

# **Software Requirements Specification for Chemistry Calculator**

Course Title: Software Maintenance Lab  
Course Code: SE4204

## Team Members

Abdullah An-Noor (ASH1825001M)  
Fazle Rabbi (ASH1825004M)  
Rahat Uddin Azad (ASH1825022M)  
Saifur Rahman (ASH1825031M)  
Anwar Kabir Sajib (ASH1825038M)

## Submitted To

Dipok Chandra Das  
Assistant Professor  
IIT, NSTU

Date of Submission  
02-March-2023

## Table of Contents

CHAPTER 1 .....	3
1.1 Introduction.....	3
1.2 Project title.....	3
1.3 Description.....	3
1.4 Objectives .....	4
1.5 Motivation.....	4
1.6 Features .....	4
1.7 Scope.....	4
1.8 Project Stakeholders.....	5
1.9 Tools and Technology.....	5
1.10 Configuration Management Plan .....	6
CHAPTER – 2 .....	7
2.1 Functional System Requirements: .....	7
2.2 Non-functional Requirements .....	8
2.2.1 Data Requirements.....	8
2.2.2 Performance Requirements .....	9
2.3 Dependability Requirements.....	10
2.3.1 Reliability & Availability Requirements .....	10
2.3.2 Robustness or Fault-Tolerance Requirements .....	10
2.3.3 Safety-Critical Requirements.....	11
2.3.4 Maintainability & Supportability Requirements .....	11
2.3.5 Maintainability Requirements.....	11
2.3.6 Supportability Requirements.....	11
2.3.7 Adaptability Requirements .....	12
2.4 Security Requirements .....	12
2.4.1 Access Requirements .....	12
2.3.2 Integrity Requirements.....	12
2.5 Usability and Human-Interaction Requirements.....	12
2.5.1 Ease of Use Requirements .....	12
2.5.2 Personalization and Internationalization Requirements .....	13
2.5.3 Understand ability and Politeness Requirements.....	13
2.5.4 Accessibility Requirements.....	13
2.5.5 Training Requirements.....	13
2.6 Look and Feel Requirements .....	13
2.6.2 Style Requirements .....	14

2.7 Operational and Environmental Requirements .....	14
2.7.1 Expected Physical Requirements .....	14
2.7.2 Requirements for Interfacing with Adjacent Systems .....	14
2.7.3 Release Requirements .....	15
2.8 Legal Requirements .....	15
2.8.1 Compliance Requirements .....	15
2.8.2 Standards Requirements.....	15
2.9 Elicitation Technique .....	16
CHAPTER 3 .....	17
DIAGRAM OF ENTITY RELATIONSHIP .....	17
CHAPTER 4 .....	18
USECASE DIAGRAM.....	18
4.1 Usecase Description.....	19
CHAPTER 5 .....	23
DIAGRAM OF ACTIVITY .....	23
5.1 Balance Equation Activity Diagram .....	23
5.2 Equivalence Check History Activity Diagram.....	24
5.3 Determine Concentration Activity Diagram .....	24
5.4 View Atomic Symbol Details Activity Diagram .....	25
5.5 Calculate Molar Mass Activity Diagram .....	26
5.6 Calculate Molar Mass With Percentage Activity Diagram.....	27
5.7 Find Unknown Value In Titration Activity Diagram.....	28
CHAPTER 6 .....	29
TRACEABILITY MATRIX.....	29
6.1 Introduction.....	29
6.2 Why RTM is Important? .....	29
6.3 Use Cases .....	29
6.4 Test Cases .....	30
6.5 Requirements Traceability Matrix Table .....	31

# **SOFTWARE REQUIREMENTS SPECIFICATION**

## **FOR**

## **CHEMISTRY CALCULATOR**

### **CHAPTER 1**

#### **1.1 Introduction**

The Chemistry Calculator project aims to develop a software application that can perform various chemistry-related calculations. The software will serve as a useful tool for students, researchers, and professionals who need quick and accurate calculations related to chemistry. The Chemistry Calculator project will follow the software development life cycle to ensure the quality of the software application. This project will start with the development of the Software Requirements Specification (SRS) document, which will capture all the necessary information and specifications required for the project.

This document will describe the functional and non-functional requirements of the software, including the features, user interface, performance, security, and other technical specifications. The document will also outline the scope, goals, and objectives of the project, along with the potential risks and challenges that may arise during the development process.

The SRS document will serve as a blueprint for the entire software development process, including design, implementation, testing, and maintenance. It will also act as a communication channel between the development team, stakeholders, and clients to ensure that the project is on track and meets all the requirements. Overall, the Chemistry Calculator project aims to develop a software application that is user-friendly, efficient, and accurate in performing various chemistry calculations. The SRS document will ensure that all the requirements are captured and documented, and the development process is streamlined to meet the project's

#### **1.2 Project title**

Chemistry Calculator.

#### **1.3 Description**

Chemistry is the Science of Matter where you learn about a matter's composition & properties. How a matter is formed and what happens when it reacts with another matter or substance. This Chemistry calculator will help you to solve equations and problems related to Electron Combination, Molarity, Mass, Percent of Composition, Balance Reaction, molecular mass, titration setup etc. In Chemical Engineering, Chemist need to find out autonomic mass, reaction balance. This also need for pharmacist.

## **1.4 Objectives**

- Learn OOP using JAVA.
- Algorithm implementation.
- Create desktop application.
- To work in group.

## **1.5 Motivation**

It's quite certain that a chemist and a pharmacist have to calculate molarity, balance, reaction and electronic format for various purposes. Even though these calculations are very much complex, these calculations can be done easily through Chemistry calculator. Our main motto is to build a chemistry calculator by which we will calculate autonomic mass, electronic format, balance reaction and titration accurately and smoothly. Through this building session, we are going to ensure our self-learning which is inevitable.

## **1.6 Features**

- Electron format.
- Mass.
- Percentage of Composition.
- Reaction Balance.
- Molecular Mass.
- Molarity.
- Titration.

## **1.7 Scope**

- This application will be available on PC

## 1.8 Project Stakeholders

The stakeholders of the "Chemistry Calculator" project are identified below:

1. **Students:** Students are the primary users of the software, as they will be using the software to solve chemistry problems, calculate chemical equations and reactions, and perform various other calculations related to chemistry.
2. **Teachers:** Teachers are the secondary users of the software, as they will be using the software to teach chemistry concepts to students. Teachers will be using the software to create assignments and worksheets, check student submissions, and verify student calculations.
3. **Pharmacists:** Pharmacists are the tertiary users of the software, as they may use the software to calculate dosages, identify drug interactions, and perform other calculations related to pharmacy.
4. **Chemists:** Chemists are the expert users of the software, as they will be using the software to perform complex calculations and analyses related to chemistry. Chemists may use the software to analyze chemical reactions, determine reaction kinetics, and perform other advanced calculations.

