

EPOKA Library System Requirements Specification

Version 2.0

June 3, 2023

**Software Modelling and Design**

**Contributors:**

Amara Çela

Anita Mjeshtri

Irisa Nazari

John Nase

Patrik Madhi

Sara Berberi

**Table of Contents**

[**EPOKA LIBRARY SYSTEM REQUIREMENTS SPECIFICATION 1**](#_heading=h.19c6y18)

[**VERSION 2.0 1**](#_heading=h.3tbugp1)

[**JUNE 03, 2021 1**](#_heading=h.28h4qwu)

[**1.**](#_heading=h.nmf14n) **EXECUTIVE SUMMARY 4**

[1.1](#_heading=h.37m2jsg) project overview 4

[1.2](#_heading=h.1mrcu09) purpose and scope of this specification 5

[**2.**](#_heading=h.46r0co2) **PRODUCT/SERVICE DESCRIPTION 5**

[2.1](#_heading=h.2lwamvv) product context 6

[2.2](#_heading=h.111kx3o) user characteristics 6

[2.3](#_heading=h.3l18frh) assumptions 7

[2.4](#_heading=h.206ipza) constraints 7

[2.5](#_heading=h.4k668n3) dependencies 8

[**3.**](#_heading=h.2zbgiuw) **REQUIREMENTS 9**

[3.1](#_heading=h.1egqt2p) functional requirements 9

[3.2](#_heading=h.3ygebqi) non functional requirements 19

[*3.2.1*](#_heading=h.2dlolyb) *Product Requirements 19*

[**3.2.1.1**](#_heading=h.44sinio) **User Interface Requirements** 19

[**3.2.1.2**](#_heading=h.2jxsxqh) **Usability** 19

[**3.2.1.3**](#_heading=h.z337ya) **Efficiency** 23

[3.2.1.3.1](#_heading=h.3j2qqm3) Performance Requirements 23

[3.2.1.3.2](#_heading=h.1y810tw) Space Requirements 23

[**3.2.1.4**](#_heading=h.4i7ojhp) **Dependability** 24

[**3.2.1.5**](#_heading=h.2xcytpi) **Security** 25

[*3.2.2*](#_heading=h.sqyw64) *Organizational Requirements 26*

[**3.2.2.1**](#_heading=h.3whwml4) **Environmental Requirements** 26

[**3.2.2.2**](#_heading=h.2bn6wsx) **Operational Requirements** 26

[**3.2.2.3**](#_heading=h.qsh70q) **Development Requirements** 26

[*3.2.3*](#_heading=h.3cqmetx) *External Requirements 27*

[**3.2.3.1**](#_heading=h.1pxezwc) **Regulatory Requirements** 27

[**3.2.3.2**](#_heading=h.49x2ik5) **Ethical Requirements** 27

[**3.2.3.3**](#_heading=h.2p2csry) **Legislative Requirements** 27

[3.2.3.3.1](#_heading=h.147n2zr) Accounting Requirements 28

[3.2.3.3.2](#_heading=h.3o7alnk) Security Requirements 28

[3.3](#_heading=h.1rvwp1q) domain requirements 28

[**4.**](#_heading=h.4bvk7pj) **SOFTWARE DESIGN 29**

4.1 User Scenarios 29

4.2 Use Cases 37

4.3 Use Cases Extended 42

4.4 BPMN 65

4.5 Data Flow Diagrams 71

4.6 Entity-Relationship Diagram 78

4.7 Relational Schema 79

4.8 Activity Diagrams 80

4.9 State Diagrams 91

4.10 Sequence Diagrams 95

4.11 Collaboration Diagrams 106

4.12 Class Diagrams 114

5. Design Patterns 115

# **Executive Summary**

## Project Overview

To facilitate and enhance a better access to the education system, an efficient solution is to digitalize the University’s Library in order to bring it closer to all the students as well as to provide better management features to the librarians and administrators of the library. Our work consists of taking all the necessary steps that will result in the creation and maintenance of the finished product: the library web application for Epoka University. As mentioned above this system will be available to three levels of users: the university’s students, the librarians and the administrators.

The account of the administrator will be provided using the backend of the web application development, meaning that the moment this application is deployed, the administrator will be able to access his/her account to which he/she can make modifications, such as changing the password. The supply of the library items and their entry in the system will be an available feature for the administrator. Therefore, he/she will be able to enter the information about any new items, such as their title, authors, genres, description, cover page and the number of copies of that particular item that are in stock. In addition to that, the administrator will also be able to update the information provided for each book, such as when errors in their registration are made or when he/she changes the number of copies. This user category is also provided with a feature of deleting library items. The administrator can find the entered items in the main page of his/her account. He/she will be able to access statistics of the books that are borrowed and which ones are requested the most in the library. The administrator account will be provided with a feature of receiving messages that come as feedback from students. An important functionality of the administrator will the ability to register librarians, using the Epoka university email of the person they are registering.

Librarians will be able to approve the registration made by the administrator, by clicking approve or reject in the email automatically sent when the administrator is attempting to make the registration. The home page of the librarian account will also contain all the available items of the library. In addition to that he/she will manage the orders made by students: approving orders which sets their status as ‘approved’, changing their status as ‘borrowed’ when the item has been picked up by the student, changing their status as ‘returned’ when the item is returned, and contacting the student when the return date is surpassed.

This web app will also provide useful features for all students, including but not limited to viewing the available items along with information and descriptions for each of them, ordering them for pickup, personalizing their account by adding items to favorites, as well as giving feedback or requests. In order to register, students are only required to create an account using their Epoka email.

The product of this ongoing process of communication, design and implementation will bring down barriers in the accessibility of education and information, helping students find the needed books without having to check the entire physical library. Furthermore, it will also help administrators keep track of the books, and librarians keep track of orders, making sure that the return deadline is respected.

## Purpose and Scope of this Specification

The main purpose of our project is to create, develop, and implement a modern and comprehensive electronic library in our university that will provide all the students, and every faculty member with seamless access to a wide range of books, where they will be able to have their own personal account, mark their favorite library items, order electronically hard copies of the books, and send feedback based on their personal experience. The project aims to enhance the overall learning experience at the university and make the process of finding items and ordering them easier for both students and the Epoka library staff.

The scope of the Epoka electronic library project includes the following:

- Development and implementation of the electronic system for the Epoka University library: The process of developing a software to adapt to specific needs of the university community includes the selection of the appropriate software as well as the customization and integration of software with different problem solutions to produce a user-friendly platform as efficiently as possible.

- Access and authentication: The Epoka electronic library system will provide secure and authenticated access to a wide range of electronic resources and will ensure that only authorized users can access the system.

- Items management and organization: The electronic library of our university will ensure that the books are easy to find electronically and can be ordered to access them physically.

- Search tools: The electronic library system will provide advanced search to allow its users easy access to books.

- User account management: The system will allow the students to create and manage their own accounts.

- Accessibility: The Epoka electronic library system will be easily accessible to all users, including those with disabilities.

Our purpose as aspiring Software Engineers is to leave a mark on our university by providing a comprehensive and user-friendly platform for them, supporting learning activities, and enhancing the overall quality of study and experience of the university community.

# **Product/Service Description**

Our team is dedicated to improving access to education by digitalizing the Epoka University library. We have designed a web application that brings the library closer to the students and provides efficient management features for its librarians and administrators. The system will allow administrators to manage the library items by adding, modifying or deleting information for each book. On the other hand, librarians will manage the orders and feedback sent from the students. Students will be able to view the available books, order them for pickup, personalize their account by adding books to favorites and provide feedback or requests. The system will provide a better communication between users, which will enhance the learning experience of the students. The system will be available only for active undergraduate and graduate students of Epoka University.

## Product Context

The Epoka University Library System will be a self-contained system and independent of other products. However, it will interface with other related systems such as the university’s student information system and database. It will be integrated with the student information system to ensure that only registered students can access the library system. It will also be integrated with the university’s authentication system to ensure secure access to the system. Our system will be designed to work smoothly with these other systems, ensuring easy operation for the users. The application will be accessible from any device with internet access.

## User Characteristics

The application will be used by three types of users: students, librarians and administrators. The user profiles for each of them are: -University Students: The primary users of the library web application are university students from different faculties like Architecture and Engineering, Economics and Administrative Sciences, Law and Social Sciences and different academic levels, including both undergraduate and graduate students. They will use the application to access information on available books, order for pickup, personalize their account and give feedback or requests. Students are expected to have basic technical knowledge and experience in using web applications; however technical expertise is not a requirement to use the system. -Librarians: The librarians who will use the library web application are part of the existing staff or the University. They will manage the orders of the students and the feedback sent from them. They will also contact students who have not returned borrowed books according to the deadline. They will need approval from the administrators to gain access to the system. They are expected to have advanced technical knowledge and experience in library management. -Administrators: The administrators will be responsible for adding, modifying or deleting the information for each of the books. They will also register librarians, so only authorized staff can take this position within the system. Administrators are expected to have advanced technical knowledge and experience in database managements and library administration.

## Assumptions

• It’s important to have a reliable internet connection available for both students and staff to access the system.

• The hardware equipment, such as computers, laptops or mobile devices, should be able to access the web application.

• The users, including librarians and administrators, have basic computer skills and are familiar with the internet.

• The users should also have access to the necessary login credentials, such as usernames and passwords, to access the web application.

• The system should be developed using programming languages and frameworks that are compatible with the university’s existing IT infrastructure.

• The web application should be designed with responsive design principles to ensure that is can be accessed on different devices with various screen sizes and resolutions.

• The system must respect the data protection regulations and provide enough security measures to protect user data and prevent unauthorized access.

•It is assumed that every active student of the university possesses an active Epoka account.

•It is assumed that the online library system will represent the actual physical items with high accuracy and real-time updates.

## Constraints

There are several factors that will limit the design options for the digitalized library system for Epoka University.

**Parallel Operation with an old system**:

This will be Epoka University’s first library system so our system will not be operating with an old system. However, our system may need to operate in parallel with other systems Epoka has. This may require our system to interface with the other systems.

**Audit Functions**:

Another constraint is the need to track user activity and maintain the integrity of the system. This will affect how data is collected, stored and analyzed.

**Access, management and security**:

The system will need to have access controls and security measures in place, to protect the data and ensure that only the authorized users can access the system. Since our product will be a web application, each user needs to have access to the internet. These constrains will affect the design of the authentication and authorization mechanisms, as well as the encryption and algorithms used to secure the data.

**Criticality of the application**:

The criticality of the web application will constrain the design options in terms of reliability, fault tolerance and disaster recovery. For example, the system is somehow critical to the functioning of the university, and it may need to have redundant servers and backup systems in place to ensure continuity of service.

**System resource constraints:**

Since the Epoka University library system will be a web application, disk space or other hardware resources may not be a significant constraint. However, there may be other resource constraints that need to be considered, such as the amount of memory and processing power required to handle user requests and to ensure that the system runs smoothly. The design of the system architecture and algorithms will need to take these constrains into account to optimize the use of resources and ensure efficient performance.

**Other design constraints:**

There may be other design constrains imposed by the university, such as programming language or framework preferences. These constrains will affect the choice of technologies and tools used to develop the system. We will be using HTML, CSS, JavaScript and PHP to develop the library web application. In addition, to connect the PHP server with the database, which will be built using MySQL, we will also use JDBC. These choices of technologies and tools are influenced by the university’s preferences.

## Dependencies

• The system may require accurate and up-to-date data from the university’s existing library.

• The system may need to be integrated with other university’s systems, such as the student information system or learning management system, to ensure seamless access to library resources and services.

• The development of the system may rely on specific technologies or tools, such as a particular programming language, database system or web framework.

• Certain features or modules may need to be developed back-to-back and may depend on the completion of other project milestones or components, such as the design of the user interface.

# **Requirements**

## Functional Requirements

| **Req#** | **Requirement** | **Comments** | **Priority** | **Date Reviewed** | **SME Reviewed / Approved** |
| --- | --- | --- | --- | --- | --- |
| BR\_01 | The system shall allow the registration of either Epoka University students, librarians and library administrator. | The users if the Epoka Library System will be either students, librarians or administrators. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_02 | The system shall provide different graphical user interfaces for each of these three types of users. | Each type of user will be provided with different features, therefore it is necessary to have a different GUI. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_03 | The student user shall be able to register by providing a valid Epoka email address, first name, last name, password and phone number. | The email address will be checked if valid, based on the specifications of Epoka Email for students, consisting of the first letter of the first name, the last name and enrollment year. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_04 | If the status of the student becomes passive from the university, his status in the library system will also change to passive. | When a student is transferred, or he/she finished his/her studies the status is changed to passive. | 2 | 06/04/23 | Anita Mjeshtri |
| BR\_05 | The student user shall be able to verify his/her registration by receiving a special code in their email. | To ensure that the email entered for registration actually belongs to the user, the code will be sent by email and entered on the website by the student registering. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_06 | The student user shall be able to log in to his/her account by providing the correct information: Epoka email and password. | To access his/her account the user should input a valid email address that matches with the entered password for that address. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_07 | The system shall grant student users the information: including title, author, genre, picture of the cover and the number of books in stock, for all the books offered by the EPOKA University Library, in the home page. | The home page will contain all the available books along with the information about each of them. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_08 | The student user shall be able to click on the desired book, to get more information about the book, including the information mentioned above as well as a short description. | The students can view the information about all the books available in the library. | 2 | 06/04/23 | Anita Mjeshtri |
| BR\_09 | The student user shall be able to order a book, that he will get to pick up within the specified time frame. | Student users can order a book, whose status will be 'ordered' until the student picks it up, in this case its status will change to 'borrowed'. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_10 | When a book is ordered by a student, its stock number shall decrease. | Each time a book is ordered by a student user its stock will decrease, until it reaches 0. In this case the book can still be viewed but not ordered. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_11 | The student user should be able to save books as favorites. | Students can save books as favorites that will be added to their personal favorites page(even when the stock of the book is 0) that is unique for each student. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_12 | The student user should be able to filter the books by the desired genres and authors that are available. | Student users can choose to view only the books of a certain genre or of a certain author. | 2 | 06/04/23 | Anita Mjeshtri |
| BR\_13 | The student user shall be able to search for a specific book title. | The student can search for the desired book by book title. | 2 | 06/04/23 | Anita Mjeshtri |
| BR\_14 | The student user should be able to give feedback or send a message about suggestions or requests. | Students can send suggestions or feedback about the books and the service. | 3 | 06/04/23 | Anita Mjeshtri |
| BR\_15 | The feedback sent by users should appear on the account of the librarian and administrator. | The messages sent by student users can be read by both the librarian and the manager. | 3 | 06/04/23 | Anita Mjeshtri |
| BR\_16 | The student user can view his/her profile. | Student users can view the information they have entered: email, password, first name, last name and phone number. | 2 | 06/04/23 | Anita Mjeshtri |
| BR\_17 | The student user can modify his/her name, last name, password, phone number and profile picture in the profile page. | Users can change their name, last name, phone number, password and modify the default profile picture. | 2 | 06/04/23 | Anita Mjeshtri |
| BR\_18 | The main administrator shall be registered from the server side, when the web application is made available. | To ensure that the library is administrated by the competent person, his account will be the first one registered. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_19 | The library administrator shall be able to log in to his/her account by providing the correct information: Epoka email and password. | The administrator must enter the valid Epoka email and password to access his/her account. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_20 | The administrator shall be able to enter new books, along with the information about each of them (title, author, genre, description, cover page and stock). | The administrator enters the information about each new book he/she enters into the system | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_21 | A list of all the available books shall be represented in the home page of the account of the library administrator. | The administrator can view all the books and their information. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_22 | The administrator shall register the librarians by inputting their information (Epoka email, password, first name, last name and phone number). | The administrator can enter the information about the librarian that will be entered in the system | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_23 | The librarian can validate the information entered by the administrator, by receiving an email with this information. | To verify the account creation the librarian can check and verify it from his/her email account. | 1 | 06/04/23 | Anita Mjeshtri |
| BR\_24 | A messages section shall be provided to the library administrator to which the incoming feedback from student users shall be stored. | Library administrators can review the messages sent by the student users. | 3 | 06/04/23 | Anita Mjeshtri |
| BR\_25 | The administrator of the library shall be able to update the information about a book. | The administrator can modify the title, author, genre, stock, description and cover page of the book. | 2 | 06/04/23 | Anita Mjeshtri |
| BR\_26 | The administrator of the library shall be able to delete books. | The administrator can delete books when they are removed from the library. | 2 | 06/04/23 | Amara Çela |
| BR\_27 | Each student shall be able to see all the books in his/her area of study by selecting his/her specific field of study. | The student can choose any of the areas of study offered at Epoka University (Software Engineering, Law, Economics, Business Informatics etc.) and search for the list of books that they will study through all the years at Epoka University. | 3 | 06/04/23 | Amara Çela |
| BR\_28 | The student shall be notified one day before the deadline for returning the borrowed book. | A notification will be sent to the students one day before he/she has to return the book. | 2 | 06/04/23 | Amara Çela |
| BR\_29 | The student shall be able to see or edit his/her favorite book list. | The students can see all the books in the favorite section, but they also can remove a specific book from their favorites list by pressing a button. | 2 | 06/04/23 | Amara Çela |
| BR\_30 | The student shall be able to see or edit his/her ordered book list. | The students can see all the books in the order section, but they also can remove a specific book from their ordered list by pressing a button, when the book has not yet been picked up by him/her. | 1 | 06/04/23 | Amara Çela |
| BR\_31 | When a book is returned, its stock number shall increase. | When a book is returned, its stock number will be increased by the librarian. If the stock of a book was zero and that book was unavailable to order for other students, that specific book will now be available for anyone who wants to order it. | 1 | 06/04/23 | Amara Çela |
| BR\_32 | Each student shall manage his/her notification settings. | The student shall be able to manage if he/she wants to receive notifications from Epoka Library, such as the notification for the book returning. | 3 | 06/04/23 | Amara Çela |
| BR\_33 | Each student shall be able to deactivate his/her account for a specified period. | Students may choose to temporarily deactivate their account for a week or a month. | 3 | 06/04/23 | Amara Çela |
| BR\_34 | Each student shall be able to delete his/her account. | Students will be able to permanently delete their account and will be asked to enter their account password and to confirm the deletion of the account, while being aware that all the history of their account (favorite and ordered lists) will be deleted as well. | 2 | 06/04/23 | Amara Çela |
| BR\_35 | Each student shall be able to find support in the “Help and Support” section. | Students should be able to find the frequently asked questions which may answer the question they may have. Otherwise, they can contact the librarian through the contact information provided for them. | 2 | 06/04/23 | Amara Çela |
| BR\_36 | The administrator shall be able to view or edit his/her profile. | The administrator can edit his/her name, surname, phone number, password. | 1 | 06/04/23 | Amara Çela |
| BR\_37 | The administrator should receive a confirmation email every time he/she tries to change his/her account password. | To change the password, the old password should be entered by the administrator and then an email confirmation about the password change will be sent to the administrator. | 2 | 06/04/23 | Amara Çela |
| BR\_38 | The administrator shall be able to view the statistics of the library. | The administrator can see the most ordered books from the students. | 1 | 06/04/23 | Amara Çela |
| BR\_39 | The administrator shall be able to discharge a librarian. | The administrator can remove a librarian from the system by deleting his/her account. | 2 | 06/04/23 | Amara Çela |
| BR\_40 | The librarian shall be able to view his/her profile. | The librarians can see their profile but cannot edit their personal information after they have confirmed it to the administrator. | 2 | 06/04/23 | Amara Çela |
| BR\_41 | The librarian shall be able to login into their account by entering their Epoka email and password. | The password is given to the librarian by the administrator who is the person who created his/her account. | 2 | 06/04/23 | Amara Çela |
| BR\_42 | When the librarian is trying to delete his/her account a permission request is sent to the administrator. | Since the account was initially created by the administrator and this post was granted to him/her by the administrator, they should first quit their post and then, after their request is approved, they may be granted permission to delete their account or their account may be deleted directly from the administrator. | 1 | 06/04/23 | Amara Çela |
| BR\_43 | The librarian shall be able to see the list of the books in the system. | The librarians can see all the books registered in the system with all their detailed information. | 1 | 06/04/23 | Amara Çela |
| BR\_44 | The librarians shall be able to confirm the returned book. | After a book is returned by a specific student, it is confirmed by the librarian and the stock number of that specific book is increased. The book is removed from the student’s ordered list. | 1 | 06/04/23 | Amara Çela |
| BR\_45 | The librarian shall be able to search books by their title or by their author. | Each librarian should be able to search for a specific book by the book name or its author. | 2 | 06/04/23 | Amara Çela |
| BR\_46 | The administrator shall be alerted when the stock number of a book falls below five. | An alert message should be displayed to the administrator whenever there are less than five copies of a book. | 2 | 06/04/23 | Amara Çela |
| BR\_47 | The student shall be able to order only one copy of the book at a time. | Students cannot order two copies of the same book. | 1 | 06/04/23 | Amara Çela |
| BR\_48 | Administrators shall be able to modify the information of the librarian account. | If the administrator considers a change in the librarian account as necessary, he/she can do it with the librarian consensus. If the librarian  would like to change something, they should ask the administrator to do it. | 2 | 06/04/23 | Amara Çela |
| BR\_49 | The librarian shall be notified when a student has not returned the borrowed book within his/her deadline. | A notification will be shown to the librarian when a specific student has not returned the book within the deadline and the librarian should contact the student. | 1 | 06/04/23 | Amara Çela |
| BR\_50 | Students with disabilities shall be able to use the library as any other student, without any difficulties. | Epoka Library System is being designed to be accessible to all Epoka students. Students with disabilities should be able to use features  such as screen reader compatibility, keyboard navigation or alternative text for images. | 1 | 06/04/23 | Amara Çela |
| BR\_51 | The Epoka electronic library shall be optimized for any device, allowing Epoka’s students to access electronic resources from their devices. | Students, administrators, librarians and anyone part of the Epoka University can access Epoka’s library electronically from any device | 1 | 06/04/23 | Amara Çela |

