

**Software Requirements Specification**

for

**RETAIL ROVER**

Technical Project (Font Size 16, Bold)

Group Name and ID (Font Size 14, Bold)

Client ABC Company

Version 1.0

Prepared by

Group Name

|  |  |
| --- | --- |
| Member Name | Registration Number |
| <<Group Leader>> |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table of Contents

[LIST OF TABLES ii](#_Toc15626866)

[LIST OF FIGURES ii](#_Toc15626867)

[Revision History iii](#_Toc15626868)

[1 INTRODUCTION 1](#_Toc15626869)

[1.1 Purpose 1](#_Toc15626870)

[1.2 Intended Audience and Reading Suggestions 1](#_Toc15626871)

[1.3 Product Scope 1](#_Toc15626872)

[2 OVERALL DESCRIPTION 2](#_Toc15626873)

[2.1 Product Perspective 2](#_Toc15626874)

[2.2 Product Functions 2](#_Toc15626875)

[2.3 User Classes and Characteristics 4](#_Toc15626876)

[2.4 Operating Environment 5](#_Toc15626877)

[2.5 Design and Implementation Constraints 5](#_Toc15626878)

[2.6 User Documentation 5](#_Toc15626879)

[2.7 Assumptions and Dependencies 5](#_Toc15626880)

[2.8 User Interfaces 5](#_Toc15626881)

[2.8.1 Customer register page 6](#_Toc15626882)

[2.8.2 Invoice Data Filling Interface 6](#_Toc15626883)

[2.9 Hardware Interfaces 7](#_Toc15626884)

[2.10 Software Interface 7](#_Toc15626885)

[2.11 Communications Interfaces 8](#_Toc15626886)

[3 System Functional Features 9](#_Toc15626887)

[3.1 Database Requirements 9](#_Toc15626888)

[3.2 Create User 9](#_Toc15626889)

[4 NON-FUNCTIONAL REQUIREMENTS 10](#_Toc15626890)

[4.1 Performance Requirements 10](#_Toc15626891)

[4.2 Safety Requirements 10](#_Toc15626892)

[4.3 Security Requirements 10](#_Toc15626893)

[4.4 Software Quality Attributes 10](#_Toc15626894)

[4.5 Business Rules 11](#_Toc15626895)

[5 OTHER REQUIREMENTS 11](#_Toc15626896)

# LIST OF TABLES

[Table 3.1.1: Use Case Description for Create User 9](#_Toc15626897)

# LIST OF FIGURES

[Figure 2.8.1.1: Architecture Diagram of the system [1] 2](#_Toc15626898)

[Figure 2.8.1.1: Dataflow diagram of the system [2] 3](#_Toc15626899)

[Figure 2.8.1.2: Class Diagram of the system [3] 3](#_Toc15626900)

[Figure 2.8.1.1: User case Diagram for the application 4](#_Toc15626901)

[Figure 2.8.1.1: Customer registration Interface 6](#_Toc15626902)

[Figure 2.8.2.1: Invoice Interface 7](#_Toc15626903)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason | Version |
|  |  |  |  |
|  |  |  |  |

# INTRODUCTION

## Purpose

In an era characterized by technological innovation and digital transformation, organizations must strive for software development excellence if they hope to remain competitive and relevant. Considering the aforementioned, we are happy to   
offer our Software Requirements Specification (SRS) report for Retail rover, a dynamic company committed to making the most of technology in the digital age.

This comprehensive SRS document has been painstakingly crafted to explain the goals, scope, and specifications of the software project. It provides a clear and organized roadmap for everyone with an interest in the business. It incorporates all of Retail rover's requirements, expectations, and insights to make sure that the development process is guided by a single, overarching vision.

This SRS report's objective is to serve as a manual for the entire software development lifecycle. Reatil rover will eventually be able to accomplish its objectives thanks to the reduction of risks and the facilitation of effective communication among project stakeholders.   
Additionally, it will be a useful point of reference for the duration of the project, guaranteeing alignment with the initial objectives and vision.

