CS2212 Group Project

Requirements Documentation

Winter 2023

Project Description

This assignment is part of the group project for CS2212. A full description of the project can be found here.

Purpose of the Assignment

The general purpose of this assignment is to read and explore the group project specification linked above, analyze and understand the requirements of this project, and produce documentation capturing your understanding of these requirements. In particular, this will give you experience in:

- Reading detailed software project specifications
- Writing requirements documentation
- Producing UML use case and activity diagrams

Due

This assignment is due on February 3rd, 2022 by 11:55PM through an electronic submission through the OWL site. If you require assistance, help is available online through OWL Help.

Late Penalty

Late assignments will be accepted for up to three days after the due date, so long as your team has late coupons remaining (every team starts the term with 6 late coupons). One late coupon is used for each day late (up to three days). After 3 days, or if your team has no late coupons remaining, you will receive a zero grade on this project component.

This assignment allows resubmissions. Only the most recent (the latest) submission will be considered and graded. This means a late penalty will be applied if you submit past the due date. The teaching assistants will only have access to your latest submission, so please ensure it is the correct one.

Group Effort

This project component is expected to be a group effort, with each member of the group contributing equally in a reasonable fashion. Feel free to discuss ideas with other groups; however, your submission must be the work of your own group (no documents or files should be shared between groups). If it is determined that you are guilty of cheating on the assignment, you could receive a grade of zero with a notice of this offence submitted to the Dean of your home faculty for inclusion in your academic record.

What to Hand in

Your submission, as noted above, will be electronically through OWL. You are to complete your documentation using our Confluence collaboration system and submit one or more exported PDF version(s) of this documentation as discussed below. This submission can be made as one large PDF or multiple smaller PDFs. You must ensure all diagrams in your PDF are legible and not cut off. Only one submission per group is necessary.

Further details on how to access and use our Confluence installation will be posted to OWL in the near future on the Project Software Page. However, you are free to start working on this component before you have access to Confluence and move your work there before submitting.

Useful Resources

If you would like to get a start on this project component right away you may need to read ahead a bit in our textbook and review some of the resources below until we cover these topics in-class.

- Wikipedia: UML Use Case
- Wikipedia: Activity Diagrams
- How to use Confluence: guides, tutorials, and demos
- Confluence: Export Content to Word, PDF, HTML and XML
- **Textbook Resources:**
  - Chapter 7
  - Chapter 8
  - Appendix 1

Assignment Task

In this assignment, you will craft requirements documentation for the group project for this course. Given the project specification that can be found here, you are to conduct a requirements analysis and produce documentation that captures your understanding of these requirements in further detail.

As noted above, you are to prepare your requirements documentation using our Confluence installation. For this course, each group is given its own project space in this wiki. For this assignment, you are to create a page under this space titled "Requirements Documentation" and you will proceed to create sub-pages under that page for the various sections of content outlined below. As necessary, you can create sub-pages of those sub-pages and so on, to provide a suitable structure to your documentation.

Until our confluence installation is setup, you are free to work on this document outside of Confluence and move it to the wiki before submitting.

Your documentation must consist of the following sections and content:

1. Main Page

The main page for this assignment, Requirements Documentation, should have the following content.

- A title for the project, and a sub-title indicating that this is the requirements documentation for that project.
- A tabular revision history for your document. It should be updated as people add to and edit your requirements documentation. It should be formatted roughly as follows:

Version	Date	Author(s)	Summary of Changes

- You must fill in the above table as you work on the document.
- A table of contents for the document. This can be created and maintained automatically by using the "Table of Contents" or "Children Display" macros in Confluence, and will enable ready access to the other sections of the documentation.

2. Introduction

This sub-page provides an introduction to your requirements documentation. As such, it should provide the following content. According to your preferences, this can either be delivered as sections on the introduction page, or as sub-pages.

- **Overview.** This section provides an overview of the problem being solved by this software and the requirements of the system. This should be a brief executive summary.
- **Objectives.** This section outlines the general objectives of the project.
- **References.** A list of references to other documents that might provide context or otherwise assist in the understanding of this document. Any referencing style can be used so long as it is used consistently in your documents.

Note that the overview and objects can be paraphrased from the original project specification provided to you. At least some rewording will be necessary in doing this; for example, to change references to "you" and "your" into something more appropriate as this is presenting things from your perspective. This specification document should be listed under your references, along with anything else you deem appropriate.

The overview and objectives sections should be at least a paragraph of text. Point form can be used if detailed, includes all required information, and is equivalent to a paragraph of text.