## 1.9 Tools and Technology

The "Chemistry Calculator" project was implemented using Java, JavaFX, and Java Swing technologies. These technologies will enable the development of a cross-platform application that can run on multiple operating systems, including Windows, Mac, and Linux.

Below are some of the tools and technologies that will be used in the project:

**Integrated Development Environment (IDE):** An IDE will be used to write, test, and debug the Java code. Eclipse, IntelliJ IDEA, or NetBeans are some of the popular IDEs for Java development.

**Java:** Java will be the primary programming language used to develop the software. Java is a widely used object-oriented programming language that provides a robust and secure development environment.

**JavaFX:** JavaFX is a framework for building rich user interfaces (UIs) for desktop and mobile applications. It is included in the Java Development Kit (JDK) and provides a powerful set of UI controls and layout containers for creating modern and responsive UIs.

**Java Swing:** Java Swing is a mature UI toolkit for building desktop applications. It provides a rich set of UI components, such as buttons, menus, text fields, and tables, and allows for easy customization and theming.

**Git:** Git is a version control system used to manage the source code of the project. It allows developers to collaborate on the codebase, track changes, and revert to previous versions if needed.

## **1.10 Configuration Management Plan**

**Version Control System:** The project will use Git as the version control system. All project files, including source code, documentation, and other assets, will be stored in a Git repository.

**Branching Strategy:** The branching strategy should define how the project's codebase will be organized and maintained. The strategy should include the main development branch, release branches, and feature branches. The main development branch will be used for ongoing development work, release branches will be created for each release, and feature branches will be created for new features or major changes.

**Change Control Process:** The change control process should outline how changes to the codebase will be managed, including the process for making and reviewing changes, testing, and approving changes. The process should include guidelines for committing code, merging changes, and handling conflicts.

**Release Management:** The release management process should outline how releases will be managed, including how releases will be numbered, tested, and deployed. It should also define how bug fixes and maintenance releases will be managed.

**Backup and Recovery:** The backup and recovery process should outline how backups of the codebase and other project files will be taken, how they will be stored, and how they will be restored in the event of a data loss or corruption.

**Access Control:** Access control should be defined to ensure that only authorized personnel have access to the project files. Access control measures may include user authentication, permissions, and role-based access control.

## CHAPTER – 2

### Requirement Collection and Analysis

#### 2.1 Functional System Requirements:

1. User can balance chemistry equation

<b>FR-01</b>	<b>User can balance chemistry equation</b>
Description	User needs to balance the left-hand side (L.H.S.) to right hand side (R.H.S.) chemistry equation. In our desktop application, user input the reactants, products. By clicking the “Balance” button, system balance the given equation.
Stakeholders	Chemist, Pharmacist
Priority	High

2. User can see his/her chemistry equation balance history

<b>FR-02</b>	<b>User can see his/her chemistry equation balance history</b>
Description	After a successful balance equation, user can see all previous balance equation history by clicking the “History” button
Stakeholders	Chemist, Pharmacist
Priority	High

3. User can clear the history

<b>FR-03</b>	<b>User can clear the history</b>
Description	User can clear the history of balance equation by clicking the “Clear” button.
Stakeholders	Chemist, Pharmacist
Priority	High

4. User can determine Concentration of compound

<b>FR-04</b>	<b>User can determine Concentration of compound</b>
Description	In our desktop application, user can determine concentration of a compound.
Stakeholders	Chemist, Pharmacist
Priority	High



## 5. User can view any atomic symbol details

<b>FR-05</b>	<b>User can view any atomic symbol details</b>
Description	If user work with different chemistry equation, then user needs to know about different atom number, mass, electron config. By using our application, user provides only atom number or symbols and gets atom name, mass, and electron config.
Stakeholders	Chemist, Pharmacist
Priority	Medium

## 6. User can calculate Molar mass of compound

<b>FR-06</b>	<b>User can calculate Molar mass of compound</b>
Description	User input the compound and system provides the compound molar mass in gram unit. System also provides each atomic mass unit in the compound.
Stakeholders	Chemist, Pharmacist
Priority	High

## 7. User can calculate Molar mass of compound with percentage

<b>FR-07</b>	<b>User can calculate Molar mass of compound with percentage</b>
Description	User input the compound and system provides each atomic molar mass in gram unit and percentage.
Stakeholders	Chemist, Pharmacist
Priority	High

## 8. User can find out unknown value in a titration equation

<b>FR-08</b>	<b>User can find out unknown value in a titration equation</b>
Description	In a titration chemistry equation, our desktop application find out an unknown value.
Stakeholders	Chemist, Pharmacist
Priority	High

## 2.2 Non-functional Requirements

### 2.2.1 Data Requirements

The process used to identify, prioritize, precisely formulate, and validate the data needed to achieve business objectives it's called data requirements. Ensures that data produced and consumed satisfies business objectives, is understood by all relevant stakeholders, and meets the needs of the business processes that create and use the data. For defining data requirements, we need to build the model.

For our application maximum data would be loaded from remote user. And for that purpose, we need to focus on some major points. Such as:

- Types of entity of the system
- Route data locations
- Capacity and resources of the data requirements
- Data source sequence
- Data availability schedules
- Quantity of data
- Availability of data

### 2.2.2 Performance Requirements

Performance requirements define how well the system performs certain functions under specific conditions. Examples are speed of response, throughput, execution time and storage capacity. The service levels comprising performance requirements are often based on supporting end-user tasks. It is very important to maintain performance of any software system. To ensure performance, we need to maintain some steps. Now, I will explain some perspective by which we are going to enhance the performance of our project.

### 2.2.3 Speed & Latency Requirements

Latency is a measure of delay. In a network, latency measures the time it takes for some data to get to its destination across the network. It is usually measured as a round trip delay - the time taken for information to get to its destination and back again. Speed Requirements means minimum internet connection speed. Speed and latency requirements must be ensured while retrieving data from the cloud server.

<b>SLR-1</b>	<b>Calculation of chemistry equation, molar mass of compound must be faster</b>
<b>Description</b>	When user calculate the molar mass of compound, balance chemistry equation must be gives faster result.
<b>Stakeholders</b>	Chemist, Pharmacist
<b>Priority</b>	High

### 2.2.4 Precision & Accuracy Requirements

Accuracy refers to closeness of the measurements to a specific value, while precision refers to the closeness of the measurements to each other. Precision is how consistent results are when measurements are repeated. Accuracy refers to how close a measurement is to the true value. Result that is to be shown to the end user is need to be accurate. Because, wrong information might be ruined the whole business process.