## Non-Functional Requirements

### Product Requirements

#### **User Interface Requirements**

The user interface of this system is divided into five (5) main sections:

* Registration interface
* Home page interface (books)
* Student Interface
* Librarian interface
* Administrator interface

**Registration Interface:**

This section outlines the requirements for the registration interface of the Epoka University Library System.  The registration interface will consist of two main processes, "Log In" and "Sign Up".

The Sign-Up process can be completed only by students since the administrator must be registered when the system is deployed, and the librarians will be added manually by the administrator. The account creation process for the students will be completed using the email address provided by Epoka University.

The students must enter their first and last name, username, phone number, and password. Before clicking sign up the system prompts the students to verify their passwords.

The Log-In interface will consist of a user entering the email address provided by Epoka University, and then the password. The Log-In interface can be applied to all user types.

A "forgot password" functionality will assist users who may forget their password or lose access to their accounts. Clicking on it will help users reset their passwords and regain access to their accounts.

**Home Page interface:**

This interface will be accessed by every registered user. The purpose of this home page is to display all of the books registered in the Epoka university library. The data will be filtered and organized to be easily navigable and user-friendly.

Initially, the user can search directly for the book they’re looking for, or the user can even specify their major by choosing a study degree and the list of courses taken from the syllabus will show on the page. The user may select a course and every book from that course will appear. The “view as” button can allow the user to show the books as a list, as small icon books, or as large icon books. The “order by” button will order the books by the number of downloads, publication time, title, author name, etc. A book is shown by its cover and title. By clicking on it the user can get more information about the book size, number of pages, name of the author, publication year, and edition. Additionally, student users have more complex and special functionalities on the homepage, which will be furtherly mentioned in the “Student Interface” section.

**Student Interface:**

Except for basic functionalities like searching and viewing books, student users have other special functionalities, such that by hovering over the book, the system can assist students to add the book to their favorites or order it by clicking on two separate buttons. Adding a book to favorites will make it appear on the “Book collection”. Furthermore, ordering or reserving the book will facilitate the students with information like library availability (borrowed, available) of the book, time, and date when the student can reserve the book. The reservation of the book by a student will cause them to get the approval if the book is on stock, otherwise the button can appear disabled, or the book will not show up as available. After approval, the book will appear as reserved automatically. Then the student must go to the library and get the book. If a student user does not show up, the librarian will wait for them for 12 hours, and after that the librarian can either deny the request or notify the students by their phone number, however, it is not mandatory to do so.

The “Book collection” functionality provides the students to see the books that are added to their favorites, and the ordered books. The students can also make their “Book collection” private or public as they wish. One week before the students get daily notifications about their due date to submit the books.

Students can give feedback on the system or even report bugs and inconveniences. These messages can be read by the librarians and the administrator. Student users can view their profiles and edit personal data.

A green dot in a student’s profile indicates an active user in terms of accessibility, while a grey dot implies that this user has no longer access to the library.

**Librarian Interface:**

To gain access to their interface, librarians first have to approve the registration of their profile by the administrator.

The librarian will have access to the default homepage without any specific features like the student users.

The “Borrowers” functionality consists of listing the borrowers of the books based on left days in ascending or descending order. If there are students that don’t submit their books in time, their records will immediately turn red. Then, the librarian can choose to click on their profile and call them by the phone number shown in their profile, or even ignore the late submission for a few days if the book has a low demand.

When a student submits a book in the library, the librarian must make a book “returned”, so the in-library availability of the book will become “available” for students to view. On the other hand, when many students reserve a book, the system will automatically approve the request of the student who applied first.

**Administrator Interface:**

The Administrator Interface is a key component of the system, as it facilitates the manual registration of librarians and the management of book records. Upon deployment, developers will manually register the administrator. The administrator is responsible for adding librarians to the system, which can be accomplished by selecting the "Add Librarians" button located in the left pane of the interface. The librarian registration process mirrors that of student registration.

The administrator will also add books and edit them. On the bottom of the “Add Librarians”, There will be a “View Books” which when clicked will show all the books as a table. The table will have columns like title, author name, quantity or stock, date of publication, date of import in the system, status (borrowed, returned, available), number of pages, etc. The administrator can edit the information of the book, delete the book from the database as a record, or even add a new book.

The administrator can read the messages that students may send, like reports about bugs, suggestions, or complaints in terms of the library service. The bug reports will be communicated to the software developers by the administrator, while other notifications will be handled properly by the administrator and even by the librarians if needed.

#### **Usability**

Usability of the software refers to the extent to which the product can be used to efficiently achieve the goals of the intended audience. There are a couple of points that make up the usability of a software:

* Learnability: the software must be easy to learn by the three different types of users. The application will be easy to understand for all students, librarians, and the administrator of the library. There will be clear instructions to each command in the different views of the user interfaces.
* Accessibility: the software will have to be available remotely as each student may use it from their own devices. The application should be available for both pc and mobile use.
* Responsiveness: the software must be highly responsive. It must be able to successfully respond to the user’s requests in real time, therefore operating on and changing data in real time on both the student and librarian side.
* Flexibility: the program should be flexible enough to accommodate different user needs and preferences. It will allow customization on certain parts such as setting or changing passwords and adding available books to the database. The developers will also be able to maintain and update the program when needed.
* Effectiveness: The app must be simple and easy to use for both students and the librarians running it therefore making it very practical and effective. It will help digitize and modernize the process of borrowing books from the school library.
* Consistency: the application must use well known terminology, so users find it easy to understand and navigate through the system.
* Efficiency: the user should find it very easy to complete every task, with easy commands and instructions even if it is the first time the user is in contact with the application or they haven't used it in a while.
* The application should be very simple to use for all students. It will approximately only take a few minutes for a student to get up to speed with the app as it will be a very well-designed, intuitive application.
* As they are essentially administering the application, librarians and administrators may need a bit more time to become completely familiar with all of its capabilities. Although it will take some time to become familiar with the app, it should be noted that a week of use should be sufficient for librarians and administrators to become skilled in using it.

#### **Efficiency**

Efficiency of the software does not mean just making sure that the software completes all the tasks given to it. There are two main groups of requirements that determine how efficient a system is, performance and space requirements.

##### Performance Requirements

* Response time: the system should be able to respond to the user requests in real-time with no lag or delay.
* Throughput: the system should be able to handle a relatively high flow of data at a time as multiple students may be using the application all at once. The program should be able to respond to every single request as soon as the command is given without compromising its speed or performance.
* Compatibility: the application should be compatible with different hardware such as computers or mobiles as long as they have a connection to the internet as most functions will require it. Regardless of the device the application is being used on, the performance and functionality should not be affected.

##### Space Requirements

Our software is a web application which means users will not have to download anything

to their device. This means that there is no required space needed for the application and

the storage of the user's device will not be affected. They will simply have to search for the

application online on a search engine to use it.

#### **Dependability**

* The data of the books in the Homepage depends on the BookList database.
* Registering in the library using the Epoka Mail depends on the Epoka Account.
* When the student users search and click courses, the courses are depending on the syllabus provided by the department of the university.
* The approval of a reservation request made by a student depends on the time the request was submitted.
* The account will be considered active and passive based on the information from EIS platform of Epoka University.

**Availability**

* The application will be available only if the device has access to the internet.
* The application will be available to users in any time, however in the weekends, holidays, or out of the university season reserving the books will not be available. Other features like viewing the books or adding them to favorites will still work.
* The application will be available at any geographic location by any device and phone number, however, only accounts with Epoka university email addresses will be able to get registered.

**Reliability**

The software has to run correctly at all times, even at nights or weekends it should produce the expected outputs for the given inputs consistently. Since this software is going to be used by a large number of students, it must be powerful enough to withhold all the students interacting with it at a time, without crashing or slowing down. It is important that the list of books updates as the students place their requests in real time.

To make sure the system is reliable, it will undergo a lot of testing before it is publicly used and it will be continuously monitored when published. To ensure that the system meets the user expectations, it will also be updated and maintained regularly after publishing

**Monitoring**

After publishing the application, the developers will continue testing out the system along with students and librarians for the upcoming days to ensure that it withholds all of the user's operating on it at a time while looking for any bugs along the way.

The developers will be in direct contact with the administrator of the system who will report any bug or fault found in the system by any user and will act on it as soon as possible with as little downtime as possible.

The system will be able to run on its own without the need of the developers and will be easily operated by the administrator and librarians. Each student can create their own account in the application and can then use the application to borrow books from the

library without the need of a librarian or developer.

**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.

**3.2.1.5 Integrity & Security**

The integrity of a software is critical for ensuring the reliability, security and trustworthiness of a software system. Therefore, the developers must take steps to maintain the software throughout its life cycle.

* Users will only be able to access the given interfaces according to the group they belong to.
* Students and librarians will only be able to change their own accounts data and information.
* The administrator can grant a librarian access to use the system.
* Students will not have direct access to the book database and will not be able to change any data on it.
* Account passwords will have to follow certain rules to make them harder to guess.
* There is no sensitive personal data stored within a user's account.
* Several security measures will be implemented on the software and will be revised continuously.
* The code will be reviewed and examined for any errors or vulnerabilities that could compromise its integrity.

### Organizational Requirements

#### **Environmental Requirements**

* Internet Access: The web app requires a reliable and stable internet connection to function properly.
* Web Browsers: The web app should be compatible with all the popular web browsers.
* Devices: The web app should be accessible on all devices like laptops, desktops, mobile phones and tablets.
* Security: The app should ensure that sensitive user data is very well protected.
* Technical Support: Any glitches or bugs that affect the app’s functionality should be dealt with quickly.
* User-friendly Interface: The app should have user-friendly features and be fairly easy to navigate and use to encourage student engagement.

#### **3.2.2.2 Operational Requirements**

* **User Authentication:** Students should be allowed to create an account and log in with the university’s email to access the app.
* **Book List:** The web app should provide the list of books the library has and the details for each book.
* **Borrowing System:** The web app should have the borrowing system that allows the users to borrow the books they need.
* **Search and Filter:** Students should be able to search and filter the book list easily by title, author, publication date, availability status and so forth.

* **Notification System**: The app should notify the students about the status of their borrowed books, or the availability of the books they want.
* **User Feedback:** the app should allow the users to provide feedback on the overall user experience.

#### **Development Requirements**

The app will be accessed through internet connection such as Wi-Fi or mobile data. Other network issues will be handled by the system.

### External Requirements

#### **Regulatory Requirements**

Personal information such as names, surnames, and borrowing history should be gathered, processed, and kept in a safe and lawful manner. The app should also follow some security standards, such as SSL and two-factor authentication, to prevent unauthorized access or security breaches.

#### **Ethical Requirements**

1. Request Permission Only When We Need It:

It is very easy to go overboard with sending access requests to the users. We might think it is easier to get those permissions out of the, but the user does not feel the same way. They feel uncomfortable.

2. Stop Promoting the Quantity of Engagement:

There’s a reason some apps are so engaging and addictive, and we will promote the opposite of that. We will advise our users to use the app only when needed.

3. Never Use Dark Patterns:  
Dark patterns are a type of design trick, and they may take many different shapes. For instance:

●  Asking ambiguous or perplexing questions in an effort to sway users' responses in favor.

●  Ads that look like ordinary content.

●  Automatically starting a subscription without reminding users the trial is about to expire.

#### **Legislative Requirements**

* **Copyright Laws:** The app should comply with copyright laws and policies, such as fair use guidelines and the Copyright Act.
* **Protection of Personal Information Laws:** The app should follow all the best practices regarding data protection and privacy. It should ensure the security of the data and protect the confidentiality of the records.
* **User Compliance:** When the students use the app, they must confirm their acceptance of the terms of use and privacy policies. If they do not agree with one or many of these terms, they are free to not use our application.

##### Accounting Requirements

Due to the nature of our web app, no accounting requirements will be implemented.

##### Security Requirements

* **Authentication:** The web app should require users to authenticate themselves before any access permission is given.
* **Authorization:** The web app should provide authorization on the users based on their roles and privileges. For example, students should only be allowed to order books and view their history, while librarians have additional privileges.
* **Encryption:** All sensitive data transmitted between the users’ devices and the app should be encrypted following a protocol like HTTPS.
* **Secure Passwords:** Password should be hashed and salted according to the industry standards.
* **Session Management:** The users should be logged out after a period of inactivity.
* **Regular Updates:** The app should regularly be updated to fix bugs and patch security flaws.

## Domain Requirements

We will use the top-level domain that educational institutions use, .edu domain. To be more specific the domain will be epoka.edu.al. Because the library will be part of Epoka University and will be used only by Epoka students, we are qualified to use that domain ending.

**4.Software design**

***4.1 User Scenarios***

STUDENT SCENARIOS

**User Scenario 1 - Student Reserving a Book**

1. The student accesses the library's online platform or website using their credentials.
2. The student navigates to the homepage or book section of the website.
3. The student scrolls through the list of books displayed on the homepage or uses the search function to find a specific book by title, author, ISBN code, or subject.
4. The student filters the list of books based on their preferences, such as popularity, number of pages, genre, title, or publisher.
5. The student clicks on a book of interest to view its details, including the title, author, publication information, and availability.
6. If desired, the student adds the book to their favorites or saves it to their collection for future reference.
7. The student checks the availability of the book by either viewing its status on the book details page or through a separate availability check feature.
8. If the book is available, the student proceeds to reserve it by selecting the "Reserve" or "Make an Order" option.
9. The student chooses a specific time during the day when they will show up at the library to collect their reserved book. They may be prompted to select an available time slot or provide additional information about their preferred pickup time.
10. The student confirms the reservation by clicking the "Reserve" or "Confirm Order" button.
11. The system verifies the availability of the book and reserves it under the student's name for the specified pickup time.
12. The student receives a confirmation message or notification with the details of their reserved book, including the pickup time and any additional instructions.
13. On the designated day and time, the student visits the library and presents their student ID or library card to the librarian.
14. The librarian retrieves the reserved book, verifies the student's identity, and completes the necessary borrowing process.
15. The student receives the reserved book and can proceed to borrow it for a specific duration.

