**Software Requirements Specification**

**for**

**<**Library Software**>**

**Prepared by <**

**1- Joseph Ashraf**

**2- Andrew Ezzat**

**3- Ziad Ali**

**4- Peter Nassrat**

**>**

**<12 / 2022>**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Intended Audience and Reading Suggestions 1

1.3 Product Scope 2

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 3

2.6 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Software Interfaces 3

**4. System Features 4**

4.1 ERD 4

4.2 Use-case scenario 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

# **Introduction**

## **Purpose**

The software requirements described in this document are introduced specifically to the managers of the system and developers and testers that are responsible for the development process and the marketing staff that will introduce the software to the user and to the users that will use the software.

This document describes the requirement specifications of the first version -1.0- of the software that will be introduced to the market as a desktop application that integrates with its private database.

This document will cover the main services that the software will introduce in version -1.0- and it will not cover the subsystems used in the software that they are prepared in advance.

The main product of this system is desktop software that integrates with other system components to help with the management of the libraries’ services.

This software will introduce two main services in the first version -1.0- of it which are buying books and borrowing books.

There will be additional services in the new versions of the system.

## **Intended Audience and Reading Suggestions**

This document targets three types of people that they are:

* Managers -Responsible for management and system approval-
* Developers and Testers -Responsible for the development process-
* Marketing staff -Responsible for deploying process-
* Users -That will use the system-

This document covers five sections that describe the system, The first section is an introduction section that introduces you to the system and the other four sections will describe the whole system in detail.

These are suggestions to each type of reader that read this document that will help you in reading this document.

* For the managers, it is necessary to read the whole document by the order of the document’s sections. If there is any ambiguity in it, they should contact the documentation writers’ team.
* For the developers and testers, their role is important and they should read it in a complete way as the managers. If there is any ambiguity in it, they should contact the documentation writers’ team.
* For the marketing staff, they should read the first two sections in a complete way and skip the remaining sections, but it is good if they read the remaining sections briefly.
* For the users, they should read the first two sections and they can skip the remaining sections. If there is any problem with reading this document they should contact the support team.

## **Product Scope**

The system introduces a desktop application for the management process of services that are introduced by the libraries, the purpose of the software is to help libraries' owners to manage the services introduced by the library easily and in a computerized way that helps in increasing the quality of the libraries' services.

The goal of the software is to help library owners to use the power of the computer to manage the libraries' services that provide high-speed search and the ability to store enormous amounts of data.

This software completes the team's goal which is to help improve the quality of services introduced by libraries that encourage the industry of printing and selling books that help in state progress.

# **Overall Description**

## **Product Perspective**

The product of the system is new and independent software, this software will start from scratch.

It will be self-contained software, there aren't other software that are depending on this software.

It provides a lot of features that will make the libraries more advanced.

It will make the services that library workers introduce more efficient and faster to them, rather than doing them manually. Thus, this saves a lot of time and reduces errors.

## **Product Functions**

The main functions that the application will provide in the first version -1.0- are:-

* User Registration
* New books entry
* Buy operation (provide search on books)
* Borrow operation (provide search on books)
* Return borrowed books operation
* Customer information Registration
* Find and display a customer’s purchases history

## **User Classes and Characteristics**

This system will be used by only one type of users:

Workers at the library (librarians) who will have the ability to manage all services related to borrowing, buying books and payment process.

## **Operating Environment**

**Hardware**: This system can run on laptops and desktop pc with minimum requirements

(2GB ram, 5GB for storage).

It requires a printer to print bills that includes all the details of the payment process.

**Software**: Software is designed to run on any platform above Microsoft Windows 7 (32bit & 64bit).

## **Design and Implementation Constraints**

* This software should work only on Windows operating system and cannot work on any other systems like MacOS or Linux.
* This software is developed by C# language and must include a SQL database to record past orders, requests and past customers’ history.
* The customer cannot borrow more than one book at a time.

