Report on Software Requirements Specifications for Chemistry Calculator

Submitted to

Dipok Chandra Das

Assistant Professor, IIT, NSTU

Submitted by

Md. Mahabub Alam (ASH1825003M)

Abrar Hossain (ASH1825005M)

Nishat Tasnim Tamanna (BKH1825006F)

Md. Faisal Ahmed (ASH1825015M)

Moon Moon Das (BKH1825027F)

Introduction

Software Requirements Specification (SRS) is a document that specifies the requirements and constraints of a software system to be developed. The SRS document typically includes the functional and non-functional requirements of the software, the system architecture and design, user interfaces, data management, and other relevant information. The SRS document serves as a communication bridge between the stakeholders and the development team, ensuring that everyone has a clear understanding of what the software system should do and how it should behave.

In this report, we will discuss how we make the SRS from a project named **Chemistry Calculator**, which provides many functionalities like-

- Equation balancing
- Electron configuration
- Molar mass
- Calculations of concentration and titration

In this document the required process are performed by following these steps-

- Set up the Environment
- Run the project
- Gather Functional & Non-functional Requirements from System Architecture and Design
- Use Case Diagram & Description
- Activity Diagram

Set up the Environment

Language: JAVA

IDE: An IDE (Integrated Development Environment) contains a code editor, a compiler or interpreter, and a debugger, accessed through a single graphical user interchange (GUI). Our IDE is Eclipse and IntelliJ IDEA.

Run the project

After set up the environment we run the project from **.jar (chemistry-calculator-main.jar)** file.

Gather Functional & Non-functional Requirements from System Architecture and Design

The functional requirements of the software system, which describe what the system should do, such as specific actions, functions, and services that the system should provide. The non-functional requirements of the software system, such as performance, security, reliability, maintainability, usability, and any other quality attributes that are important to the stakeholders. We explore the project to gather requirements.

- Understand the business goals
- Analyze existing documentation
- Observe user workflows
- Analyze user data
- Identify use cases
- Create functional specifications

Use Case Diagram & Description

After identifying the different use cases & actors for the system and the key features required to support them. Use cases can help us identify the functional requirements needed to support each use case. Then we draw a use case diagram and write descriptions based on the use cases we found.

Activity Diagram

Create an activity diagram for each use case, showing the different activities that need to be performed.

Identify the inputs and outputs: Identify the inputs and outputs for each activity from the project. Inputs are the data required for an activity to be performed, while outputs are the results of an activity.

Determine the conditions: Determine the conditions that need to be met for an activity to be performed, such as data validation.

Identify the decision points: Identify the decision points in the project, where the system needs to make a decision based on the input data.

Review and refine: Review and refine the activity diagrams to ensure they accurately represent the functionality required by the system.

Conclusion

In conclusion, creating a Software Requirements Specification (SRS) for an old software project without any proper documentation can be a challenging but necessary task. The process of creating an SRS involves several important steps, such as gathering information about the project, analyzing user needs and requirements, defining functional and non-functional requirements, and creating a detailed design document. By following these steps, it is possible to create a comprehensive SRS that accurately captures the needs and requirements of the project. This SRS can be used as a blueprint for future development efforts, helping to ensure that the software is developed according to user needs and specifications. Moreover, the creation of an SRS for an old software project can provide several benefits, such as improving the maintainability and extensibility of the software, making it easier to identify and fix bugs, and enabling easier collaboration between development teams and stakeholders.

In short, while creating an SRS for an old software project may require a significant investment of time and resources, the benefits of having a comprehensive and well-documented set of requirements can make it well worth the effort.