The software requirements specified in this document pertain to the development and enhancement of a Point of Sale (POS) system for a small retail supermarket. This system is intended to replace the existing, outdated POS system currently in use.

## Intended Audience and Reading Suggestions

This SRS document is intended for a diverse audience, including but not limited to:

* Developers
* Project Managers
* Marketing Staff
* Users
* Testers

**For Developers:**

Begin with the Introduction and System Description to grasp the overall project goals and system architecture.

Proceed to Functional Requirements and User Interface sections for a detailed understanding of system behavior and design expectations.

Reference System Interfaces, Hardware Requirements, and Software Requirements sections for integration details.

**For Project Managers:**

Start with the Introduction and System Description to gain insights into the project's objectives.

Focus on Functional Requirements, Non-functional Requirements, and System Constraints for project planning and resource allocation.

Review the Revision History for updates and changes made during development.

**For Users:**

Begin with the Introduction for a general understanding of the upcoming changes.

Explore the User Interface section for insights into the system's appearance and user interactions

**Functional Requirements:** Developers and testers should closely examine the functional requirements section to comprehend the functions that the software must be able to execute.   
**Non-Functional Requirements:** Documentation writers, testers, and developers should consider non-functional requirements such as security, compliance, and performance.

## Product Scope

The scope of this SRS extends to the comprehensive replacement of the existing POS system in the small retail supermarket. The aim is to address the limitations of the current system by introducing a modernized solution that not only enhances the user interface but also improves overall system functionality.’

A significant focus within the scope is the redesign of the user interface (UI). The new UI is expected to be intuitive, user-friendly, and visually appealing. The redesign aims to streamline the checkout process, minimize training requirements for staff, and enhance the overall user experience.

Beyond the UI, the scope covers the enhancement of core functionalities, including but not limited to:

* Efficient transaction processing
* Inventory management and tracking.
* Real-time reporting and analytics
* Supplier, customer and employee management

The scope allows for the incorporation of additional features and improvements based on stakeholder requirements and industry best practices. These may include loyalty programs, promotional features, and support for emerging payment methods.

While the primary focus is on replacing the POS system, the scope includes seamless integration with existing systems, such as inventory management and customer relationship management. The interoperability ensures a cohesive and interconnected technology infrastructure.

# OVERALL DESCRIPTION

## Product Perspective

**Product Origin:**

The Retail Supermarket POS System specified in this SRS is a replacement for the existing, outdated POS system currently in use at the small retail supermarket. The decision to develop a new POS system arises from the need for modernization, improved functionality, and an enhanced user interface to meet evolving business requirements.

**product Context:**

Follow-on Member: While this POS system is a replacement for the current one, it is also a follow-on member of the broader technology infrastructure aimed at improving the supermarket's operational efficiency.

Self-contained Product: The POS system is designed to be a self-contained product, capable of handling all aspects of point-of-sale transactions, inventory management, and reporting.

**Relationship with Larger System:**

Larger System Description: The larger system encompasses the entire technological framework supporting the retail operations, including inventory management, customer relationship management, and backend databases.

**Subsystem Interconnections:**

The POS system communicates with the Inventory Management and CRM subsystems through well-defined interfaces.Data flow ensures seamless integration between the POS system and larger system components.

Draw your architecture Diagram here

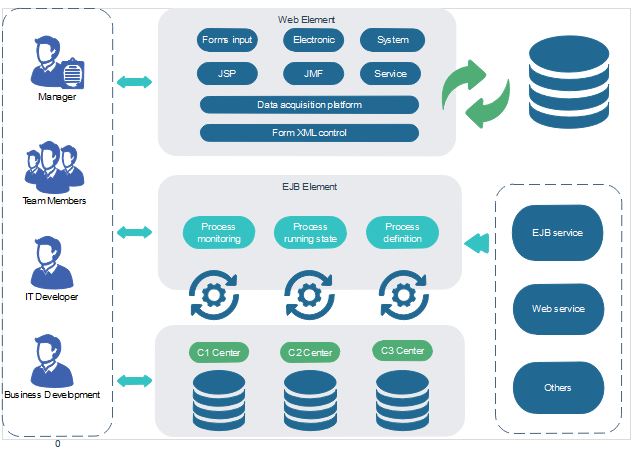


Figure 2.8.1.1: Architecture Diagram of the system [1]

