

Deliverable #1 Template

SE 3A04: Software Design II – Large System Design

1 Introduction

- This section of the SRS should provide an overview of the entire SRS.

1.1 Purpose

- Specify the purpose of the SRS
- Specify the intended audience for the SRS

1.2 Scope

- Identify the software product(s) to be produced, and name each (e.g., Host DBMS, Report Generator, etc.)
- Explain what the software product(s) will do (and, if necessary, also state what they will not do)
- Describe the application of the software being specified, including relevant benefits, objectives, and goals
- Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist

1.3 Definitions, Acronyms, and Abbreviations

- Provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS
- This should be in alphabetical order.

1.4 References

- Provide a complete list of all documents referenced elsewhere in the SRS
- Identify each document by title, report number (if applicable), date, and publishing organization
- Specify the sources from which the references can be obtained
- Order this list in some sensible manner (alphabetical, or something else that makes more sense)

1.5 Overview

- Describe what the rest of the SRS contains
- Explain how the SRS is organized

2 Overall Description

- This section of the SRS should describe the general factors that affect the product and its requirements.
- It does not state specific requirements.
- It provides a *background* for those requirements and makes them easier to understand.

2.1 Product Perspective

- Put the product into perspective with other related products, i.e., context
- If the product is independent and totally self-contained, it should be stated here
- If the SRS defines a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to the functionality of the software being developed. Identify interfaces between that larger system and the software to be developed.
- A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful

2.2 Product Functions

- Provide a summary of the major functions that the software will perform.
 - **Example:** An SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.
- Functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time
- Present the functions in a list format - each item should be one function, with a brief description of it
- Textual or graphical methods can be used to show the different functions and their relationships
 - Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables

2.3 User Characteristics

- Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise
- Since there will be many users, you may wish to divide into different user types or personas

2.4 Constraints

- Provide a general description of any constraints that will limit the developer's options

2.5 Assumptions and Dependencies

- List any assumptions you made in interpreting what the software being developed is aiming to achieve
- List any other assumptions you made that, if it fails to hold, could require you to change the requirements
 - **Example:** An assumption may be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

2.6 Apportioning of Requirements

- Identify requirements that may be delayed until future versions of the system

3 Use Case Diagram

- Provide *one* use case diagram for the most important Business Event.
- The text of all use cases will be specified under "Highlights of Functional Requirements"

4 Highlights of Functional Requirements

- Specify the "use cases" organized by Business Event. (The Global Scenario is what you might think of as a use case). Be sure to consider Business Events that aren't just triggered by users with goals (e.g. something happens in the environment that your system needs to respond to)
- Your focus should be on what the system needs to do, not how to do it. Specify it in enough detail that it clearly specifies what needs to be accomplished, but not so detailed that you start programming or making design decisions.
- Keep the length of each use case (Global Scenario) manageable. If it's getting too long, you need to condense your steps and give a name to what's accomplished by that sequence of steps. (e.g. "Authenticate user" in one line, instead of a list of steps of how to; that's a design decision anyways)
- You are *not* specifying a complete and consistent set of functional requirements here. (i.e. you are providing them in the form of use cases/global scenarios, not a refined list). For the purpose of this project, you do not need to reduce them to a list; the global scenarios format is all you need.

Below, we organize by Business Event.

BE1. Business Event name

VP1.1 Viewpoint name

- S_1 : Initial response of the system to the Business Event
- E_1 : Reaction of the environment to S_1
- S_2 : Response of the system to E_1
- E_2 : Reaction of the environment to S_2
- ...
- S_n : Response of the system to $E_{(n-1)}$
- E_n : Reaction of the environment to $E_{(n-1)}$
- $S_{(n+1)}$: Final response of the system concluding its function regarding the Business Event

VP1.2 Viewpoint name

- S_1 : Initial response of the system to the Business Event
- E_1 : Reaction of the environment to S_1
- S_2 : Response of the system to E_1
- E_2 : Reaction of the environment to S_2
- ...
- S_k : Response of the system to $E_{(k-1)}$
- E_k : Reaction of the environment to $E_{(k-1)}$
- $S_{(k+1)}$: Final response of the system concluding its function regarding the Business Event

VP1.3 ...

VP1.4 ...

VP1.5 ...

...

Global Scenario of *Business Event Name*: It is the scenario corresponding to the integration of all the above scenarios from the different Viewpoints of the Business Event BE1.

- S_1 : Initial response of the system to the Business Event
- E_1 : Reaction of the environment to S_1
- S_2 : Response of the system to E_1
- E_2 : Reaction of the environment to S_2
- ...
- S_m : Response of the system to $E_{(m-1)}$
- E_m : Reaction of the environment to $E_{(m-1)}$
- $S_{(m+1)}$: Final response of the system concluding its function regarding the Business Event

BE2. Business Event name

VP1.1 Viewpoint name

- S_1 : Initial response of the system to the Business Event
- E_1 : Reaction of the environment to S_1
- S_2 : Response of the system to E_1
- E_2 : Reaction of the environment to S_2
- ...
- $S_{n'}$: Response of the system to $E_{(n'-1)}$
- $E_{n'}$: Reaction of the environment to $E_{(n'-1)}$
- $S_{(n'+1)}$: Final response of the system concluding its function regarding the Business Event

VP1.2 Viewpoint name

- S_1 : Initial response of the system to the Business Event
- E_1 : Reaction of the environment to S_1
- S_2 : Response of the system to E_1
- E_2 : Reaction of the environment to S_2
- ...
- $S_{k'}$: Response of the system to $E_{(k'-1)}$
- $E_{k'}$: Reaction of the environment to $E_{(k'-1)}$
- $S_{(k'+1)}$: Final response of the system concluding its function regarding the Business Event

VP1.3 ...