## **Assumptions and Dependencies**

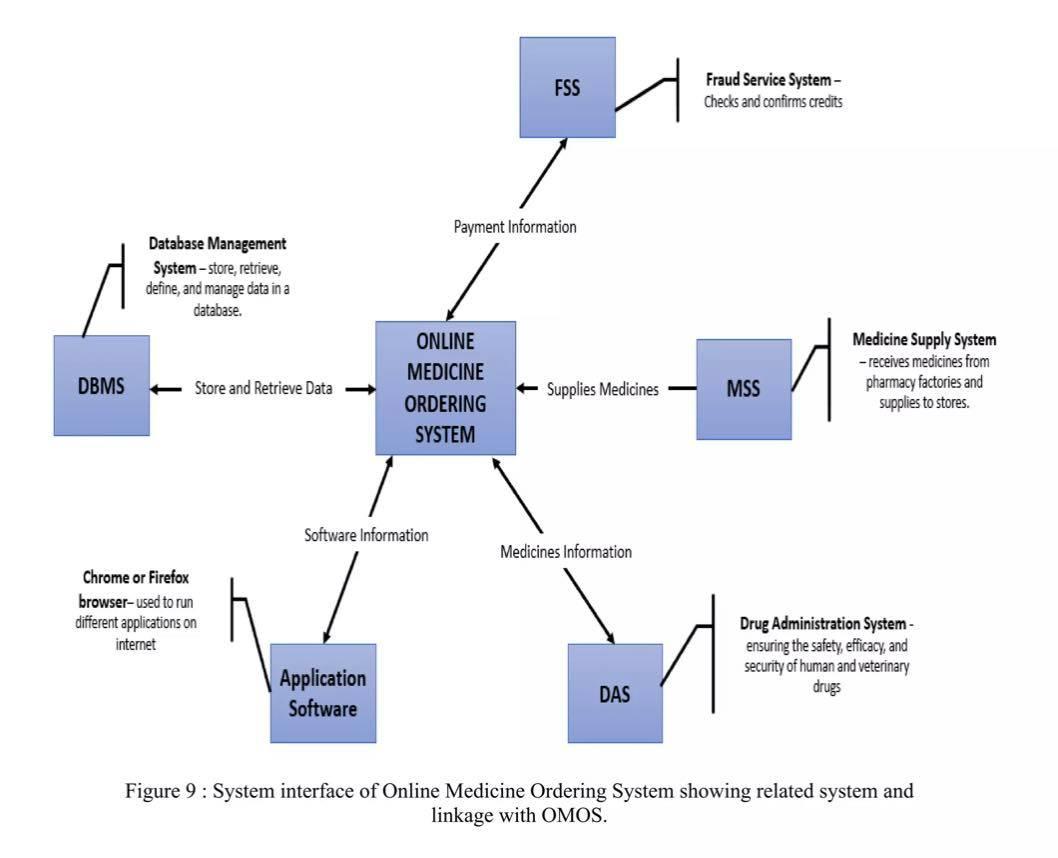
* The maintenance of the software after delivery is not free and it is for a limited period of time. The fees and the due date for the available maintenance is determined in contract.
* The software cannot be distributed by the customer by any means, free or paid. The software is owned by the developer only and has the only right to sell it.