**User Scenario 2 - Student Returning a Book**

1. The student visits the library in person to return a borrowed book.
2. The student presents their student ID or library card to the librarian or library staff.
3. The librarian manually records the book return in the library system.
4. The librarian verifies the book's condition, ensuring it is not damaged or missing any pages.
5. If the book is in good condition, the librarian acknowledges the return and updates the student's account accordingly.
6. The student receives a confirmation in email indicating the successful return of the book.

**User Scenario 3 - Student Browsing and Filtering Books by Qualities**

1. The student accesses the library's website by login.
2. After logging in, the student is directed to the homepage where they can view a list of books available in the library.
3. The student scrolls through the list of books on the homepage.
4. The student selects one or more filtering options such as popularity, number of pages, genre, title, or publisher to refine the book list.
5. The system applies the selected filters and presents the student with a narrowed-down list of books that match the chosen qualities.
6. Additionally, the student clicks on the course filter option to discover books relevant to their course.
7. The system retrieves books associated with the chosen course.
8. While browsing, the student comes across a book that captures their interest and clicks on it to access its detailed information.

**User Scenario 4 - Student Writing a Bug Report**

1. The student encounters a bug while using the library's website or platform.
2. The student finds the bug report section on the website.
3. They provide their name and contact information.
4. The student describes the bug, including steps and any error messages.
5. They may attach screenshots or videos.
6. The student includes device and browser information.
7. They submit the bug report.

**User Scenario 5 - Student Searching for a Book**

1. The student accesses the library's online platform or website using their credentials.
2. They navigate to the search feature or search bar on the website.
3. The student enters relevant search terms such as the book's title, author, ISBN code, or subject.
4. They initiate the search and wait for the system to retrieve the matching results.
5. The system displays a list of books that match the provided search criteria.
6. The student scans through the search results and clicks on a specific book of interest.
7. They are directed to the book's details page, which provides information about the title, author, publication, and availability.
8. The student can further explore the book's details, such as its description, table of contents, or reviews posted by other users.
9. If the book meets their requirements, the student can add it to their favorites or save it to their personal collection for future reference.
10. If the book is currently available, the student can proceed to reserve or borrow it based on the library's borrowing policies and procedures.
11. If the book is currently unavailable, the student can either add it to their waiting list or explore other similar books suggested by the system.
12. The student can continue their search or repeat the process to find additional books based on their needs and interests.
13. Once they have completed their browsing and search activities, the student can log out of the library's online platform.

ADMINISTRATOR SCENARIOS

**User Scenario 1 - Administrator Adding a New Book to the Library**

1. The administrator logs into the library management system.
2. They navigate to the "Add New Book" section.
3. The administrator enters the required details of the new book, such as title, author, publisher, and ISBN code.
4. They may also provide additional information like the number of pages, genre, and any other relevant data fields.
5. The administrator clicks the "Add" button to save the book information into the library's database.
6. The system validates the entered data and confirms the successful addition of the book.
7. The book is now available for students to search, reserve, and borrow.

**User Scenario 2 - Administrator Modifying Existing Book Data**

1. The administrator accesses the library management system and selects the "Modify Book Data" option.
2. They search for the specific book by title, author, ISBN code, or any other relevant criteria.
3. The system displays a list of matching results.
4. The administrator selects the book they want to modify from the list.
5. They make the necessary changes to the book's data fields, such as updating the ISBN code, title, author, publisher, or number of pages.
6. The administrator saves the modifications by clicking the "Update" or "Save" button.
7. The system validates the changes and updates the book's information in the database.
8. The modified book data is now reflected in the library's catalog and accessible to users.

**User Scenario 3 - Librarian Registration by Admin**

1. The admin accesses the library management system using their authorized credentials.
2. The admin navigates to the "User Management" or "Staff Management" section of the system.
3. The admin selects the option to register a new librarian.
4. The system presents a registration form for the admin to enter the librarian's details.
5. The admin fills in the required information, including the librarian's full name, email address, contact number, and any other necessary fields.
6. The admin assigns a unique username and password for the librarian or requests the system to generate one automatically.
7. If applicable, the admin sets the librarian's access permissions, specifying which sections or functionalities of the system the librarian can access.
8. The admin reviews the entered information to ensure accuracy.
9. The admin submits the registration form by clicking the "Register" or "Add Librarian" button.
10. The system validates the entered information, checks for any conflicting usernames or duplicate entries, and confirms the successful registration.
11. The system displays a confirmation message or provides a unique identification number for the newly registered librarian.

**User Scenario 4 - Administrator Removing a Book from the Library**

1. The administrator logs into the library management system.
2. They navigate to the "Manage Books" or "Book Inventory" section.
3. The administrator searches for the specific book they want to remove by title, author, ISBN code, or other relevant criteria.
4. The system displays a list of matching results.
5. The administrator selects the book they want to remove from the list.
6. They click on the "Remove" or "Delete" button associated with the selected book.
7. The system prompts a confirmation message to ensure the administrator's intention to remove the book.
8. The administrator confirms the removal, and the book is permanently deleted from the library's database.
9. The book is no longer available for students to search, reserve, or borrow.

**User Scenario 5 - Administrator Viewing Reviews Posted by Users on Books**

1. The administrator logs into the library management system.
2. They navigate to the "Book Reviews" or "User Feedback" section.
3. The administrator selects a specific book they want to view reviews for.
4. The system displays a list of reviews posted by users for that book.
5. The administrator can read the reviews, including the content and rating provided by the users.
6. They may have the option to sort or filter the reviews based on criteria such as date, rating, or user information.
7. The administrator can gain insights from the user feedback to assess the book's quality, popularity, or areas of improvement.
8. They may take appropriate actions or respond to the reviews if necessary.

**User Scenario 6 - Administrator Modifying Book Data**

1. The administrator logs into the library management system.
2. They navigate to the "Manage Books" or "Book Inventory" section.
3. The administrator searches for the specific book they want to modify by title, author, ISBN code, or other relevant criteria.
4. The system displays a list of matching results.
5. The administrator selects the book they want to modify from the list.
6. They click on the "Edit" or "Modify" button associated with the selected book.
7. The system opens the book's details page, allowing the administrator to make changes.
8. The administrator updates the necessary fields such as ISBN code, title, author, publisher, number of pages, etc.
9. After making the modifications, they save the changes by clicking the "Save" or "Update" button.
10. The book's data is updated in the library's database, reflecting the modifications made by the administrator.

**User Scenario 7 - Administrator Handling User Messages/Reports**

1. The administrator logs into the library management system.
2. They navigate to the "User Messages" or "Reports" section.
3. The system displays a list of user messages or reports received by the library.
4. The administrator can sort or filter the messages/reports based on criteria such as date, user, or message type.
5. They select a specific message/report from the list to handle.
6. The administrator reads the content of the message or report, understanding the user's concern or issue.
7. Depending on the nature of the message/report, the administrator takes appropriate action, such as responding to the user, investigating the reported issue, or escalating it to the relevant department.
8. They may compose a reply or initiate further communication with the user if necessary.
9. After handling the message/report, the administrator may mark it as resolved or take any other required actions to ensure proper follow-up and closure.

**User Scenario 8 - Administrator Checking Book Availability**

1. The administrator logs into the library management system.
2. They navigate to the "Book Availability" or "Book Inventory" section.
3. The administrator may have the option to search for a specific book by title, author, ISBN code, or other relevant criteria.
4. They enter the search criteria and initiate the search.
5. The system displays the search results, showing the availability status of the books.
6. The administrator can see whether the book is currently available for borrowing or if it is already reserved by a student.
7. They can view additional details such as the book's location in the library or the expected return date if it is currently checked out by a student.
8. The administrator can use this information to assist students in finding available books, managing reservations, or planning library operations efficiently.

LIBRARIAN SCENARIOS

**User Scenario 1 - Librarian Searching for a Book**

1. The librarian accesses the library management system or website by logging in with their credentials.
2. The librarian is directed to the homepage, where a list of books is displayed.
3. The librarian notices a search bar on the homepage and enters specific keywords such as a book title, author, ISBN code, or subject to search for a particular book.
4. The library management system presents the librarian with relevant search results based on the entered keywords.
5. The librarian clicks on a book from the search results or the displayed book list.
6. The system displays detailed information about the selected book, including its title, author, genre, publisher, and other relevant details.
7. The librarian reviews the book's data, including availability status, number of copies, and location within the library.
8. If necessary, the librarian navigates through different tabs or sections to access additional information related to the book, such as its borrowing history or related user reviews.
9. The librarian concludes the search and either proceeds with other tasks or makes a note of the book's details for future reference.

**User Scenario 2 - Librarian Handling User Messages/Reports**

1. The librarian logs into the library management system using their credentials.
2. Upon logging in, the librarian is presented with a dashboard or menu that includes various options and functionalities.
3. The librarian notices a section or tab specifically dedicated to user messages or reports.
4. The librarian clicks on the user messages/reports section to access the list of messages or reports received from library users.
5. The system displays a list of user messages/reports, along with relevant information such as the sender's name, date, and subject of the message/report.
6. The librarian scans through the list to identify the messages/reports that require attention or further action.
7. For each message/report, the librarian has two options: handle it themselves or delegate it to the administrator.
8. If the librarian decides to handle the message/report personally, they click on the respective message/report to open and review its contents.
9. The librarian reads the message/report carefully, understanding the user's concern or inquiry.
10. Based on the nature of the message/report, the librarian takes appropriate actions to resolve the issue or provide the necessary information to the user.
11. The librarian may need to communicate with the user through the library management system or other communication channels to gather additional details or clarify any ambiguities.
12. After resolving the message/report, the librarian updates the status of the message/report in the system to indicate that it has been addressed or resolved.
13. If the librarian decides to delegate the message/report to the administrator, they select the message/report and choose the appropriate option to transfer the responsibility.
14. The librarian may provide any necessary context or notes to the administrator regarding the message/report.
15. The librarian ensures that the system accurately records the transfer of responsibility for the message/report.
16. If required, the librarian communicates with the administrator to provide any additional information or context about the message/report.
17. The librarian follows up with the administrator to ensure that the message/report is being handled promptly and appropriately.

**User Scenario 3 - Librarian: Confirming Book Return**

1. The Librarian receives a book from a library patron at the book return counter.
2. The Librarian visually inspects the book for any damages.
3. The Librarian checks the due date to ensure it is not overdue.
4. The Librarian will mark the book as "returned".
5. The Librarian thanks the student for returning the book and confirms that the return is successfully processed.
6. The Librarian places the book on a designated shelf or cart for returned books.
7. The Librarian updates the book's status in the library's inventory management system, indicating that it is available for circulation again.

**User Scenario 4 - Librarian Creating and Confirming Book Order**

1. The librarian logs into the library management system.
2. They navigate to the "Orders" or "Book Requests" section.
3. The system displays a list of pending book orders or requests made by students.
4. The librarian reviews the orders and identifies the book that needs to be processed.
5. They click on the specific order to view the details of the requested book.
6. The librarian verifies the availability of the book in the library's inventory.
7. If the book is available, they proceed to confirm the order by clicking the "Confirm" or "Process" button.
8. The librarian updates the order status as "Confirmed" in the system.
9. They may assign a due date or notify the student about the availability and pickup details.
10. The librarian ensures that the book is prepared for the student's pickup and updates the necessary records accordingly.

**User Scenario 5 - Librarian Handling Late Book Return**

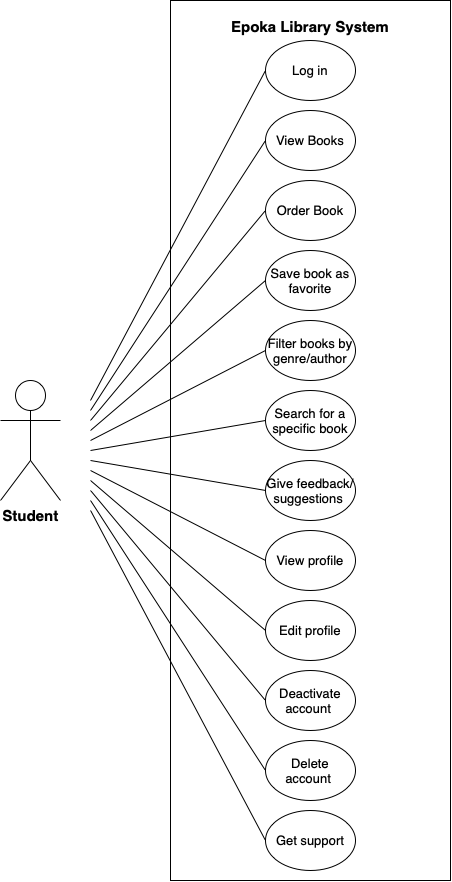
1. The librarian logs into the library management system.
2. They navigate to the "Returns" or "Book Returns" section.
3. The system displays a list of recently returned books.
4. The librarian identifies the book(s) that were not returned within the specified deadline.
5. They click on the specific book to view the borrower's information and return details.
6. The librarian checks the student's profile to gather their contact information.
7. Based on the library's policies, the librarian decides on the appropriate action to take.
8. If the librarian chooses to provide extra time, they update the return date in the system and make a note of the extension.
9. If the librarian decides to report the late submission, they notify the student via email, phone call, or any other preferred communication method.
10. The librarian updates the necessary records to reflect the action taken and ensures proper follow-up.

**User Scenario 6 - Librarian Viewing Book Reviews**

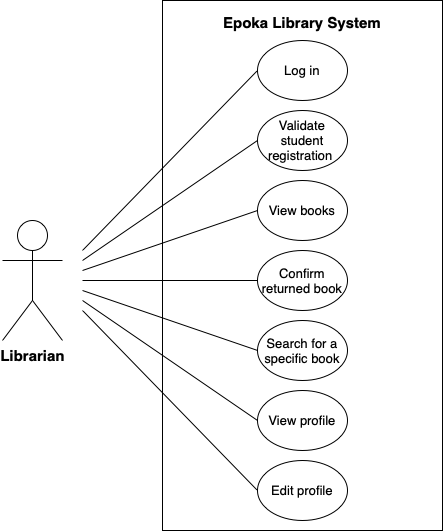
1. The librarian logs into the library management system.
2. They navigate to the "Book Reviews" or "User Reviews" section.
3. The system displays a list of book reviews posted by library users.
4. The librarian can sort or filter the reviews based on criteria such as book title, rating, or user feedback.
5. They select a specific book review to view the details.
6. The librarian reads the review, paying attention to the user's comments, suggestions, or feedback.
7. Based on the review content, the librarian may take actions such as noting down the feedback, responding to the user if necessary, or considering the feedback for future book selection or improvement purposes.
8. The librarian may also share the review with other library staff or administrators for further discussion or decision-making.
9. They ensure that the review records are appropriately maintained for reference and analysis.

