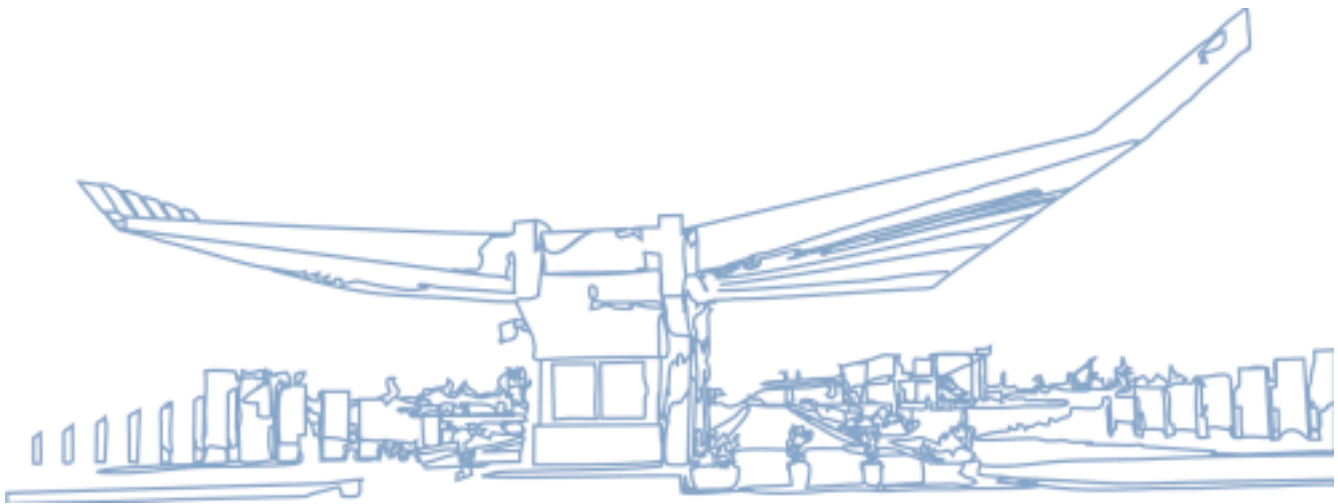


INTRODUCTION TO SOFTWARE ENGINEERING

LIBRARY MANAGEMENT SYSTEM



Team Members:

Layan Alateeq [1]

Tea Disho [2]

Era Fejza [3]

Rosilda Bajrami [4]

Library Management System Requirements Specification



❖ **Table of Contents**

1. EXECUTIVE SUMMARY 3

1.1 PROJECT OVERVIEW 3

1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION 3

2. PRODUCT/SERVICE DESCRIPTION 3

2.1 PRODUCT CONTEXT 3

2.2 USER CHARACTERISTICS 3

2.3 ASSUMPTIONS 3

2.4 CONSTRAINTS 3

2.5 DEPENDENCIES 4

3. REQUIREMENTS 4

3.1 FUNCTIONAL REQUIREMENTS 5

3.2 NON-FUNCTIONAL REQUIREMENTS 5

3.2.1 User Interface Requirements 5

3.2.2 Usability 5

3.2.3 Performance 6

3.2.4 Manageability/Maintainability 6

3.2.5 Security 8

3.2.6 Standards Compliance 8

3.2.7 Other Non-Functional Requirements 9

3.3 DOMAIN REQUIREMENTS

4. DESIGN THINKING METHODOLOGIES

10 4.1 Negotiation

11 4.2 Empathy

11 4.3 Noticing

11 4.4 GUI 11

5. SOFTWARE DESIGN

12 5.1 Use Case

12 5.2 State Diagram

12 5.3 Class Diagram 12

APPENDIX 13

APPENDIX A. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS 13 APPENDIX B. REFERENCES 13 APPENDIX C. ORGANIZING THE REQUIREMENTS 13
1. Executive Summary

Library Management System Requirements Specification

1.1 Project Overview

Describe this project and its intended audience.

The library management system is a software created to satisfy the needs of both users and librarians. It is a system that easily helps librarians on accessing and edit every sort of data inputted, with all their respective details but also navigate through the functions specified and the library's resources.

The need for such a system was first felt after the overflow of information available. Local libraries started having such a hard time keeping track of it all manually, so it was crucial to develop such kind of software to help them organize and manage their resources in a quicker and more effective way. As time went by, the system consequently got updated and it made different processes easier such as the process of borrowing and returning books easier, keeping track of the resources in such a way to minimize the possibility of loss or mistakes. It also was able to revolutionize many aspects such as the one where you now are able to include digital resources alongside physical books.

The library management system offers a wide range of features, making it smooth and user-friendly. By using their components, such as book availability, quantity, and users ID or phone number, etc., users may easily search for books and other users. This approach makes sure users can find their requested books quickly, promoting effective use of the library's resources. Users may handle their borrowing demands without needless delays or problems of any kind by having access to these types of operations. Additionally, the system sends a message to users to remind them of their due date, guaranteeing easier management of borrowed goods.

A vital tool for modern libraries, the library management system serves as a link between librarians and users. Its efficient features make library resources easily accessible and improve the entire library experience thanks to its user-friendly design and useful features. This system provides smooth and seamless interactions, resulting in an orderly and delightful library environment/community by bringing together librarians and users.

1.2 Purpose and Scope of this Specification

Describe the purpose of this specification and its intended audience. Include a description of what is within the scope and what is outside of the scope of these specifications.

The purpose of specifying a library management system is to define the requirements, features, and functionality of the system. It's a software that manages to draw an outline of what the system must accomplish, how it should respond and the way of implementation.

