Deliverable 1

Start Assignment

- Due Sep 26 by 11:59p.m.
- Points 16
- Submitting a text entry box

For any general queries about the deliverable, use Piazza. For your team-specific or project-specific queries, contact your Mentor TA or Lead TA (pritish.mishra@mail.utoronto.ca (mailto:pritish.mishra@mail.utoronto.ca)).

The main goal of this deliverable is to give you a chance to:

1. Get a basic understanding about your partner organization, why this project matters to them, and build a relationship and process to work with them.
2. Refine and articulate the project idea and answer key questions.
3. Think about your collaborative work process.
4. Identify potential risks and mitigation strategies.
5. Agree on an MVP as a team (and with your partner if you have one).
6. Agree on the coding standard and guidelines to be followed.
7. Agree on IP terms with your partner (if you have one).

In addition to that, it gives us a chance to ensure that your plan is reasonable (i.e. neither too much, nor too little) and that everybody (team members, TA, partner) is on the same page in terms of the expectations.

DO NOT LEAVE THIS DELIVERABLE TO THE LAST MINUTE. COMPLETING THE DOCUMENT SUCCESSFULLY REQUIRES, POSSIBLY MULTIPLE, MEETINGS WITH EXTERNAL PARTIES AND REFINING CERTAIN DETAILS THAT REQUIRE CAREFUL THINKING.

Deliverables

The following files should be submitted to your team repo inside *D1* folder. The folder structure should be like /deliverables/D1 in the GitHub root folder.

1. Planning.md: (https://github.com/csc301-2025-f/deliverable-documents/blob/main/deliverables/D1/planning.md) This starter template should be included in the folder deliverable-1 when you create your repository.
2. **First mockup:** This can be done using prototyping software like Figma.com (http://figma.com/) , Marvel Pop (https://marvelapp.com/pop) , Draw IO (http://draw.io/) , Balsamiq (https://balsamiq.com/) , or any other software. The goal is that you discuss how your software looks with your partner and come to a clear understanding. The prototypes can change over the term.

3. **Create another '/deliverables/team/' folder and add the following items to it:**
    1. Team-[Team_Number]-[Team_Name].csv. (e.g., Team-5-Code Busters.csv). This will include a comma-separated file with the following values *in each line* and sorted ascending by the last name:
        I. Member number (1-6),
        II. utorid,
        III. Student name,
        IV. Githubid,
        V. Role(s)
        VI. Status,

        For example,
        "1, djorjani12, David Jorjani, jorjani, Scrum Master & Frontend Developer, enrolled."

        Note: You are responsible for maintaining this list if someone is added to your team OR drops the course (status = dropped) or becomes a CR/NCR student (status = CR).

        2. Meeting minutes: Your meeting minutes should go under */deliverables/team/minutes/* folder.

        3. Stakeholder information should go under */deliverables/team/Stakeholders.txt*. This should cover the following information:

        I. Who are your partners? What's their name, title, and email?
        II. Identify the primary partner contact (e.g., developer you are working with and meeting regularly) or secondary partner contact (e.g., executive director who you met once)
        III. Who is your TA and what's their email?

You should also have a readme.md file **in the main folder of your repo.** Readme.md will contain all the relevant information for your TA, partner or any new person to be able to run, use, and maintain your software and will evolve throughout the term. It will eventually include things like:

1. Where and how are tasks managed? (e.g., are you using Linear.app, Jira, GitHub Project...)
2. How to access and use the system
3. Key development requirements for installing and running the application
4. External dependencies and third-party software

NOTE: For D1, you are laying out the foundation to capture the required information. You may not have the details above yet.

Expectations

Here is what we expect (generally) from your deliverable:

1. You must join your team group on Quercus. You should go to the 'People' tab, then Project Groups' tab and search for your team number. You can find your team number in this **sheet** (https://www.notion.so/software-partnership/1f7aaedc973280fcaaf6db583f1f8898?v=1f7aaedc97328031b1f6000c2fdc4491&source=copy_link) (Check the tab "2025 Fall Teams"). These groups are created by your teaching team but you will need to sign up to them yourself.

2. You must create your team repository or join an existing team repository. You can use the **following link** (https://classroom.github.com/a/uT2MKyLA) . You will use this repository for the duration of the semester. Note that the template (.md document) for Deliverable 1 will automatically be included in your repository at '**/deliverables/D1**' directory by default. You must use the team name from Step 1. Note that '***project***' is automatically included in your Github team name. In other words, only enter <***team-number***>-<***partner***>**.** For example:

> If your team number is '2' and your partner organization is 'Athena Learning', then you should enter 2-Athena-Learning.

