3- Class oriented models: that represent object-oriented classes (attributes and operations) and the manner in which classes collaborate to achieve system requirements.

4- Behavioral models: that depicts how the software behaves as a consequence of external "events".

In some cases, a clear distinction between the analysis and design models is possible. While in other cases, the analysis model slowly blends into the design, and no clear distinction is obvious.

The design model elements use many of the same UML diagrams used in the analysis model. The difference is that these diagrams are refined and elaborated as part of the design, more details are provided, and more focus is on components and interfaces.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

After the analyst has gathered all the required information regarding the software to be developed and has removed all incompleteness and inconsistencies from the specification, he starts to systematically organize the requirements in the form of an SRS document. Software requirements specification is the final production of the requirements stage of the software development process.

Among all the documents produced during a software development life cycle, the SRS document is probably the most important document and is the hardest to write.

Users of SRS Document

Usually, a large number of different people need the SRS document for very different purposes. Some of the important categories of users of the SRS document and their needs for use are as follows:

1- Users, customers, and marketing personnel: These stakeholders need to refer to the SRS document to ensure that the system as described in the document will meet their needs.

2- Software developers: The software developers refer to the SRS document to make sure that they are developing exactly what is required by the customer.

3- Test engineers: The test engineers use the SRS document to understand the functionalities, and based on this write the test cases to validate it's working. They need that the required functionality should be clearly described, and the input and output data should have been identified exactly.

4- User documentation writers: The user documentation writers need to read the SRS document to ensure that they understand the features of the product well enough to be able to write the users' manuals.

5- Project managers: The project managers refer to the SRS document to ensure that they can estimate the cost of the project easily and that it contains all the information required to plan the project.

6- Maintenance engineers: The SRS document helps the maintenance engineers to understand the functionalities supported by the system. A clear knowledge of the functionalities can help them to understand the design and code.

Characteristics of a Good SRS Document

1- brief: The SRS document should be brief and at the same time unambiguous, consistent, and complete.

2- Implementation-independent: The SRS should be free of design and implementation decisions unless those decisions reflect actual requirements. It should only specify what the system should do and not how to do these. This means that the SRS document should specify the externally visible behavior of the system and not discuss the implementation issues.

3- Traceable: It should be possible to trace a specific requirement to the design elements that implement it and vice versa. Traceability is also important to verify the results of a phase with respect to the previous phase and to analyze the impact of changing a requirement on the design elements and the code.

4- Modifiable: Customers frequently change their requirements during software development due to a variety of reasons. Therefore, in practice, the SRS document undergoes several revisions during software development. Also, an SRS document is often modified after the project completes to accommodate future enhancements and evolution.

5- Identification of response to undesired events: The SRS document should discuss the system responses to various undesired events and exceptional conditions that may arise.

6- Verifiable: All requirements of the system as documented in the SRS document should be verifiable. This means that it should be possible to design test cases based on the description of the functionality as to whether or not requirements have been met in an implementation.

7- A requirement such as “the system should be user-friendly” is not verifiable. On the other hand, the requirement—“When the name of a book is entered, the software should display whether the book is available for issue or it has been loaned out” is verifiable. Any feature of the required system that is not verifiable should be listed separately in the goals of the implementation section of the SRS document.

A good SRS document should properly categorize the requirements into different sections:

Functional Requirements

The functional requirements capture the functionalities required by the users of the system. It is useful to consider software as offering a set of functions {fi} to the user. These functions can be considered similar to a mathematical function f: I → O, meaning that a function transforms an element (i) in the input domain (I) to a value (oi) in the output (O).

Non-Functional Requirements

The non-functional requirements capture those requirements of the customer that cannot be expressed as functions like Performance, Reliability, Security, and Portability.

EXAMPLE:

Software Requirements Specification for ***Library Management System***.