Library Management System Requirements Specification

The scope of the specification for a library management system includes:

1. **Functional Requirements:** These are requirements that describe the features and functions it's going to provide. It covers such activities as book searching, user management, borrowing and returning books, overdue dates etc.
2. **User Roles and their permissions:** In the library system you have options that determine the different roles you have; so if you are a librarian or just a simple user
3. **Data Management:** It refers to each type of information such as info about the book, by authors, users, dates, phone numbers, and other relevant data.
4. **User Interface and User Experience:** In here we specify the outline of the design and how the user interface is going to look. It includes information about different search functionalities, notifications to send to users, and many different other elements needed to implement a user-friendly experience.
5. **Performance:** A section that contains info and defines the expected performance characteristics of the system, such as response times, and the ability to handle a specific max number of users and books.

2. Product/Service Description

2.1 Product Context

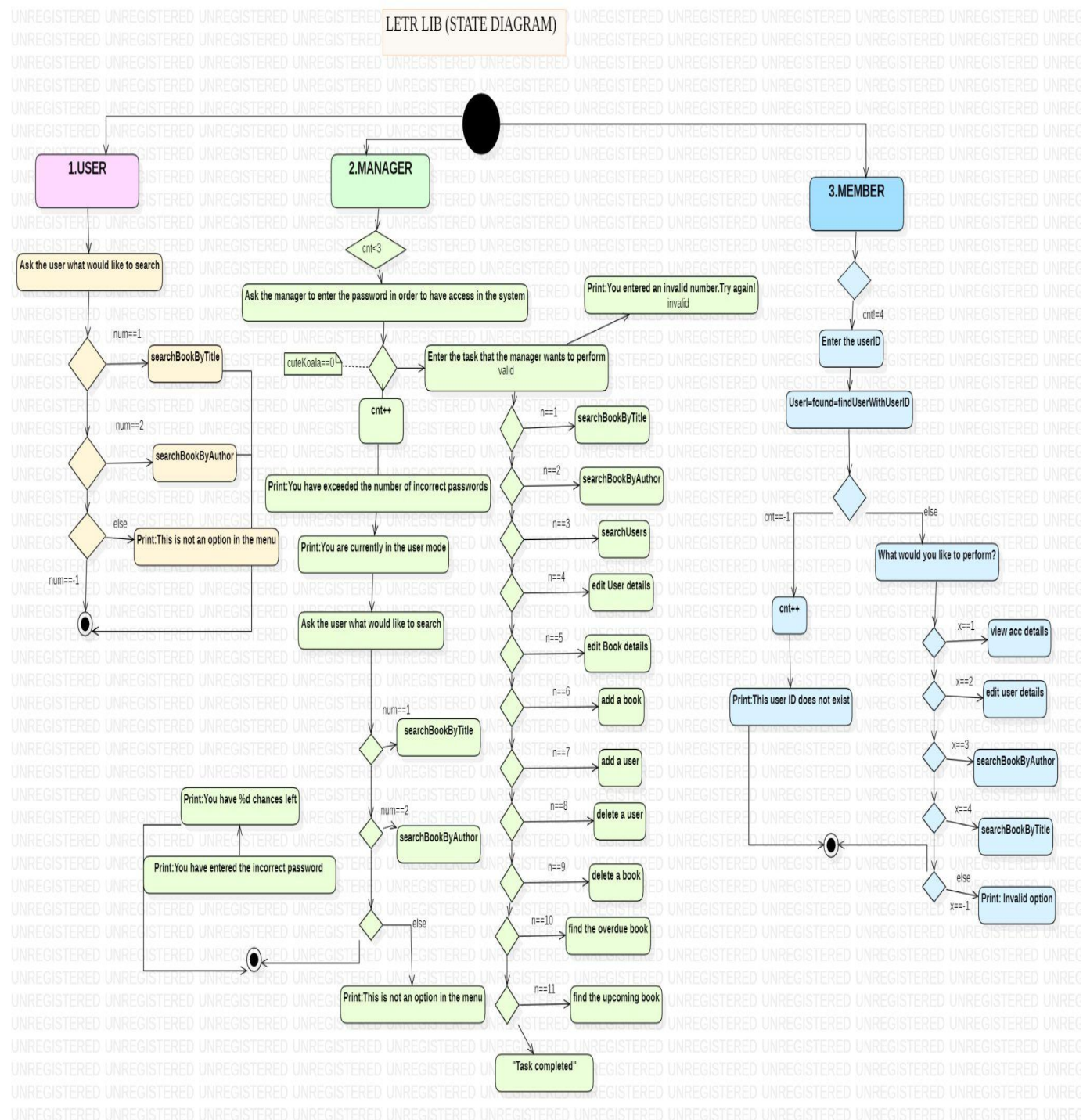
How does this product relate to other products? Is it independent and self-contained? Does it interface with a variety of related systems?

- Library Management System is an application which has a purpose of providing the main functionalities for managing daily activities inside a library. The automated system saves a considerable amount of time as opposed to the manual system. This System allows its stakeholders which consist of: *Library administrator/manager and the Library members/users*, to perform many operations, based on their requirements and needs. Library Manager can perform tasks like managing users (including/adding/editing/deleting user accounts) and users can borrow or add books, based on their preferences. They have the ability to see overdue books or upcoming dates, so users can pre plan their orders.
- It is a *self-contained* product that functions by a collaboration of many independent separated systems, concretely: Borrowing system, Reservation System, Reporting System.
- This system interfaces with “File Systems” interfaces, as our data is saved in several files and then accessed by a file interface (C language functions and library for files) as they can read from or write to files (.txt format)
- It meets users needs by being easy to use and having a welcoming and friendly interface for anyone who wants to perform actions on it, due to the fact that *manager and user interface are nicely intertwined*. The language used is kind and friendly.

Library Management System Requirements Specification

Describe these relationships or use a diagram to show the major components of the larger system, interconnections, and external interfaces.