## Product Functions

The primary goal of Retail rover software is to automate and optimize business processes.   
It includes inventory management , customer management ,Employee management, supplier management, sales analysis and reporting to facilitate efficient decision-making. The software's intuitive interface makes it easy for users to access and interact with their data. Additionally, it offers data security measures to protect personal information.

Inventory Management:

This functionality involves tracking and managing the supermarket's inventory efficiently. It includes features such as real-time stock updates, automatic reorder triggers, and product categorization to streamline restocking processes. The goal is to minimize stockouts, reduce excess inventory, and enhance overall inventory control.

Customer Management:

Customer management focuses on creating and maintaining a database of customer information. It includes features like customer registration, purchase history tracking, and loyalty program management. This functionality aims to improve customer engagement, personalize marketing efforts, and foster long-term relationships.

Employee Management:

Employee management functionality involves tools for managing staff information, roles, and schedules. It may include features like time tracking, payroll management, and access control. The goal is to optimize workforce efficiency, ensure proper resource allocation, and streamline HR-related processes.

Supplier Management:

Supplier management enables the monitoring and control of relationships with product suppliers. It includes features like supplier information tracking, order management, and performance analysis. This functionality aims to enhance procurement processes, optimize costs, and ensure reliable and timely product supply.

Sales Analysis and Reporting:

Description: Sales analysis and reporting functionality involve the generation of comprehensive reports and analytics related to sales performance. It includes features like sales trend analysis, product profitability assessment, and customizable reporting tools. The objective is to provide actionable insights for strategic decision-making and business growth.

Include your Top-level dataflow diagram or Class diagram here.

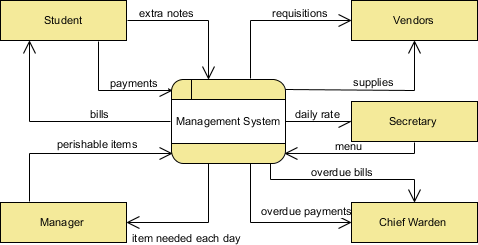


Figure 2.8.1.1: Dataflow diagram of the system [2]