<b>PAR-1</b>	<b>Calculation of chemistry equation result must be accurate</b>
<b>Description</b>	When user calculate any titration equation or any equation, system given result must be accurate.
<b>Stakeholders</b>	Chemist, Pharmacist
<b>Priority</b>	High

## 2.3 Dependability Requirements

A system may be considered to be dependable if it operates without interruption, delivers the services that are expected by stakeholders does not have adverse effects on the system's environment and does not damage its data or the system itself. The term dependability is measured based on four dimensions. Such as:

- Availability
- Reliability
- Safety
- Security

If we want to say that our application system is dependable then it must fulfil the four dimensions. But there are other tasks. Like there is no way to make mistakes or our system should have the ability to detect and then remove errors. Besides that, it is also very important to limit the damage which might be caused by system failure.

### 2.3.1 Reliability & Availability Requirements

Availability is the probability that a system, at a point in time, will be operational and able to deliver the requested services. Reliability is the probability of failure-free system operation over a specified time in a given environment for a given purpose. Functional reliability requirements define system and software functions that avoid, detect or tolerate faults in the software and so ensure that these faults do not lead to system failure. Now, we will mention requirements which are related to reliability and availability.

<b>RAR-1</b>	<b>The system must be available on 24 X 7</b>
<b>Description</b>	Our system must be available all day long, every day in a week <ul style="list-style-type: none"> <li>• The system must be updated regularly</li> <li>• System must be malware free</li> </ul>
<b>Stakeholders</b>	Chemist, Pharmacist
<b>Priority</b>	High

### 2.3.2 Robustness or Fault-Tolerance Requirements

Robustness is the ability of a computer system to cope with errors during execution and cope with erroneous input. Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components. An app is robust when it can work consistently with inconsistent data. An app is fault-tolerant when it can work consistently in an inconsistent environment. To ensure robustness and fault-tolerance facilities to the end users, it is urgent to ensure 0% crash. Moreover, it must show accurate results.

<b>RFT-1</b>	<b>The system handles all user access without system errors</b>
<b>Description</b>	Thousands of users might hit our application system at a time. All their requests must be handled without any fault.
<b>Stakeholders</b>	N/A
<b>Priority</b>	High

### 2.3.3 Safety-Critical Requirements

A safety-critical system or life-critical system is a system whose failure or malfunction may result in one (or more) of the following outcomes death or serious injury to people, loss or severe damage to equipment/property, environmental harm. Safety critical software systems are defined to be those systems that should unanticipated failure occur, there could be the harm to life or property. There are no safety-critical requirements in our project.

### 2.3.4 Maintainability & Supportability Requirements

Maintainability involves a system of continuous improvement - learning from the past in order to improve the ability to maintain systems, or improve reliability of systems based on maintenance experience. Supportability refers to the degree by which the characteristics, design and functions of products or services meet the standards of a particular system or organization. It is very important to provide after service or support to the end users.

### 2.3.5 Maintainability Requirements

<b>MR-1</b>	<b>System helps to update features</b>
<b>Description</b>	It is important to update features for future.
<b>Stakeholders</b>	Chemist, Pharmacist
<b>Priority</b>	Medium

### 2.3.6 Supportability Requirements

Supportability requirements may have related to some extends like:

- Testability
- Extensibility
- Adaptability
- Maintainability
- Compatibility
- Configurability
- Serviceability
- Install ability

Our application meets all of the above requirements related to supportability.

### 2.3.7 Adaptability Requirements

Adaptability is to be understood here as the ability of a system to adapt itself efficiently and fast to changed circumstances. An adaptive system is therefore an open system that is able to fit its behavior according to changes in its environment or in parts of the system itself. There are no adaptability requirements in our system software.

## 2.4 Security Requirements

Making software security as a requirement is very important. Software security requirements should be its functional requirement. Software security enforces security of an application system.

Functionality related to software security can either be directly tested or observed. Some security related requirements are given below:

- Atom name, number, electron config stored in HashMap
- Each atom details stored as key, value pair

### 2.4.1 Access Requirements

For accessing to our application system, there remains no authentication and authorization techniques. And every module of our system will provide free access. So, there are no access requirements.

### 2.3.2 Integrity Requirements

Integrity refers to methods of ensuring that data is real, accurate and safeguarded from malicious user modification. Data integrity maintenance is an information security requirement. Integrity requirements refers to a security system which ensures an expectation of data quality. It also ensures that all data of the system would never be exposed to the malicious modification or accidental destruction. For that reason, we will store atom data in HashMap such as key, value pair.

## 2.5 Usability and Human-Interaction Requirements

Usability and Human Computer Interaction are becoming core aspects of the system development process to improve and enhance system facilities and to satisfy user's needs and necessities. Human Computer Interaction will assist designers, analysts and users to identify the system needs from text style, fonts, layout, graphics and color, while usability will confirm if the system is efficient, effective, safe, utility and to evaluate, practical visible and provide job satisfaction to the users. Usability is about assuring users safe to use, easy to use and evaluate, enjoyable, and satisfying. To ensure usability, the user should participate in the development process to prevent future user frustration and error and meet the user's requirements. The main target of developing any system is to make the system user friendly and easy to usable for the end users.

### 2.5.1 Ease of Use Requirements

Our application is easy to use and also easily understandable. The system must be efficient for the frequent user. The system must be easy to remember for the casual user. The user must understand what the system does. The user must feel satisfied with the system.

<b>EUR-1</b>	<b>Application must be usable for the end users</b>
<b>Description</b>	This website is enough usable to the chemist or pharmacist by which they can operate this system easily.
<b>Stakeholders</b>	Chemist, Pharmacist
<b>Priority</b>	High

### 2.5.2 Personalization and Internationalization Requirements

Internationalization is the process of designing a software application so that it can be adapted to java languages and regions without engineering changes. Personalization is the process of tailoring pages to individual users' characteristics or preferences. There are not any personalization and internationalization requirements to our system. This maiden version of our application is be operated by anywhere.

### 2.5.3 Understand ability and Politeness Requirements

It is already said that the application which we are going to develop, is understandable enough. The system provides hints to users whether any error occurred or wrong. By reading those errors users can be able to operate the system easily.

### 2.5.4 Accessibility Requirements

Accessibility means more than putting things online. It means making your content and design clear and simple enough so that most people can use it without needing to adapt it, while supporting those who do need to adapt things. Accessibility is essential for developers and organizations that want to create high quality websites and web tools, and not exclude people from using their products and services. There are no specific accessibility requirements associated to our system yet.

### 2.5.5 Training Requirements

Training requirements involved in after service of any application. First of all, select all of the end user than it's easy to train up them. It is very necessary to properly train up end users to the system so that they would be capable to operate easily. After launching the full package to the market, firstly we provide training to the different end users like chemist and pharmacist etc. So that every end user can well know about the application.

## 2.6 Look and Feel Requirements

In software design, look and feel is a term used with respect to a graphical user interface and comprises aspects of its design, including elements such as colors, shapes, layout, and typefaces, as well as the behavior of dynamic elements such as buttons, boxes, and menus. The look and feel requirements describe the intended spirit, the mood, or the style of the product's appearance. These requirements specify the intention of the appearance, and are not a detailed design of an interface. Look and feel requirements mainly refers how the system will look like and how the user interface or graphical user interface of our system will display to the user.

