1.Introduction

1.1.Purpose

This document aims to give a detailed description of the requirements for the “Library Management System” (LMS) software. The document will cover each of the system’s intended features, hardware, software, and various other technical dependencies.
Borrowing books, returning books or viewing the available books at the library is currently done manually where in the student has to go to the library. Issuance of books is carried out by the librarian based on availability.
The “Library Management System” solves the above stated problem by automating the entire process wherein the students can borrow, return or view availability of books by logging in to the system.

1.2.Document Conventions

Term	Definiton
LMS	Library Management System
UML	Unified Modeling Language
SRS	Software Requirement Specification
GUI	Graphical User Interface

1.3.Intended Audience and Reading Suggestions

This document is intended for developers, testers and stakeholders. Some of the content of this document may not be easily understandable by the end user as it contains technical terms and UML diagrams that require knowledge of few software engineering concepts.

Readers interested in a brief overview of the software should focus on the rest of Section 1 (Introduction), as well as Section 2 of the document (Overall Description), which provide a brief overview of each aspect of the project as a whole.
Section 3 (External Interface Requirements) offers further technical details, including information on the user interface as well as the hardware and software platforms on which the application will run.
Readers who wish to explore the features of LMS in more detail should read on to Section 4 (System Features), which expands upon the information laid out in the main overview. Readers interested in the non-technical aspects of the project should read Section 5, which covers performance, safety, security, and various other attributes that will be important to users.
Readers who have not found the information they are looking for should check Section 6 (Other Requirements), which includes any additional information which does not fit logically into the other sections.

1.4.Product Scope

The LMS is a web application which helps the user query for books and check their availability through a simplified web-portal. It makes looking for information about availability of books much easier through automation. It initially registers the user, providing each new one with a unique user ID. The user subsequently logs in using his/her credentials to search for and borrow books. Upon receiving a request for an available book, the web-portal assigns the book to the corresponding user and updates the information in the database. Further, the admin possess the ability to update information about the books i.e add or delete books. Users can similarly return books back to the system allowing others to borrow them.

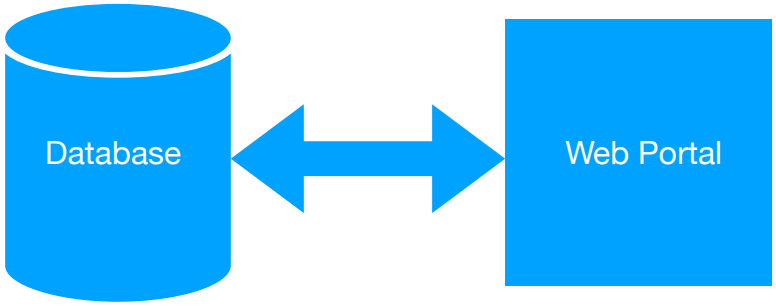
1.5.References

[1] IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998
[2] Karlsson J, "A Cost-Value Approach for Prioritizing Requirements", Norges Teknisk-Naturvitenskapelige Uni.1997
[3] Davis M A, "Just Enough Requirements Management: Where Software Development Meets Marketing", New York, Dorset House Publishing, 2005.
[4] "Design and Implementation of a Library Management System Based on the Web Service" by Yujun Li, Hao Zheng, Tengfei Yang, Zhiqiang Liu. http://ieeexplore.ieee.org/document/6405716/

2.Overall Description

2.1.Product Perspective

This software consists of a single web application providing various functionalities to a registered user. Since this is a data-centric product it will need somewhere to store the data. For that, a database will be used. The admin possesses the permission to add and delete books from the database. A new user must first register himself in the system after which he is assigned a unique user ID. The user can then login to the system and query for books. The query will display information about the presence or absence of the book in the LMS and if present, displays information about the availability of the book. The user can subsequently place a request for the book upon which the availability information is updated in the database and the user is assigned the book. A logout functionality is also implemented.



2.2.Product Functions

The LMS software will be able to provide the following functionalities:

1. Functionalities available to non-registered users

- Registration for users to set up their account after which each user is assigned a unique user ID