***4.2 Use cases***

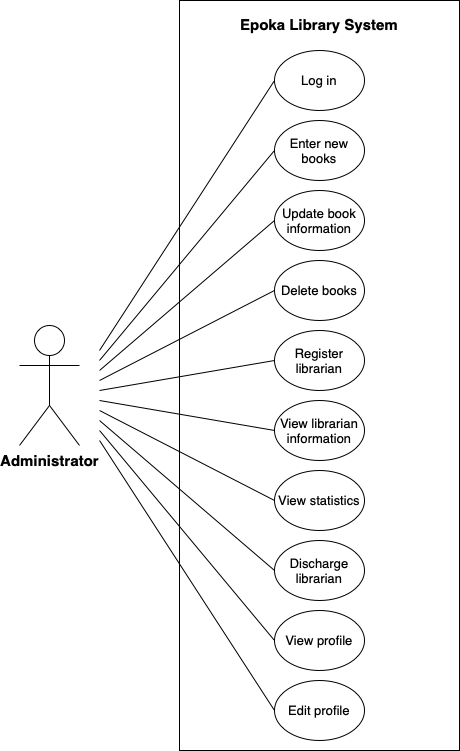
Use case 1: Student



Use case 2: Librarian



Use case 3: Administrator



Use case 4: General use case



***4.3 Use case extended***

|  |  |
| --- | --- |
| **Use Case (UC\_1.1):** | Student log in |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The student needs to log in to the system in order to browse the books and benefit from the offered features, such as finding specific library items, ordering items, saving them as favorites, and sending any feedback. |
| Minimum Guarantees: | The student provides the needed data but can not log in. |
| Success Guarantees: | The student fills out the login form and can successfully log in. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | To provide easy access to the library, in order to help students save time by finding the items they need through a process of just a few clicks. Keeping track of orders in a more accessible way. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.2):** | Student order item |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | One of the main features provided to the students is the easy access to the entire catalog of library items, from which they can place orders without having to search for them in-person. The orders are then picked up by the students within the specified time frame up to two days. |
| Minimum Guarantees: | The student tries to order the library item but the latter has run out of stock. |
| Success Guarantees: | The student successfully places the order, which automatically saves it for the student who shall pick it up within the specified time frame. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | The students can order books in real time, from anywhere they are. The interest of the librarian and the administrator is to keep the orders more organized and easier to keep a track of. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.3):** | Student save library item |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | This feature gives the opportunity to all the student users to personalize their accounts more, by saving the library items so they can view them at a later time, or when the are out of stock at the moment. |
| Success Guarantees: | The student presses the button and the library item is saved successfully. From there it can be accessed in the saved items section. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | To help students have more individual access to the Web Application, in order to make their experience better within the system, as well as helping them keep track of books they think they might want to view later for some reason or another. |
| Precondition: | The student must have a stable connection to the Internet. He/she must be logged in successfully. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.4):** | Student search library item |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The search feature is one of the main reasons why students would prefer to utilize the electronic library system instead of physically searching for a library item in the physical library, which in most cases results to be very time consuming. The students can find whether an item they want is available or not, as well as finding the necessary information about it, by just a few clicks. |
| Minimum Guarantees: | The students does not find the item he/she is looking for, |
| Success Guarantees: | The student is able to find the item he/she searched for, which would also provide them with the necessary information about the specific item. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | To reduce the time it takes to find whether items are or are not part of the library. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.5):** | Student send feedback |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The student can send feedback to the librarian, who can pass it on to the administrator, regarding any suggestions they may have about the available item, or requests about items they would like to be available. |
| Minimum Guarantees: | The student tries to order the library item but the latter has run out of stock. |
| Success Guarantees: | The student successfully sends the feedback to the librarian. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | The students can help improve the system by making suggestions, offering to bring in library items, or requesting specific items. These can help other students have a more enjoyable experience within the application. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app and send the feedback. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.6):** | Student view profile |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The student may need to view his/her profile to access the information they have provided and fix any inconsistencies. |
| Success Guarantees: | The student can get access to his/her profile and all the data he/she has provided. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | The students can access their information easily and keep track of the data they have provided. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.7):** | Student edit profile |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The student may detect inconsistencies in the information he/she has previously provided. |
| Minimum Guarantees: | The student is not able to change some specific information, such as the email address. |
| Success Guarantees: | The student is able to modify the certain data he/she chose to edit. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | The information about the students stays up-to-date and it is easier to get rid of inconsistencies. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.8):** | Student delete profile |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The student wants to delete his/her profile and not be able to access the Library System anymore. |
| Minimum Guarantees: | The student provides the needed data but can not delete his profile. |
| Success Guarantees: | The student fills out the delete profile form and can successfully delete his/her profile. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | To only have users that are interested in using the app registered, students that have no interest have the option to delete their profile. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.9):** | Student deactivate profile |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The student wants to deactivate his/her profile and not be able to access the Library System anymore until he/she activates it again. |
| Minimum Guarantees: | The student provides the needed data but can not deactivate his/her profile. |
| Success Guarantees: | The student clicks on the deactivate profile form and can successfully deactivate his/her profile. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | To only have users that are interested in using the app registered, students that have no interest have the option to deactivate their profile. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.10):** | Student log out |
| Scope: | Epoka Library System |
| Level: | Student user level |
| Intention Context: | The student wants to log out of his/her profile and not be able to access the Library System anymore until he/she logs in again. |
| Minimum Guarantees: | The student presses the log-out button but can not log out of his/her profile. |
| Success Guarantees: | The student clicks on the log-out button and successfully logs out of his/her profile. |
| Primary Actor: | Student |
| Stakeholder’s Interest: | To give the users that have no particular interest in the meantime the option to log out of the app. |
| Precondition: | The student must have a stable internet connection, and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.11):** | Librarian receive feedback |
| Scope: | Epoka Library System |
| Level: | Librarian user level |
| Intention Context: | The librarian receives feedback from students, who can pass it on to the administrator, regarding any suggestions made by the students, such as the availability of an item. |
| Minimum Guarantees: | The librarian tries to fix it but there is a shortage of stock. |
| Success Guarantees: | The item restocks and the students can order it. |
| Primary Actor: | Librarian |
| Stakeholder’s Interest: | To give the librarian and administrator the opportunity to improve the system according to suggestions made by the students. |
| Precondition: | The librarian must have a stable internet connection and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.12):** | Administrator log in |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to log in to the system in order to do the things that he/she should do, such as add a library item, delete a library item, register librarians, etc. |
| Minimum Guarantees: | The administrator provides the needed data but can not log in. |
| Success Guarantees: | The administrator fills out the login form and can successfully log in. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To provide the administrator easy access to the library, in order to improve the system and do the necessary changes and updates that need to be done. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.13):** | Administrator add library item |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to log in to the system and add the library items that are out of stock, short of stock, or a completely new library item. |
| Minimum Guarantees: | The administrator adds the library item, but the change is not reflected in the system. |
| Success Guarantees: | The administrator adds the library item and the change is reflected in the system. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To provide the administrator easy access to the library, in order to improve the system and add the library items that should be added. This makes it easier and more convenient for students too. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.14):** | Administrator delete library item |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to log in to the system and delete the library items that are meant to be deleted. |
| Minimum Guarantees: | The administrator deletes the library item, but the change is not reflected in the system. |
| Success Guarantees: | The administrator deletes the library item and the change is reflected in the system. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To provide the administrator easy access to the library, in order to improve the system and delete the library items that should be deleted. This makes it easier and more convenient for students too. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which he/she can access the web app. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.15):** | Administrator registers librarian |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to register a new librarian in the system to provide them access and privileges to manage the library. |
| Minimum Guarantees: | The administrator fills out the registration form for the librarian, but the librarian is not successfully registered in the system. |
| Success Guarantees: | The administrator fills out the registration form for the librarian and the librarian is successfully registered in the system with appropriate access and privileges. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To efficiently manage the library staff and ensure that librarians have the necessary access and privileges to perform their duties effectively. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be logged in to the Epoka Library System as an administrator user. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.16):** | Administrator confirms librarian request |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to review and confirm a librarian request made by a user to become a librarian in the system. |
| Minimum Guarantees: | The administrator reviews the librarian request but does not confirm or reject it and the librarian status is not updated. |
| Success Guarantees: | The administrator reviews the librarian request and either confirms or rejects it, updating the librarian’s status accordingly. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To efficiently manage librarian requests and ensure that qualified individuals are granted librarian access in the system. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be logged in to the Epoka Library System as an administrator user. There must be at least one pending librarian request in the system. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.17):** | Administrator views statistics |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to access statistical information and reports related to the library system’s usage and performance. |
| Minimum Guarantees: | The administrator attempts to view statistics but encounters errors or is unable to access the required information. |
| Success Guarantees: | The administrator successfully views the desired statistics and reports related to the library system’s usage and performance. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To monitor and analyze the library system’s usage and make decisions for system improvements. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be logged in to the Epoka Library System as an administrator user. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.18):** | Administrator views profile |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs t access and view their own profile information and settings within the Epoka Library System. |
| Minimum Guarantees: | The administrator attempts to view their own profile but encounters errors or is unable to access the required information. |
| Success Guarantees: | The administrator successfully views their profile information and settings within the Epoka Library System. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To have access to and review their own profile information, including personal details and settings, within the library system. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be logged in to the Epoka Library System as an administrator user. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.19):** | Administrator edits profile |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to edit and update their profile information within the system. |
| Minimum Guarantees: | The administrator attempts to edit their profile but encounters errors or is unable to save the changes. |
| Success Guarantees: | The administrator successfully edits and saves their profile information. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To have the ability to update and maintain accurate profile information within the system. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be logged in to the Epoka Library System as an administrator user. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.20):** | Administrator deletes profile |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to delete their profile from the Epoka Library System. |
| Minimum Guarantees: | The administrator attempts to delete their profile but encounters errors or is unable to complete the deletion process. |
| Success Guarantees: | The administrator successfully deletes their profile from the system, resulting in the removal of their account and associated data. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To have the ability to permanently remove their profile and account from the library system. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be logged in to the Epoka Library System as an administrator user. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.21):** | Administrator receives feedback |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to receive feedback from the library users. |
| Minimum Guarantees: | The administrator may not receive feedback or encounter difficulties in accessing and reviewing the feedback. |
| Success Guarantees: | The administrator successfully receives and accesses feedback from the library users. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To gather feedback from library users to improve the Epoka Library System and address any concerns or suggestions. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be logged in to the Epoka Library System as an administrator user. |

|  |  |
| --- | --- |
| **Use Case (UC\_1.22):** | Administrator logs out |
| Scope: | Epoka Library System |
| Level: | Administrator user level |
| Intention Context: | The administrator needs to log out from the system to ensure the security and privacy of their account. |
| Minimum Guarantees: | The administrator attempts to log out but encounters errors or is unable to complete the logout process. |
| Success Guarantees: | The administrator successfully logs out from the system, ensuring the termination of their session and protecting their account. |
| Primary Actor: | Administrator |
| Stakeholder’s Interest: | To ensure security and privacy of the administrator’s account and prevent unauthorized access. |
| Precondition: | The administrator must have a stable internet connection and access to a web browser from which they can access the web app and the administrator must be currently logged in to the Epoka Library System as an administrator user. |

|  |  |
| --- | --- |
| Use Case (UC\_23) | Log In |
| Scope: | Electronic Library System. |
| Level: | Second user level (Librarian). |
| Intention Context | In order for the Librarian to have access to the application, he/she must log in to their existing account first. |
| Minimum Guarantees: | The librarian fills in his personal data, but cannot login. |
| Success Guarantees: | The librarian fills in his personal data and successfully passes the data validation, while signing in to his account. |
| Primary Actor: | Librarian. |
| Stakeholders Interest: | To have Librarians who are familiar with the application and are able to provide exceptional service. |
| Precondition: | The Librarian must download the app or access its web page. |

|  |  |
| --- | --- |
| Use Case (UC\_24) | Deactivate Account |
| Scope: | Electronic Library System. |
| Level: | Second user level (Librarian). |
| Intention Context | The librarian shall be able to send a request to the administrator to deactivate his/her account temporarily. |
| Minimum Guarantees: | The librarian sends the request to the administrator, but his request is not approved. |
| Success Guarantees: | The request of the librarian sent to the administrator is approved and his account is successfully deactivated temporarily. |
| Primary Actor: | Librarian. |
| Secondary Actor: | Administrator |
| Precondition: | The librarian must send the request to the administrator first and must wait for his approval. |

|  |  |
| --- | --- |
| Use Case (UC\_25) | Delete Account |
| Scope: | Electronic Library System. |
| Level: | Second user level (Librarian). |
| Intention Context | The librarian shall be able to send a request to the administrator to delete his/her account permanently. |
| Minimum Guarantees: | The librarian sends the request to the administrator, but his request is not approved. |
| Success Guarantees: | The request of the librarian sent to the administrator is approved and his account is successfully deleted permanently. |
| Primary Actor: | Librarian. |
| Secondary Actor: | Administrator |
| Precondition: | The librarian must send the request to the administrator first and must wait for his approval. |

|  |  |
| --- | --- |
| Use Case (UC\_26) | Edit Account |
| Scope: | Electronic Library System. |
| Level: | Second user level (Librarian). |
| Intention Context | The librarian shall be able to send a request to the administrator to edit his/her personal data of his/her account. |
| Minimum Guarantees: | The librarian sends the request to the administrator, but his request is not approved. |
| Success Guarantees: | The request of the librarian sent to the administrator is approved and his/her personal data is successfully edited. |
| Primary Actor: | Librarian. |
| Secondary Actor: | Administrator |
| Precondition: | The librarian must send the request to the administrator first and must wait for his approval. |

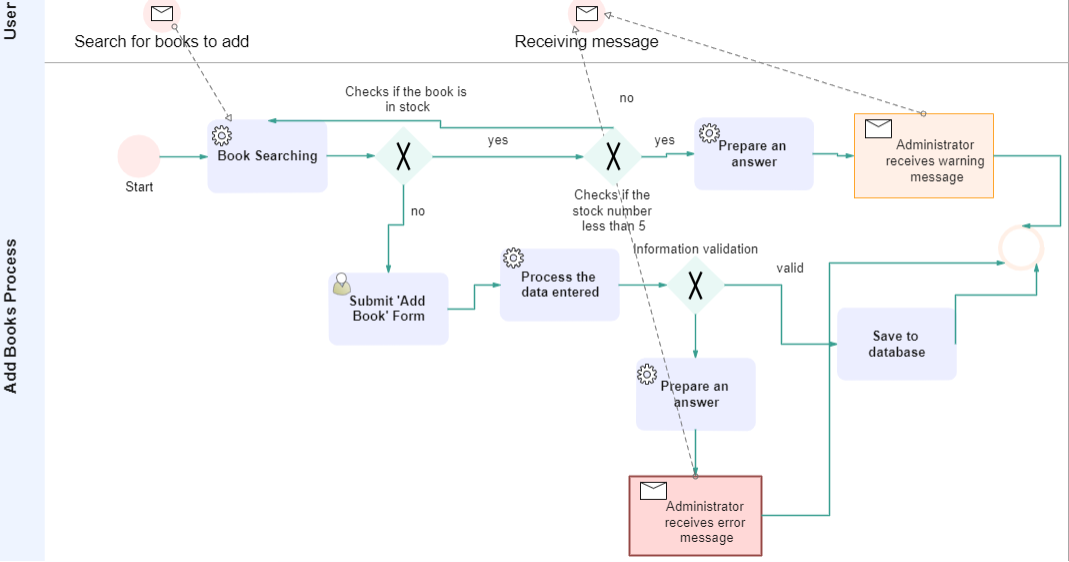
|  |  |
| --- | --- |
| Use Case (UC\_27) | View Order |
| Scope: | Electronic Library System. |
| Level: | Second user level (Librarian). |
| Intention Context | The librarian shall be able to see the orders of the students in order to keep track of the library items records, and the library items returning deadline. |
| Minimum Guarantees: | There may be orders who are not presented to the system as a result of web traffic. |
| Success Guarantees: | Detailed information is provided for every student order, including the library item’s borrowed information as well as the hour of the borrowing and the deadline for its return. |
| Primary Actor: | Librarian. |
| Secondary Actor: | User (Student) |
| Precondition: | The user (student) must have made an order. |

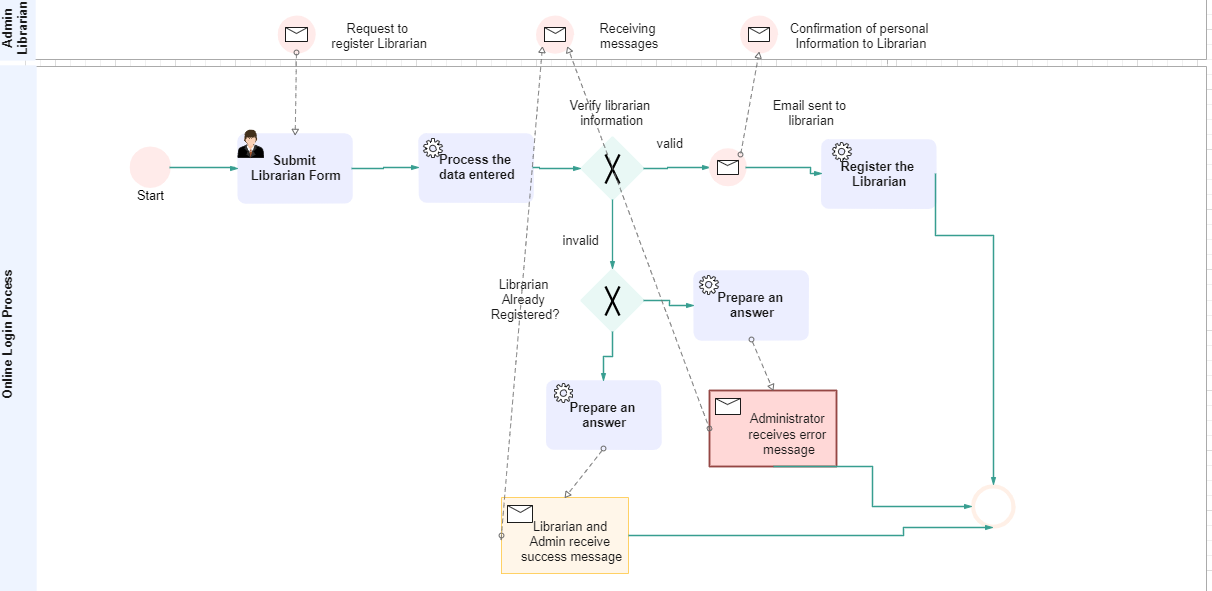
|  |  |
| --- | --- |
| Use Case (UC\_28) | Cancel Order |
| Scope: | Electronic Library System. |
| Level: | Second user level (Librarian). |
| Intention Context | The librarian shall be able to cancel the order if the person who has made that specific order doesn’t borrow it. |
| Minimum Guarantees: | The user borrows the book and the librarian does not have to cancel any order. |
| Success Guarantees: | The user does not come within 1-2 days to borrow the ordered book, so the librarian proceeds with the order cancellation. |
| Primary Actor: | Librarian. |
| Precondition: | The user does not come physically to borrow his/her order. |