### 2.6.1 Appearance Requirements

Chemist, pharmacist and all other user must know which input fields are required and which are not. For that reason, we will use labels for all input fields. Input fields might be text type, radio, checkbox, spinner etc.

<b>AR-1</b>	<b>Labels of mandatory fields must be bold</b>
<b>Description</b>	The mandatory field's label must be bold and all input fields must have placeholder to make it easier for the users.
<b>Stakeholders</b>	Chemist, Pharmacist
<b>Priority</b>	Low

### 2.6.2 Style Requirements

After keeping all contents, it is very essential to load stylesheet to the application. For desktop application like desktop system, java swing language. It is to be said that we are going to develop our system at desktop platform. Style makes the system lucrative.

<b>SR-1</b>	<b>The appearance must be controllable using stylesheet file</b>
<b>Description</b>	For desktop application style sheet files are .css So, all stylesheet must be controllable by the JPanel, JTable etc.
<b>Stakeholders</b>	Software developer
<b>Priority</b>	High

## 2.7 Operational and Environmental Requirements

Operational and environmental requirement refers to the capabilities, performance measurements, process, measurements of effectiveness, measurements of performance, measures of sustainability, measurements of technical performances etc. The operational requirements focus on how the system will be operated by the users, including interfaces and interoperability with other systems. The requirements establish how well and under what conditions the system must perform. Operational requirements are the basis for system requirements. Environmental Requirement means any Environmental Law, agreement or restriction, as the same now exists or may be changed or amended or come into effect in the future, which pertains to any Hazardous Material or the environment.

### 2.7.1 Expected Physical Requirements

Physical Requirements is the physical ability to perform the job, and these criteria often get neglected. There are no expected physical requirements in our system.

### 2.7.2 Requirements for Interfacing with Adjacent Systems

The adjacent systems are those pieces of work that supply your work with information or that receive information and services from your work. An adjacent system might be an organization, an individual, a computer system or some other piece of technology, or a combination of any of these. For interfacing the system client needs qualification or ability. But there are no requirements for interfacing with adjacent system for our project.

### **2.7.3 Release Requirements**

Release requirements means the deadline of the application. Project manager set up the time line of the application and show the time line to the client and inform the client when the application should be release. Release deadline are sometimes extending for testing and validating the application. There are no specific release requirements in our system.

## **2.8 Legal Requirements**

Legal requirements normally refer to the terms and conditions or privacy policy of any organizations. The terms and condition of our application is that, no third-party software or person are allowed to engage to use our data for their business purpose.

### **2.8.1 Compliance Requirements**

Compliance requirements are only guidelines for compliance with the hundreds of laws and regulations applicable to the specific type assistance used by the recipient, and their objectives are generic in nature due to the large number of federal programs. There are no specific compliance requirements for our system.

### **2.8.2 Standards Requirements**

Quality standards are defined as documents that provide requirements, specifications, guidelines, or characteristics that can be used consistently to ensure that materials, products, processes, and services are fit for their purpose. There are no specific standards requirements for our system.



## **2.9 Elicitation Technique**

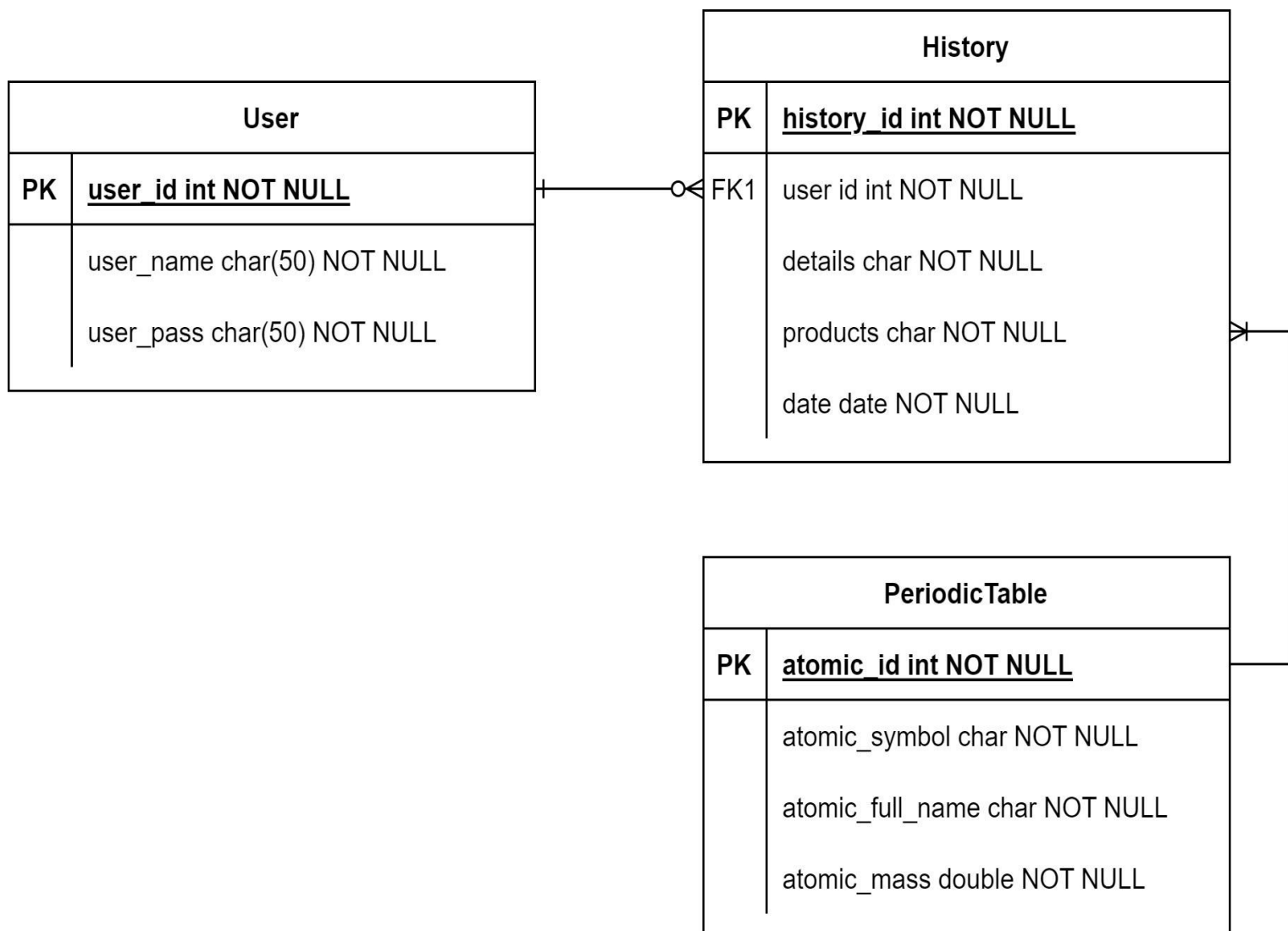
Using program comprehension techniques for eliciting requirements from existing source code can be an effective approach for preparing an SRS document. Program comprehension techniques involve analysing and understanding existing software systems to extract valuable information and requirements.