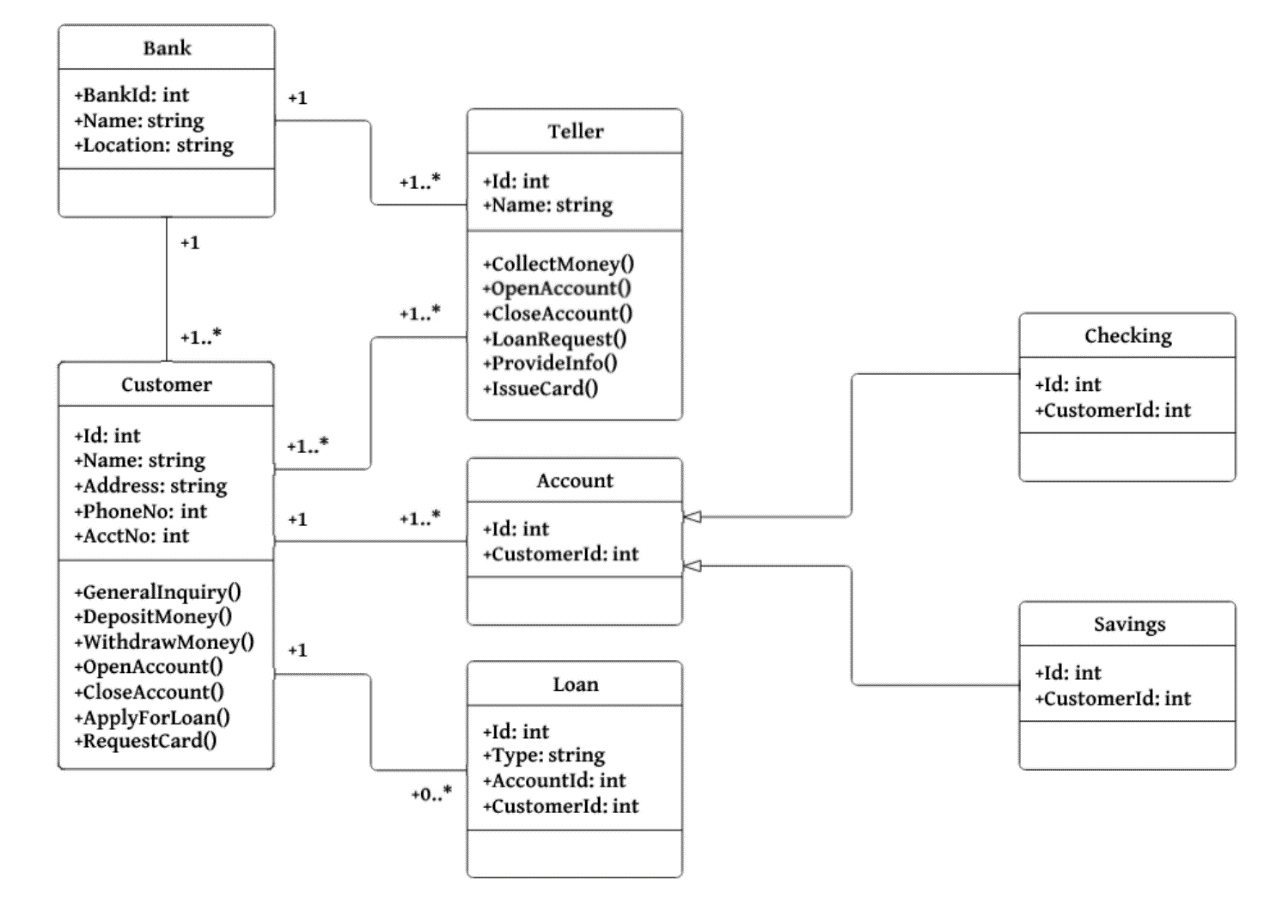


Figure 2.8.1.2: Class Diagram of the system [3]

## User Classes and Characteristics

We provide a Use Case Diagram to show how the Retail Rover software interacts with users and outside systems. This diagram is a helpful tool for both technical and non-technical users because it gives a high-level overview of how the system operates and interacts with various stakeholders.

Provide your Use case Diagram here. List down your potential users.

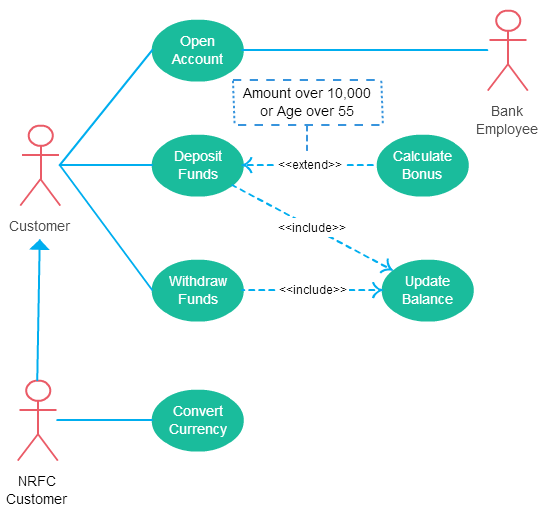


Figure 2.8.1.1: User case Diagram for the application

## Operating Environment

The following is the environment in which the software will operate:

Hardware Platform: • The program will run on standard server hardware for the backend, which consists of a CPU, RAM, and storage suitable for launching applications.   
Operating System and Versions: The backend will be powered by a compatible operating system Windows

The database management solution for inventory management , customer management ,Employee management ,supplier management , and report generation will be postgre SQL.