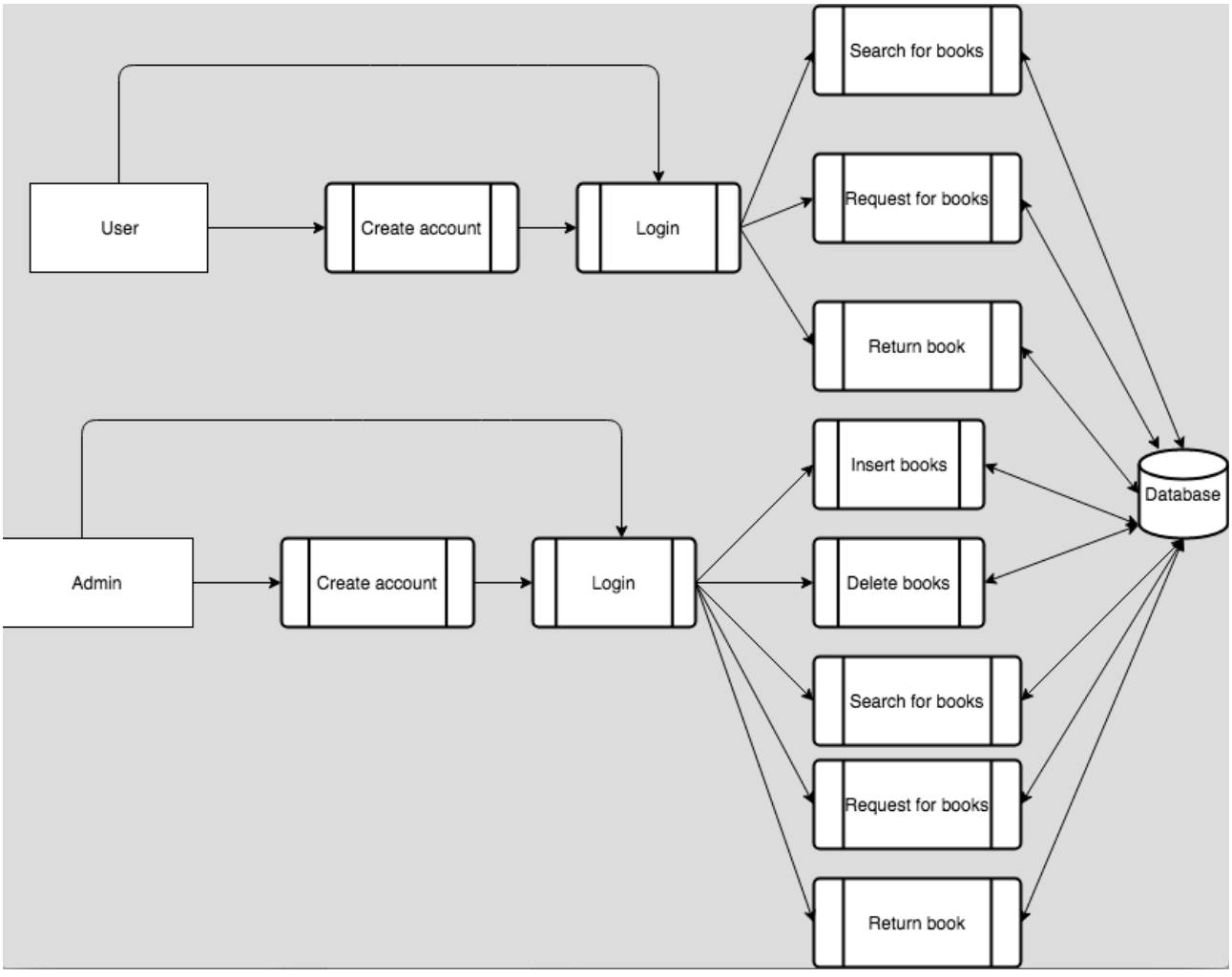
2. Functionalities available to registered users

- Search functionality for registered users to look up books available in the library
- Functionality for registered users to place a request for an available book after which the user is assigned the designated book
- Functionality to return a book.

3. Functionalities available to the admin

- Functionality to place a request for an available book after which the admin is assigned the designated book
- Functionality to return a book.
- Functionality to add/remove a books to/from the library database

The data flow diagram depicting the above mentioned functionalities is as below:



2.3.User Classes and Characteristics

The software will support two types of user privileges - user and admin. Users will have access to user functions, and the admins will have access to both user and admin functions. The admins of the system are expected to have more knowledge about the internal working of the system and possess the ability to rectify problems that may arise due to disk crashes, power failures and other catastrophes. The user interface, must be sufficient to educate the users on how to use the system without any problems and hence do not require any technical expertise.

2.4.Operating Environment

The software will be developed to run on any *nix platform or Windows 8+ platform with XAMPP (or corresponding LAMPP/MAMPP) installed. The software will developed using Flask micro framework, a python framework to develop web applications with ease. Since the product is data-centric , a database that satisfies the required storage needs is used. The LMS software will use the MySQL database to store data.

2.5.Design and Implementation Constraints

The entire application will be developed using the Flask microframework for Python. The system shall use the MySQL database for storage. However the amount of information that can be stored for the LMS will be constrained by the capacity of the database. The TCP/IP protocol is followed for internet communication. For working on the coding phase of the LMS- Python is used as the primary programming language and supplemented by HTML,CSS,jQuery for the web development part of the software development process.

2.6.User Documentation

The software will not provide any user documentation and assumes that the user is qualified enough to navigate through the application without any external help.

2.7.Assumptions and Dependencies

It is assumed that the user machine has enough RAM to run the web application and a working internet connection.

3.External Interface Requirements

3.1.User Interfaces

The users interact with the system through the web-portal.The web portal contains a home page with GUI buttons to navigate the user to Register or Login pages. Upon successful login, the user should be able to search for books and view their corresponding information. The user can subsequently place a request for issuing the book by pressing the Issue GUI button and can also return a book he possesses using the Return GUI button. The information about books present in the database is listed in the form of a table. The login page also contains a GUI button to implement the

log out function. The admin has two additional functionalities to allow insertion and removal of books from the database which are again implemented using GUI buttons.

3.2.Hardware Interfaces

Since the web portal does not have any designated hardware, it does not have any direct hardware interfaces. The hardware connection to the database server is managed by the underlying operating system on the web server.

3.3.Software Interfaces

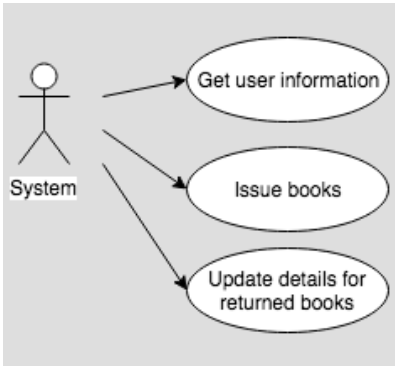
The software interfaces include:
- MySQL version 5.7.19 or greater
- *nix based based operating system / Windows 8 or greater
- XAMPP/LAMPP server

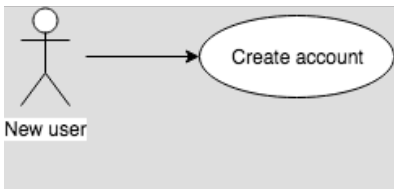
3.4.Communications Interfaces

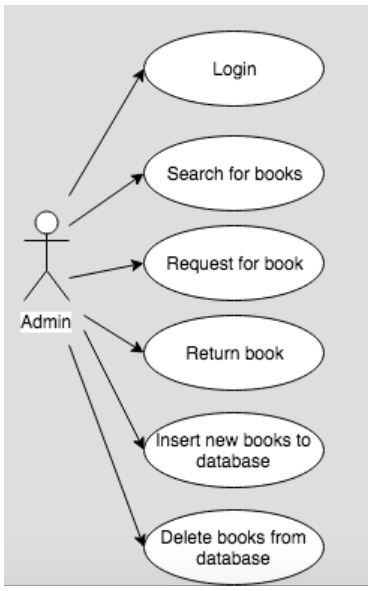
The communication of the software will be through HTTP.

