

**Software Requirements Specification**

for

**Project Name (Font size 18, Bold)**

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

Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.

## Intended Audience and Reading Suggestions

Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.

## Product Scope

Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.

# OVERALL DESCRIPTION

## 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 a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful

Draw your architecture Diagram here

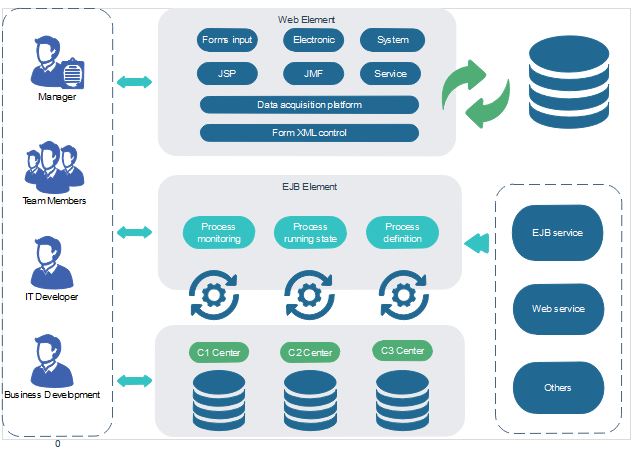


Figure 2.8.1.1: Architecture Diagram of the system [1]

## Product Functions

Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high-level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top-level data flow diagram or object class diagram, is often effective.

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

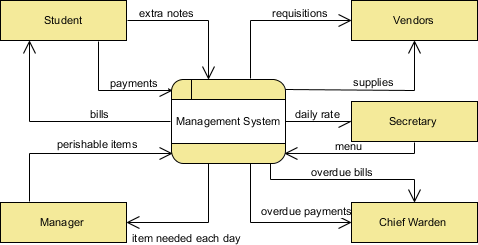


Figure 2.8.1.1: Dataflow diagram of the system [2]

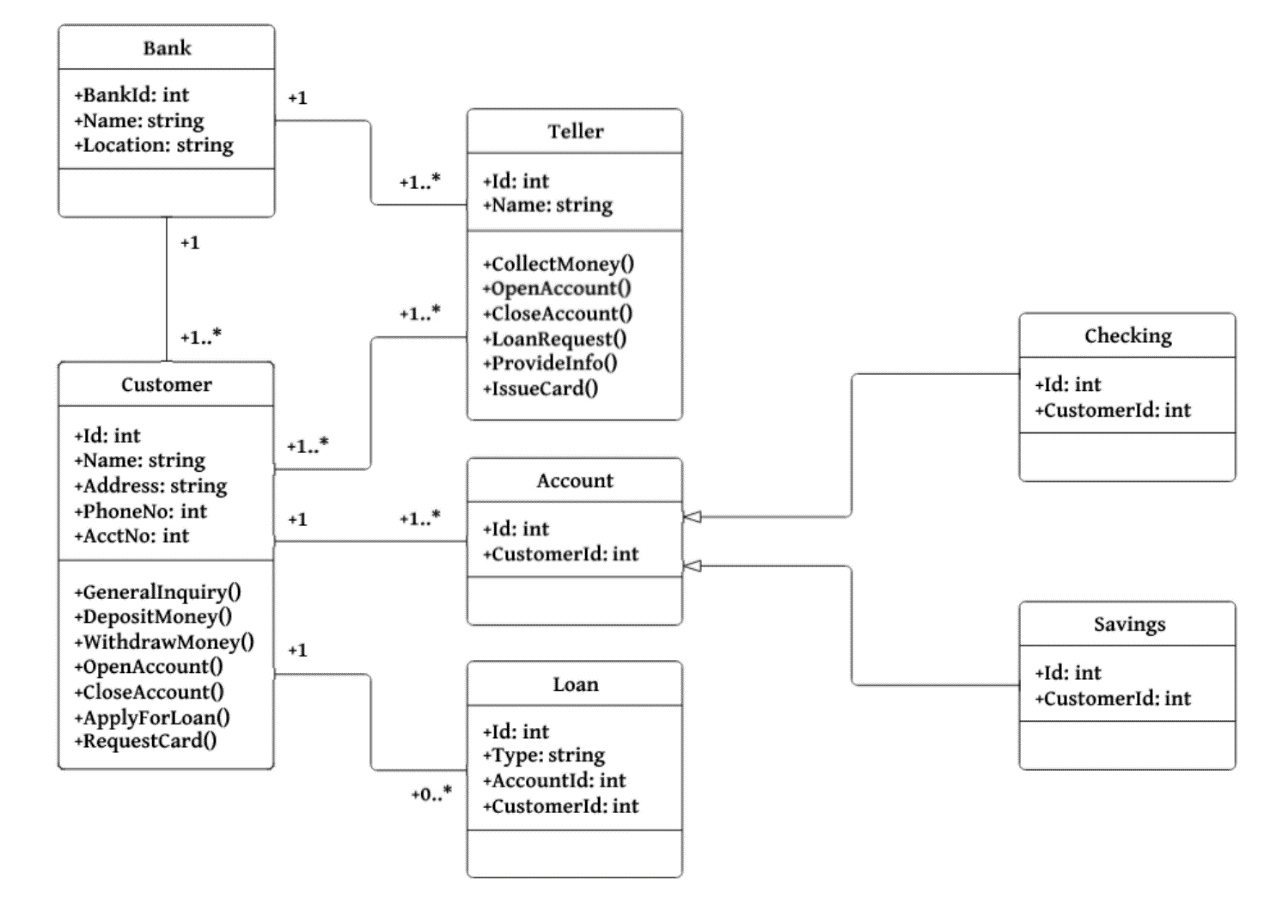


Figure 2.8.1.2: Class Diagram of the system [3]

## User Classes and Characteristics

Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.