-This is a visual representation(State Diagram) of how Library Management System works and which are its main interconnections between users, external interfaces and main purpose. The Product context and development are easily understood by this diagram.



Library Management System Requirements Specification

2.2 User Characteristics :

This System allows its users which consist of : *Library administrator/manager, the Library members* and Library (new) users to perform many operations, based on their requirements . Their actions consist as following:

- Members
 - Must have an account/membership (unique identifier,name, address, phone number);
 - Can search for the books; by author or by title(book name);
 - Navigate through the list of books (file of books);
 - Can borrow books;
 - Can return books;

- Library manager/staff
 - Must have an account/membership (unique identifier,name, address, phone number);
 - Navigate through the list of books (file of books);
 - Search for the books; by author or by title(book name);;
 - Search for the users by name or by ID;
 - Edit user details ;
 - Edit book details ;
 - Add books ;
 - Add users;
 - Delete users;
 - Delete books;
 - Keep track of due date;
 - Keep track of overdue books;
 - Provide reports on book availability;

- Library User (*New user/ not a member yet*)
 - Can create an account/membership (unique identifier,name, address, phone number)
 - (In order to continue other operations in our library, they must firstly be a member, so they can have extended access in the library)

2.3 Assumptions

- It is assumed that the library already has a physical space, a collection of books and a basic organizational structure.
- The system is designed for a specific user base, such as library members/users and library staff/manager.
- The library management system will be implemented in a computerized environment with appropriate hardware and software resources.
- It is assumed that the library has access to the internet for various functionalities, such as online catalog searches, database updates etc.
- The library management system will include a database to store and manage information about books, borrowers, and other relevant data.
- The library management system will be user-friendly and 24 hours/day available.

Library Management System Requirements Specification

2.4 Constraints

- The project may be constrained by the available hardware resources, such as servers, storage capacity, or network infrastructure. Limited resources can impact the system's scalability, performance, and capacity to handle concurrent users or large volumes of data.
- The project may have budgetary constraints that limit the available funds for development, implementation, and maintenance of the library management system. This can impact the selection of technologies, the scope of features, and the overall project timeline.
- The project may have specific time constraints or deadlines that need to be met. This can influence the project schedule, development methodology, and the prioritization of features to ensure timely delivery.
- The project must consider the requirements and constraints of the library's users. This can include factors such as user preferences, language support, accessibility needs, and the need for intuitive and user-friendly interfaces.

2.5 Dependencies

List dependencies that affect the requirements.

1. This software has a maximum number of users that it can hold so the amount of users that you can input is limited.
2. This software has a maximum number of books that it can hold so the amount of books that you can input is limited.
3. Each file, the book and also the user one is saved in an array so once its size capacity reaches the user and librarian can't perform any more actions.
4. Books cannot be taken if the specific book intended to order is out of stock/is not available.
5. Users can only access the library functions by logging in with their ID.
6. Librarians can access the library functions only by entering a password only they can know.

3. Requirements

Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.

❖ A good requirement is:

- Correct
- 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 and specify measurable quantities)
- Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
- Does not specify any particular design
- Traceable (cross-reference with source documents and spawned documents).

3.1 Functional Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME/Reviewed /Approved
R_01	The program stores the data in separate files	The data gets saved in different arrays so there are two separate arrays; one for books and one for borrowers	High/1	24.05.2023	Tea Disho
R_02	The data gets displayed for the manager to check that every field is consistent	An efficient way to make sure the manager is in coordination with everything. Ensure effective coordination and communication between the manager and all relevant aspects.	2	23.05.2023	Tea Disho
R_03	Based on users request the program gives an option to delete their account	A necessary implementation to assure the library puts borrowers and their experiences first.	1	24.05.2023	Tea Disho
R_04	Displaying and giving the necessary notifications for the user's upcoming due dates	Ensures that the data is properly managed and users will not misuse the library's functionalities.	1	22.05.2023	Tea Disho
R_05	Both the user and manager can search by books author	Enables program users to utilize various search methods to create a customized user interface according to their preferences.	1	23.05.2023	Tea Disho

Library Management System Requirements Specification

R_06	Manager has to enter the password and only has 3 tries if they fail all 3 they are switched to user mode	To ensure the security and only those who have the password have access	High	29.05.2021	Layan Alateeq
R_07	The members of the library have 3 tries to enter their member ID before they get switched to user mode	To enable security for which only the user can access their data	High	29.05.2021	Layan Alateeq
R_08	Managers can edit all aspects of the member's information	For when users data is changed, managers change it easily	Medium	29.05.2021	Layan Alateeq
R_09	Managers can allow users to borrow and return books from the system	So that members can borrow books	high	29.05.2021	Layan Alateeq
R_10	Users, members and managers can search for books by their title	To get access/information on certain books	high	29.05.2021	Layan Alateeq
R_11	Managers can search for users by, name, surname, ID and phone number	To be able to change and view certain users data	high	29.05.2021	Layan Alateeq
R_12	Managers can check those whose due date is near (3 days or less)	In order to send them a reminder to return their books.	low	29.05.2023	Layan Alateeq
R_13	Library managers can add new books in the library	Managers can edit book details(quantity) if the book is currently existent or add new book's data in the library	High/1	29.05.2023	Era Fejza

Library Management System Requirements Specification

R_14	Library managers can delete a book by its title	In order to delete books, the program automatically distincts the selected book's titles entered by the manager and deletes them from the database.	High/1	29.05.2023	Era Fejza
R_15	Library managers can delete books by their author	When assigned to delete by the author, the automated system deletes all books written by inserted authors.	High/1	29.05.2023	Era Fejza
R_16	The program searches for books by the title in order to perform pre mentioned actions (deleting and adding books)	In order to add or delete books, the program firstly searches by book title if the book is existent, and then performs the desired action by the library manager.	Medium/2	29.05.2023	Era Fejza
R_17	The program gets data from the file and stores them in an array.(So the number of books is limited)	After reading data, all of it is stored in an array of structures, and then is accessed by the program to perform the action of inserting/adding(option is inserted by the user)	Low/3	29.05.2023	Era Fejza