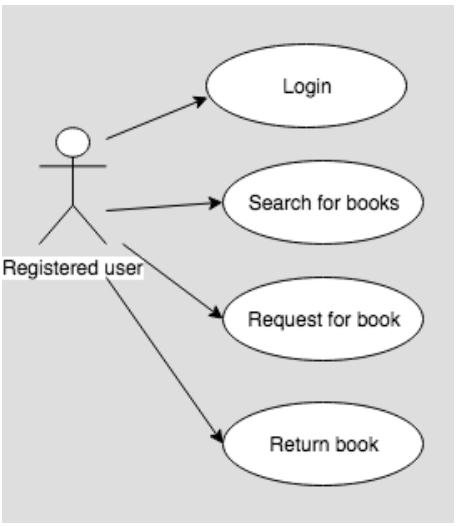
4.System Features

4.1.Use cases









4.2.Functional Requirements

4.2.1 Functional requirements associated with a non-registered user

Requirement ID - R1.01.01
Title - Sign Up/Register
Description - Allow non-registered users to create their account by providing the necessary details. After it's creation, the user is assigned a unique UserID

4.2.2 Functional requirements associated with a registered user (including admin)

Requirement ID - R1.02.01
Title - Validate user account and search for books
Description - Registered users must be able to visit their profile once the credentials provided are validated. Upon successful validation, the user must be able to search for books in the LMS database.

Requirement ID - R1.02.02
Title - Issuance of available books
Description - A user upon placing a request for an available book must be issued the book with details being updated correspondingly in the LMS database. The corresponding UserID is mapped with the BookID.

Requirement ID - R1.02.03
Title - Return of books
Description - A user must be able to return books that were issued against his/her UserID. The LMS database is updated correspondingly.

4.2.3 Functional requirements associated with the admin

The admin along with the functionalities provided to a regular user also has the following additional functional requirements:

Requirement ID - R1.03.01
Title - Insertion/Deletion of books
Description - The admin should be able to insert new entries or delete older ones from the library database.

5.Other Nonfunctional Requirements

5.1.Performance Requirements

The system shall accommodate a decent number of books and users without any fault. Responses to view information shall not take long durations of time. The speed of the system should be good enough to retrieve information requested by the user and also to make the process of borrowing and returning books easier, less time consuming and more efficient. The software functionalities should

be easy to understand and use for the end users since it's a system which is intended for an audience that may have varying level of domain knowledge/technical knowledge.

5.2.Safety Requirements

There are no safety requirements with this application, other than any normal hazards of a computer device.

5.3.Security Requirements

Database has to be reached securely and its data should not be lost. It also should not change or manipulate the data unnecessarily except updates. Since our database contains some personal information of user security design is important to see that no information is misuse.

5.4.Software Quality Attributes

5.4.1 Usability

The scope of the product is large. Age of users doesn't matter. The LMS will be developed keeping users in mind.

5.4.2 Correctness

The software should be correct in terms of its functionality and hence the internal processing must be immaculate. This means that the application should adhere to functional requirements.

5.4.3 Completeness

The software aims to cover all necessary requirements that it is expected out of it.

5.5.Business Rules

The admin is given database access and modification rights. Other users have no access to the database and can make use of other functionalities available to registered and nonregistered users.

6.Other Requirements

There are no other requirements associated with this project.

Appendix A: Glossary

User: Someone who interacts with the application

Web-Portal: A web application which presents special facilities for the user

Admin: System administrator who is given specific permission for managing and controlling the system

TCP/IP: TCP/IP stands for Transmission Control Protocol/Internet Protocol, which is a set of networking protocols that allow two or more computers to communicate

HTTP: HTTP stands for Hyper Text Transfer Protocol which is an application protocol for distributed, collaborative, and hypermedia information systems.