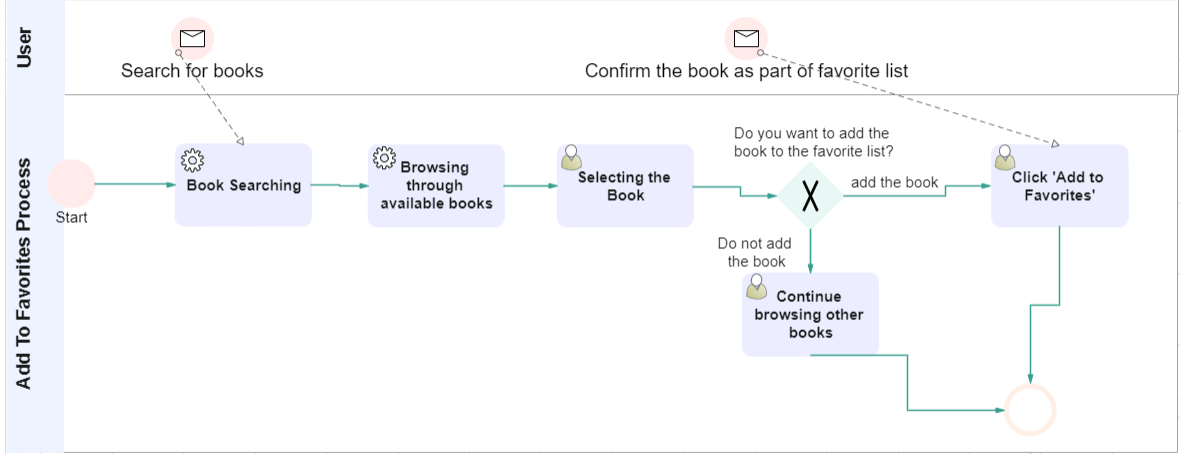
|  |  |
| --- | --- |
| Use Case (UC\_29) | Mark order as returned |
| Scope: | Electronic Library System. |
| Level: | Second user level (Librarian). |
| Intention Context | The librarian shall be able to mark a specific user order as returned when that user returns the borrowed library item within its deadline. |
| Minimum Guarantees: | The user does not return back the borrowed library item within its deadline. |
| Success Guarantees: | The user does return the library item within its deadline. |
| Primary Actor: | Librarian. |
| Precondition: | The user does not come physically to borrow his/her order. |

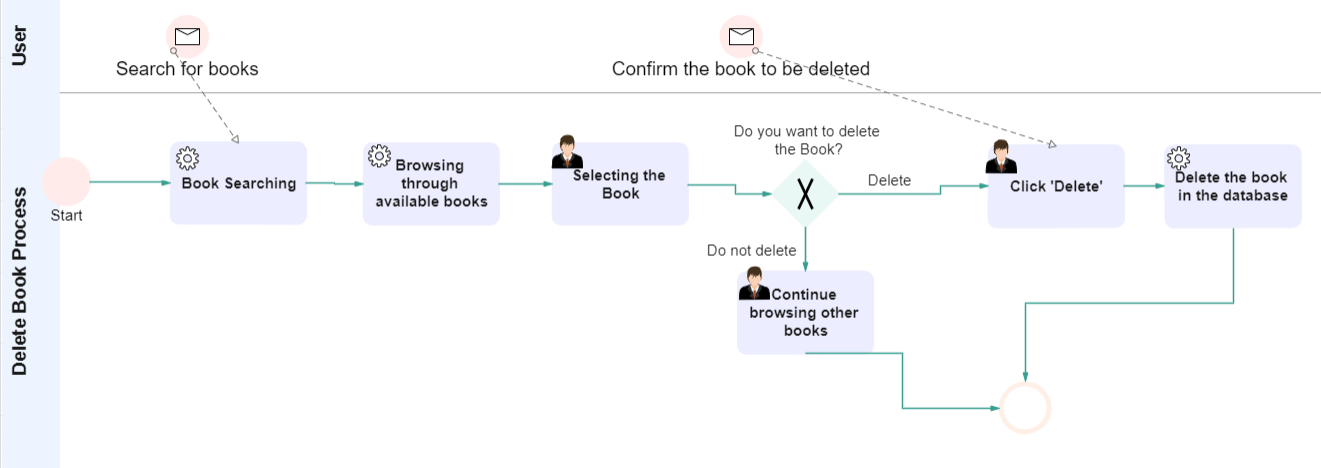
|  |  |
| --- | --- |
| Use Case (UC\_30) | Receive Notification |
| Scope: | Electronic Library System. |
| Level: | Third user level (Student). |
| Intention Context | The students shall be able to receive notifications (1 day before) regarding their deadline for returning the library item. |
| Minimum Guarantees: | The student does not receive any notification about their deadline. |
| Success Guarantees: | The user does return the library item within its deadline. |
| Precondition: | The user does not come physically to borrow his/her order. |