R_18	Library managers can add new users in the library.	Managers can add a new user information and store it in the next index of the array.	High	30.05.2023	Rosilda Bajrami
R_19	Library managers can edit book details by its name.	If the book exists, managers can edit it by its name such as author name,quantity and amount .	High	30.05.2023	Rosilda Bajrami
R_20	The program searches for books by name in order to edit them.	In order to edit book details,the program first searches by book name and checks if the book exists ,and then performs other steps.	High	30.05.2023	Rosilda Bajrami

Library Management System Requirements Specification

R_21	Library managers can make a choice to select which detail they want to edit .	If the book is found,managers should be presented with a menu to select which details they want to edit.The manager's choice should be read and appropriate actions should be taken to edit book details.	Medium	30..05.2023	Rosilda Bajrami
R_22	The program gets data from the file and stores them in an array.	After reading data, all of it is stored in an array of structures, and then is accessed by the program to perform the action of editing a book.	Low	30.05.2023	Rosilda Bajrami

3.2 Non-Functional Requirements

1. **Performance:** The library management system should efficiently handle a large number of users and reservations without experiencing significant slowdowns. It should provide quick responses to user requests, ensuring fast search results display.
2. **Scalability:** The system should be capable of accommodating the library's growth in terms of book collections and user base. It should effectively manage increased data volume and user traffic without compromising performance.
3. **Reliability:** The system should be dependable and available for uninterrupted use. It should be capable of recovering from any system malfunctions without data loss or disruption to user activities.
4. **Usability:** The system should be user-friendly and intuitive for both library managers and users. It should offer a clear interface with easy navigation and well-organized features.
5. **Maintainability:** The system should be designed and implemented in a way that facilitates straightforward maintenance and future enhancements. It should possess well-documented code, a well-structured architecture, and support mechanisms for efficient issue resolution and the addition of new features.

Product Requirement

❖ Performance:

1. Response time
2. The capacity or rate of production of the system
3. Scalability
4. Concurrent user handling

❖ Reliability:

1. System availability
2. Fault tolerance
3. Disaster recovery

❖ Security:

1. Data confidentiality
2. User authentication
3. Authorization controls
4. Logging

❖ Usability:

1. User-friendly interface
2. Easy navigation
3. Accessibility compliance

❖ Maintainability:

1. Modular and reusable code
2. Configuration ease
3. Documentation

❖ Compatibility:

1. Integration with other systems
2. Support for various IDEs

Organizational Requirements:

Governance and Compliance:

1. Compliance with laws and regulations
2. Data protection and privacy

Organizational Policies:

1. User access policies
2. Information management guidelines

External Requirements:

Interoperability:

1. Integration with external databases or library networks
2. Compliance with data exchange standards

Performance Standards:

1. Performance metrics

Ethical Considerations:

1. Data privacy and protection

3.2.1 User Interface Requirements

Screen Formats/Organization:

- The layout of the screens is clear and organized, with important information presented in a structured way. Sections are separated by dashed lines to make them easily distinguishable. The beginning and end of the code are highlighted to stand out from the rest.
- The menus and options are well-arranged, making it easy for users to navigate and interact with the system. They provide clear choices for users to select from.
- The design of the screens is consistent across the system, creating a unified experience for the users.

Report Layouts:

- Reports have clear headers, sections, and columns that make it easy to understand and interpret the information presented.

Structure of menu:

1. The menus are organized in a logical and structured way, reflecting the system's functionality.
2. The menus are categorized, grouping related features or tasks together for easier navigation and access.

Error and such type of messages:

1. The system provides clear and helpful messages when errors occur, guiding users to understand the issue and find a solution.
2. The error messages are designed to stand out and catch the user's attention, with consistent formatting and style.

Function Keys(the password):

- Function keys are designed to be easy to remember and use, allowing for efficient navigation or execution of tasks within the system.

3.2.2 Usability

Include any specific usability requirements.

1. User-Friendly Interface:

- The system should be easy to use with clear labels and organized features.
- Menus and options should be intuitive and easy to understand.

2. Efficient Search and Navigation:

- Users should quickly find books and information in the library catalog.
- Moving between options should be simple and straightforward.

3. User-Focused Data Entry:

- Data entry forms should be user-friendly with clear instructions.
- The system should help prevent errors and provide relevant suggestions.

4. Responsive Design:

- The system should work well on different devices (desktops, tablets, and mobiles).

5. Clear Feedback and Error Handling:

- ❖ Users should receive clear feedback after their actions.
- ❖ Error messages should explain the problem and offer solutions.
- ❖ Users should be redirected to the correct section if necessary.

3.2.3 Performance

❖ Simplified Static Numerical Requirements:

System Capacity: The system should support up to 300 books and 150 registered users.

Storage Capacity: The system should have enough space to store all library data, including book details, user information, and borrowing history.

❖ Simplified Dynamic Numerical Requirements:

Response Time: The system should quickly display search results within 2 seconds.

Transaction Processing Time: The system should promptly process book checkouts or returns for efficient service.

Maximum Wait Time: Users should experience a maximum wait time of 4 seconds for tasks like logging in or searching for books.

Library Management System Requirements Specification

3.2.1.1 Monitoring

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

Logging and Error Reporting: The system generates logs that record important events, errors, and exceptions. It has a mechanism to report errors to administrators or support teams for timely resolution.

Performance Monitoring: The system monitors response times, throughput, and other performance metrics to ensure optimal performance.