Appendix B: Analysis Models

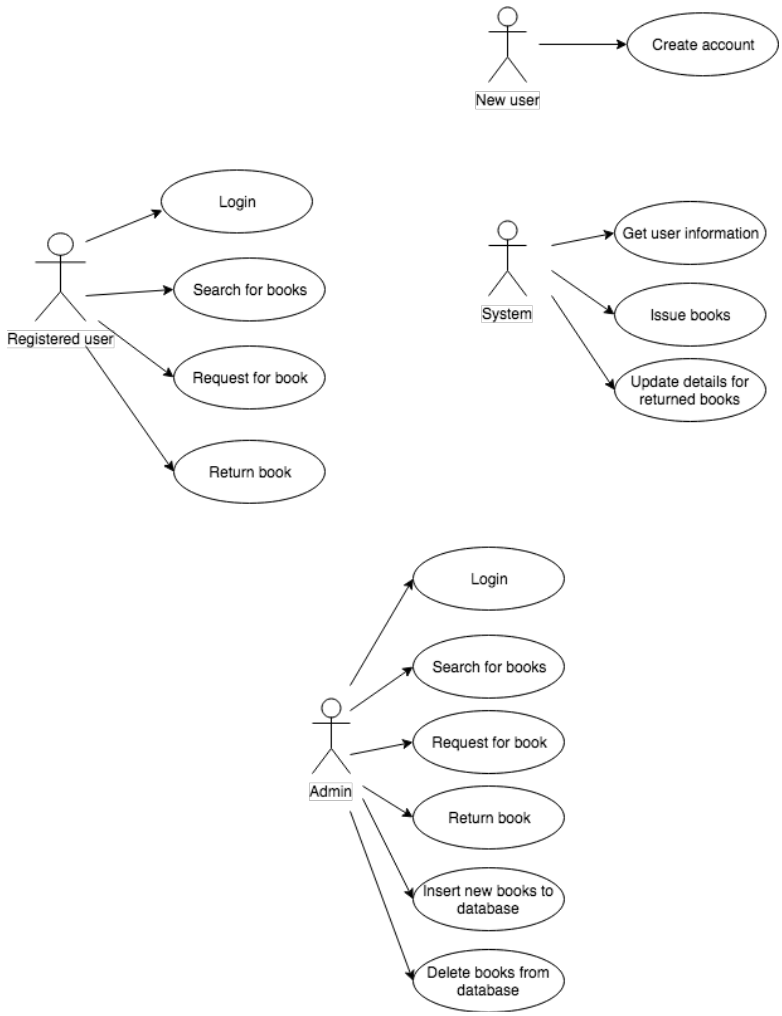


Figure 1 - Use Case Diagram

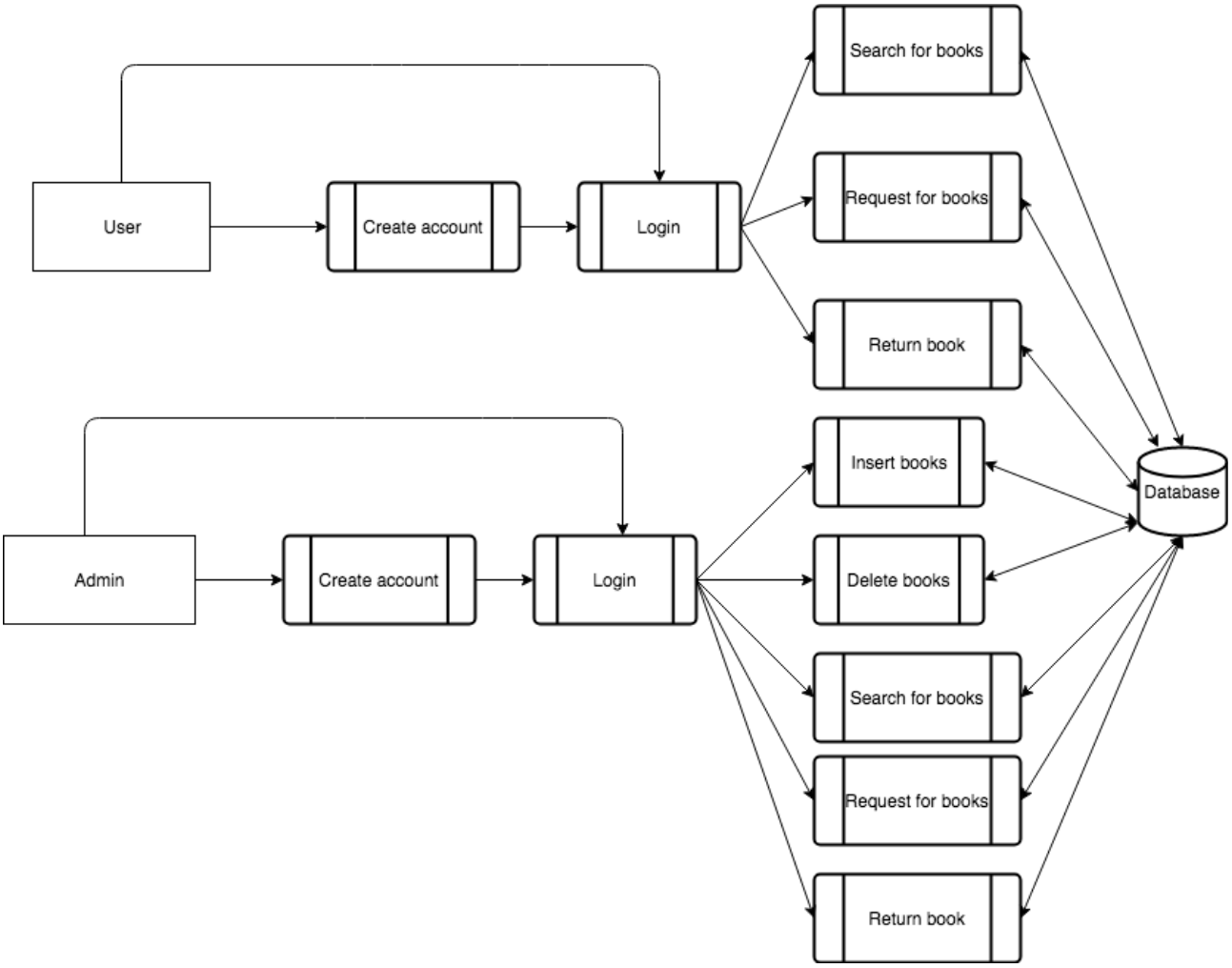


Figure 2 - Flow Diagram