## Design and Implementation Constraints

Corporate and Regulatory Policies:

Developers must adhere to corporate policies and regulatory standards governing the retail industry. This includes compliance with data protection regulations, accounting standards, and any other legal requirements relevant to the supermarket's operations. Adherence to these policies is non-negotiable and may influence system design and functionality.

Hardware Limitations:

Timing Requirements: The system must operate within specified timing constraints to ensure responsive and efficient performance during peak transaction periods.

Memory Requirements: The POS system should be optimized to operate within the available memory constraints of the existing hardware infrastructure.

Specific Technologies, Tools, and Databases:

Technology Stack: The development team is constrained to a predefined technology stack, including specific programming languages, frameworks, and tools. This limitation ensures consistency with the organization's technology standards and facilitates future maintenance.

Database Requirements: The POS system must utilize a designated database system for data storage and retrieval, adhering to the organization's database management policies.

Security Considerations:

Access Control: The system must enforce strict access controls to ensure that users have appropriate permissions based on their roles.

## User Documentation

List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

## Assumptions 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 change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).

## User Interfaces

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.

Provide your sample interfaces and there description here.

### Customer register page

<sample> Customer register page description goes here

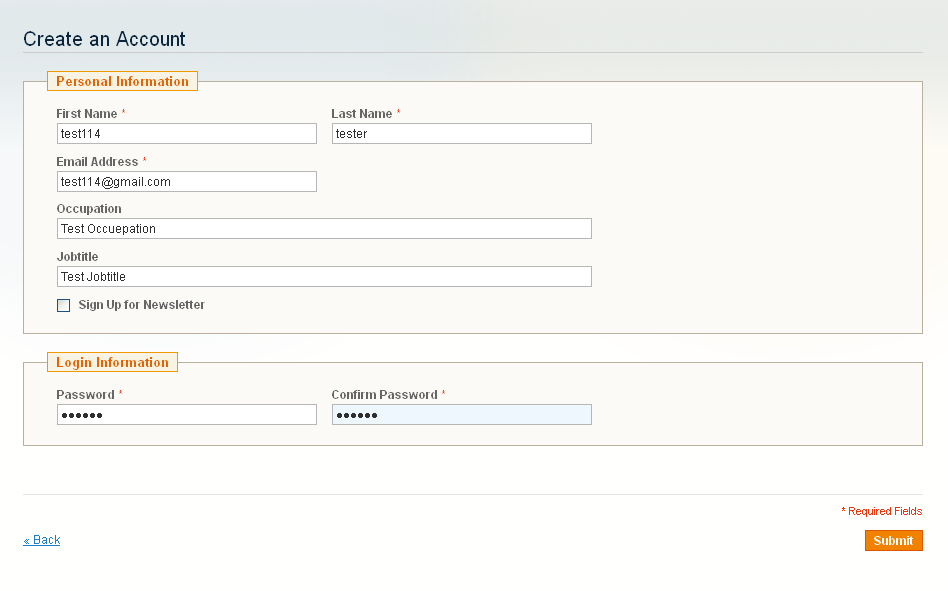


Figure 2.8.1.1: Customer registration Interface

### Invoice Data Filling Interface

Description about the interface goes here

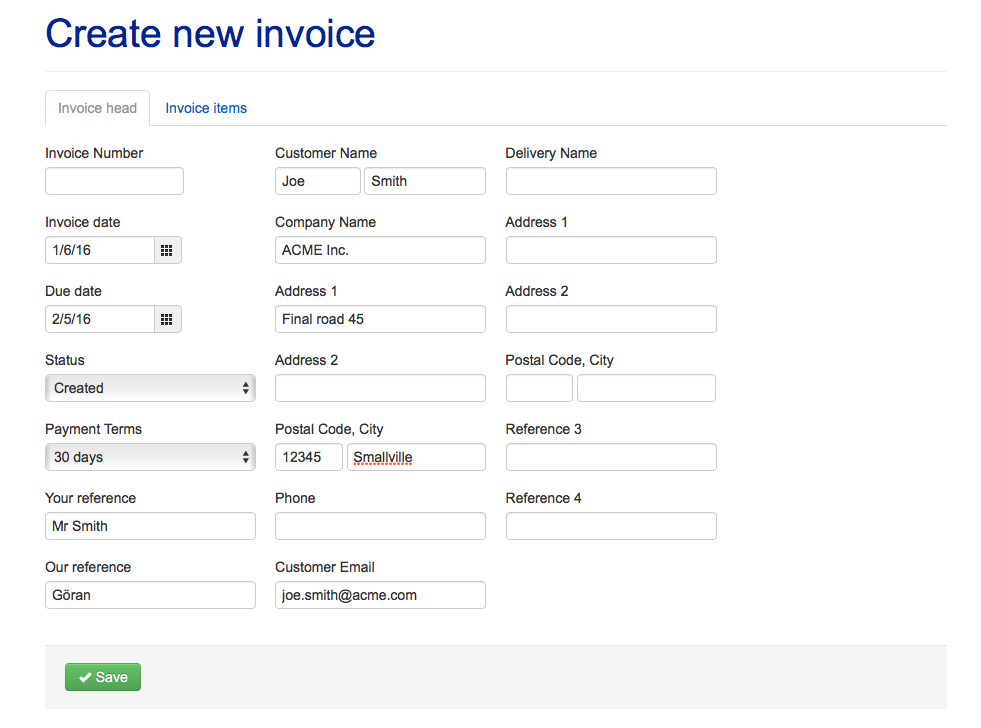


Figure 2.8.2.1: Invoice Interface

## Hardware Interfaces

Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used. (Add this section only if necessary)

## Software Interface

Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.)

Provide diagrams if necessary

