google.com/spreadsheets/d/1422Rr_2YRKFydVoz7m8bNor666bmfIwj/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true) document for Sprint 1 (10 % of Deliverable 1) * [ ] Create [Sprint Backlog](https://docs.google.com/spreadsheets/d/18AXZZxBCle6GE3ufvEgtNAKRVHvWBjZV/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true) document for Sprint 1 (10 % of Deliverable 1) * [ ] Create the first version of your [Software Requirements Specification (SRS)](https://docs.google.com/document/d/1vvQOsI9HOc245abbvkirbRIGhm9VoLls/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true) document ([template](https://docs.google.com/document/d/1vvQOsI9HOc245abbvkirbRIGhm9VoLls/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true)) (40% of Deliverable 1)   o 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.  o 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.  o You need to have some mock-ups of the UI. This is here so you get started. Put these in Section 3.1  o You do not need to fill out anything about use cases in this document.  o You do not need to fill out other parts of the document that are not listed here.   * [ ] Create [Sprint Review 1](https://docs.google.com/document/d/1QkzW3zqLNUdav9vv93DhmefqulpcLdep/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true) document (10% of Deliverable 1).   o Follow the [template](https://docs.google.com/document/d/1QkzW3zqLNUdav9vv93DhmefqulpcLdep/edit?usp=sharing&ouid=108788083057609121398&rtpof=true&sd=true). 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](https://docs.google.com/spreadsheets/d/14o4rCgWxjn1sYmZsUQwFYF_3-0gaB21mgPigcL0G2P8/edit#gid=0) 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). | |