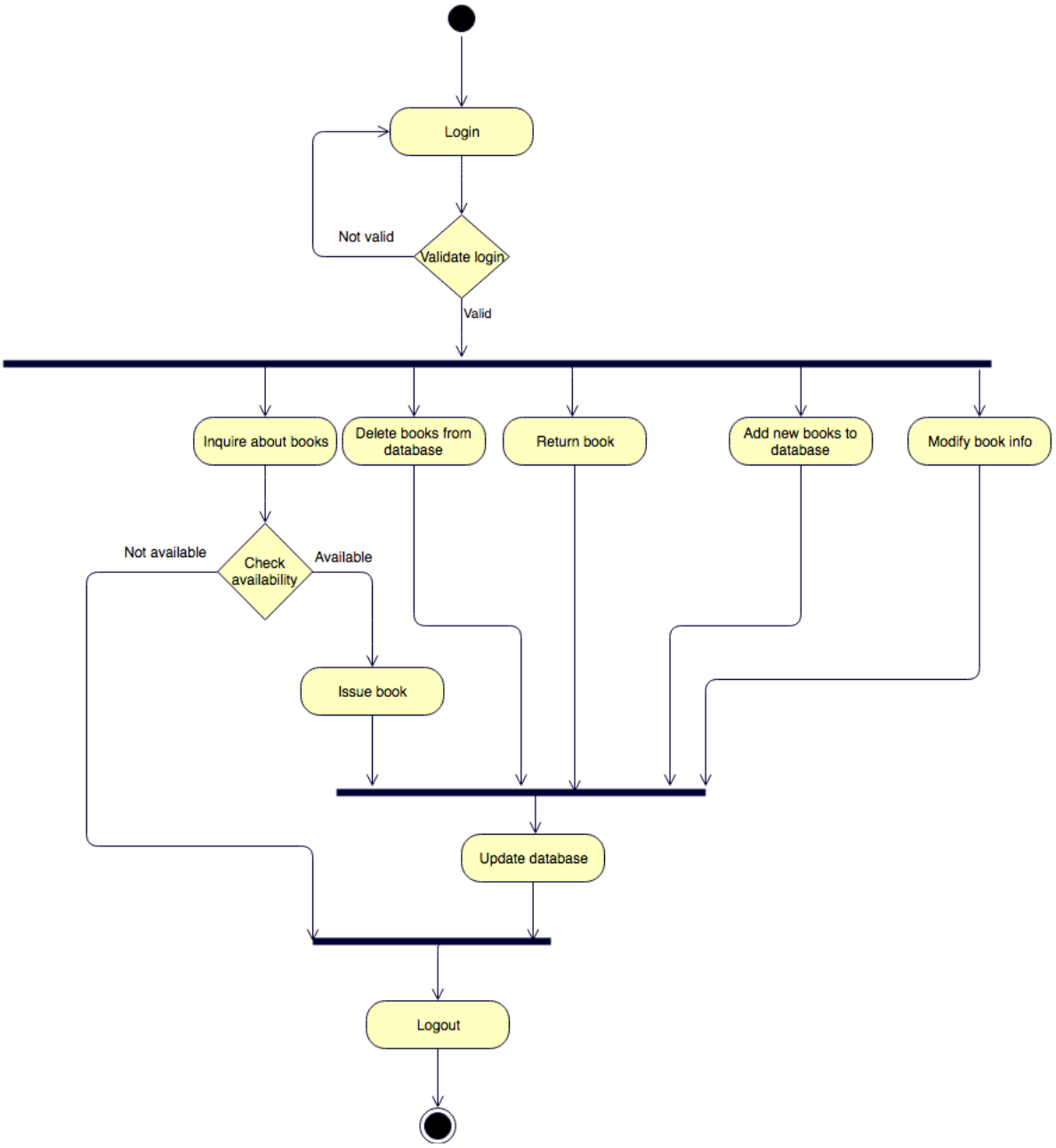


Figure 3 - Activity diagram for admin

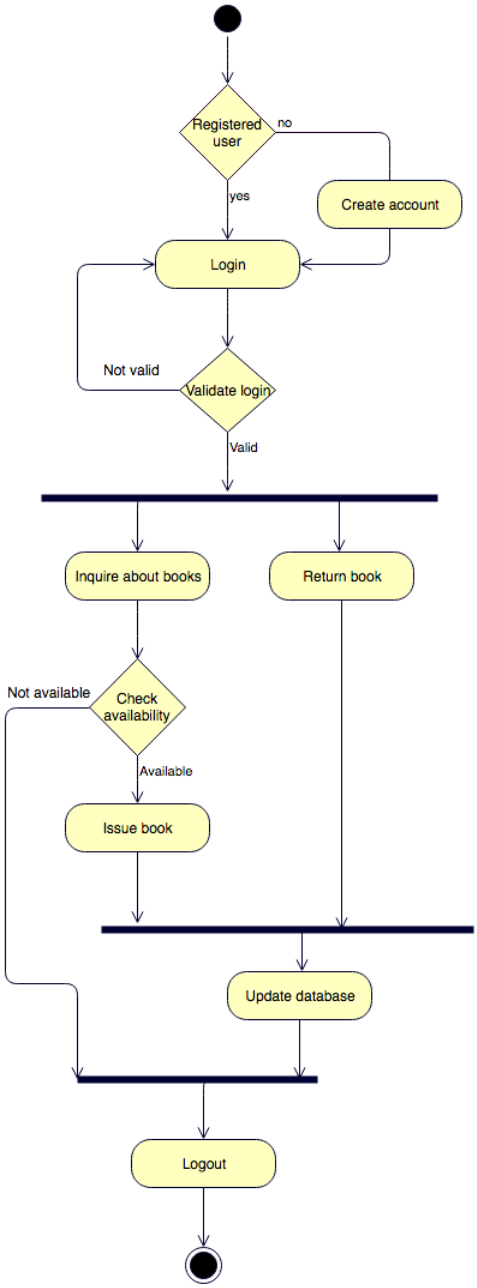


Figure 4 - Activity diagram for user

