

Project #2. Software Requirements Specification (SRS)

Open: 2021-09-29 (Sep. 29th, 2021)

Due: 2021-10-17 (Oct. 17th, 2021)

In Idea proposal, your team selected a topic and proposed some ideas and solutions. Also, some functional and non-functional requirements were identified. In this project, your team will refine and specify those requirements to articulate goals and scope of your project based on the elicited requirements. Software Requirements Specification (SRS) will include (i) Title Page, (ii) Introduction, (iii) Overall Description, (iv) System Features, (v) Preliminary User Manual, (vi) Non-functional Requirements, and (vii) Acknowledgement.

(no more than 25 pages)

1. Title Page

Include the name of the document, team number, team members, and date. Also, prepare a Table of Contents (including page numbers and so on) of your document.

2. Introduction

Identify the product whose software requirements are specified in this document. Describe the scope of the product that is covered by this SRS. A quick description of the project, to communicate how your 'product' (application) supports the need of the stakeholder. (i) stakeholder & user scenario, (ii) the overall goal of the product, (iii) the most top-level requirements associated with the system goal, which you think the most important requirements from section 4-A.

3. Overall Description

A. Product Perspective

Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of product family, a replacement for a certain existing system, or a new, self-contained product. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.

B. Product Features

Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 4, so only a high-level summary is needed here. Organize the functions to make them understandable to any reader of the SRS.

C. Operating Environment

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or

applications with which it must peacefully coexist.

D. Assumption and Dependencies

List any **assumed factors** (as opposed to known facts) **that could affect the requirements stated in the SRS**. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. **The project could be affected if these assumptions are incorrect, are not shared, or modified.**

4. System Features

A. Functional Requirements

Specify **at least ten functional requirements** of your system in a table (See Appendix A for a guide). They could be defined hierarchically, and they **should include specific actors and functional features**. Also, the **specified requirements should be prioritized** based on your team's own criteria.

B. Use Case Diagram & Descriptions

Draw a **use case diagram** in which the **product boundary (scope of the product)** is **clearly identified**, and all the **major use cases** are shown.

Include actors relevant to your product, such as customer (user), external system, etc. **For each use case, give a priority for product development**. For **at least six** main use cases, describe them based on the use case description template provided (See Appendix B and C).

- If there are six main use cases in a diagram, then six use case description tables must be made.
- When you properly and correctly use *use-case relationships* such as <<include>>, <<extend>>, and <<generalization>>, there will be extra credits.

C. Sequence Diagrams

For **at least six** main use cases, which were selected for 4-B, **show expected dynamic behavior (ordered interactions) of your product using a sequence diagram** (See Appendix D for a guide).

- When you use *sequence fragments* effectively, there will be extra credits.

5. Preliminary User Manual

Identify who the user of the manual should be (state assumptions about the user). Give the user a good idea of the human interface. For a happy case scenario of your application, show some expected menus, command lines, interaction screens, and/or report formats.

- If you use some *images to describe preliminary GUI (Graphical User Interfaces)*, there will be extra credits.

6. Non-functional Requirements (Quality Attribute)

State important non-functional requirements. You may make reasonable assumptions about the scope of your project. You can refer to Appendix E for the detail about nonfunctional requirements.

7. Acknowledgement

Include authors of this document, by section, and editor of the document.

8. Presentation Video

Upload the presentation video to YouTube that summarizes your SRS document. We recommend you to include the overall introduction of your system, main functional requirements, use-case and sequence diagram, and main quality attributes. The presentation video needs to be finished within 15 minutes. Every member needs to present at least one important part in the presentation. We will make the submission page for the link to your presentation video in KLMS.

** Attached link may be helpful in making the presentation material.*

- <https://slideplayer.com/slide/763712/>
- <https://www.slideshare.net/Lynnylu/presentation1-20887067>

9. In-class Presentation

There will be a 3-minutes presentation on October. 27th (tentative). The details of the presentation will be notified in KLMS.

