*Software Requirement Specification*

Project Name

Date

Version 1.0

Revision

|  |  |  |  |
| --- | --- | --- | --- |
| version | Author | Description | Date |
| V1.0 |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Introduction 3](#_Toc136081045)

[General Description 3](#_Toc136081046)

[Requirements 3](#_Toc136081047)

[Business Requirements 3](#_Toc136081048)

[Functional Requirements 3](#_Toc136081049)

[Non-Functional Requirements 3](#_Toc136081050)

[Data Requirements 3](#_Toc136081051)

[Definitions Acronyms Abbreviations 3](#_Toc136081052)

[Technical Specification 3](#_Toc136081053)

[High-level Architectural diagram 3](#_Toc136081054)

[Wire-frames 3](#_Toc136081055)

[Use-Case Diagram 4](#_Toc136081056)

[Use-Case Description. 5](#_Toc136081057)

[Sequence Diagram 7](#_Toc136081058)

[Activity Diagram 8](#_Toc136081059)

[State Diagram 8](#_Toc136081060)

[Class Diagram 8](#_Toc136081061)

[Entity Relation Diagram 8](#_Toc136081062)

[Testing procedure 8](#_Toc136081063)

[Appendix 8](#_Toc136081064)

# Introduction

Describe the problem that this software sets out to solve.

# General Description

Discuss the different technologies that you could use to solve the identified problem. Include the reason why you have settled for that approach that you are going to take to solve that problem. Provide a block diagram to explain the how you see your software working.

|  |  |  |
| --- | --- | --- |
|  |  |  |

# Requirements

A short paragraph introducing this part of the report.

Define your product purpose. Describe what you are building. Set out the product scope by detailing what it will not do.

## Business Requirements

Discuss the results of the initial meeting that took place between the different stakeholders. Identify the different roles and provide a synopsis of what they want the software to do.

These include the high-level statements of goals. They do not include any detail of specific features they just state the problem and business objectives to be achieved.

* Increased revenue
* Reduced expenses
* Reduced Errors
* Improved Customer service

Who will be using the product? Are they a primary or secondary user? What is their role within their organization?  What need does the product need to fulfil for them?

## Functional Requirements

Describe in detail what functionality the software must provide.

## Non-Functional Requirements

Describe the general properties of the system under development define how it should perform. Provide Smart criterial so that there is a baseline measure that can be checked to ensure that these requirements have been met.

## Data Requirements

Discuss any data that may be used in the system. How will this data be used will the system be in compliance with, GDPR Regulations or California consumer privacy act or any other global legal requirements that the product may need to be in compliance. It is better to be prepared during the initial design of the system than having to redesign the software at a later stage.

## Definitions Acronyms Abbreviations

Any Acronyms or abbreviations used in the document should be explained here.

# Technical Specification

General introduction statement

## High-level Architectural Diagram

This high-level diagram should provide a big picture overview of the software under development.

Provide a few sentences to describe the diagram.

## Wire-frames

The inclusion of wireframes will help the reader visualise the document in practice.

high-level wireframes can be used to assist the implementation of the code provide the naming conventions that should be used for buttons, text inputs text area checkboxes. Colour schemes how the GUI should be implemented. Remember the author of this documentation may not be the person that will implement the code provide enough information so that the final product will be as you imagined it to be.

|  |  |
| --- | --- |
|  |  |

## Use-Case Diagram

A diagram of a system

Description automatically generated with medium confidence

Keep use-case diagrams simple the important things to remember.

* Use-Cases are described from the perspective of the main user.
* What’s inside the system box is under your control.
* Use-case name Verb Noun if the most important aspect of this UML it should be obvious to the reader what is its function.
* For every use-case in the use-case diagram there must be a use-case description. DO NOT include ovals in this diagram that parr of the use case description this is a high-level overview of the system.
* The Use-Case name must match in the use case description.
* Incluses Use-Case is part of the DRY Don’t repeat yourself principle and they must always be called.
* Extends Use-Case are use cases that may or may not be called.
* If you find yourself with too many use-cases in the diagram you are defeating the purpose of the diagram. As a rule of thumb keep the number of use-cases to around seven if your system requires more consider splitting in to more than one use-case diagram

Provide a table to keep track of the use-cases and the use-case descriptions.

|  |  |  |
| --- | --- | --- |
| UC-Number | UC-Description |  |
| UC-1 | Login |  |
| UC-2 | Register Reader |  |
| UC-3 |  |  |
|  |  |  |

## Use-Case Description.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: |  | | |
| Use Case Name: | Register User | | |
| Created By: |  | Last Updated By: |  |
| Date Created |  | Last Updated Date |  |

|  |  |
| --- | --- |
| Description: |  |
| Actors:  Primary  Secondary |  |
| Triggers: |  |
| Pre-conditions: |  |
| Post-Conditions: |  |
| Normal Flow: |  |
| Alternate Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Priority: |  |
| Special Conditions: |  |
| Assumptions: |  |
| Notes and Issues: |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: |  | | |
| Use Case Name: | Register User | | |
| Created By: |  | Last Updated By: |  |
| Date Created |  | Last Updated Date |  |

|  |  |
| --- | --- |
| Description: |  |
| Actors:  Primary  Secondary |  |
| Triggers: |  |
| Pre-conditions: |  |
| Post-Conditions: |  |
| Normal Flow: |  |
| Alternate Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Priority: |  |
| Special Conditions: |  |
| Assumptions: |  |
| Notes and Issues: |  |
|  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: |  | | |
| Use Case Name: | Register User | | |
| Created By: |  | Last Updated By: |  |
| Date Created |  | Last Updated Date |  |

|  |  |
| --- | --- |
| Description: |  |
| Actors:  Primary  Secondary |  |
| Triggers: |  |
| Pre-conditions: |  |
| Post-Conditions: |  |
| Normal Flow: |  |
| Alternate Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Extends: |  |
| Priority: |  |
| Special Conditions: |  |
| Assumptions: |  |
| Notes and Issues: |  |
|  |  |

## Sequence Diagram

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

Create the sequence diagram and write a short paragraph explaining what is happening the sequence diagram should be covered as part of the flows in the use-case description. iF necessary revisit the use-case description or amend the sequence diagram.

## Activity Diagram

## State Diagram

Use any UML diagrams that will assist the reader with the understanding of the software under development.

## Class Diagram

A screenshot of a computer

Description automatically generated

## Entity Relation Diagram

When using a database provide an ERD diagram this will assist the developer when it comes to coding the project and provide a better understanding of the data that will be held by the system. Refer back to the data classification described earlier and see how. Security features for this data can be implemented.

# Testing procedure

TBA

# Appendix

TBA

The Business Model Canvas can assist in the development of the software but can be removed before submitting.

A picture containing text, screenshot, diagram, rectangle

Description automatically generated