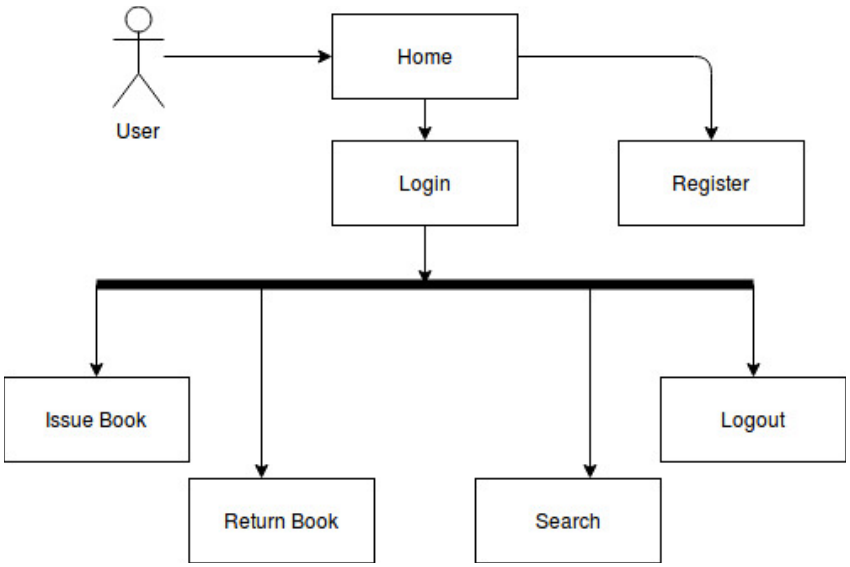


Figure 5 - Navigation diagram for user

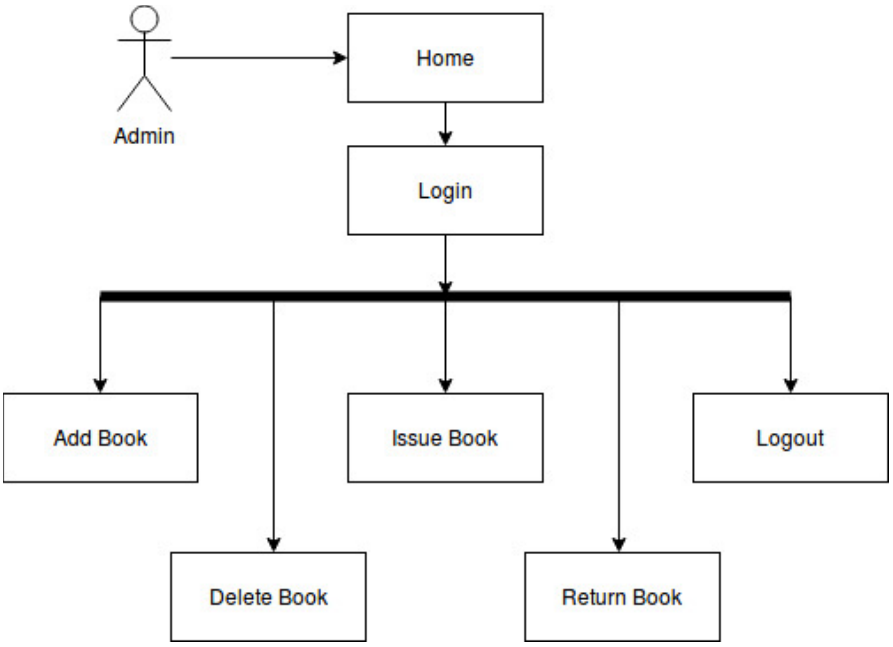


Figure 6 - Navigation diagram for admin

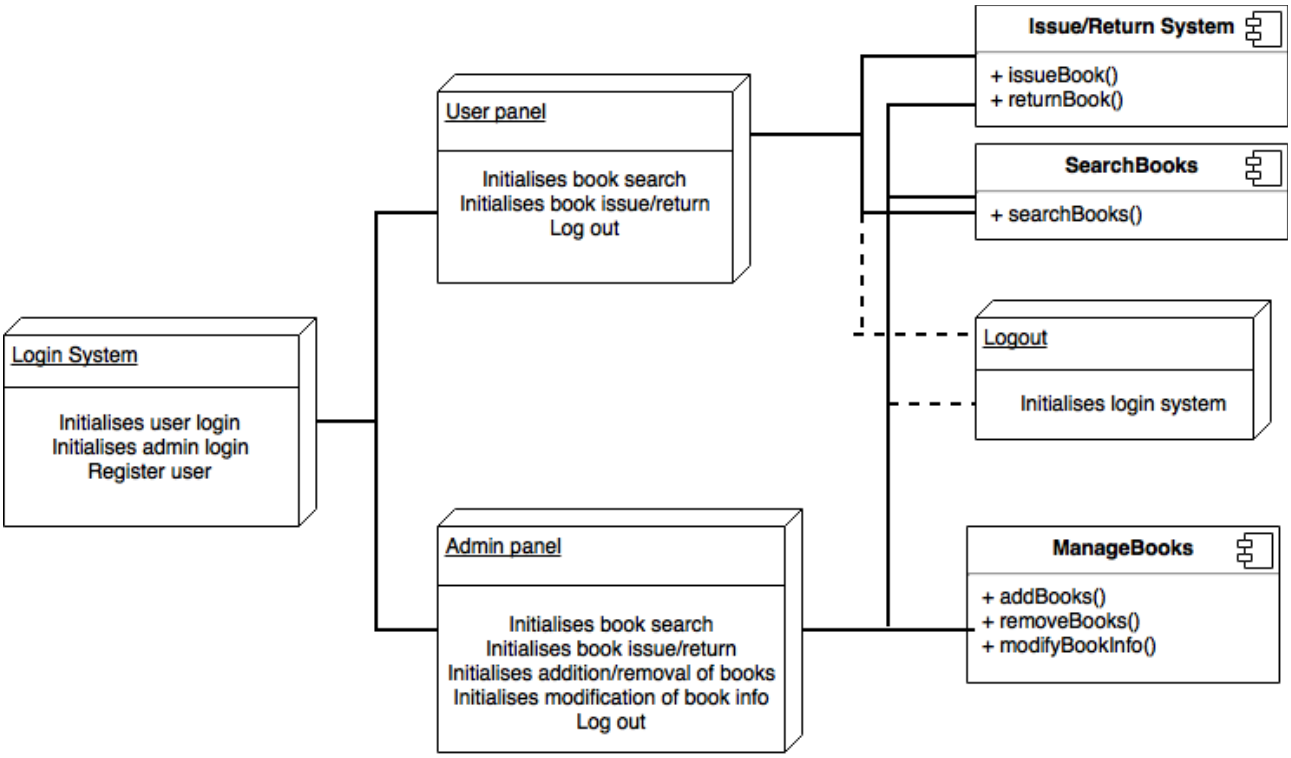