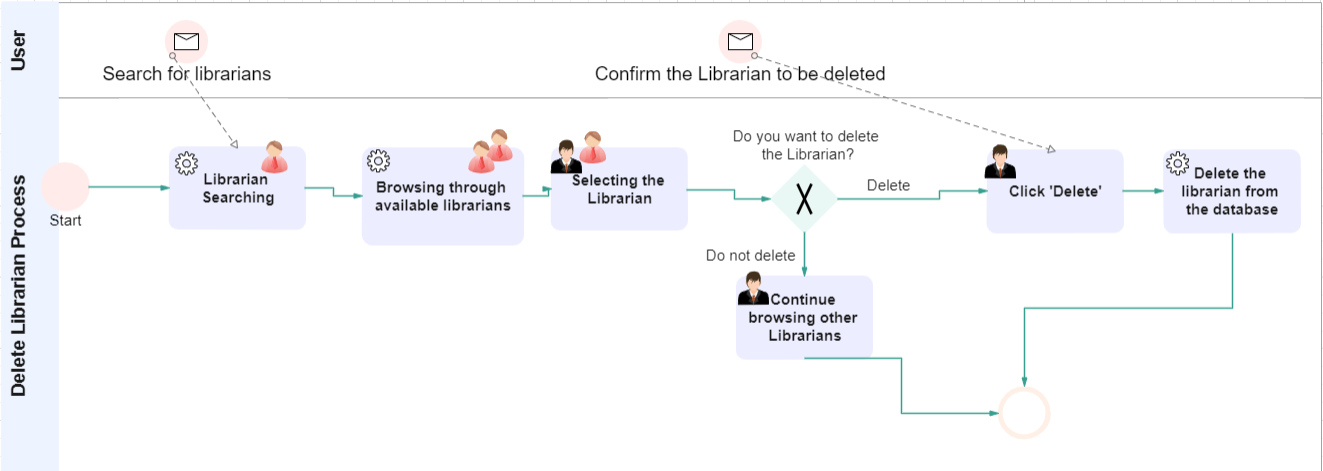
***4.4 BPMN***

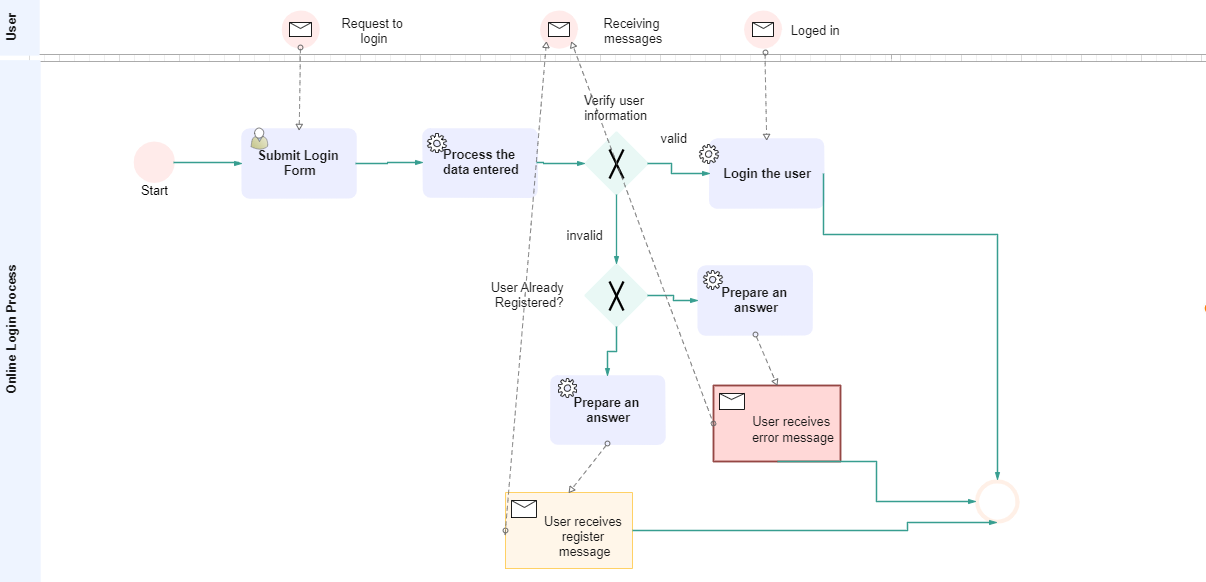


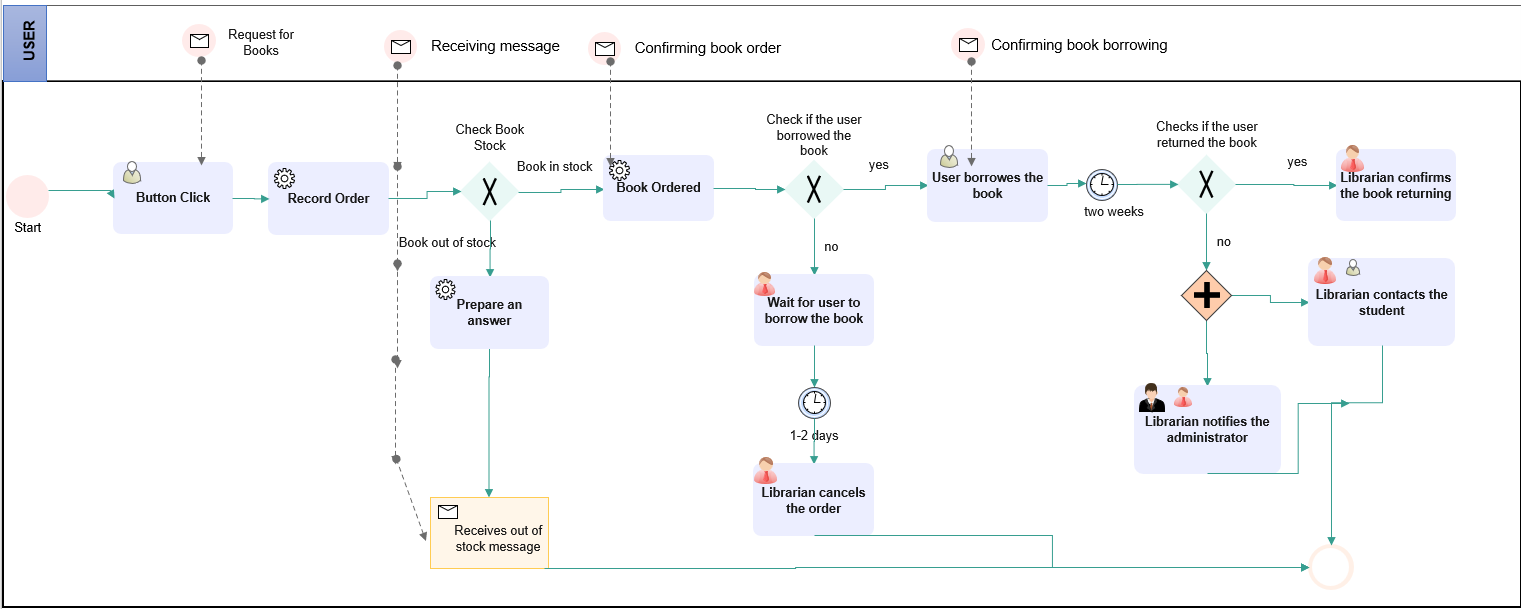


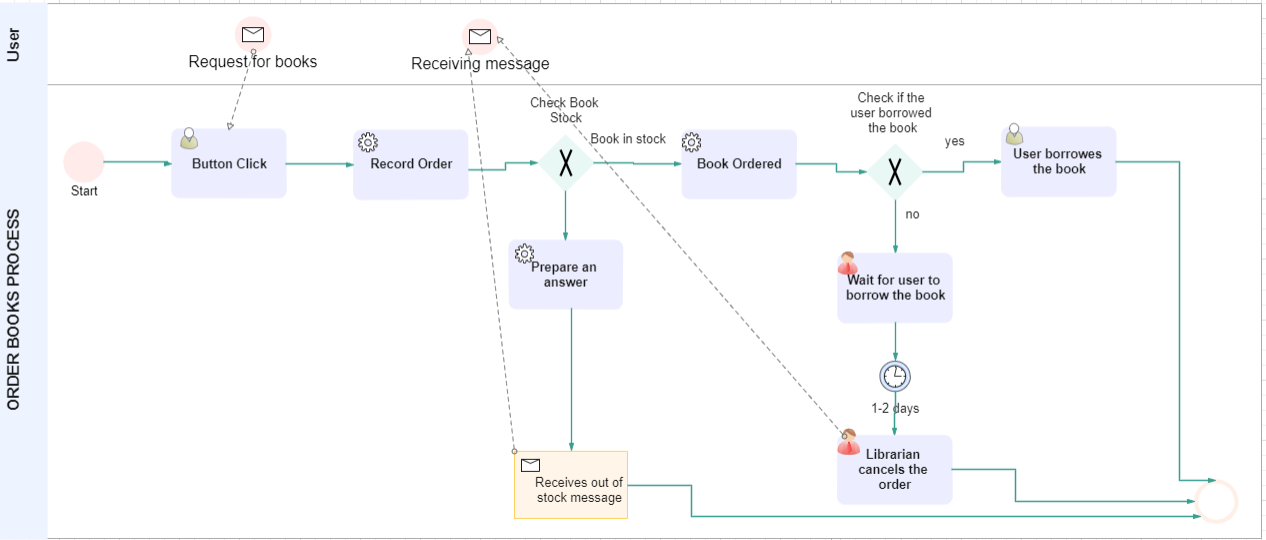


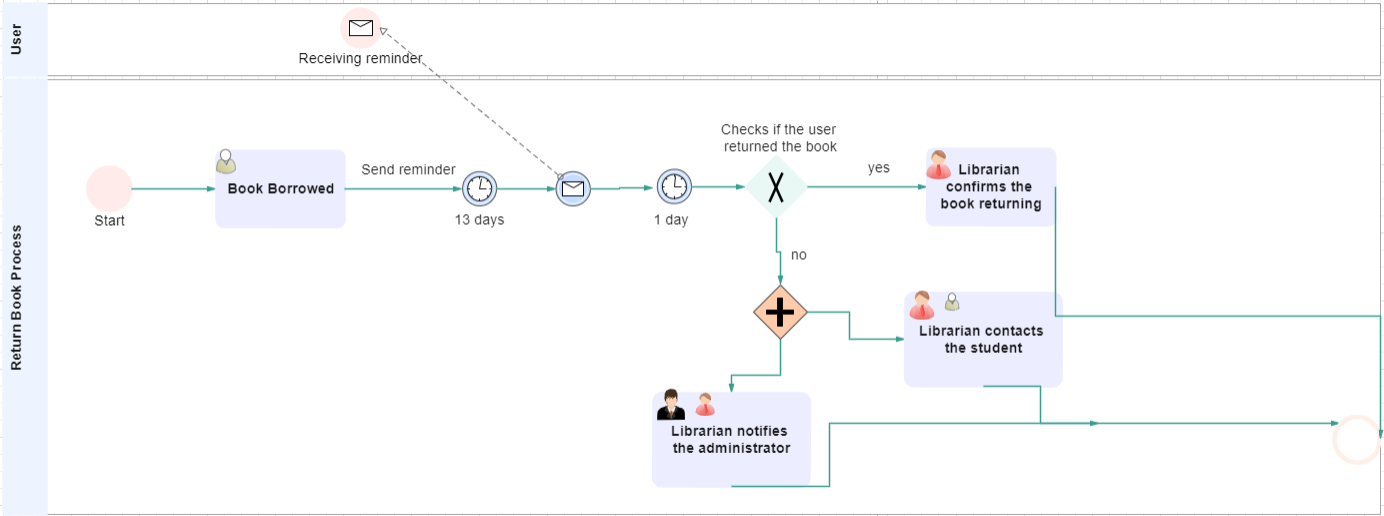


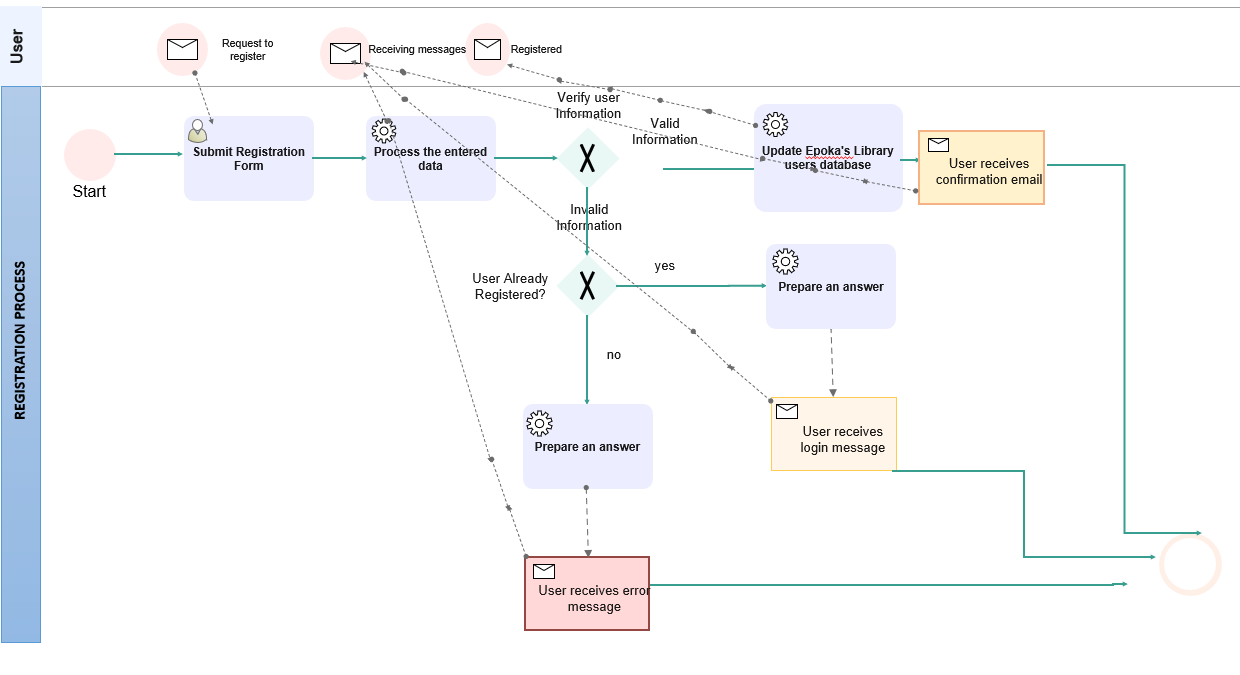




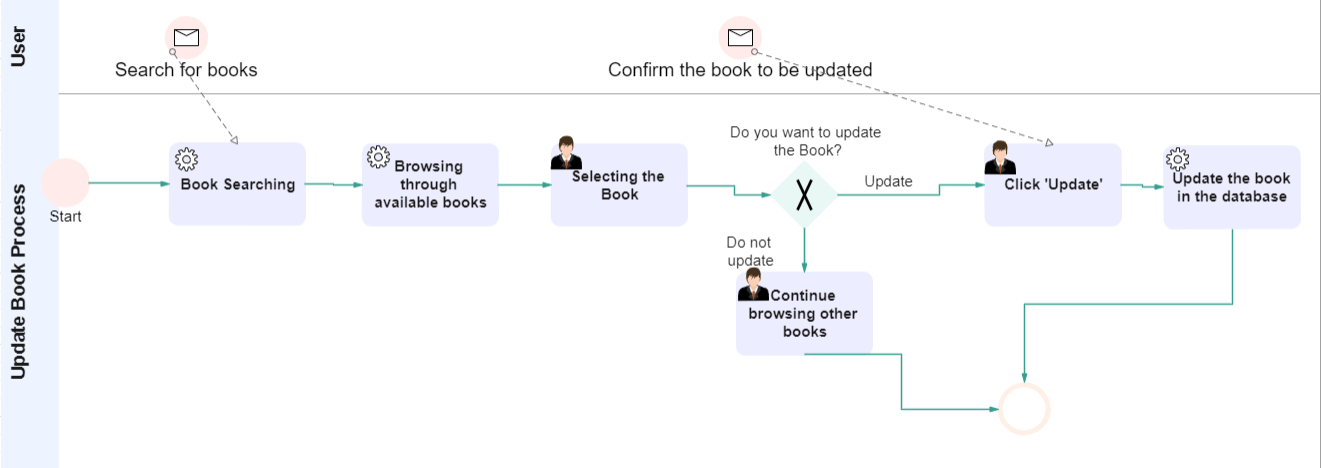






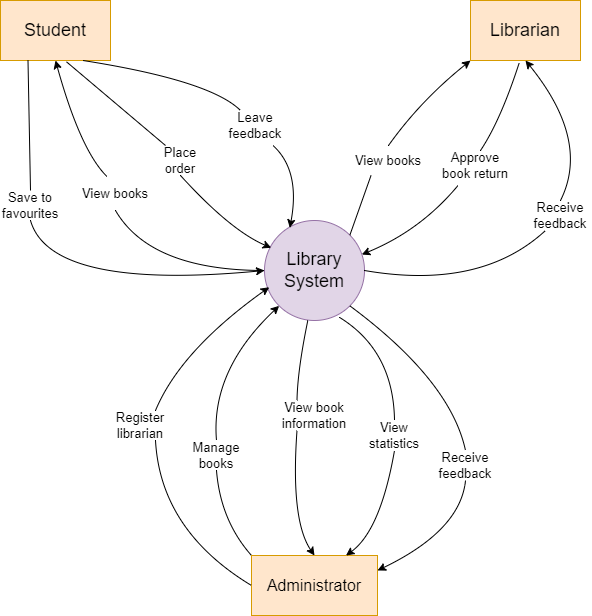




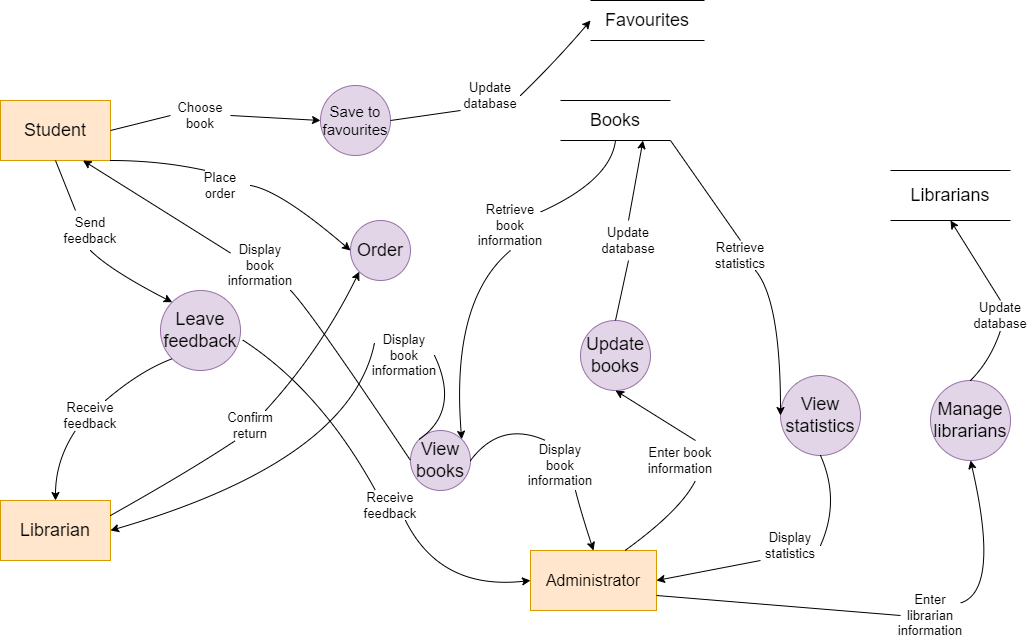


***4.5 Data Flow Diagrams***

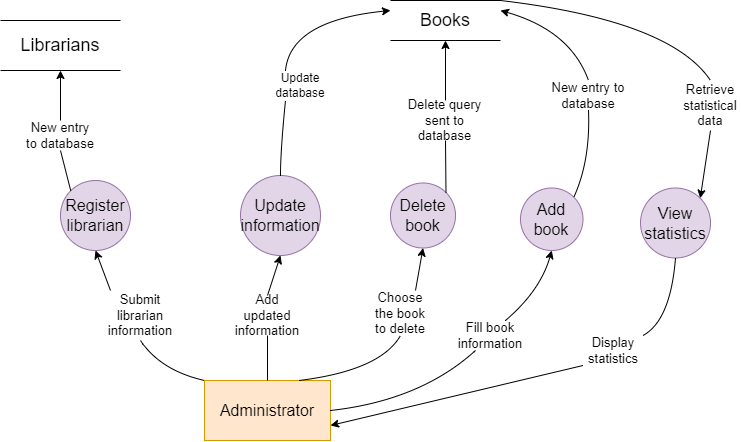
DFD Level 0



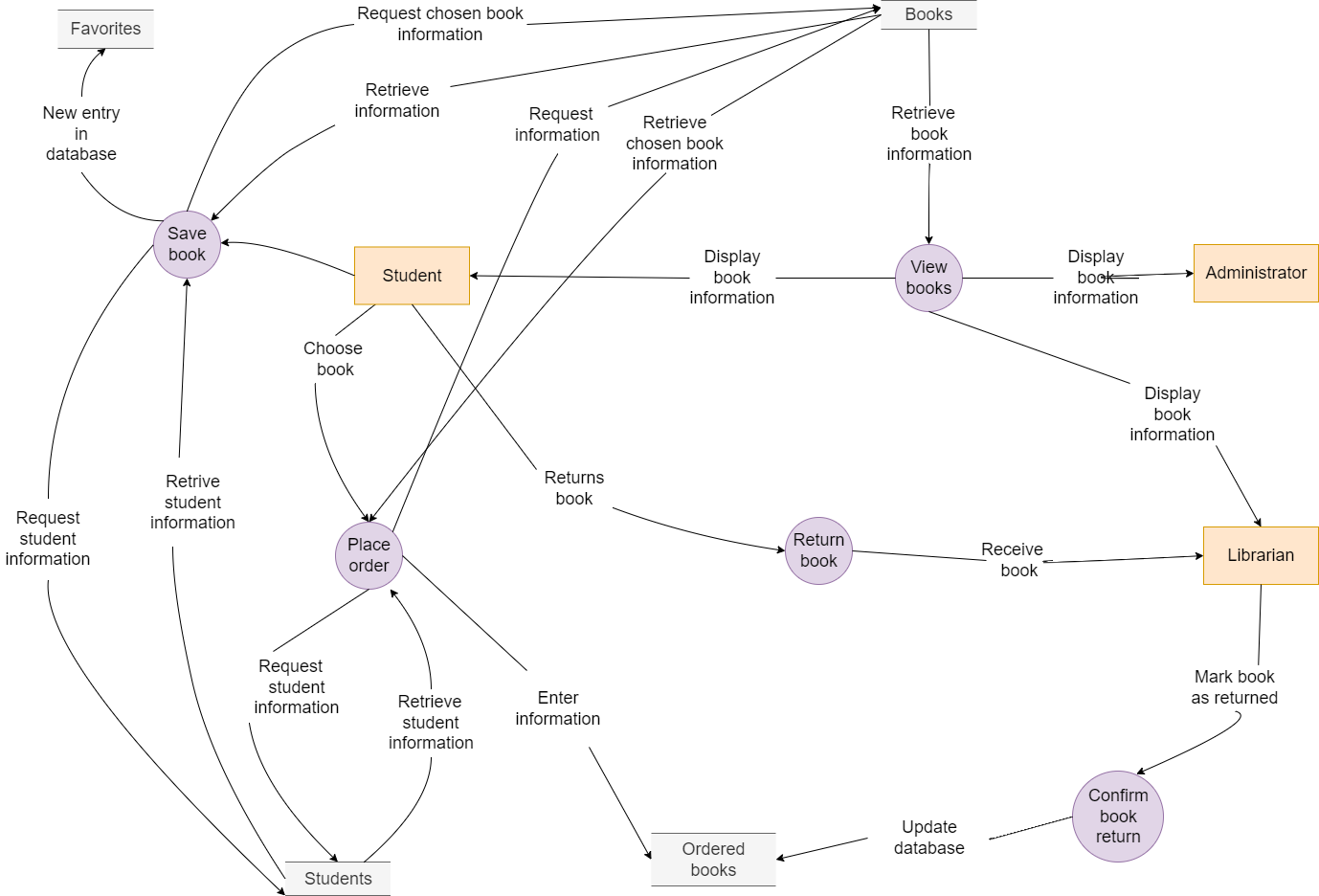