Fault Tolerance and Redundancy: The system is designed to handle failures gracefully and has built-in redundancy mechanisms. This helps minimize downtime and ensures continuous availability of the service.

Correction and Recovery: The system has processes and procedures in place to correct errors, recover from failures, and restore normal operation as quickly as possible.

3.2.1.2 Maintenance

Specify attributes of the system 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.

1. **Modularity:** The system should be divided into independent modules, making it easier to maintain each module without affecting the entire system.
2. **Code Maintainability:** The system's code should be well-organized, readable, and follow coding conventions. This simplifies understanding, debugging, and modification during maintenance.
3. **Low Complexity:** The system should be designed with simplicity in mind, avoiding excessive dependencies, code duplication, and using clear algorithms and data structures. Lower complexity eases troubleshooting and maintenance.
4. **Interface Design:** The system's interfaces, both internal and external, should be well-designed, consistent, and user-friendly. This simplifies maintenance tasks, such as adding or modifying features.

3.2.1.3 Operations

Specify any normal and special operations required by the user, including:

- **Normal Operations:**

Search and Retrieval: Users can search for books and resources using criteria like title, author, subject, or keywords. The system shows available items and their locations.

User Management: Librarians can manage user accounts (add, edit, or delete), update personal information, provide information about upcoming events, and handle penalties for overdue items.

- **Special Operations:**

Reporting and Analytics: Generates detailed reports on different aspects of library operations. These reports help with decision-making and future planning.

3.2.5 Security

3.2.5.1 Protection

- **User Authentication:** The system checks user identity using secure methods like personal ID for clients and passwords for managers before allowing them to access the system.
- **Access Control:** The system applies rules to control user access based on their roles and permissions. This ensures that users can only access the features and information that are appropriate for their assigned roles, keeping sensitive data secure and preventing unauthorized access.

3.2.5.2 Authorization and Authentication

Specify the Authorization and Authentication factors.

- **User Authentication :**

The code should ask users to provide their username / password as credentials to confirm their identity. Input from users should be handled securely and compared with stored credentials in a safe manner, ensuring protection of sensitive information.

- **Access Control :**

The code should include checks for access control to limit user access to specific functionalities or data based on their roles or permissions. User roles and permissions should be clearly defined and enforced in the code to prevent unauthorized access to sensitive operations.

3.2.6 Standards Compliance

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

3.3 Domain Requirements

The library management system by using their components and features, such as book availability, quantity, and users ID or phone number, etc., users may easily search for books and other users. This approach makes sure users can find their requested books quickly, promoting effective use of the library's resources. Users may handle their borrowing demands without needless delays or problems of any kind by having access to these types of operations. Additionally, the system sends a message to users to remind them of their due date, guaranteeing easier management of borrowed goods. This is only done by having a stable internet connection.

Library Management System Requirements Specification

4. Design thinking methodologies

4.1 Negotiation:

-As the term “negotiation” implies an agreement between groups, the same idea is followed in software design methodologies. The aim of negotiation in design methods is to achieve a design that is compatible with different requirements, in order for the final product to be balanced regarding the expectations of all groups involved. By collaborating, discussing and sharing ideas with teammates, we ended up with the current version of design for our project.

-Even though C language is limited in the designing process, and doesn't fully support an interactive interface for the user (usage of dashboards and buttons to click), we “negotiated” with those requirements in accordance with the opportunities and the knowledge we had. The interface is easy to navigate, very welcoming and friendly for the users. *The welcoming and title design is made by using different patterns made of symbols such as; #, *, -, _ , ! ~ etc. Their combination concludes a nice simple design.*

-The user who participates in our Library Management System software, has the opportunity to come across all the possible choices they can make regarding the list of the requirements we were given to follow, so *by a simple click of the corresponding number, they can decide on their next operation on the system. The steps are very clear and easy to understand, as every message is clearly displayed on the screen for the user to see.*

4.2 Empathy

-In a software product like Library Management System, dealing with users is one of the most crucial aspects, as they are the main focus for this system's wellgoing. For our team, as developers of this product, it was very important for us to conduct a very user friendly product so we included the empathy part in the design aspect by:

- bringing up quotes and questions in the main page
- inviting the users to discover more about our library
- giving an outline for our system's background and history etc.
- By imagining ourselves in users shoes and collaborating and discussing, it was easier for us to invent a product that aims for a great user experience.

Even though C-language has limited opportunities to provide a colorful and interactive Graphical User Interface, we tried to balance that aspect by using questions and a very kind and inviting language, in order to make the user feel very comfortable and leave our software product with the greatest opinion and a strong desire to receive a service by our product again!

4.3 Noticing

-As developers, while building the product “Library Management System” we tested each function separately in order to notice every single design inconvenience that happened. By checking the functions one by one and not the main function directly, we provided a software product which offers the best experience for a user who wants to perform any action inside a library.

4.4 GUI (Screenshots)

```

*****
*****
*****      Library Management System Project      *****
*****
*****
*****
-----
-----

**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**

=--==--==--==--==--==--==--==--==--==
=
=      WELCOME TO      =
=      LIBRARY MANAGEMENT SYSTEM      =
=
=      "LETR LIB"!      =
=
=--==--==--==--==--==--==--==--==--==

**--**--**--**--**--**--**--**--**--**--**--**--**--**--**--**

Hello! Welcome to "LETR LIB"!
-----

We are a local library which currently holds 100 books and has the capacity to store 300 books in total.
We have around 75 different users, and if you would like to join our loving community we are open to applications. <3

Enter
1. if you are a user
2. if you are library manager.
3. if you are member
enter your option: █