Figure 7 - Deployment diagram

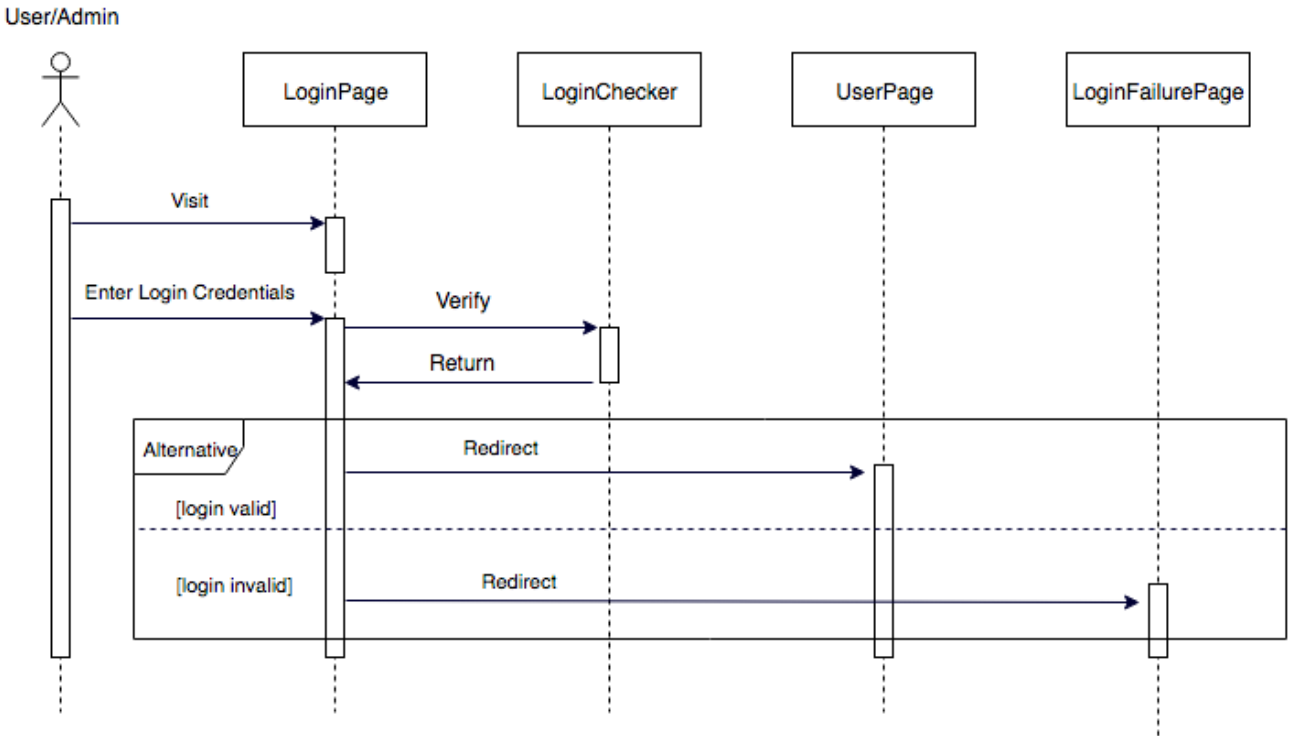


Figure 8 - Sequence diagram for login

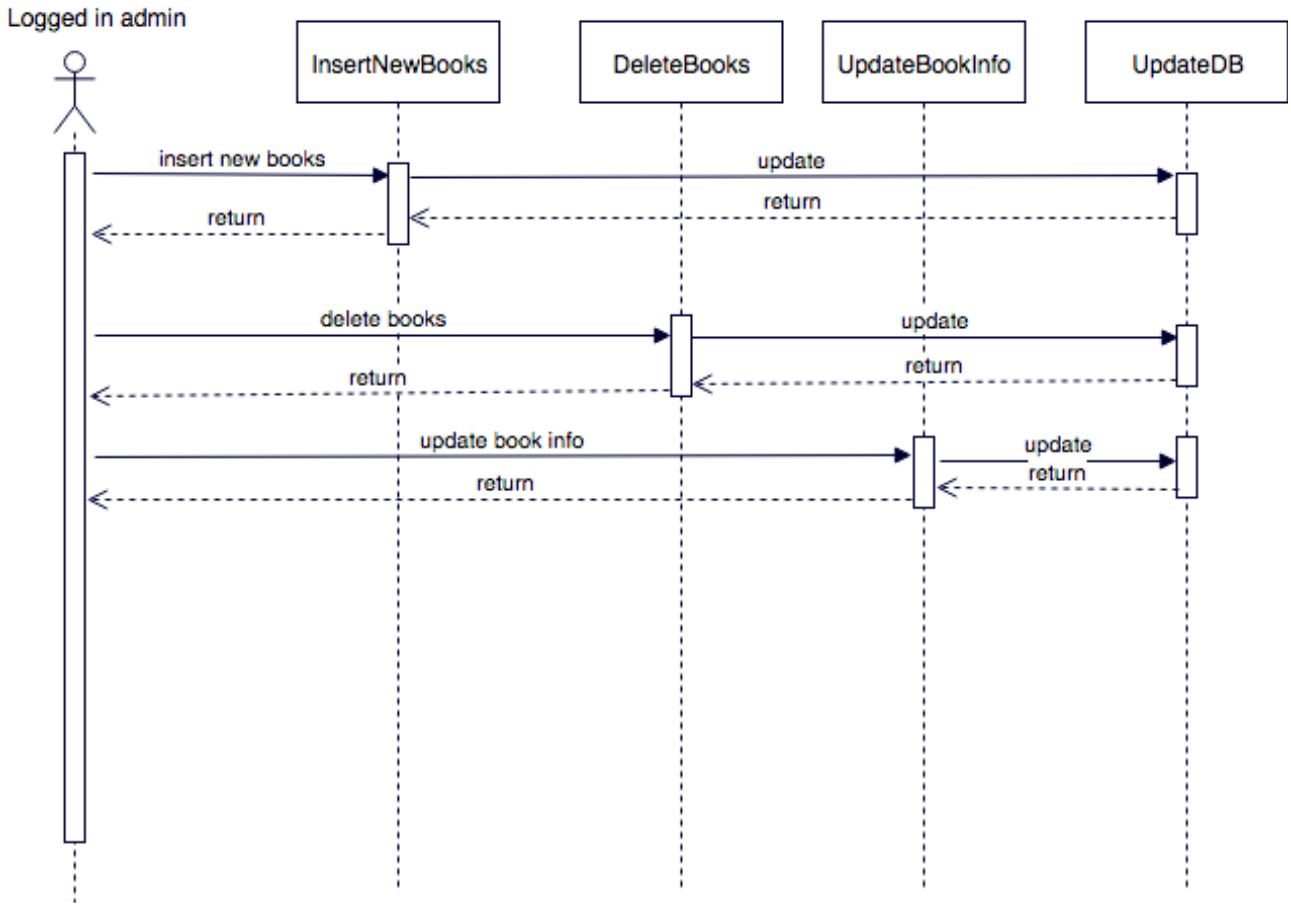


