Software Requirements Specification

<COMP401 Note: You can remove this page and add your own cover including your team name (logo), product details, and the members of you team>

<COMP401 Note: Please leave all section in the document. If you don't/can't fill any section, leave it there, empty and shortly explain why you did not filled it>

This document outline is based on the IEEE Standard 830-1993 for Software Requirements Specifications.

This document was created in part by Steve Mattingly (smattingly@computer.org).

This document should specify what functions are to be performed on what data to produce what results at what location for whom.

A properly written SRS limits the range of valid designs, but does not specify any particular design.

A good SRS is

- Correct (accurately captures the "real" requirements)
- Unambiguous (all statements have exactly one interpretation)
- Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
- Consistent
- Ranked for importance and/or stability
- Verifiable (avoid soft descriptions like "works well", "is user friendly"; use concrete terms specify measurable quantities)

This document is adapted from *Software Project Survival Guide* by Steve McConnell (Microsoft Press, 1998). The document template used to create this document, related documents, plans, and other materials can be downloaded from the survival guide website at http://www.construx.com/survivalguide/.

- Modifiable (evolve the SRS only via a formal change process, preserving a complete audit trail of changes)
- Traceable (cross reference with source documents and spawned documents)

The paragraphs written in the "Comment" style are for the benefit of the person writing the document and should be removed before the document is finalized.

VERSION: DRAFT

SEPTEMBER 11, 1998, CHANGES: AUGUST 2013

Revision Chart

This chart contains a history of this document's revisions. The entries below are provided solely for purposes of illustration. Entries should be deleted until the revision they refer to has actually been created.

The document itself should be stored in revision control, and a brief description of each version should be entered in the revision control system. That brief description can be repeated in this section. Revisions do not need to be described elsewhere in the document except inasmuch as they explain the development plan itself.

Version	Primary Author(s)	Description of Version	Date Completed
Draft	TBD	Initial draft created for distribution and review comments	TBD
Preliminary	TBD	Second draft incorporating initial review comments, distributed for final review	TBD
Final	TBD	First complete draft, which is placed under change control	TBD
Revision 1	TBD	Revised draft, revised according to the change control process and maintained under change control	TBD
Revision 2	TBD	Revised draft, revised according to the change control process and maintained under change control	TBD
etc.	TBD	TBD	TBD

CONTENTS

1. INTRODUCTION

"Welcome to the future"

1.1 Description of the client

Our client was found by our company, through people we know and was stopping by that shop. First of all, the shop is selling books and stationery. At the same time, the shop gives special importance to the copy-center, since the copy-center is the most profitable part of the company. In this section of the company, their clients can print professional business cards, copies or laminations of documents. Furthermore, our client inside the company has professional painting materials such as pens, pencils, notepads, files etc. Finally, they have a different section for the books that they sell.

1.2 Scope

The main scope for this cooperation with the client is to have the opportunity for better control of their storage and supplies. Also, they will have the ability to order books that they don't have at the company and re-order past orders, easy and fast. In addition to this, they will be a separate database with loyal customers, that will take points each time that their customer makes purchases.

1.3 Definitions, Acronyms, and Abbreviations

ClientTsingDel - Name Of Our Software Based Project

GitHub - Cloud Service regarding programming scripts

Google Drive - Cloud Service

1.4 References

Did not need to reference any book since it is a work by our own!

2. Overall Description

"Welcome to the future"

2.1 Product Perspective

The product developed can be considered independent but it also needs the user to add / remove information etc. For instance, the product will automatically inform the customers for any updated when available. Furthermore, it is self-contained due to the fact that the information regarding the books, painting materials and anything else included, are gonna be stored inside the software through a database which will be created. So that way the program will know about what needs to be done and when. Compared now with other similar products, the only thing that can be said is that it is a typical program service for a specific purpose of managing data and importing/exporting data when needed and suitable towards the customer.

2.1.1 System Interfaces

List each system interface and identify the related functionality of the product.

2.1.2 User Interfaces

The User of the system will be provided with the Graphical User Interface (GUI), there is no need for a command line interface for any use or function for the program.

There is only administrator (Staff) access.

The User will be able to scan products (books), and also interact with the system regarding the products, for example the user can check or change the availability of the sock for a specific product.

Also, the User can add at the cash-out the ID number of the customer and add it to the receipt for points and send informative emails and messages.

2.1.3 Hardware Interfaces (maybe we need to add our own hardware as well)

Regarding the hardware requirement, our client will need:

- A basic computer with basic characteristics.
- Internet access, modem/router.
- A barcode reader (POS).
- Lan (Local Area Network) cable.

2.1.4 Software Interfaces:

DATABASES

- Microsoft Access 2016
- Version 16.0
- Microsoft 365 Suits

The purpose for the use of databases is to keep track of the books purchased, "rented?" or requested to copy and any other material included in the shop. The database will also includes an ID for every customer, every book and for any other material such as the painting ones (UNIQUE ID NUMBERS PROVIDED). Lastly, it will include prices.