DFD Level 1



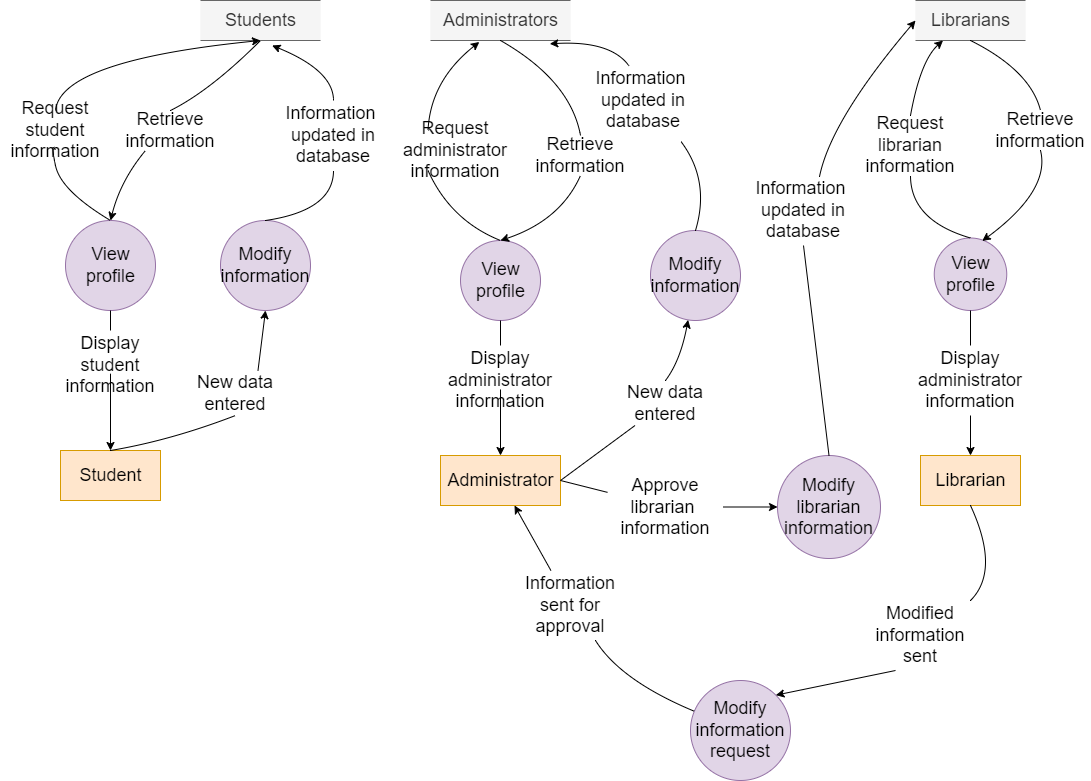
DFD Level 2 Diagram 1



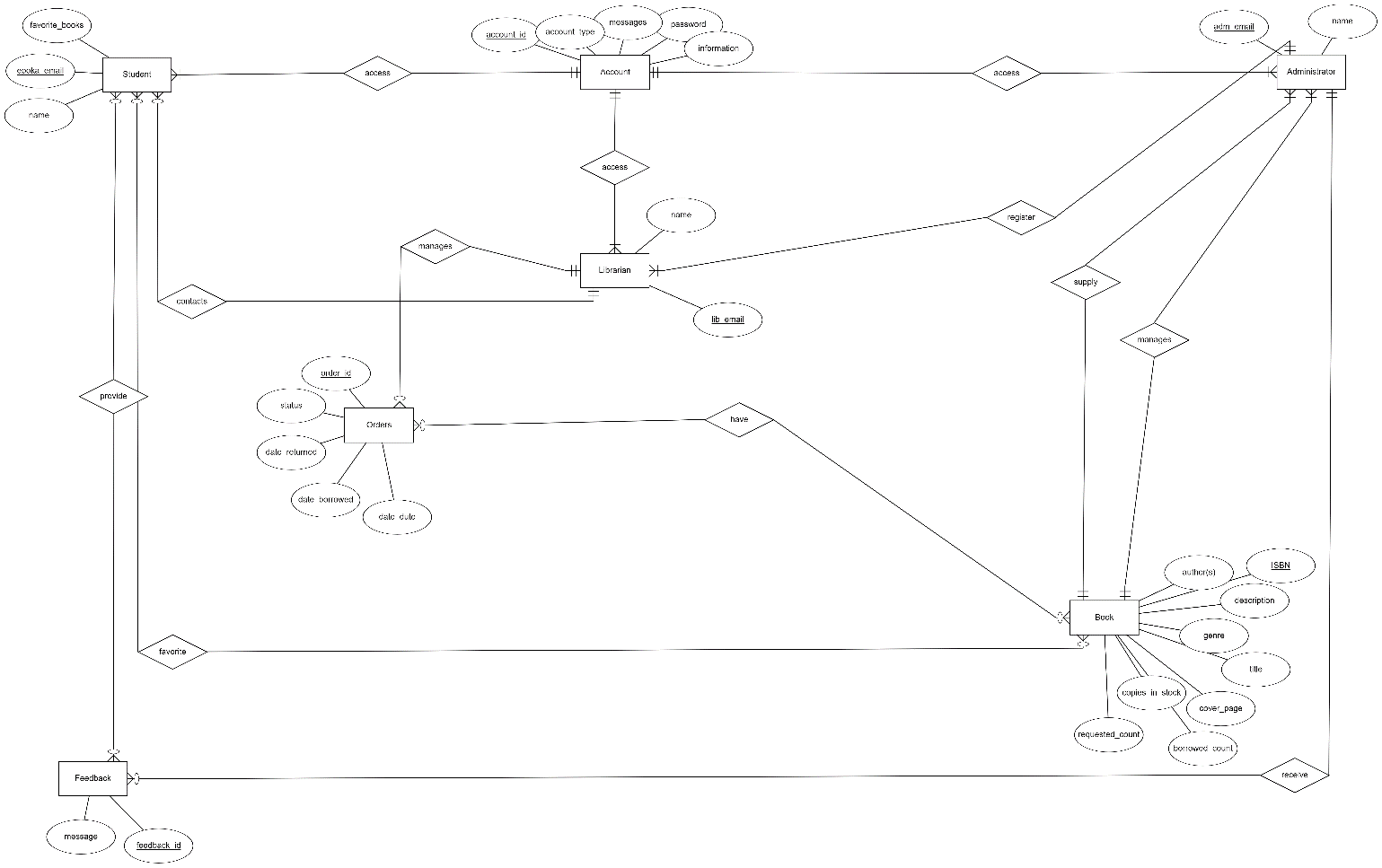
DFD Level 2 Diagram 2



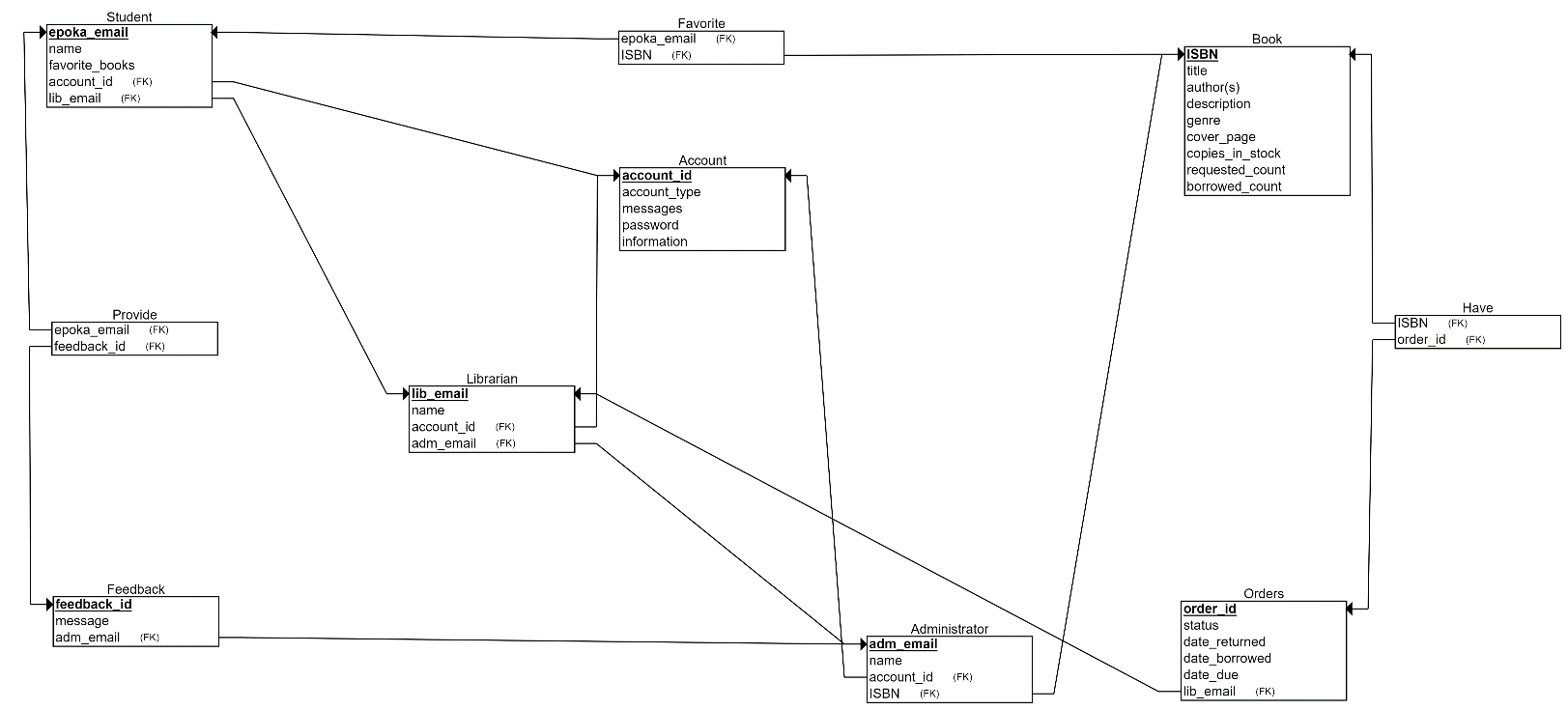
DFD Level 2 Diagram 3



***4.6 Entity relationship (ER) diagram***

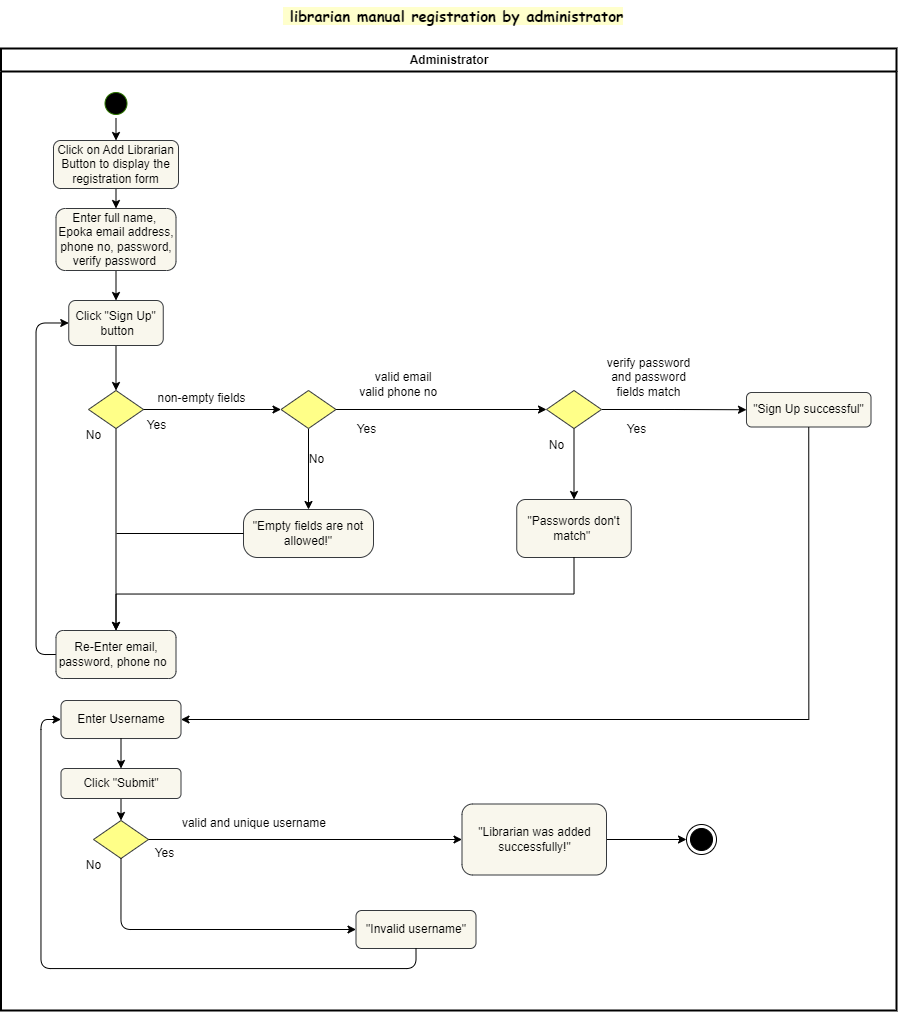


***4.7 Relational Schema (RS) Diagram***

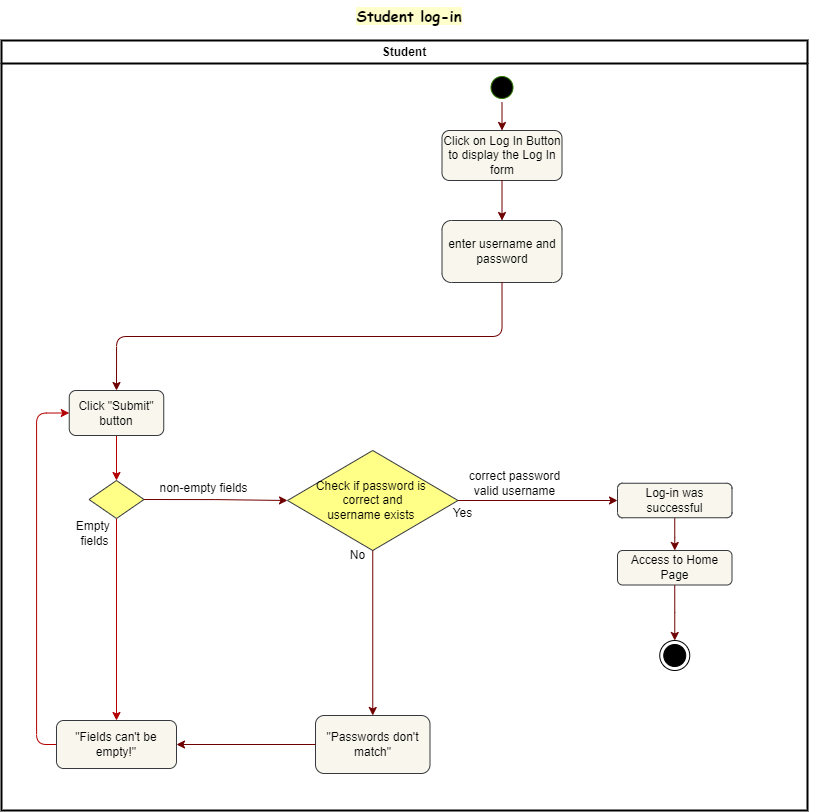


***4.8 Activity diagrams***

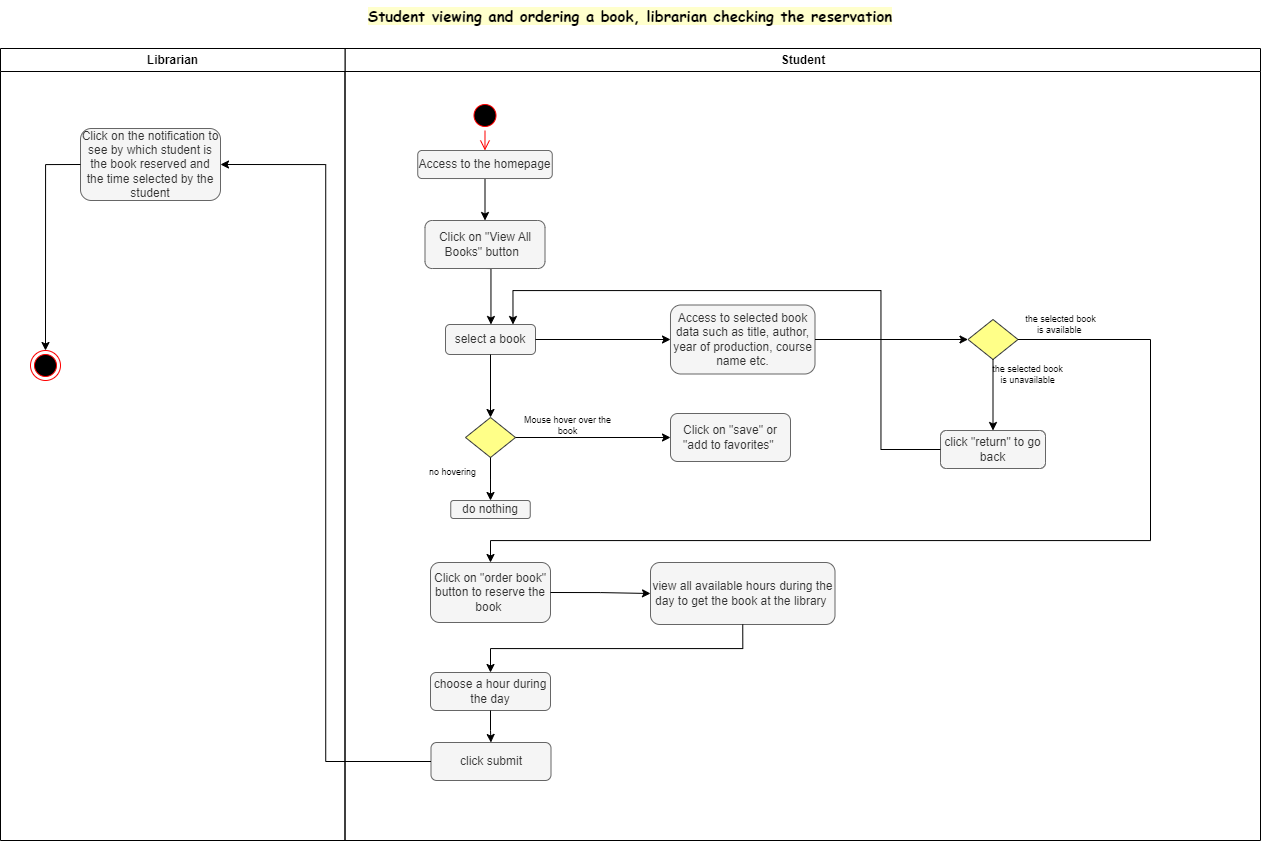
AC\_01 Librarian registration by administrator