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

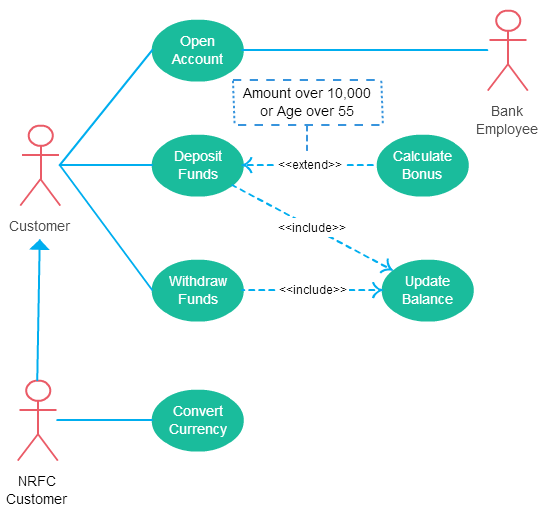


Figure 2.8.1.1: User case Diagram for the application

## 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.

## Design and Implementation Constraints

Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software.

## 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

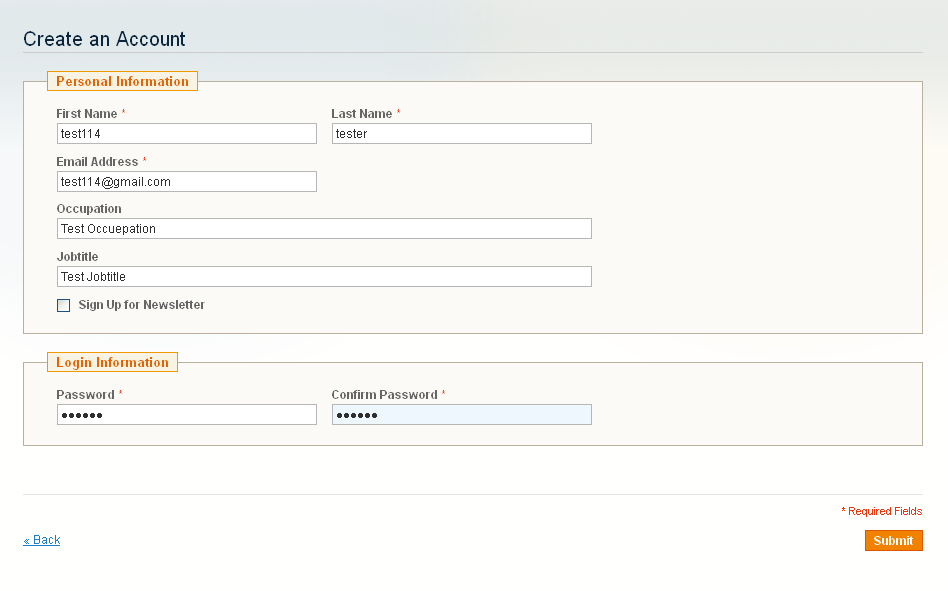


Figure 2.8.1.1: Customer registration Interface

### Invoice Data Filling Interface

Description about the interface goes here

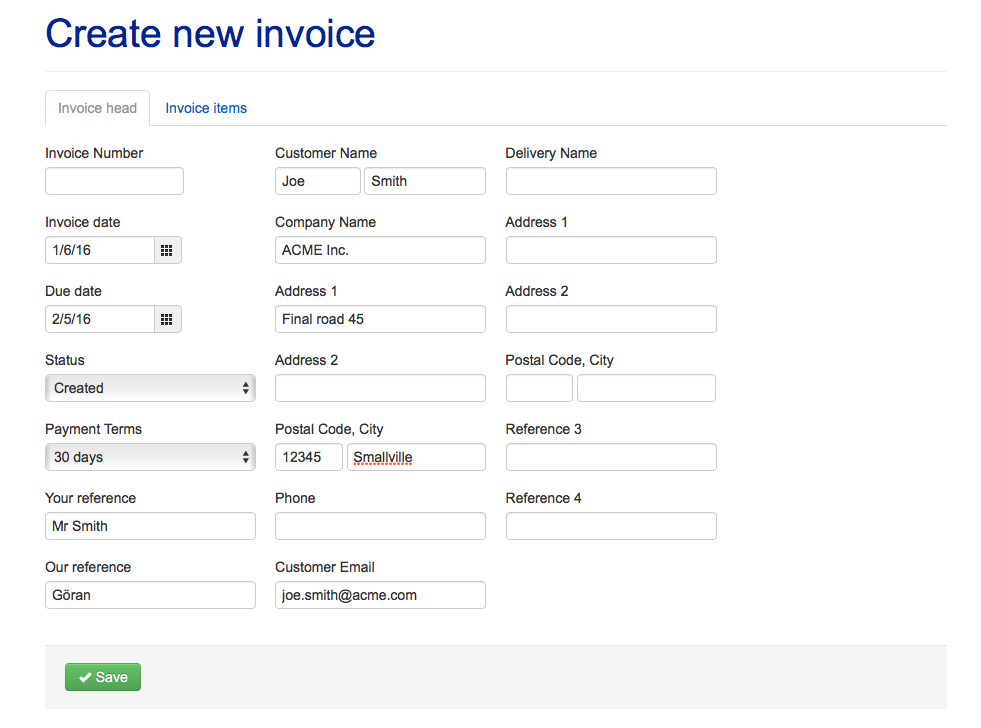


Figure 2.8.2.1: Invoice Interface

## 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)