SOFTWARE SYSTEM

- Visual Studio 2022
- Version 17.0.5
- Microsoft
- For the object oriented programming language used is C#, using a visual studio extension pack, which goes by the name "**.NET Desktop development**".

.NET Desktop development Used For

- windows forms
- Console Applications
- .NET framework

For each interface, discuss the purpose of the interfacing software, and define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

2.1.5 Communications Interfaces

Regarding communication interfaces, we have a LAN (local area network) as well as an internet service provider connected to the LAN, who of course provides access to the internet and so for the several machines connected to LAN (through ethernet cables) that can communicate with each other.

2.1.6 Memory Constraints

For memory constraints the minimum requirements that we will need are minimum 16 GB of RAM on the pc and at least 3 internal storage devices(SSD,HDD). Also every 2 weeks the database will be updated automatically in the Cloud.

2.1.7 Site Adaptation Requirements

The interface that we are creating is used by all the employers of the company. Its very simple and easy to use so all the employees no matter their educational background can easily learn to use it properly without any previous knowledge with the software.

2.2 User Characteristics

As a company, we involve people with a high educational level and expertise in both programming and designing. For the reason above, we strive to make good and respectable results. Our client might hold a higher educational level of degree with a higher expertise, but the people working for him not that much. Hence, our objective is to create a software that's applicable and easily workable for all! That way we can ensure a positive and productive corporation among us. Same stands for the technical expertise part.

2.3 Constraints

Describe any other items that will constrain the design options, including

- regulatory policies
- hardware limitations
- interfaces to other applications
- parallel operation
- audit functions
- control functions
- higher-order language requirements
- signal handshake protocols
- reliability requirements
- criticality of the application
- safety and security considerations

2.4 Assumptions and Dependencies

List factors that affect the requirements. These factors are not design constraints, but areas where future changes might drive change in the requirements.

3. SPECIFIC REQUIREMENTS

This section should describe all software requirements at a sufficient level of detail for designers to design a system satisfying the requirements and testers to verity that the system satisfies requirements.

Every stated requirement should be externally perceivable by users, operators or other external systems.

At a minimum, these requirements should describe every input into the software, every output from the software, and every function performed by the software in response to an input or in support of an output.

All requirements should be uniquely identifiable (e.g., by number).

The remainder of this sample document is organized according to A.5 Template of SRS Section 3 Organized by Feature shown in the Annex of Std 830-1993. For alternative organizational schemes by system mode, user class, object, stimulus, functional hierarchy, and combinations, see the standard, which is available from IEEE Standards Office, P.O. Box 1331, Piscataway, NJ 08855-1331.

No text is necessary between the heading above and the heading below unless otherwise desired.

3.1 External Interface Requirements

Provide a detailed description of all inputs into and outputs from the software. This section should complement the interface descriptions under section 2.1 and should not repeat information there. Include both content and format as follows:

- name of item
- description of purpose
- source of input or destination of output
- valid range, accuracy, and/or tolerance
- units of measure
- timing
- relationships to other inputs/outputs
- screen formats/organization
- window formats/organization
- data formats
- command formats
- end messages

These requirements may be organized in the following subsections.

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communications Interfaces

3.2 Software Product Features

3.2.1 Feature 1

Repeat subsections at this level and below for each feature.

3.2.1.1 Purpose

3.2.1.2 Stimulus/Response Sequence

3.2.1.3 Associated Functional Requirements

3.2.1.3.1 FUNCTIONAL REQUIREMENT 1

Repeat subsections at this level and below for each associated functional requirement.

Each functional requirement may be described in natural language, pseudocode, or in four subsections as follows. Functional requirements include:

- validity checks on inputs
- exact sequencing of operations
- responses to abnormal situations, including error handling and recovery
- effects of parameters
- relationships of inputs to outputs, including input/output sequences and formulas for input to output conversion

3.2.1.3.1.1 Introduction

3.2.1.3.1.2 Inputs

3.2.1.3.1.3 Processing

3.2.1.3.1.4 Outputs

3.3 Performance Requirements

Specify static and dynamic numerical requirements placed on the software or on human interaction with the software.

Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.

Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form.

3.4 Software System Attributes

The following items provide a partial list of system attributes that can serve as requirements that should be objectively verified.

3.4.1 Reliability

Specify the factors needed to establish the software's required reliability.

3.4.2 Availability

Specify the factors needed to guarantee a defined level of availability.

3.4.3 Security

Specify the factors that will protect the software from accidental or malicious access, misuse, or modification. These factors may include:

- cryptography
- activity logging
- restrictions on intermodule communications
- data integrity checks

3.4.4 Maintainability

Specify attributes of the software that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

3.4.5 Portability

Specify attributes of the software that relate to the ease of porting the software to other host machines and/or operating systems.

3.5 Logical Database Requirements

Specify the requirements for any information that is to be placed into a database, including

- types of information used by various functions
- frequency of use
- accessing capabilities

- data entities and their relationships
- integrity constraints
- data retention requirements

3.6 Other Requirements

4. APPENDICES

Include supporting detail that would be too distracting to include in the main body of the document.