[Scoring criteria]

- **Readability:** readability of the descriptions and diagrams
- **Understandability:** understandability of descriptions and diagrams
- **Completeness:** completeness of the descriptions and diagrams
- **Consistency:** consistency between requirements, use-case diagram and description, and sequence diagram
- **Unambiguity:** unambiguity of the UML diagrams
- **Understanding of basic skills for UML modeling:** syntax and semantic understanding for UML

[Submission Deadline]

Submission Deadline: **Oct. 17th on KLMS.**

** Submit your report to KLMS (not by e-mail)*

** TAs will answer the questions about project 2 only if you asked before Oct. 15th.*

Appendix A. Functional Requirements

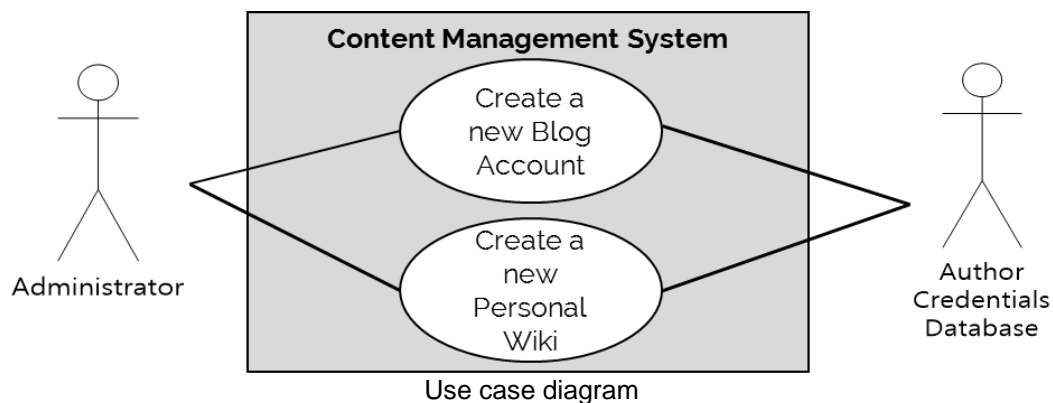
Hierarchy	Requirement ID	Requirement Description	Actors	Priority
User Management	A_1	The content management system shall allow an administrator to create a new blog account, provided the personal details of the new blogger are verified using the author credentials database.	Database	Critical
Data Management	B_1	The data from the sensors shall be collected in the database and analyzed.	Database, sensors	Critical

Appendix B. Use Case Description Template

Use case name	Name of a use case to be described
Related Requirements	Some indication as to which requirements this use case partially or completely fulfills.
Goal in Context	The use case's place within the system and why this use case is important.
Preconditions	What needs to happen before the use case can be executed.
Successful End Condition	What the system's condition should be if the use case executes successfully.
Failed End Condition	What the system's condition should be if the use case fails to execute successfully.
Primary Actors	The main actors that participate in the use case. Often includes the actors that trigger or directly receive information from a use case's execution.
Secondary Actors	Actors that participate but are not the main players in a use case's execution.
Trigger	The event triggered by an actor that causes the use case to execute.
Main Flow	The place to describe each of the important steps in a use case's normal execution.
Extensions	A description of any alternative steps from the ones described in the Main Flow.

Appendix C. Use Case Description Example

(<Create a new blog account> use case of Content Management System (CMS))