3. Domain Analysis

This sub-page provides an analysis of the **software domain** for the project. That is, the general field of business or technology in which the software will be used. By analyzing and better understanding the domain, you will have a better understanding of the needs and expectations of the software to function and deliver value to users in this domain. As such, it should at least answer the following questions; you can feel free to add further discussion as you deem necessary. According to your preferences, this can either be delivered as sections on the introduction page, or as sub-pages.

- What is the software domain for this project?
- Does this project fall into a specific subdomain or category within this software domain?
- What do we know about the domain?
- What are the common issues encountered in this domain (list at least 3)?
- What are the common solutions to the above issues in this domain?
- How can we use this domain understanding to improve or accelerate development of this project?

4. Functional Requirements

This sub-page discusses the functional requirements of the software project (in essence, what the software needs to do). As such, it should provide the following content. According to your preferences, this can either be delivered as sections on the introduction page, or as sub-pages.

- **Functionality to be delivered.** You must list and number each functional requirement for your project. Note that much of this information can be copied/paraphrased from the original project specification provided to you. Again, at least some rewording will be necessary in doing this, as discussed above. You may split the requirements listed in the specification into multiple requirements but at a minimum you should have one for each requirement listed in the specification. This must include at least one extra feature your team has chosen. If you can identify additional functional requirements, feel free to list them here as well.

- **Scenario model.** This section presents a model representing the functional requirements, in terms of actors, use cases, and activity diagrams. To carry out this modelling, you will need to analyze and understand these requirements, better preparing yourselves for what is to come. For this modelling, you need to document the following:

    - **Actors.** You will need to describe each actor (user, external system, etc.) involved in the use of the software. If your team is undertaking an extra feature that adds any additional actors, it must be included here. For each actor *(you should have at least 2)*, you will need to provide the following information in the tabular format given below. Keep in mind that actors are **not** parts of the system it's self (such as databases, GUI, etc.) and must be external entities such as a category of user, external systems, etc.

Description	*Brief description of the actor and its role in the system. This description should be no more than a small paragraph and should give the reader an understanding of the role of the actor in the organization.*
Aliases	*Any other names by which this actor may be known. A simple list is sufficient. Say 'None' if there are none.*
Inherits	*The ancestors for the actor. Some actors may be specialised types of other actors. A simple list of the names of the ancestors will suffice. Say 'None' if there are none.*
Actor Type	*Whether the actor is a person or an external system.*
Active/Passive	*Whether the actor is active or passive in the software system.*

- **Use cases.** Use cases describe specific usage scenarios of the software in question from the point of view of a particular actor. For each use case you identify, you will need to provide a formal definition of the use case. In most cases you should have at least one use case per functional requirement. For the definition of the use case, you will need to provide the following information, and can use this tabular format:

Name	*A name given to the use case.*
Primary actor	*The main actor in the use case; the use case is from their perspective.*
Secondary actors	*Other actors involved in the use case.*
Goal in context	*The overall scope of the use case, and provides a brief description of the use case and its purpose.*
Preconditions	*What is known to be true before the use case is initiated.*
Trigger	*What event or condition gets the use case started or invoked.*
Scenario	*A numbered series of steps that capture the narrative of the use case, outlining what the actors do in this use case and what happens as a result.*
Alternatives	*If alternative behaviour is possible at any of the steps outlined in the Scenario, it should be described here in a similar fashion. Say 'None' if there is no such alternative behaviour.*
Exceptions	*Identify potential issues or situations that may arise from the various steps of this use case.*
Priority	*One of: Highest, High, Medium, Low, Lowest. This is the priority your team is assigning to this use case (how important it is to implement, how essential it is to the function and goals of the application, etc.).*
...	*Additional headings and information may be provided as you deem necessary.*

- **Activity diagrams.** For each use case defined above, you must provide an activity diagram that provides a graphical representation of the flow of interaction within that specific scenario. In a way, this can better capture the steps of the scenario (and their alternatives) using standard flowcharting symbols.

- **Use Case Diagram:** You must provide at least one Use Case Diagram that includes all of the actors and use cases in your system. You may split this diagram into multiple diagrams so long as it is clear how they are connected. Don't over complicate the diagram and try to keep it clear to even stakeholders who may not be familiar with UML.

5. Non-Functional Requirements

This sub-page will list and number the non-functional requirements of the software project (in essence, how you and the software will go about doing things). Note that much of this information can be paraphrased from the original project specification provided to you. Again, at least some rewording will be necessary in doing this; as discussed above.

If you can identify additional non-functional requirements, feel free to list them here as well.

6. Summary

Provide a brief summary wrapping up your document, at least one paragraph in length. Also include a table of terms, notations, and acronyms you have introduced in your document with concise definitions. These definitions should be written at a level for a stakeholder that may not have a technical background.