3. Once you have your repo, give your partner access to it so they can access your work at any point needed.

4. Your team must do one team-building activity of your choice in-person or online. This can be playing a game, doing a puzzle, getting ice cream or pizza, or anything else that allows you to get to know each other better.

5. You must have met your partner, ideally twice. If your partner agrees, use an AI tool (e.g., Granola, Fathom, Fireflies, etc.) to record and transcribe your meeting. You could use your first meeting to understand the problem and requirements of your partner, as well as discuss the licensing, IP, and software stack. You could use the second meeting to get feedback on your proposal. Document the minutes of this meeting and submit them as mentioned above in the Deliverables section.

6. In some cases, teams may have to work with partner repos that are outside of CSC301 org. For such cases, please fork the partner repo into the CSC301 org and work on the forked repo. If you face any issues in this, please email us or raise a question on Piazza. All student project repos must be part of the CSC301 org.

Note:

A. You must create this repository regardless of if you want to work in a partner organization or not. The teaching team needs to have access to your repo at all times during and after the course and be able to easily navigate to any other relevant repos without asking you. You may be penalized if your teaching team (TA, Lead TA, Instructor) don't have access to this repo.
B. An important point to keep in mind when writing user stories is that we expect user stories for projects that introduce new features to a product. If your project falls under the maintenance bracket, it's likely that you won't be able to write user stories. In this case, provide a clear

explanation of the goals that your project will accomplish.

C. Following the previous point, it is important to clarify and form a deeper understanding of the user stories. Preferably, use your second partner meeting to understand how the user stories fit into the overall project and what exactly you are expected to do. This will help you identify the scope of your project – development or maintenance.

7. For **planning.md:** Complete all sections of the template. You should remove the prompts, keep the questions, and add your answers. Provide clear, specific answers that show you have thought through the questions and discussed them.

8. For **Mockup:** You must provide an interactive prototype of your application (without backend or persistence) so you can show how your software would work. You can use platforms like v0.dev, lovable.dev, Figma or any other tool to create this. You can use prototypes provided by your partners (but they are not expected to give you this). Please specify if they have done it. Captures some user stories in the mockup (enough to provide the reader with an understanding of your product).

We expect all written deliverables to have a high-quality presentation:

- Clear and concise writing that is easy to read and understand
- No typos and/or grammatical errors.
- Template instructions and placeholders are removed.
- Make good use of images, videos, links and/or any other aid that makes your deliverable easier to read and understand.
- Using the markdown format appropriately - numerous GUI based markdown editors exist that make this easy to do

Submission

When you want to submit your work, you should tag a version of your repo as a "**deliverable-1**" release. You can learn more about **creating releases here. (https://docs.github.com/en/repositories/releasing-projects-on-github/managing-releases-in-a-repository)**

Please submit a link to your "deliverable-1" release on Quercus.

Note: Deliverable 1 is your first version. We don't expect you to get everything right. You will update all of the above as you move forward in your project. We don't grade you on the accuracy of your predictions but on the depth, relevance, and thoughtfulness of your responses.

Evaluation Rubric

Component	Mark	Notes
Mockup/ Architecture	4	• Did you provide an ***interactive*** mockup and clarify the partner's role in creating it? Static or paper prototypes will not receive a full mark. • Is your mockup accessible to your TA? • Is your mockup easy to follow? If your project involves no graphical user interface or is building over an existing project, you are still required to clarify how the user engages with your software and how your software engages with other components, integrations, and pieces. For example, if you are working on an API, expectations around requests and responses should be clear. Please provide an architecture of your software with major components showcasing the key workflow.
Product Spec(Q1-5)	5	• Did you cover all questions and prompts clearly and thoughtfully?
Teamwork(Q6-10) Organization (Q11-12)	5	• Did you cover all questions and prompts clearly and thoughtfully? • Did you cover items required under /team folder well? • Did you provide evidence of doing at least 1 team-building activity? • Did you clearly specify who does what? especially communications with partner?
Risks (Q13-14)	1	• Did you cover all questions and prompts clearly and thoughtfully?
Readme	0.5	• Do you have a well-organized and up-to-date readme?
Overall Quality, Clarity & Completeness	0.5	• Is your work easy to follow, read, and understand? • Did you follow the instructions properly? • Is your writing clear and high-quality? • Did you organize your answers well?