VP1.4 ...

VP1.5 ...

...

Global Scenario of *Business Event Name*: It is the scenario corresponding to the integration of all the above scenarios from the different Viewpoints of the Business Event BE2.

- S_1 : Initial response of the system to the Business Event
- E_1 : Reaction of the environment to S_1
- S_2 : Response of the system to E_1
- E_2 : Reaction of the environment to S_2
- ...
- $S_{m'}$: Response of the system to $E_{(m'-1)}$
- $E_{m'}$: Reaction of the environment to $E_{(m'-1)}$
- $S_{(m'+1)}$: Final response of the system concluding its function regarding the Business Event

5 Non-Functional Requirements

- For each non-functional requirement, provide a justification/rationale for it.

Example:

SC1. *The device should not explode in a customer's pocket.*

Rationale: Other companies have had issues with the batteries they used in their phones randomly exploding [insert citation]. This causes a safety issue, as the phone is often carried in a person's hand or pocket.

- If you're making a guess because you couldn't really talk to stakeholders, you can say "We imagined stakeholders would want...because..."
- Each requirement should have a unique label/number for it.

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF-A1. The colour scheme of the application should match the branding colours of the company

Rationale: For marketing purposes, the colour scheme within the app need to match the branding colours of the company so the brand becomes well-recognized and there is no confusion among users about the brand.

5.1.2 Style Requirements

LF-S1. The app should have an attractive, responsive UI

Rationale: According to <https://www.ishir.com/blog/9633/why-design-is-the-most-important-factor-in-a-mobile-app-development.htm>, users are more likely to use and come back to attractive, responsive mobile apps which helps increase the user base and branding power of the app.

LF-S2. The app should have a minimalist design

Rationale: Minimalist designed mobile applications have faster loading times, easier navigation within the app, less maintenance and time spent on design/frontend and allow businesses to more clearly give messages to users within the application. <https://ugem.design/blog/minimal-app-design>

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH-EOU1. The product should be easy to use for people aged 18 years or older

Rationale: The intended audience of this application is adults who need to carpool, so anyone 18 or older should find the app easy to understand and use. People under this age may use the app but most likely their rides would be booked by parents/guardians, so the focus is on this age range.

UH-EOU2. The user should not encounter any unclear prompts from which they do not know how to proceed

Rationale: If the app is prompting the user to do something, that prompt must be clear and easy to understand so it is easy for the user to use the app.

UH-EOU3. The user should not encounter any errors while using the app, and if they do, what to do next should be clear

Rationale: The app should not show the user any errors while they are using it to prevent confusion and to make the app easy to use. On the chance an error does appear to the user, it should be clear on what the user has to do next to remove the error or continue. If the error is generic and confusing to understand, the app is no longer easy to use and leaves the user in confusion.

5.2.2 Personalization and Internationalization Requirements

UH-PI1. The product will operate in both English and French and the user will be able to set their preferred language

Rationale: The app will be available only in Canada, so it needs to contain support for both the national languages of Canada and allow the user to choose their preferred language

UH-PI2. The product will allow the user to set their preference for dark mode and light mode

Rationale: To make the app more personal, users should be able to customize the appearance to their liking. Its important for users to enjoy the appearance of the app so allowing them to personalize dark mode or light mode will help contribute to that.

5.2.3 Learning Requirements

UH-L1.

5.2.4 Understandability and Politeness Requirements

UH-UP1.

5.2.5 Accessibility Requirements

UH-A1.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR-SL1.

5.3.2 Safety-Critical Requirements

PR-SC1.

5.3.3 Precision or Accuracy Requirements

PR-PA1.

5.3.4 Reliability and Availability Requirements

PR-RA1.

5.3.5 Robustness or Fault-Tolerance Requirements

PR-RFT1.

5.3.6 Capacity Requirements

PR-C1.

5.3.7 Scalability or Extensibility Requirements

PR-SE1.

5.3.8 Longevity Requirements

PR-L1.

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE-EPE1.

5.4.2 Requirements for Interfacing with Adjacent Systems

OE-IA1.

5.4.3 Productization Requirements

OE-P1.

5.4.4 Release Requirements

OE-R1.

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS-M1.

5.5.2 Supportability Requirements

MS-S1.

5.5.3 Adaptability Requirements

MS-A1.

5.6 Security Requirements

5.6.1 Access Requirements

SR-AC1.

5.6.2 Integrity Requirements

SR-INT1.

5.6.3 Privacy Requirements

SR-P1.

5.6.4 Audit Requirements

SR-AU1.

5.6.5 Immunity Requirements

SR-IM1.

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP-C1.

5.7.2 Political Requirements

CP-P1.

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR-COMP1.

5.8.2 Standards Requirements

LR-STD1.

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

IMPORTANT NOTES

- Be sure to include all sections of the template in your document regardless whether you have something to write for each or not
 - If you do not have anything to write in a section, indicate this by the *N/A*, *void*, *none*, etc.
- Uniquely number each of your requirements for easy identification and cross-referencing
- Highlight terms that are defined in Section 1.3 (**Definitions, Acronyms, and Abbreviations**) with **bold**, *italic* or underline
- For Deliverable 1, please highlight, in some fashion, all (you may have more than one) creative and innovative features. Your creative and innovative features will generally be described in Section 2.2 (**Product Functions**), but it will depend on the type of creative or innovative features you are including.