To collect requirements from existing source code, we first understand the structure and design of the code. This can be achieved by using program comprehension tools such as source code analysis and visualization tools. These tools help in identifying the different modules, classes, functions, and variables in the code, and understanding the relationships between them. Once we understood the code structure, next we identify the requirements that are already implemented in the code. This can be achieved by analysing the code logic and identifying the functionalities and features that are implemented. The requirements can then be documented and incorporated into the SRS document.

Additionally, program comprehension techniques can also help in identifying the limitations and potential improvements of the existing code. By analysing the code, we identify areas where the code can be optimized, updated, or improved to meet the desired requirements. Overall, using program comprehension techniques for eliciting requirements from existing source code can save time and effort in preparing the SRS document. It also ensures that the requirements are accurately documented and reflect the current functionality of the software system. However, it is important to note that program comprehension techniques should not be the sole method for eliciting requirements, and other approaches such as stakeholder interviews and surveys should also be used to ensure comprehensive requirement gathering.

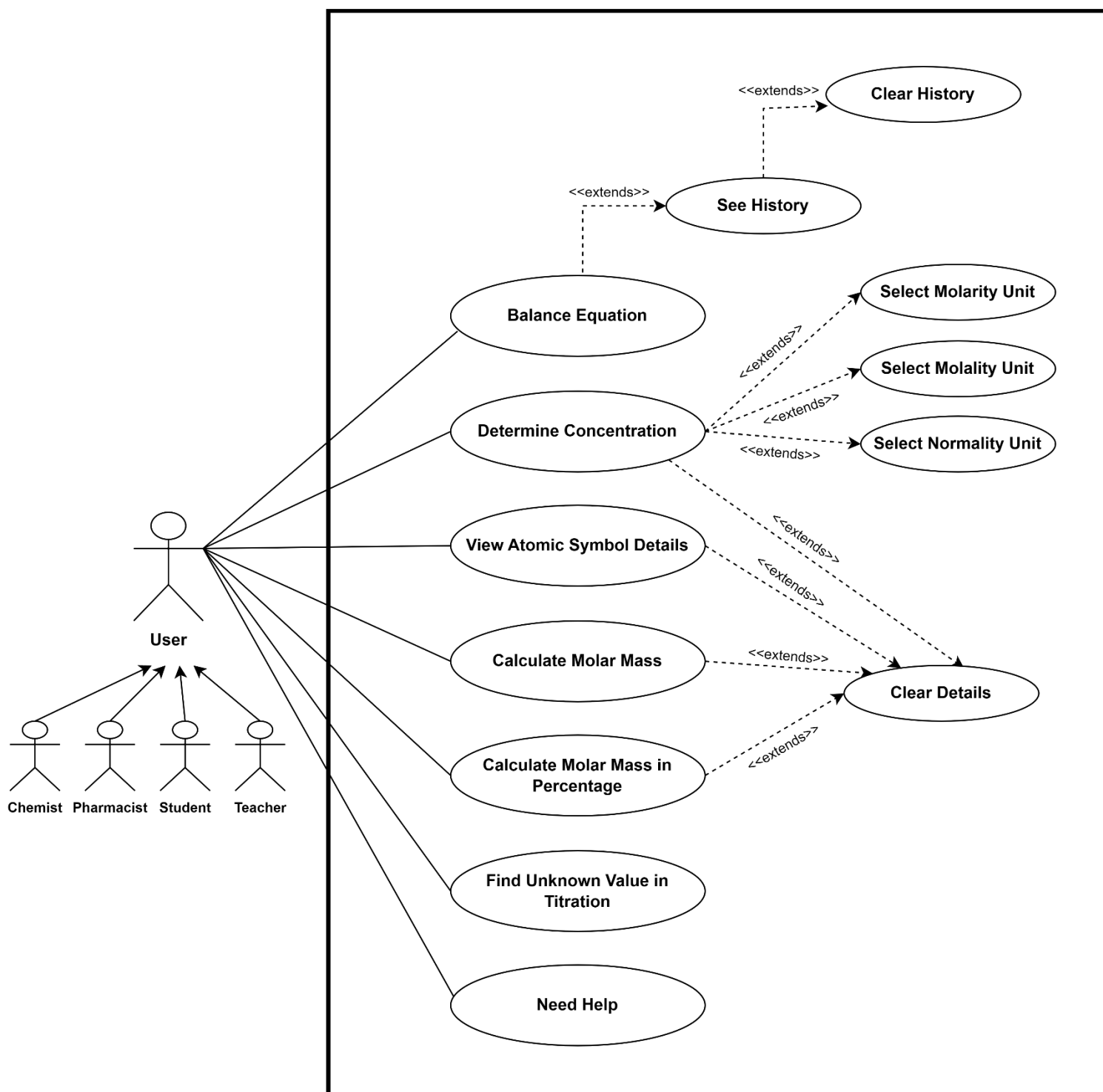
## CHAPTER 3

### DIAGRAM OF ENTITY RELATIONSHIP



## CHAPTER 4

### USECASE DIAGRAM



## 4.1 Usecase Description

<b>Use Case Id</b>	01	
<b>Use Case</b>	User can balance chemistry equation	
<b>Goal</b>	User able to balance chemistry equation	
<b>Preconditions</b>	Users should have fill up Reactants and products	
<b>Success End Condition</b>	Users successfully balanced chemistry equation	
<b>Failed End Condition</b>	Users can't successfully balance chemistry equation	
<b>Primary Actors:</b>	Chemist, Pharmacist	
<b>Secondary Actors:</b>	Users	
<b>Trigger</b>	Click on Balance Button	
<b>Main Success Flows</b>	<b>Step</b>	<b>Action</b>
	1	User click the Equations balance side bar
	2	User gives Reactants and products
	3	System checks the valid symbol
	3.1	System finds that symbols is valid
	4	System successfully provides the balanced equations
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	3a	System finds that symbols is not valid, and information are wrong
	3a1	System doesn't provide the balanced equations
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirement</b>
	1	

Table 4.1: Use Case Description of balance chemistry equation

<b>Use Case Id</b>	02	
<b>Use Case</b>	User can see his/her chemistry equation balance history	
<b>Goal</b>	User able to see his/her chemistry equation balance history	
<b>Preconditions</b>	Users should successfully do some previous balance equation	
<b>Success End Condition</b>	Users successfully see his/her chemistry equation balance history	
<b>Failed End Condition</b>	Users can't successfully see his/her chemistry equation balance history	
<b>Primary Actors:</b>	Chemist, Pharmacist	
<b>Secondary Actors:</b>	Users	
<b>Trigger</b>	Click on History button	
<b>Main Success Flows</b>	<b>Step</b>	<b>Action</b>
	1	User click the Equations balance side bar
	2	User then click the History button
	3	System successfully provides all of the previous balanced equations
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	3a	System doesn't provide any of the previous balanced equations because there is no previous balance equation
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirement</b>
	1	