Figure 9 - Sequence diagram for logged in admin

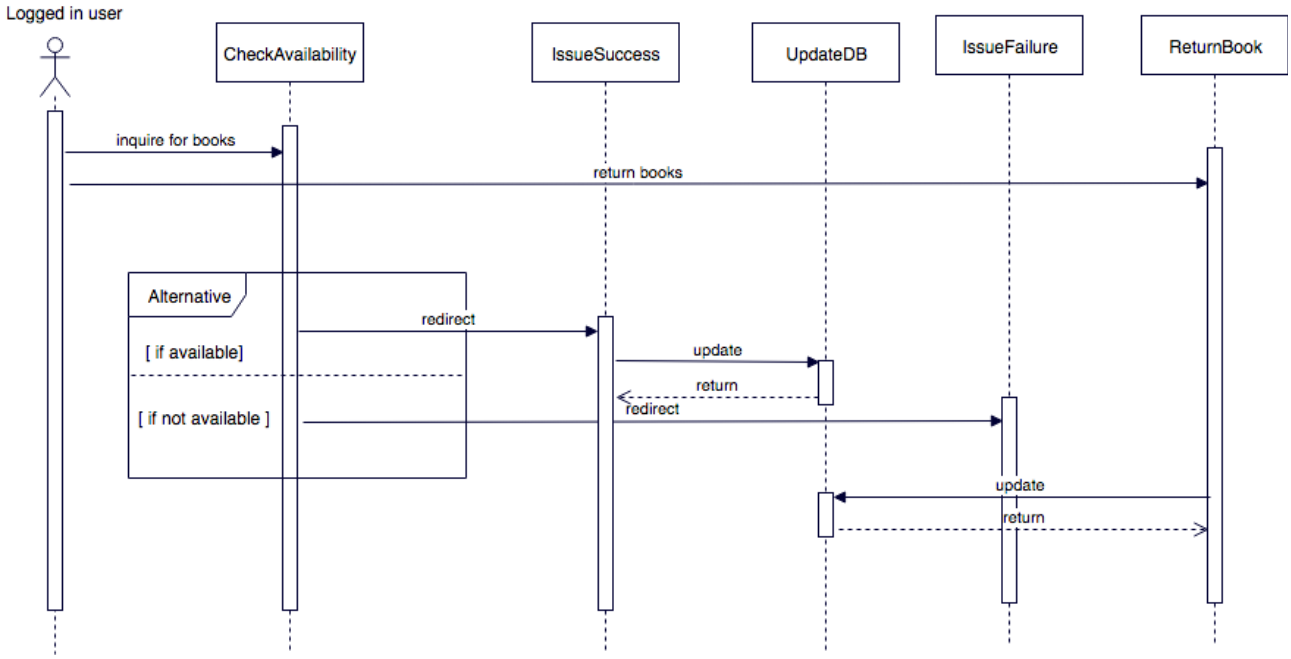


Figure 10 - Sequence diagram for logged in user

Appendix C: To Be Determined List