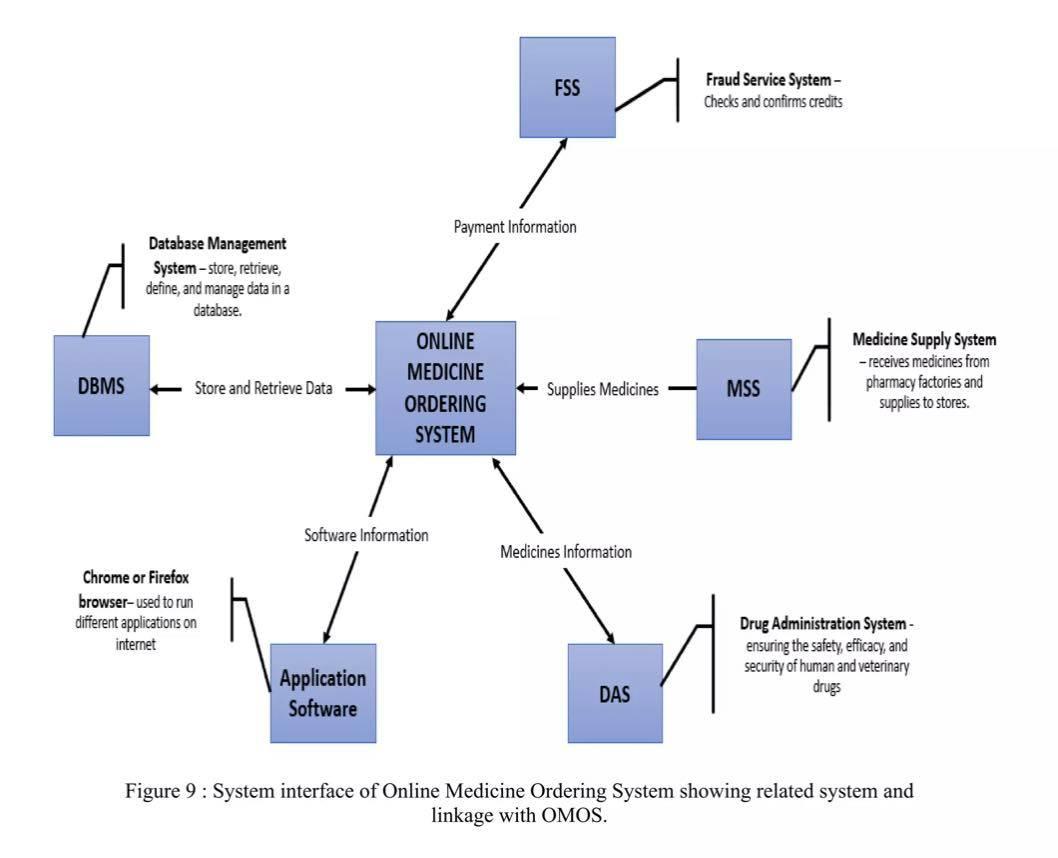
# **External Interface Requirements**

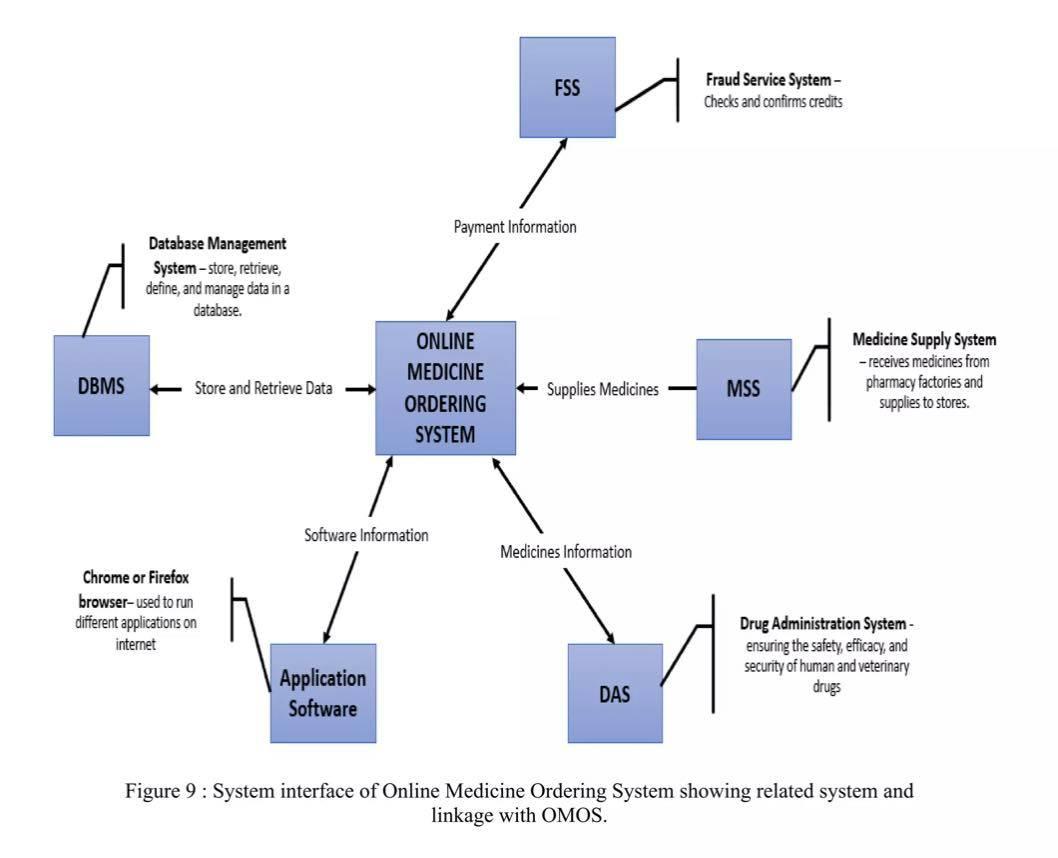
## **User Interfaces**

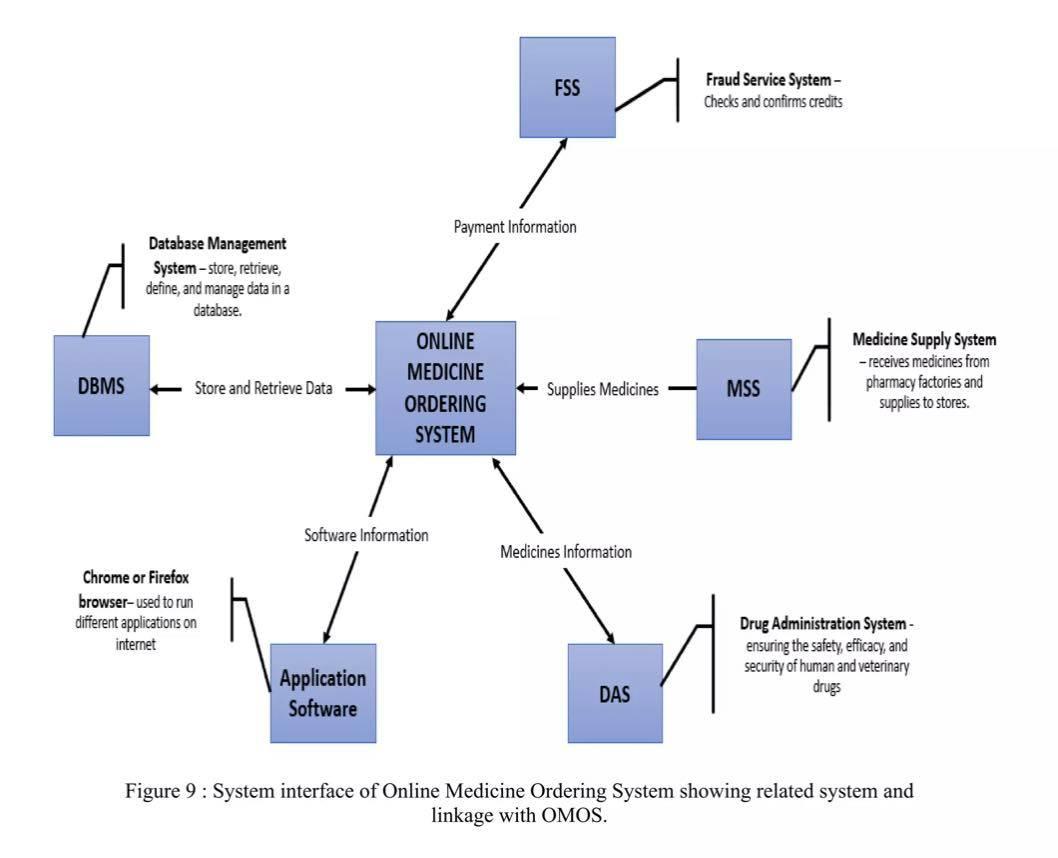
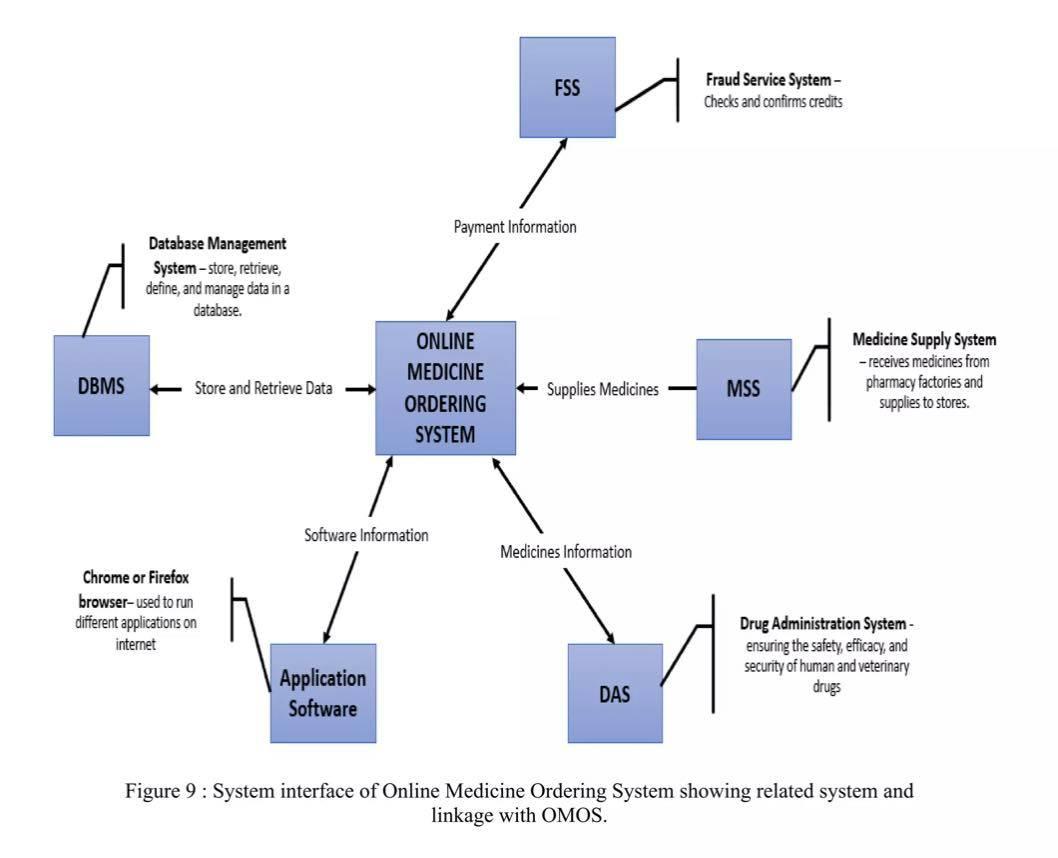
## **Software Interfaces**

<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.>

**

**





# **System Features**

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## **System Feature 1**

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## **System Feature 2 (and so on)**

Examples,

* 1. ER Diagram
  2. Use case Diagram
  3. Use case Scenario
  4. Activity Diagram
  5. ….. So on

# **Other Nonfunctional 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.>

Example,

* The log in information shall be verified within five seconds causes’ efficiency of the system.
* Returning query results within five seconds makes search function more accurate.

## **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.>

Example,

* Maintaining backups ensure the system database security.
* System can be restoring in any case of emergency.

## **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.>

Example,

* Access to the various subsystems will be protected by a user log in screen that requires a user name and password. (Owner, manager, secretary, receptionist)

## **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.>

# **Other Requirements**

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

**Appendix A: Glossary**

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

**Appendix B: Analysis Models**

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

**Appendix C: To Be Determined List**

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>