

CSE 403, Project Phase 1b: SRS (40 points)

Software Requirements Specification and UI Prototype

The SRS milestone artifact is a set of documents related to your project's requirements and high-level user interface (UI) design. Your SRS must contain the following three (3) major items:

1. Requirements Outline

Create a **wiki page** within your Phabricator project named "Requirements Outline", whose content is roughly equivalent to **2-3 written pages in total length**, with concise yet descriptive answers to each of the following areas of questions:

- **Product Description:** What is your product? What generally does it do? What is the target audience you expect to use the product? Why is it interesting?
- **Software Toolset:** Will you use any software tools or libraries in addition to the required ones? How well do your group members already know the tools and languages you will be using, versus how much will they have to learn? Briefly explain why you chose these tools and languages, as well as why they are suitable for use on this project.
- **Feature Set:** What are your product's major core features? Include at least 3-4 major features your product will provide, along with at least 2 other minor features or aspects you hope to complete.
- **Alpha, Beta, V1 Breakdown:** You will have roughly 2 weeks to work on each real coding phase of the project: the initial Alpha version, the second Beta version, and the third "Version 1.0" (V1) completed version. How will you divide up the work between these phases? For each of the three phases, which features do you expect to be complete, which will be started, and which will not yet be started? Why have you chosen this particular division of the labor?
- **Risks, Cuts, Adjustments:** It's very difficult to predict exactly how much work a group will finish in a given amount of time. Describe at least three specific adjustments you will make if the project begins to fall behind schedule. Two of the adjustments you list can be feature cuts, but at least one must be some other change or cutback, such as changing specific areas of testing, adjusting your group dynamics or time schedule, etc.
- **Group Dynamics:** For the most part, your group organization is up to you. However, we require that you choose a single person to serve as your **Project Manager (PM)**. Who will be your project manager? What will be the other members' roles? Will you split up the work by features? Will each person work separately, or will you have sub-groups within the team? Why have you chosen these roles? If a disagreement arises, how will it be resolved?

Please format your wiki post nicely with headings and sections so that it is clear which content is answering which of the above question areas.

2. Use Cases

Create additional wiki pages on Phabricator representing the following **use case** documents:

- **Two (2) formal use cases** for scenarios you think are two of the most important to your product. They should be similar to Use Cases 1-3 and 5 from Cockburn's paper, following a similar syntax and actor/system interaction. The cases should include: primary actor, level, preconditions, minimal/success guarantees, a list of steps to the success scenario, a list of properly numbered and meaningful extensions, and a failure-handling remedy for each extension as appropriate. It is impossible to think of every possible failure case ahead of time. But you should list a reasonable set of extensions and remedies if reasonable ones exist. If you do not have a known remedy, your use case should state this and explain what will be done to investigate possible remedies.
- **One (1) casual use case** in paragraph form for one other scenario that you think is important to your product, perhaps of lower importance than the two you chose to depict formally. This use case should be similar to Use Case 4 from Cockburn's paper, describing the main success scenario first in paragraph form, then listing each extension and its remedy (if known) in a second paragraph.

3. User Interface Prototype

Submit diagrams containing rough sketches of your product's user interface. These diagrams should depict the major UI used to complete the use cases you submit. For example, if one of your use cases is to Purchase Stocks, you should draw the initial UI that is presented when the user wishes to purchase a stock, along with any other major windows, messages, etc. that appear as the user navigates through this use case. From looking at these UI diagrams, the customer should be able to clearly see how your product will be able to successfully complete each use case you are submitting. Briefly represent paths through some major extension(s) as appropriate in your UI diagrams.

Submit as many prototype screens as are necessary to cover your use cases; this must be **at least two (2) UI diagrams**. A "diagram" is a depiction of a complete major screen, web page, window, etc. of your system. The diagrams can be drawn by hand or computer, or can be screenshots of an actual prototype. (Since it often requires a lot of time to create an actual code prototype, we generally do not recommend doing so for this deliverable phase.)

If a window leads to a dialog box, drop-down box, etc., include it as a sub-diagram within one of your two overall diagrams. For full credit, there should be at least some dynamic content to your prototype, such as expanding lists, pop-up dialogs, etc. as described in the reading.

To help the customer understand how to "use" your prototype, label which items and sheets go together, for example, by putting numbers or names on the back of papers. For sub-diagrams such as pop-up windows, drop-down boxes, etc., it is helpful to put a corresponding number/name in the place that sub-diagram appears. If there is anything non-obvious about how your UI is used, include brief written instructions about how the user is expected to walk through the UI.

Your diagrams do not need to be beautiful to get full credit, but they should be legible and reflect some forethought about what options will need to be shown and how the user will use the software. Part of your grade comes from choosing appropriate UI elements to effectively accomplish the desired task.

(Your group will not be "locked in" to using the exact UI represented by these diagrams on your actual final project. The purpose of a prototype is to try out a set of user interface ideas, see how they work, and revise/modify the UI as needed.)

Submission and Grading:

Submit your Requirements Outline and Use Cases online by checking them into your version control system on **GitHub** by the due date. You may submit your UI prototype through GitHub as well if you like, but since it is likely to be on paper, you may also bring it to lecture on its due date to turn it in at the beginning of lecture.

Part of your grade will come from the plausibility, thoughtfulness, and level of detail of your work. For example, if you are listing features, take care not to forget important aspects that would reasonably be required of a project such as yours. When listing software tools, list a plausible set of tools for all aspects of the project and defend your choices. When listing group dynamics and a schedule, choose a group strategy that makes sense for your team and for the time given.

In use cases, sloppy or incomplete work often leads to deductions. We want to see that you have given due thought by choosing substantial use cases that are important to the core functionality of your product. You should also list a reasonable set of steps in the various scenarios that can occur in these use cases. Take care not to omit important steps or details. Make sure that the state of the system at the end of any path through the use case matches the guarantees that the use case claimed would occur. A portion of your grade will come from the format of your use cases. It should match that described in the reading in content structure, ordering, numbering, and style.

A small part of your grade comes from the looks or aesthetics of your documents. They do not need to be beautiful or excessively formatted, but your customers need to be able to read them and extract information from them. This means they should be clearly written in a professional writing style in complete sentences as appropriate, with proper spelling and grammar, clear wording, and formatted with a enough organization to present your ideas clearly to the reader.

Remember that part of your grade comes from having a meaningful in-person **interaction with your customer TA** before the phase is due to show your progress, ask questions, get feedback, and generally make sure you are on the right track.