Your grade will be the weighted average of each one of the components above.

100%	**Excellent** • Excellent quality, organization, or clarity of work (e.g. easy to read, understand, and navigate, highly visual or complex artifacts, creative approach, etc.) when answering a question • Answered every single prompt clearly, concisely, and thoughtfully • Clear evidence of detailed discussions around the topics
80%	**Great** • Meets all expectations and addresses each point in the question template • The answer has both content and justification (i.e., what/how and why)
70%	**Adequate** • Some expectations were not fully met within the question either because the answer is unclear or no clear justification was provided for any point in the template (e.g., you mention what/how but not why) • One major (or a relatively small number of minor) point(s) were not addressed in the question
60%	**Below expectations** • More than one major point was missed or not addressed within the answer to the question • Multiple minor points lack clarity or clear justification
50%	**Minimal** • The provided answer does not cover any of the required details or is too high-level (i.e., an answer was provided but met no expectations)

0%	**Missing** (or extremely low quality) • Multiple (or all) points missed or very little content provided for the answer to the question

Examples

Some good examples of this deliverable from the previous course offering -

Project 1: **Mockup** (https://www.figma.com/design/1ZD8h7knalGfjoXKqp8oWO/Savi-AI-Designs?node-id=700-15841&node-type=section) , **ReadMe** (https://q.utoronto.ca/courses/395569/files/38736139?wrap=1) (https://q.utoronto.ca/courses/395569/files/38736139/download?download_frd=1) , **Planning Document** (https://q.utoronto.ca/courses/395569/files/38736138?wrap=1) (https://q.utoronto.ca/courses/395569/files/38736138/download?download_frd=1)

Project 2: **Mockup** (https://docs.google.com/document/d/1fA4bLSfhvU3tJZVde7es9Dum87uw9Tau7lo2bS_zTME/edit?tab=t.0) , **Planning Document** (https://q.utoronto.ca/courses/395569/files/38736149?wrap=1) (https://q.utoronto.ca/courses/395569/files/38736149/download?download_frd=1)

FAQs:

1. Do we have to stick to whatever we say we will do in D1 or can we change our plan?
A: You don't have to but do your best. That means do your best to come up with a detailed enough plan and cover as many gaps as you can. However, we do not hold you against what you said you will do in your D1 and you do not have to do everything you say. Doing the work will simply make the rest of the deliverables easier for yourself.

2. Are we judged based on the complexity of the user stories presented as part of D1?
A: Yes, and No. No, we don't explicitly grade the deliverable based on complexity. A project with more complex features doesn't get a higher grade than a project with less complex features. Having said that, we require you to be creative and thoughtful in planning the user stories. Keep in mind that the user stories you identify should give an impression of a project that needs 2-3 months of effort for a team of 5-6 developers. Hence, don't limit yourself with only low-hanging fruits.

3. Does one person submit this deliverable or should we all do it?
A: Submitting the URL to your teams GitHub repository by one person on behalf of their project team

is enough.

4. What should the project repository look like when we make a submission?
A: You should have a Readme.md file in the root folder, and **all the other documents, including the planning.md** file should go under the directory '**/deliverables/D1**'.

5. We don't have an answer to question 'X' yet. What should we do?
A: Share what you know and tell us what you are doing to come up with an answer. You can also identify this as a potential risk.

6. We couldn't meet with our partner twice before the due date. What should we do?
A: We strongly encourage you to best utilize the first meeting and try your best to meet again with the partner to review your plans. It's fine if this doesn't end up happening due to the partner's restrictions. As long as you have met with your partner once and planned the next meetings, you should be good.

7. After meeting with the partner, we want to change our project for XYZ reason. What should we do?
A: As soon as you know you want a different project, first talk with your partner and let them know why you want to change (if you are comfortable). Your next plan of action is to send an email to the email address specified on the handout and explain why you want to change your project/partner and what other projects your team is considering.
We strongly encourage you to email the teaching team as soon as possible if you want to do this to avoid further delay in your future deliverables.

8. There was a delay in meeting with the partner, and we need more time to submit our first deliverable. Can we get an extension?
A: Yes. Explain the situation in an email to your TA and request more time from them. If they accept, you are fine.

2024F-D1

Criteria	Ratings	Pts
Product Specification		5 pts
Teamwork & Organisation		5 pts
Readme		0.5 pts
Overall Quality		0.5 pts
Mockup		4 pts
Risks		1 pts
		Total Points: 16