```

```
C:\Users\HP\Desktop\SWE PROJECT\LETR LIB project\bin\Debug\LETR LIB project.exe"
```

```
*****  
*****  
***** Library Management System Project *****  
*****  
*****  
*****  
-----  
  
-----  
*****  
  
=====
```

```
= WELCOME TO =  
= LIBRARY MANAGEMENT SYSTEM =  
=  
= "LETR LIB"! =  
=  
=====
```

```
*****  
Hello! Welcome to "LETR LIB!"  
-----  
We are a local library which currently holds 100 books and has the capacity to store 300 books in total.  
We have around 75 different users, and if you would like to join our loving community we are open to applications. <3  
  
Enter  
1. if you are a user  
2. if you are library manager.  
3. if you are member  
enter your option: 2  
  
-----  
Enter the password to access the Library Management System: cuteKoala  
1. Search for the book data by title  
2. search for the books by author  
3. search the users  
4. edit a user  
5. edit a book  
6. add a book  
7. add a user  
8. delete a user  
9. delete a book  
10. find the overdue books
```

```

C:\Users\HP\Desktop\SWE PROJECT\LETR LIB project\bin\LETR LIB project.exe"
-----
We are a local library which currently holds 100 books and has the capacity to store 300 books in total.
We have around 75 different users, and if you would like to join our loving community we are open to applications. <3

Enter
1. if you are a user
2. if you are library manager.
3. if you are member
enter your option: 3

-----
enter the user ID you want to find: 65017
what would you like to perform
1. view your account details
2. search for books by author
3. search for book by title
-1. to exit the menu
your choice: 1
user name: name1
surname: surname1
phone: 3614222652
address: address1
ID: 65017
borrowed books:
book 1: name2

-----
what would you like to perform
1. view your account details
2. search for books by author
3. search for book by title
-1. to exit the menu
your choice: -

```


❖ Visual representation for the “user” option:

```
Enter
1. if you are a user
2. if you are library manager.
3. if you are member
enter your option: 1

-----
What would you like to search?
1. Search for the book by title
2. search for the books by the author
-1. to exit
enter your option: 1

-----
enter the title of the book you want to find: book78
book title: book78
Author: author3
books available: 2
quantity: 5

Task completed!
-----
What would you like to search?
1. Search for the book by title
2. search for the books by the author
-1. to exit
enter your option: 2

-----
enter the name of the author: author3
Book: book7
Book: book24
Book: book28
Book: book42
Book: book62
Book: book78
```

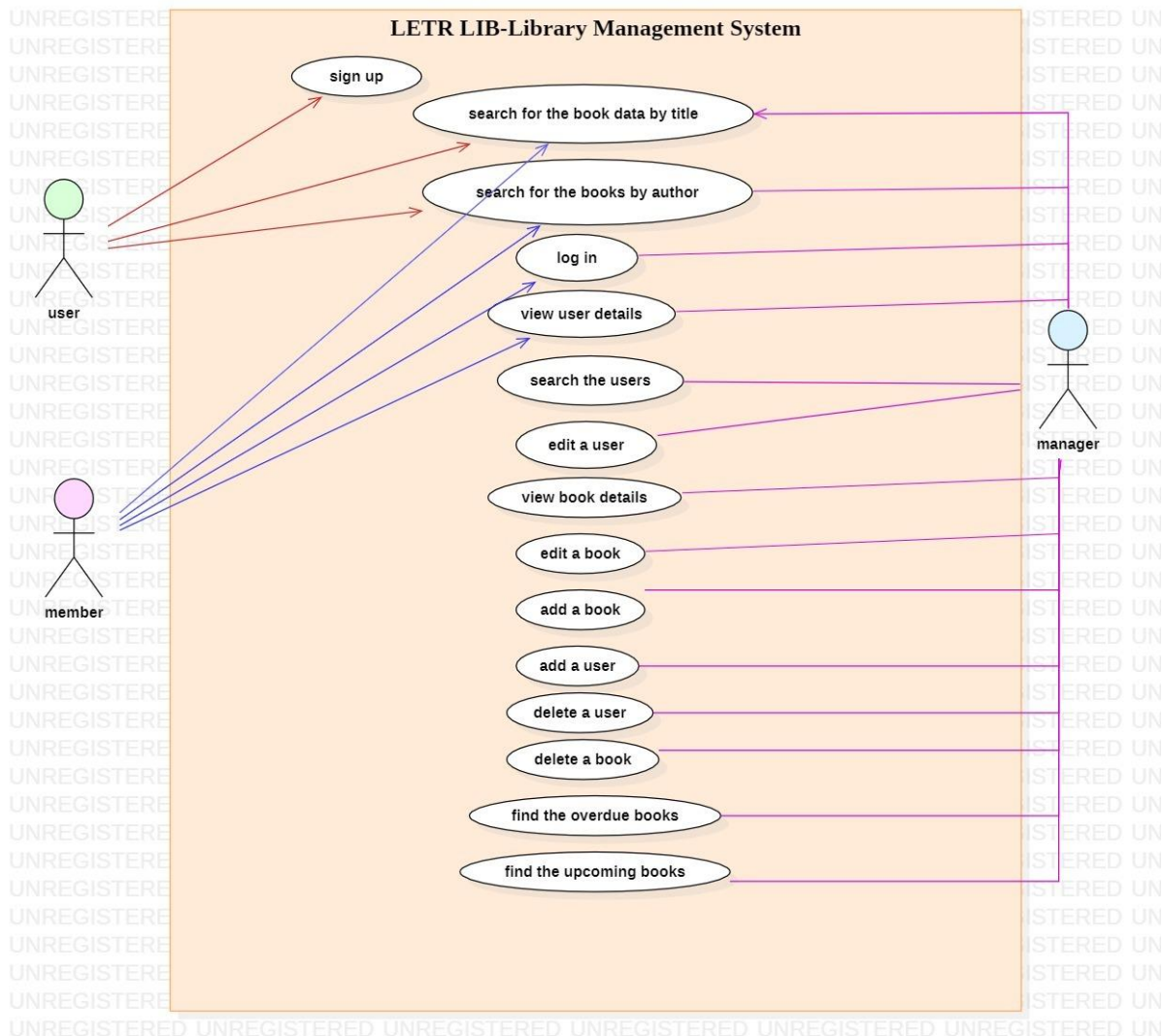
5. Software Design

5.1 Use Case

A use case diagram consists of use cases (represented by oval shapes) that describe the interactions between actors (in this case are **Manager**, **User** and **Member**) and the system. Actors are represented by stick figures and are connected to the use cases with lines that indicate the type of interaction.