Table 4.2: Use Case Description of see chemistry equation balance history

<b>Use Case Id</b>	03	
<b>Use Case</b>	User can determine Concentration of compound	
<b>Goal</b>	User able to determine Concentration of compound	
<b>Preconditions</b>	Users should successfully enter compound, compound mass, volume of solution, Equivalent number	
<b>Success End Condition</b>	Users successfully able to determine Concentration of compound	
<b>Failed End Condition</b>	Users can't successfully able to determine Concentration of compound	
<b>Primary Actors:</b>	Chemist, Pharmacist	
<b>Secondary Actors:</b>	Users	
<b>Trigger</b>	Click on Get Concentration button	
<b>Main Success Flows</b>	<b>Step</b>	<b>Action</b>
	1	User click the Concentration from the side bar
	2	User Enter the Compound name
	3	User Enter the Compound mass
	4	User Enter the volume of solution
	5	User Enter the Equivalent number
	6	User check format Molarity
	7	User then click the Get concentration Button
	8	System provide concentration in Molarity format
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	6a	User check format Molality
	8a	System provide concentration in Molality Format
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	6b	User check format Normality
	8b	System provide concentration in Normality Format

Table 03: Use Case Description of determine Concentration of compound

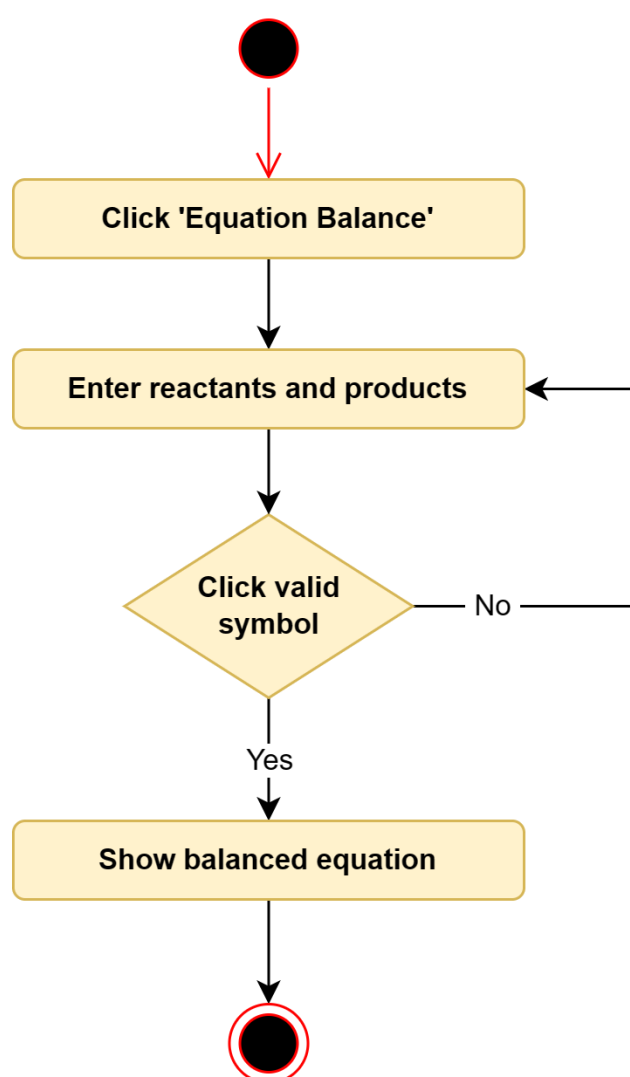
<b>Use Case Id</b>	04	
<b>Use Case</b>	User can view any atomic symbol details	
<b>Goal</b>	User able to view any atomic symbol details	
<b>Preconditions</b>	Users should have given any valid atomic symbol	
<b>Success End Condition</b>	Users successfully view given atomic symbol details such as Atomic Name, Atomic Number, Atomic Mass, Electron Config	
<b>Failed End Condition</b>	Users can't view any atomic symbol details	
<b>Primary Actors:</b>	Chemist, Pharmacist	
<b>Secondary Actors:</b>	Users	
<b>Trigger</b>	Click on Get config	
<b>Main Success Flows</b>	<b>Step</b>	<b>Action</b>
	1	User click the Electron Config side bar
	2	User gives valid Atomic Symbol
	3	System checks the valid symbol
	4	System finds that symbols is valid
	5	System successfully provides atomic symbol details
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	User doesn't give valid Atomic Symbol
	5a	System does not provide any information about atomic symbol
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirement</b>