## Communications Interfaces

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

Provide diagrams if necessary

# System Functional Features

Need to add use case descriptions here. For each use case create separate use case descriptions.

## Database Requirements

Use your E-R diagram and explain the database requirements

## Create User

Add the explanation about this and create following table to describe the details.

Table 3.1.1: Use Case Description for Create User

|  |  |  |
| --- | --- | --- |
| Use case ID |  | |
| Use case Name: |  | |
| Actors: |  | |
| Pre- Conditions |  | |
| Post Condition |  | |
|  | | |
|  | Action | System Response |
| Success Path |  |  |
| Alternate Path |  |  |
| Exception Path |  |  |
| Special Requirements |  | |

For Each use case need to provide this table

# **NON-FUNCTIONAL REQUIREMENTS**

## Performance Requirements

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

Use your activity diagram here to explain these things

## Safety Requirements

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.

## Security Requirements

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

## Software Quality Attributes

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

## Business Rules

List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.

# References

|  |  |
| --- | --- |
| [1] | DE Draw, "Architecture Diagrams," 2019 Edrawsoft, 2019. [Online]. Available: https://www.edrawsoft.com/architecture-diagram.php. [Accessed 31 July 2019]. |
| [2] | Visual Paradigm, "What is Data Flow Diagram?," Visual Paradigm Corp, 2018. [Online]. Available: https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/. [Accessed 31 July 2019]. |
| [3] | Medium Corporation, "Medium , UML Class Diagrams Tutorial, Step by Step," 02 Sep 2017. [Online]. Available: https://medium.com/@smagid\_allThings/uml-class-diagrams-tutorial-step-by-step-520fd83b300b. [Accessed 31 July 2019]. |
| [4] | M. Lotz, "Waterfall vs. Agile," Segue Technologies Inc, 05 July 2018. [Online]. Available: https://www.seguetech.com/waterfall-vs-agile-methodology/. [Accessed 17 July 2019]. |

# OTHER REQUIREMENTS

Appendix A: Glossary (If Necessary)

Appendix B: Analysis Models (If Necessary)

Appendix C: To Be Determined List (If Necessary)

Appendix D: Personal Contribution (One page for each member of the group)