> Use cases describe the various ways in which actors interact with the system. For instance, in this case they interact by logging in, signing up, searching for the book by title or author, viewing user details, searching the users, editing the users, viewing book details, editing books, adding books, adding users, deleting users, deleting books, finding the overdue and upcoming books.

> Use case diagrams are useful for understanding the requirements of a system and identifying the features and functionality that need to be developed. They can help developers and stakeholders to communicate and understand the interactions between the system and its users.



5.2 State Diagram

In a state diagram, the states of the system or object are represented by rectangles, while the transitions between states are represented by arrows. Each transition is labeled with the event that triggers it and the action that occurs when the transition happens.

> State diagrams are useful for modeling systems that have multiple states, such as user interfaces, control systems, and protocol handlers. They can help developers to understand the behavior of the system and to identify potential issues, such as race conditions or deadlocks. State diagrams can also be used to generate code and to test and verify the behavior of a system before implementation.

In our case the initial start is LOGIN or SignUp .Then there are some options for the user to enter:

1.If the user wants to sign up as a USER

If yes ,there are some other options/conditions (option is declared as num) for the user to perform the actions ,otherwise the system terminates.

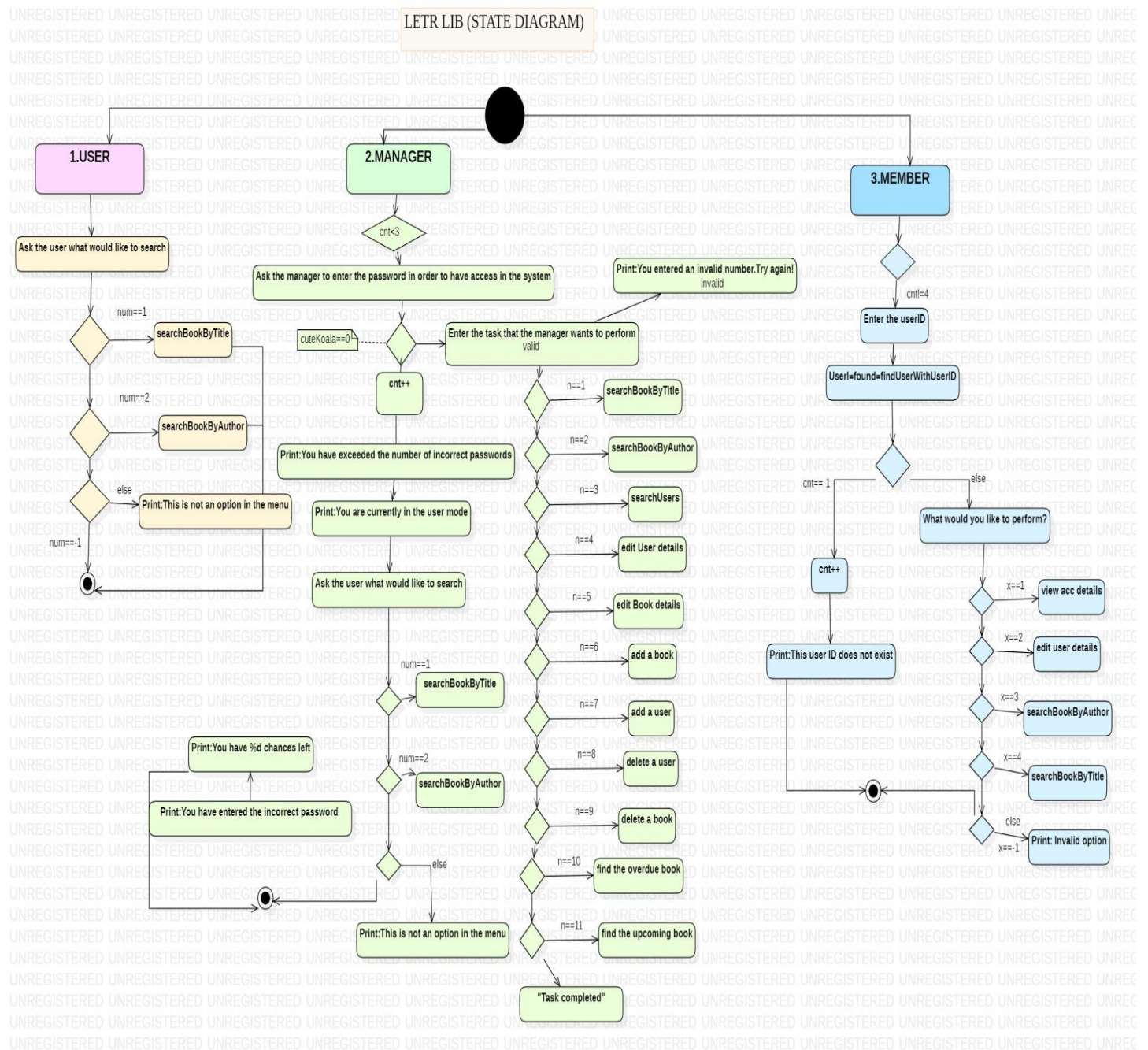
2.If the user wants to login as a MANAGER

If yes,firstly the manager has to enter the right password in order to have access in the system.If the manager enters the password incorrectly more than 3 times ,the system will pass in user mode and then it will perform user's mode functions and it will terminate in case the user enters another option or -1.If manager enters the right password (cuteKoala) there are some option/conditions (option is declared as n) for the manager to perform the actions,otherwise it will print a message that says "You entered an invalid number.Try again!".