Table 04: Use Case Description of view any atomic symbol details

## CHAPTER 5

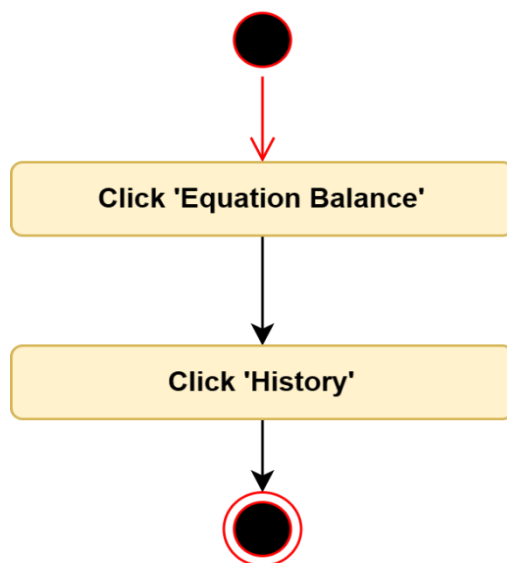
### DIAGRAM OF ACTIVITY

#### 5.1 Balance Equation Activity Diagram

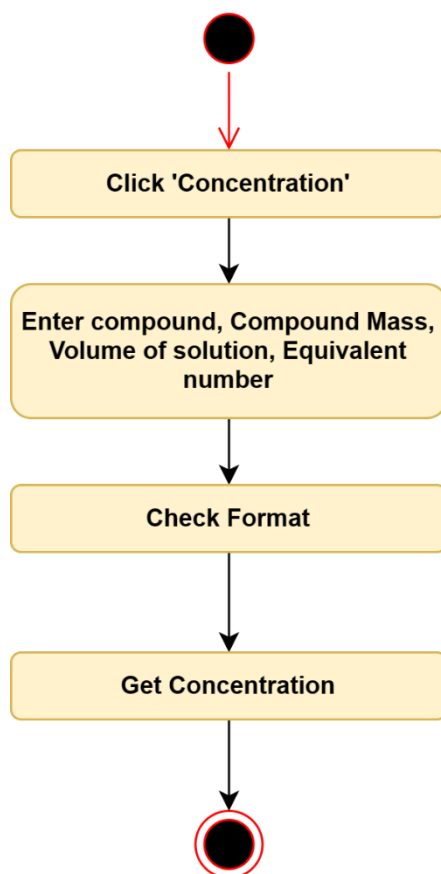




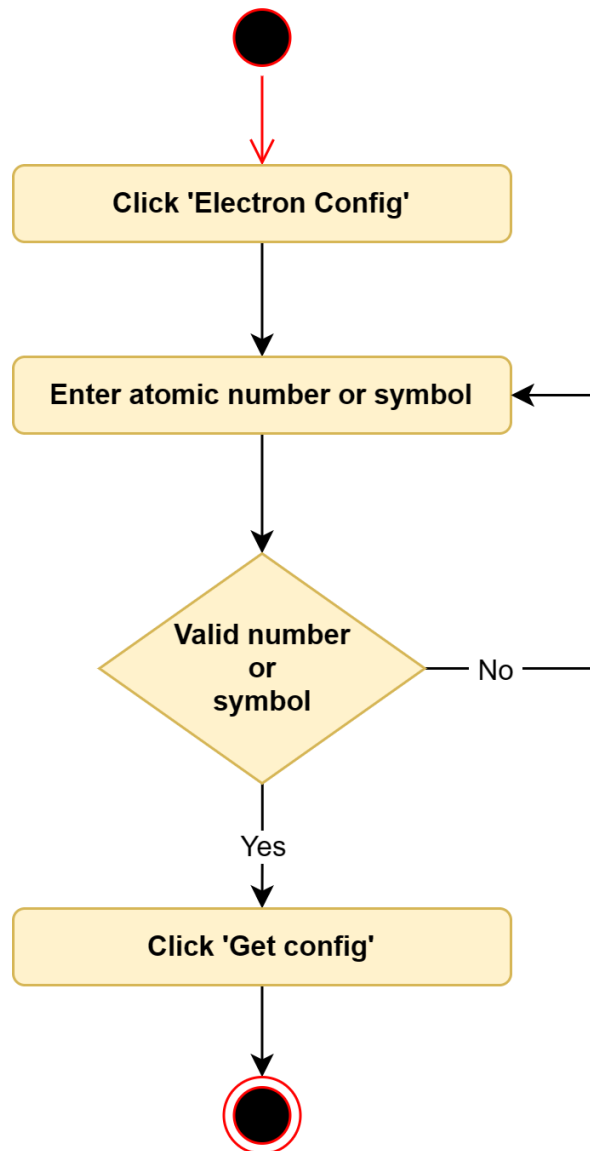
## 5.2 Equivalence Check History Activity Diagram



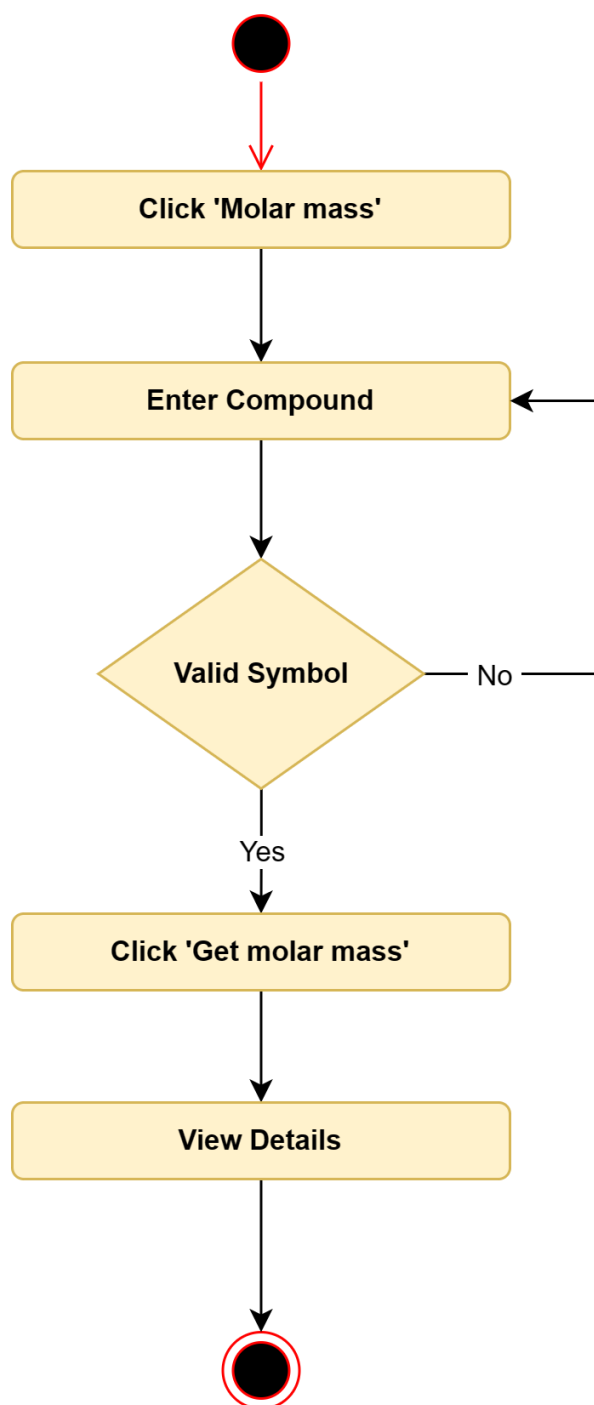
## 5.3 Determine Concentration Activity Diagram



