Software Requirements Specification Document

Due Thursday, September 19, 2019 by 10:30 AM KST

Goal: your goal for this assignment is to create a System Requirements Specification to define your project. This document will be used as the basis for your full design and implementation.

In this assignment, you will document the major features and use cases your product will support, as well as a non-functional requirements and a high level overview of the user interface. Additionally, you will make an initial schedule and risk analysis.

This document is an extension of your initial project statement and will be a living document that gets modified as the course progresses.

Please note that I expect your product to be as usable as possible. This means having an intuitive user interface and appropriate tutorials, clarifying text, and help documentation as needed, while also being a robust system that can handle common error cases that users will face (e.g. invalid input, network connectivity issues).

Deliverables

Please create a Software Requirement Specification document by filling out the requested information in each section below.

1. Introduction
    - 1.1 Product description
        - This is a more detailed version of your problem statement and solution. What is the product, who is the target audience, and what problem does it solve? This should be several paragraphs long and sell why your product is needed.
    - 1.2 Existing alternatives
        - What alternatives exist for this problem? What are their strengths and weaknesses? How will your system be different?
    - 1.3 Scope
        - State the intended scope of the product, what it will do, who it will serve, and how generalizable it will be. Also note anything it will not do. Describe what platform(s) the product will run on and what range of devices it will support.
    - 1.4 Documentation & Support
        - What sort of documentation and support will you provide for users to understand your product? (e.g. user manual, first-run walkthrough, video tutorial, help files, help text throughout the app)
2. Features & Requirements
    - 2.1 What are your product’s major features?

- Briefly describe your product’s major features. Include at least 4 major features you’ll provide and at least 2 “stretch” features that you hope to do but could ship without if needed.

2.2 Use cases

- Create use cases for all the scenarios that you will be supporting in your product. Do not forget to include use cases for different actors on your system. How you specifically format the use cases is up to you, but these components must clear and included:
    - Use case title (short description of the use case)
    - Primary actor (the role/description of the primary actor)
    - Goal in Context (the goal of the primary actor in this context)
    - Priority (whether this use case working is *essential*, *nice to have*, or some other classification for your final product)
    - Scenario (an enumerated list of steps describing the primary actor interacting with the system that ends in a success scenario)
    - Extensions (write out the alternative scenarios that may occur, such as alternative success scenarios or potential errors and issues that may occur. Be sure to include any corrective actions taken by the system or the primary actor, if applicable)
    - You may optionally include the following, if applicable.
        - Secondary actors (any other people or systems that may be involved in the scenario beyond the primary actor and the main system)
        - Preconditions (conditions that must be true for this use case to occur)
        - Triggers (the action that starts the use case – this may be obvious from step 1 of the scenario and thus not need to be explicitly stated)
        - Open issues (any notes, concerns, or other materials regarding this use case)

2.3 User Interfaces

- What will the user interface be like? Create mockups of your product’s interface. **Show each distinct screen** of your project and annotate it with the use cases that are applicable on that screen.
    - Note that these mockups are not expected to be the final design, but to force you to think about the general layout and design of each screen now as you are considering all of your requirements. The diagrams do not need to be pretty or professionally designed (handwritten sketches are fine), the goal is just to make these clear and understandable to your team and also third parties (e.g. the “customer”).

2.4 Non-functional Requirements

- What are the non-functional requirements of your product? Any performance, security, or other requirements? Make sure that these requirements are written in a way that is verifiable.

3. Schedule and Risks

   3.1 Schedule

   - Create a rough schedule for what you plan to have completed for each project milestone (dates listed on the course schedule). For each milestone, list the title of each use case you expect to have completed and who you think will be working on each use case.
   - Note: You will not be graded on the accuracy of these predictions, though it is interesting to review later how accurate your predictions were.

   3.2 Risk Analysis

   - What are the major risks involved in completing your project? List at least 3 items. Why are you most worried about these risks? Describe how you plan to gather more information or reduce these risks (such as running experiments using some software components early on). Describe what you will do if you are unable to overcome a problem, such as you cannot get a 3rd party component to work? This can include cutting features, but feature cutting should not be your solution to all of your major risks.

   3.3 Acquiring Feedback

   - At what points will feedback from people outside of your team (potential users) be most useful and how you will get that feedback?

How to Submit

Make a “Documents” folder in your GitHub repository. Save the file as ProjectName-RequirementsSpec.pdf – Replace “ProjectName” with the name of your project. Make sure your project’s name and all group members’ names + their SBU emails are on the first page of the document.

Have one member of the team email the instructor a link to the document.

Grading (8 possible points)

Your document should be clearly and concisely written, with proper spelling and grammar. Use appropriate formatting to communicate the ideas clearly. Part of your grade will be on the thoughtfulness and appropriate level of detail in your work for the reader to understand the requirements of your proposed system.

- 1 point for creating a clear and concise Introduction section.
- 5 points for describing the major features, use cases, user interfaces, and non-functional requirements. Your requirements and use cases should be complete (covering all the requirements needed to make the product and the use cases should describe common variants scenarios and error conditions that your users will encounter), unambiguous and

consistent (no requirements or use cases should conflict with each other), and be verifiable. The UI Diagrams should be complete in covering all of the major screen types presented during your use cases and be clear to the reader.

- 2 points for having a reasonable schedule and thoughtful risk analysis plan.