**Software Requirement Specification (SRS)**

1. INTRODUCTION

##### *NOTE: General background and reference information*

1.1 Purpose of this Document

Full description of the main objectives of this SRS document in the context of this project (e.g. “This SRS describes the functions and performance requirements allocated to the XYX system. The XYX is a stand-alone component of a system ”)

1.2 Scope of the Development Project

Identifies the product to be developed by name and function.

Lists limitations (if any), highlights distinct features.

2. GENERAL DESCRIPTION

***NOTE: This section gives an “executive overview” and is very client-oriented.***

2.1 Glossary (Definitions, Acronyms, and Abbreviations)

Include any specialized terminology dictated by the application area or the product area. This will help the reader understand the rest of the text. Be sure to alphabetize the terms!

2.2 User Characteristics

This section considers the needs of the anticipated users. List critical characteristics of the system’s human interfaces based on the anticipated users’ characteristics. “Who will use the system”?

2.3 Product Perspective

* If the product is part of a larger product, then identify its interface to the other products.
* If the product uses existing hardware, describe the hardware.
* Any other relevant information.

2.4 Overview of Functional Requirements

A short description of the functions to be performed by the software, i.e. what the product should do. This description must be in a form understandable to users, operators, and clients. The detailed requirements specifications are left to Section 3.2 in this SRS. ***Number the Functional Requirements in a systematic manner so your team can refer to them in Section 3.2 of the SRS, in the SDD, and in the testing documents.*** This section should *not* be design-oriented, a common mistake.

2.5 General Constraints and Assumptions

This can include hardware limitations or requirements, the amount of memory available, response times, policies, interfaces to other application software, networks, environmental limitations, and compliance with relevant standards. This section can also provide guidance in situations when there may be more than one implementation strategy. Examples: “The product will only work with certain operating systems or a particular network environment.” “The product must be Web-based.”

2.6 User View of Product Use

This section will provide a user’s-eye-view of the product. This may include aspects such as narrative to describe the setting, sketches to show possible appearance of the screen, samples of the data that is stored, entered, or output, and scenarios that demonstrate the product in operation.

###### NOTE: Technical information needed to design the software

3.1 Interface Requirements

3.1.1 User Interface

The user interface will be Java-capable web browser

3.1.2 Hardware Interface

A work station connected to the internet plus mouse and mousepad.

3.1.3 Software Interface

Java-capable web browser with access to the internet, the Java Development Kit (JDK) from Sun Microsystems or Integrated Development Environment (IDE), and a text editor for preparing HTML files.

3.1.4 Communication Interfaces

Internet access

3.2 Detailed Description of Functional Requirements

3.2.1 Template for describing functional requirements

This lists the exact template your SRS will apply in describing each of the functional components that were identified in Section 2.4. This section should have ***(for each requirement)*** at least the following:

• Purpose / description

• Inputs: which inputs; in what form/format will inputs arrive; from what sources input will be

derived, legal domains of each input element

• processing: describes the **outcome** rather than the **implementation**; include any validity

checks on the data, exact timing of each operation (if needed), how to handle unexpected or abnormal situations.

• Outputs: the form, shape, destination, and volume of the output; output timing; range of

parameters in the output; unit measure of the output; process by which the output is stored or destroyed; process for handling error messages produced as output

3.2.2 Data Dictionary

Data dictionary supplies information such as data item, data type, how data is used.

3.3 Non-Functional requirements

Issues such as number of connections to the system, number of simultaneous users, response time, number of files, size of files and tables, number of files, size of files and tables, number of transactions per interval, security, and performance issues.

4. Object Oriented Analysis (OOA) -UML

4.1. For OOA, Complete the following:

* + 1. Draw use case diagram.
    2. Describe most Use Cases (the more dynamic and interesting ones) as shown in the example below:

Use Case Name: Buying a Book Online

Use Case Number: UC32

Authors: John Adams

Actors: Customer (initiator), credit- card authorization service, book seller

Overview: This use case captures the process of purchasing one or more books from an online book seller

References: R23, R34, and R45.

Related Use Cases: UC11

Typical Flow Description: (include precondition & post-condition)

Alternative Flow Description: (include precondition & post-condition)

* + 1. List potential / analysis classes based on the problem statement and use cases.
* **Provide clean drawings of your models.**
* **Include explanatory text as needed.**

5. Special remarks or comments

6. References or resources used