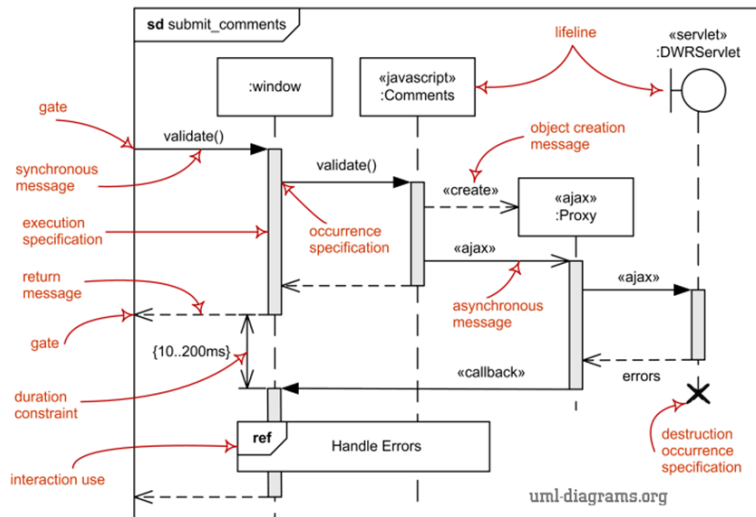


Use case name	Create a new blog account	
Related Requirements	Requirement A.1.	
Goal in Context	A new or existing author requests a new blog account from the Administrator.	
Preconditions	The system is limited to recognized authors and so the author needs to have appropriate proof of identity.	
Successful End Condition	A new blog account is created for the author.	
Failed End Condition	The application for a new blog account is rejected.	
Primary Actors	Administrator.	
Secondary Actors	Author Credentials Database.	
Trigger	The Administrator asks the CMS to create a new blog account.	
Main Flow	Step	Action
	1	The Administrator asks the system to create a new blog account.
	2	The Administrator selects an account type.
	3	The Administrator enters the author's details.
	4	The author's details are verified using the Author Credentials Database.
	5	The new blog account is created.
	6	A summary of the new blog account's details are emailed to the author.
Extensions	Step	Branching Action
	4.1	The Author Credentials Database does not verify the author's details.
	4.2	The author's new blog account applications is rejected.

Appendix D. Sequence Diagram

• (Refined) Sequence Diagram

- To draw refined sequence diagrams.
 - At this sprint, it is important to consider your team's design/development patterns & methods & styles that you will actually use.
 - If there were ambiguous participants or interactions (i.e., messages/ordering), your team should describe them as clearly (i.e., unambiguously) as possible by the refinement.



Appendix E. Non-functional Requirements

1. User Interface and Human Factors

- What type of user will be actually using the system?
- Will more than one type of user be using the system?
- What sort of training will be required for each type of user?
- Is it particularly important that the system be easy to learn?
- Is it particularly important that users be protected from making errors?
- What sort of input/output devices for human interface are available, and what are their characteristic?

2. Documentation

- What sorts of documentation are required?
- What audience is to be addressed by each document?

3. Hardware Considerations

- What hardware is the proposed system to be used on?
- What are the characteristics of the target hardware, including memory size and auxiliary storage space?

4. Performance Characteristics

- Are there any speed, throughput, or response time constraints on the system?
- Is there size of capacity constraints on the data to be processed by the system?

5. Error Handling and Extreme Conditions

- How should the system respond to input errors?
- How should the system respond to extreme conditions?

6. System Interfacing

- Is input coming from systems outside the proposed system?
- Is output to go to systems outside the proposed system?
- Are there restrictions on the format or medium that must be used for input or output?

7. Quality Issues

- What are the requirements for reliability?
- Must the system trap faults?
- Is there a maximum acceptable time for restarting the system after a failure?
- What is the acceptable system downtime per 24-hour period?
- Is it important that the system be portable (able to move to different hardware or operating system environments)?

8. System Modifications

- What parts of the system are likely candidates for later modification?
- What sorts of modifications are expected?

9. Physical Environment

- Where is the target equipment to operate?
- Will the target equipment be in one or several locations?
- Will the environmental conditions in any way be out of the ordinary (for example, unusual temperatures, vibrations, magnetic influences, and so on)?

10. Security Issues

- Must access to any data or the system itself be controlled?
- How often will the system be backed up?
- Who will be responsible for back up?
- Is physical security an issue?

11. Resources and Management Issues

- What materials, personnel, computer time, and other resources will be required to build, install, and maintain the system?
- What skills or knowledge must the developers have to develop the system?
- What are the proposed intermediate and final deadlines for system development costs?

- What is the proposed budget for hardware, personnel, and other development costs?
- Who is responsible for system installation?
- Who will be responsible for system maintenance?