3.If the user wants to login as a MEMBER

If yes,firstly the member logs in by ID and if the member is found ,there are some other options/conditions (option is declared as x) for the member to perform the tasks,if the member enters -1 it will terminate.If the member is not found ,it will print "The user ID does not exist" and will terminate.

Library Management System Requirements Specification



5.3 Class Diagram

In class diagram, classes are represented by rectangles with three sections: the top section contains the name of the class (in our case there are 4 classes : User ,Books,Member,Manager), the middle section contains the attributes of each class (such as name,surname,phone number,address,ID etc), and the bottom section contains the methods of the class (such as signing up, logging in , searching Books, searching Users etc).

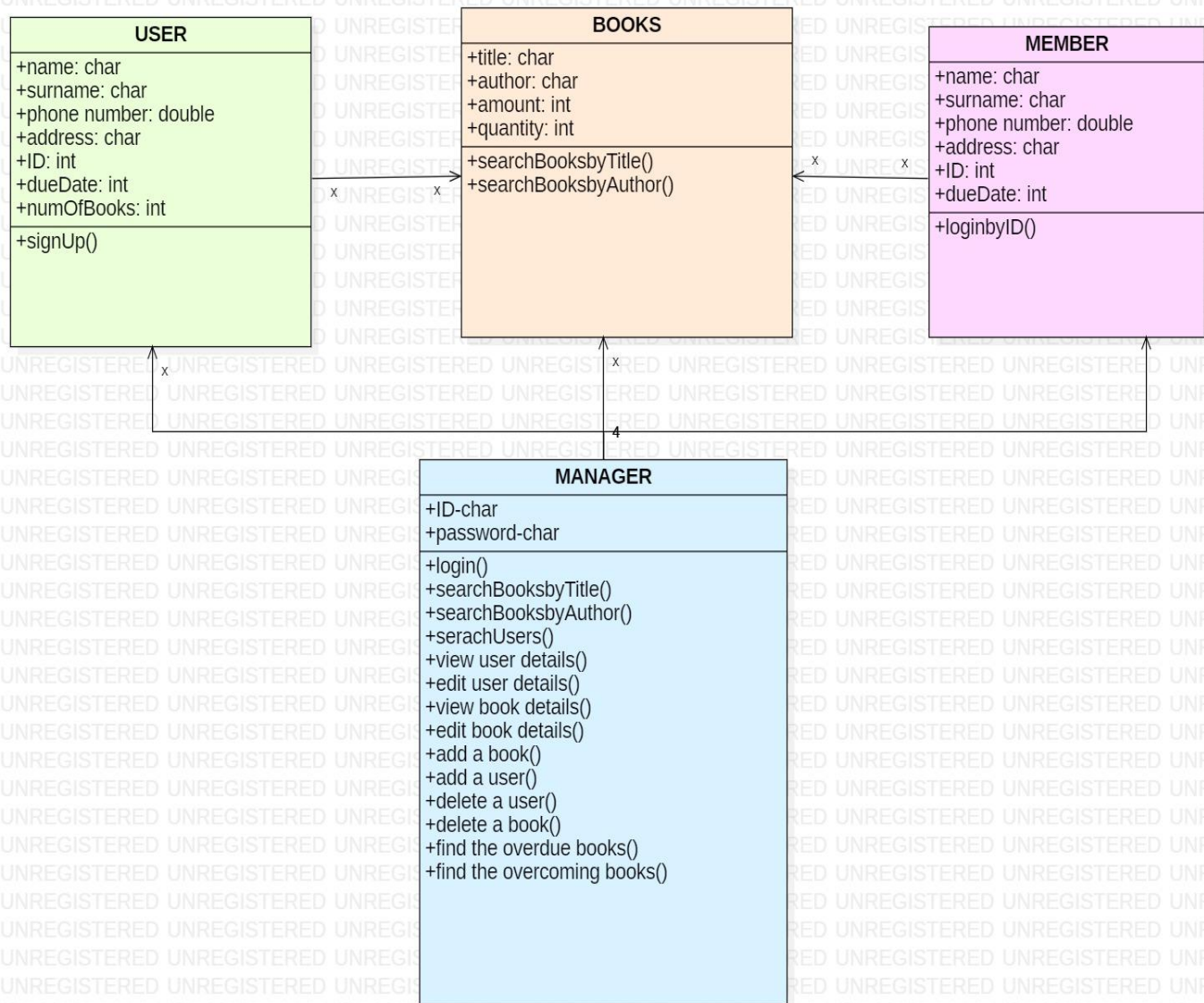
>Relationships between classes are represented by lines, and in this case they are represented with association with multiplicities.

>In our case every class is associated with the BOOKS CLASS since actors have in common two functions (searching Books By Title and searching Books By Author).The only class associated with every class is MANAGER CLASS because the library manager has the option to perform all the tasks/functions .

> Class diagrams are useful for designing and documenting the structure of a system, and for communicating the design to stakeholders. They can help developers to identify potential design issues and improve the overall structure of the system. Class diagrams can also be used to generate code and to test and verify the design of a system before implementation.

Library Management System Requirements Specification

LETR_LIB(CLASS DIAGRAM)



Library Management System Requirements Specification

APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys; •
- Supporting or background information that can help the readers of the Requirements Specification; •
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

Appendix A. Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

Appendix B. References

List all the documents and other materials referenced in this document.

Appendix C. Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may 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.

By Stimulus

Library Management System Requirements Specification

[YourProject] Requirements Specification Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.