TABLE OF CONTENTS

1 .Introduction

1.1 Purpose

1.2 Scope

1.3 Intended Audience Definitions, and abbreviations

2 .Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 Operating Environment

2.4 User Characteristics

2.5 Design and Implementation Constraints

3 .External Interfaces Requirements

3.1 User Interfaces

3.2 Hardware Interfaces

3.3 Software Interfaces

4 .Functional Requirements

5. Behavioral Requirements

6. Non-Functional Requirements

1 .Introduction

1.1 Purpose

The purpose of this document is to describe the Library Management System.

1.2 Scope

Library Management System is basically updating the manual library system into an internet-based application so that the users can know the details of their accounts, the availability of books, and the remaining time for borrowing .

1.3 Audience Definitions and Abbreviations

1.3.1 Audience Definitions

The intended readers of this document are the developer, testers, library owners, managers, and coordinators .

Any suggested changes to the requirements listed in this document should be included in the last version of it so it can be a reference for developing and validating teams.

1.3.2 Abbreviations

Abbreviations	Meaning
MS SQL	Microsoft Structured Query Language
ISBN	International Standard Book Number
IEEE	Institute of Electrical and Electronics Engineers

1.4 References

- IEEE 830-1998 standard for writing SRS documents .
- I Sommerville, Software Engineering, 8th ed, Addison-Wesley, 2007.

2 .Overall Description

2.1 Product Perspective

Library Management System is a replacement for the ordinary library management systems which depend on paperwork for recording books and users information .

2.2 Product Functions

2.2.1 Administrators

- ✓ Admin should be able to insert, modify and delete books .
- ✓ Can accept or reject a new user according to the library policy or payment methods .
- ✓ Increase the period for borrowing a book.
- ✓ Can get the information (status report) of any member who has borrowed a book .
- ✓ Add and edit book categories and arrange books by categories .
- ✓ Add and edit authors and publishers' information .
- ✓ Can send lateness warnings to people who have exceeded the deadline date .
- ✓ Can record books returned by users.

2.2.2 Normal Users (Library Members)

- ✓ The member should be provided with updated information about the book's catalog .
- ✓

- ✓ Members are given access to check their account information and change it .
- ✓ Members have the ability to search through books by subject, title, authors, or any information related to the book .
- ✓ Can extend the period of borrowing books according to the library policy .

2.3 Operating Environment

The Library Management System shall operate in all famous browsers, like Google Chrome, Mozilla, and Firefox.

2.4 User Characteristics

Users of this Library Management System are members ,librarians and administrators.

Members and librarians are assumed to have basic knowledge of computers and Internet browsing. Administrators of the system should have more knowledge of the internal modules of the system and be able to correct small problems that may arise like power failures.

2.5 Design and Implementation Constraints

- ✓ The information of all users, books, and libraries must be stored in a database .
- ✓ MS SQL Server will be used as an SQL engine and database .
- ✓ Users may access from any computer that has Internet browsing capabilities and an Internet connection .
- ✓ Users must have their correct usernames and passwords to enter into their online accounts and do actions .

3. External Interfaces Requirements

3.1 User Interfaces

Login Interface, search, and Librarian's Control Panel.

3.2 Hardware Interfaces

The existing Local Area Network (LAN) will be used for collecting data from the users and also for updating the Library Catalogue.

3.3 Software Interfaces

- ✓ Browser to load and view the web pages
- ✓ Operating System

4. Functional Requirements

4.1. Librarian:

Login/ Insert book/ Delete book / modify book /Delete member/ Return book.

4.2 Normal User:

Login or Register/ Search for book/Extend borrowing deadline.

5. Behavioral Requirements

Like Use case diagram.

6 .Non-functional Requirements

6.1 Error handling

- Library Management System shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period .

6.2 Performance Requirements

- The system shall accommodate high number of books and users without any fault .
- Safety Requirements
- System use shall not cause any harm to human users Security Requirements
- System will use secured database
- System will have different types of users and every user has access constraints.