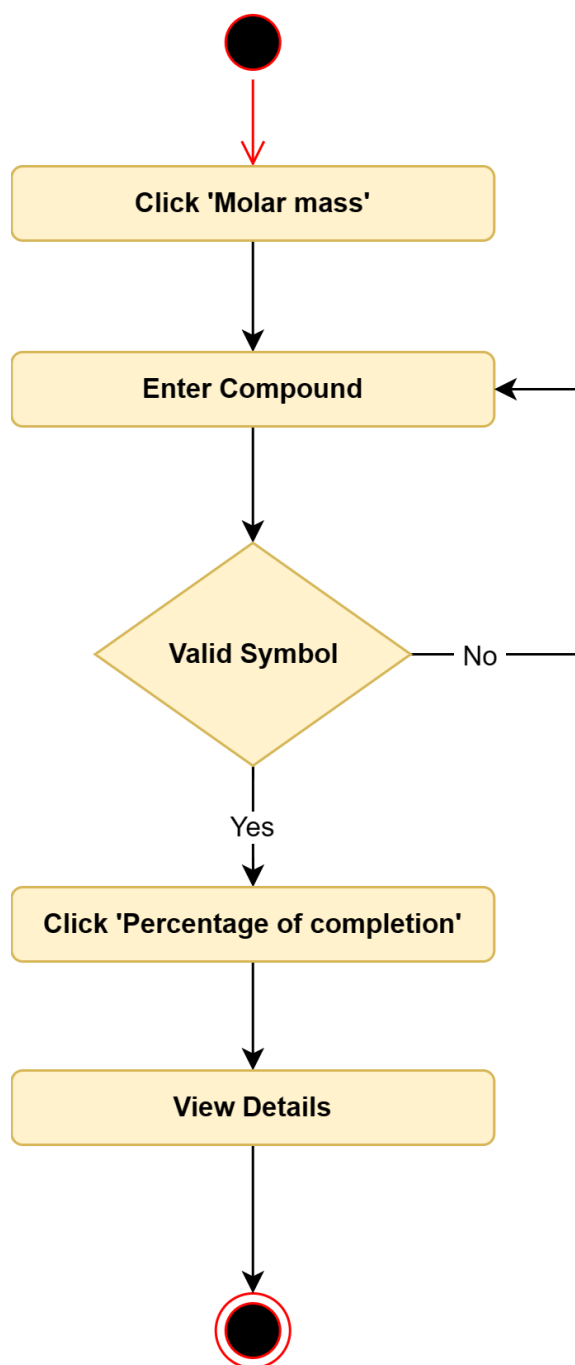
## 5.4 View Atomic Symbol Details Activity Diagram



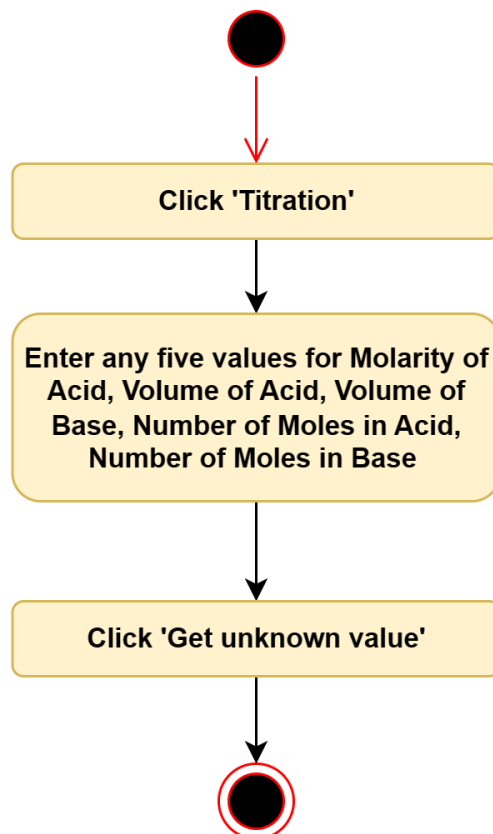
## 5.5 Calculate Molar Mass Activity Diagram



## 5.6 Calculate Molar Mass With Percentage Activity Diagram



### 5.7 Find Unknown Value In Titration Activity Diagram



## CHAPTER 6

### TRACEABILITY MATRIX

#### 6.1 Introduction

A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. It is used to track the requirements and to check the current project requirements are met.

Requirement Traceability Matrix (RTM) is a document that maps and traces user requirement with test cases. It captures all requirements proposed by the client and requirement traceability in a single document, delivered at the conclusion of the Software development life cycle. The main purpose of Requirement Traceability Matrix is to validate that all requirements are checked via test cases such that no functionality is unchecked during Software testing.

#### 6.2 Why RTM is Important?

The main agenda of every tester should be to understand the client's requirement and make sure that the output product should be defect-free. To achieve this goal, every QA should understand the requirement thoroughly and create positive and negative test cases. This would mean that the software requirements provided by the client have to be further split into different scenarios and further to test cases. Each of this case has to be executed individually. A question arises hereon how to make sure that the requirement is tested considering all possible scenarios/cases? How to ensure that any requirement is not left out of the testing cycle? A simple way is to trace the requirement with its corresponding test scenarios and test cases. This merely is termed as 'Requirement Traceability Matrix.' The traceability matrix is typically a worksheet that contains the requirements with its all possible test scenarios and cases and their current state, i.e. if they have been passed or failed. This would help the testing team to understand the level of testing activities done for the specific product.

#### 6.3 Use Cases

Use case ID	Use case Name
UC1	Balance Chemistry Equation
UC2	See History
UC3	Determine concentration
UC4	Select Molarity unit
UC5	Select Molality unit
UC6	Select Normality Unit
UC7	View atomic symbol details
UC8	Calculate Molar Mass
UC9	Calculate Molar Mass with percentage
UC10	Find Unknown value in titration
UC11	Need help

## 6.4 Test Cases

Test Case ID	Test Case
TC1	H <sub>2</sub> + O <sub>2</sub>
TC2	H <sub>2</sub> O
TC3	Hh
TC4	Verify if user can view history
TC5	5
TC6	Na
TC7	Verify if user can able to clear the input
TC8	23 milligram
TC9	12 milliliter
TC10	Check if user can select Molarity
TC11	Check if user can select Molality
TC12	Check if user can select Normality
TC14	NH <sub>3</sub>
TC15	Hk
TC16	12 molar
TC17	23 molar
TC18	32
TC19	12
TC20	Verify if user able view details
TC21	Verify if user able to view Need helps

## 6.5 Requirements Traceability Matrix Table

Requirements Traceability Matrix							
<b>Project Name</b>	Chemistry Calculator	<b>Business Area</b>		N/A			
<b>Project Leader</b>	Rahat Uddin Azad	<b>Business Analyst Lead</b>		Fazle Rabbi, Anwar Kabir			
<b>QA Lead</b>	Saifur Rahman, Abdullah An-Noor	<b>Target Implementation Date</b>					
<b>Category/ Functional Activity</b>	<b>Requirement Description</b>	<b>Use Case Reference</b>	<b>Design Document Reference</b>	<b>Code Module/ Reference</b>	<b>Test Case Reference</b>	<b>User Acceptance Validation</b>	<b>Comments</b>
FR-01	User can balance chemistry equation	UC1			TC1,TC2	Pass	
FR-02	User can see his/her chemistry equation balance history	UC2			TC4	Verified	
FR-03	User can clear the history	UC12			TC7	Verified	
FR-04	User can determine Concentration of compound	UC3, UC4, UC5, UC6			TC8, TC9, TC10, TC11, TC12	Pass	
FR-05	User can view any atomic symbol details	UC7			TC5, TC6	Pass	
FR-06	User can calculate Molar mass of compound	UC8			TC14	Pass	
FR-07	User can calculate Molar mass of compound with percentage	UC9			TC14	Pass	
FR-08	User can find out unknown value in a titration equation	UC10			TC16, TC17, TC18, TC19	Pass	
FR-09	User can view need help	UC11			TC20	Verified	