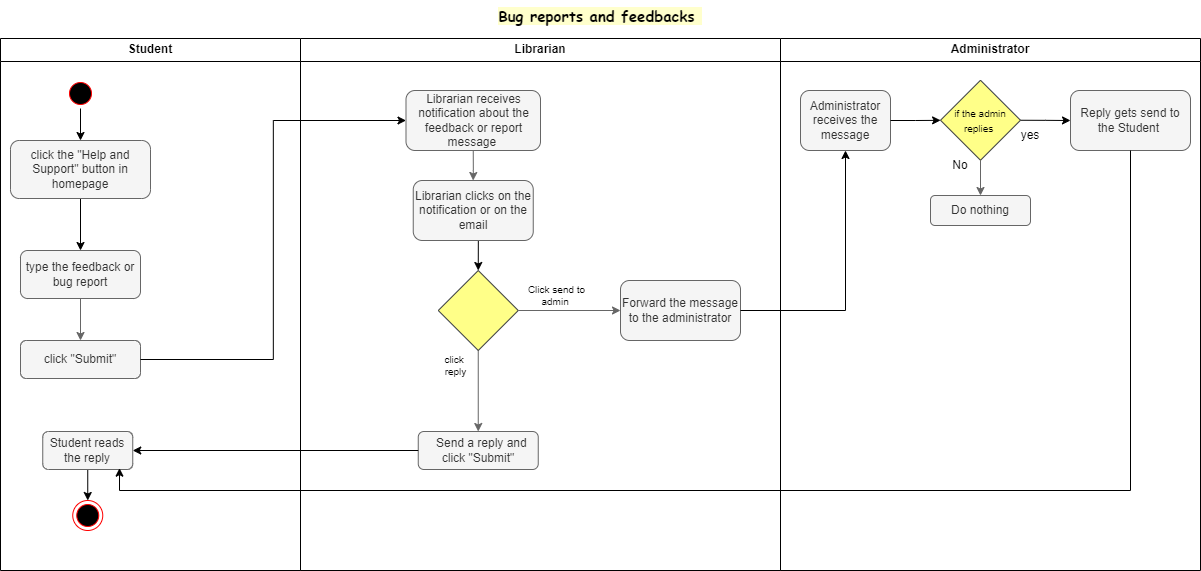
AC\_02 Log in for students



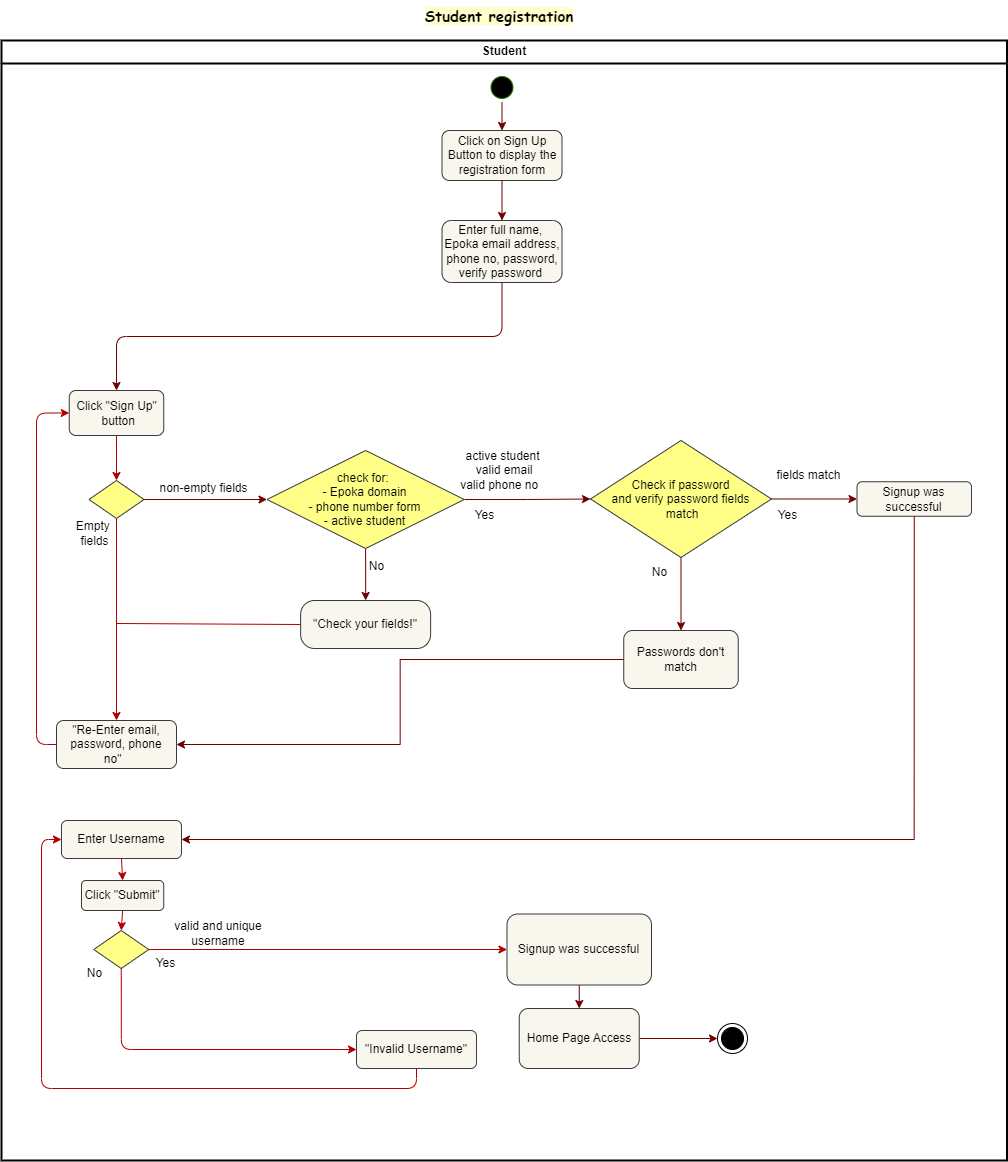
AC\_03 Item reservation

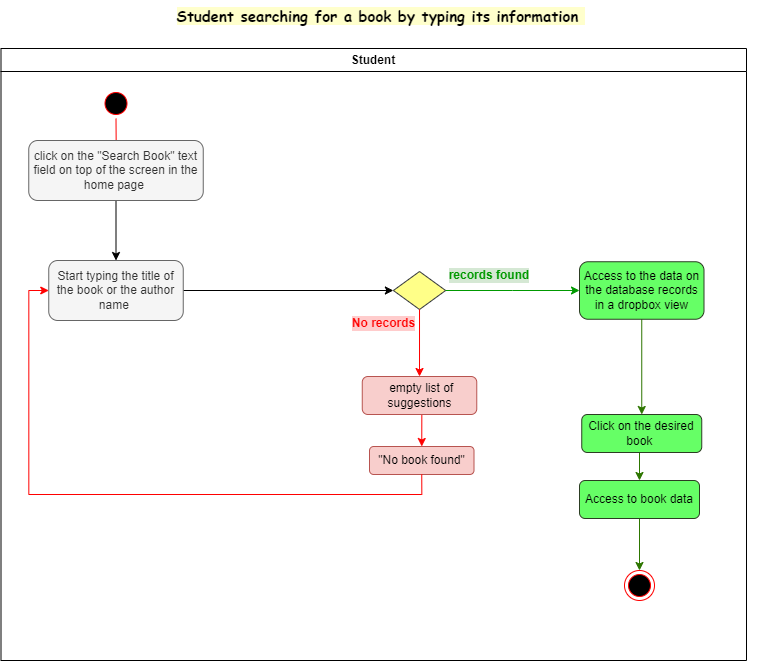


AC\_04 Feedback

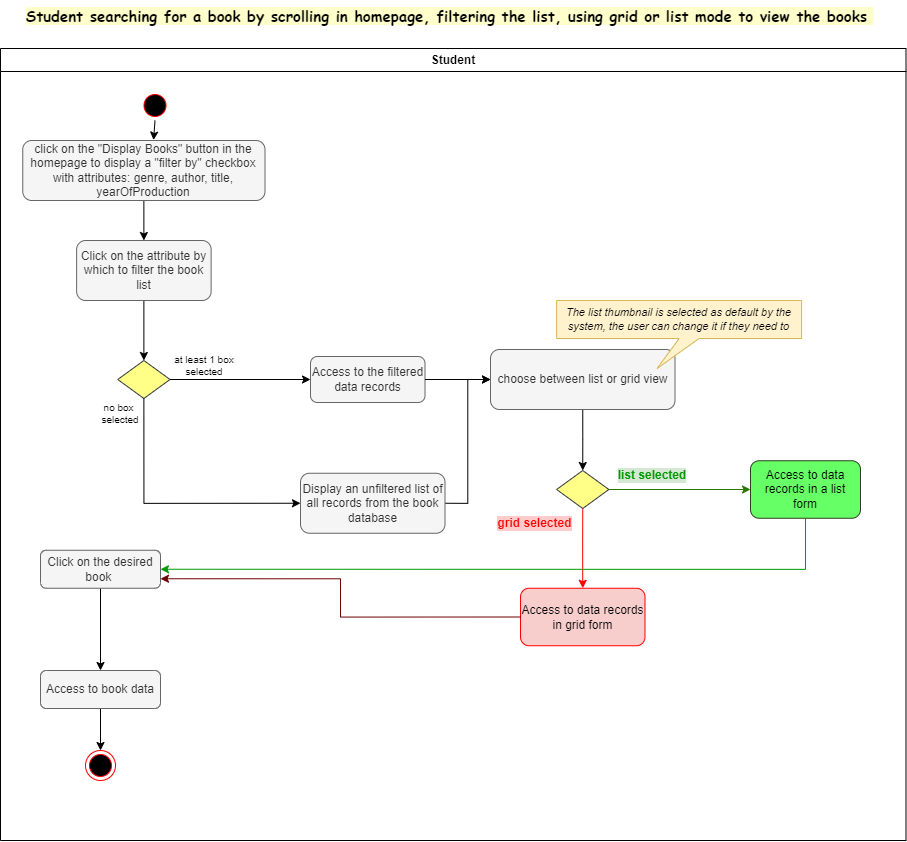


AC\_05 Student registration

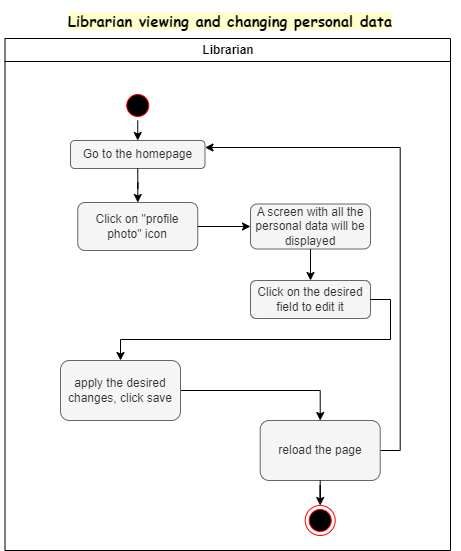


AC\_06 Search item

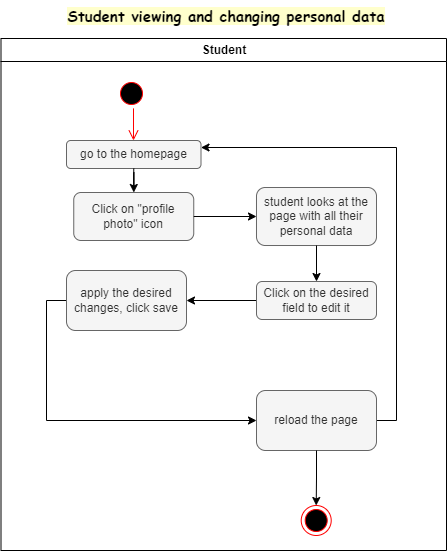
AC\_07 Student library item search



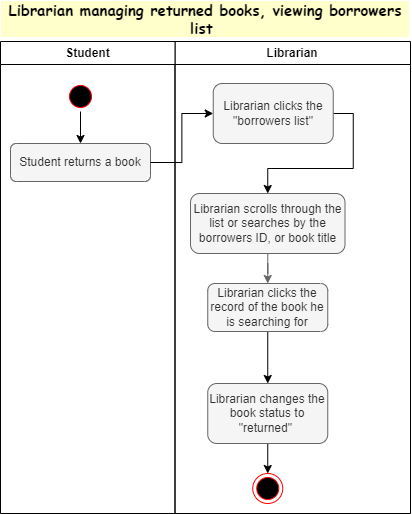
AC\_08 Edit personal information - librarian



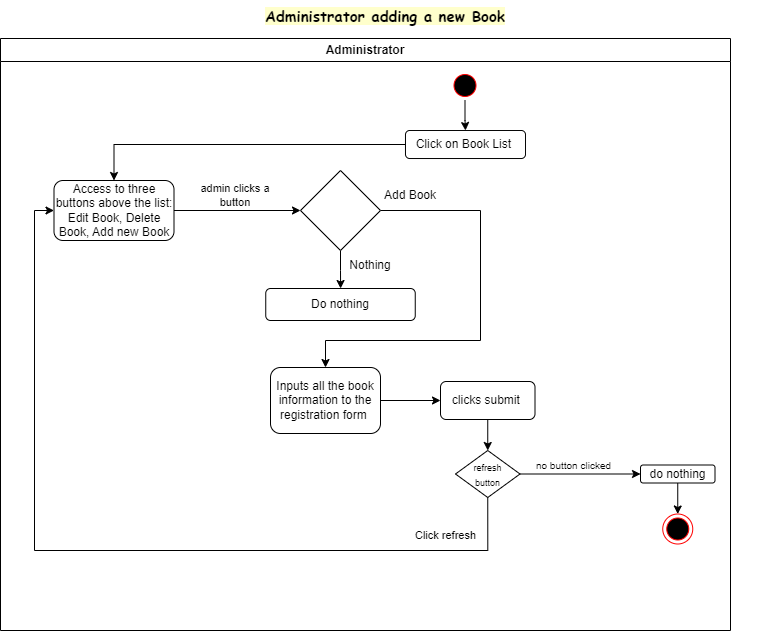
AC\_09 Edit student information



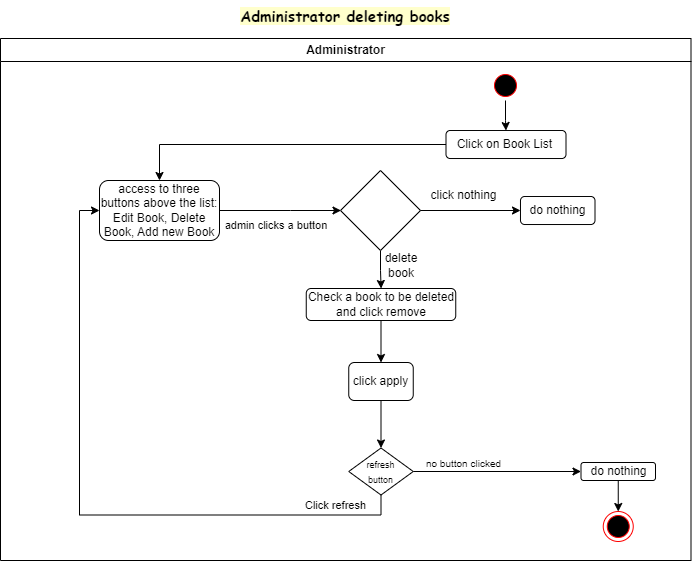
AC\_10 Returned Library Item



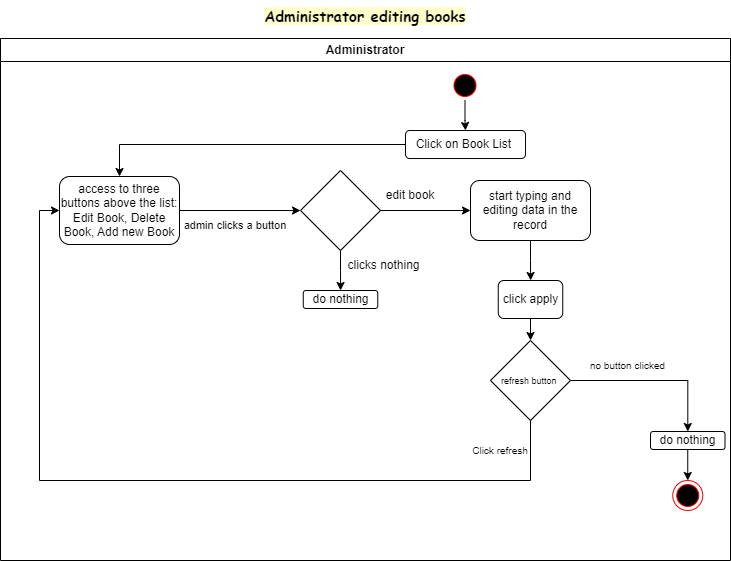
AC\_11 Adding a library item



AC\_12 Deleting library items



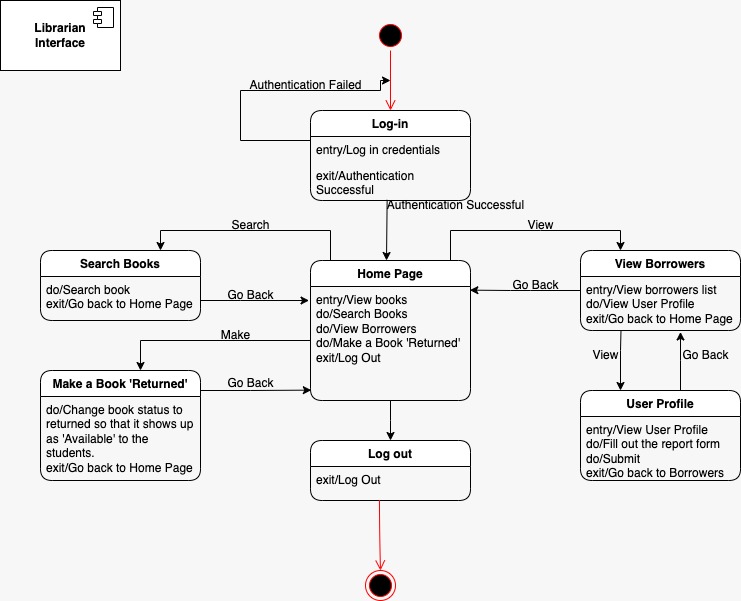
AC\_13 Editing library items

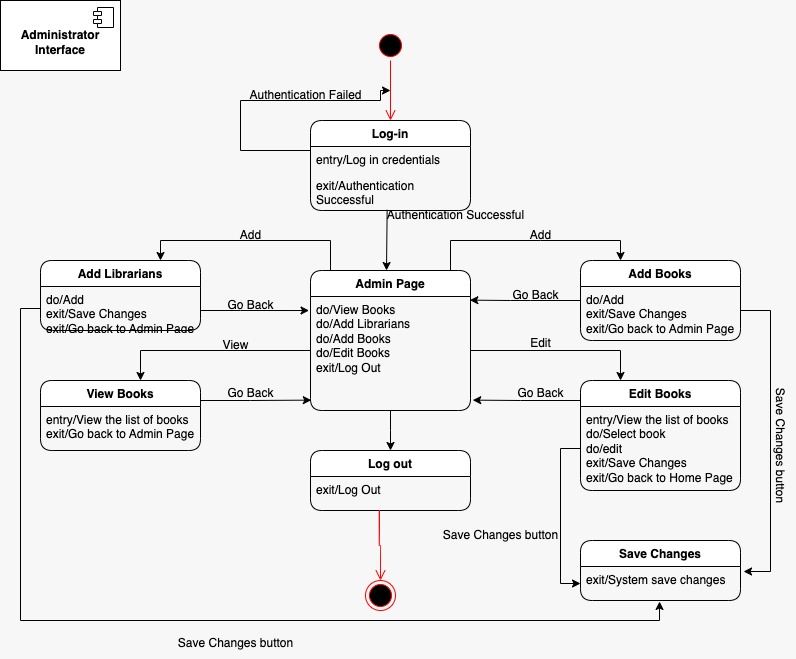


***4.9 State diagrams***

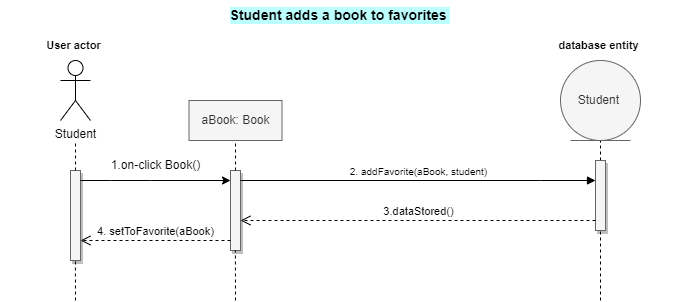


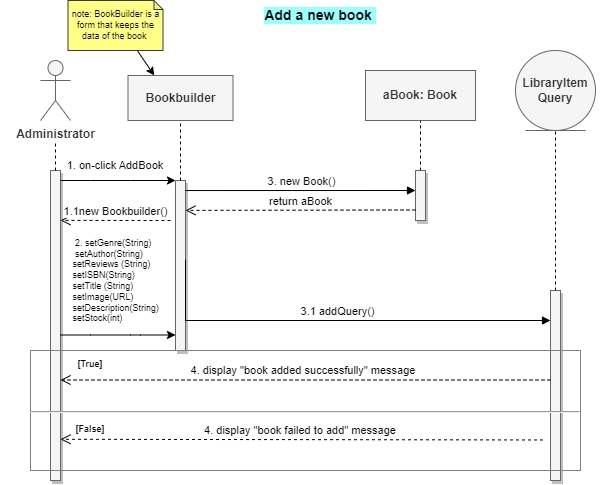


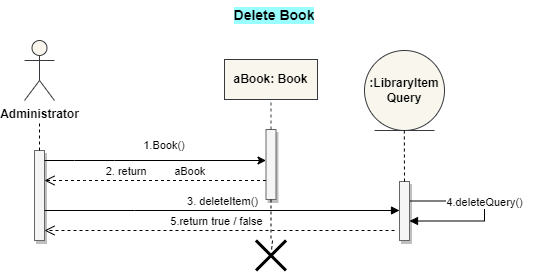


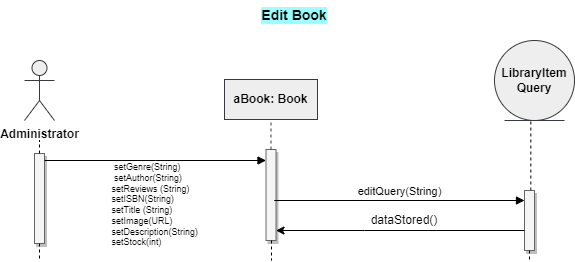


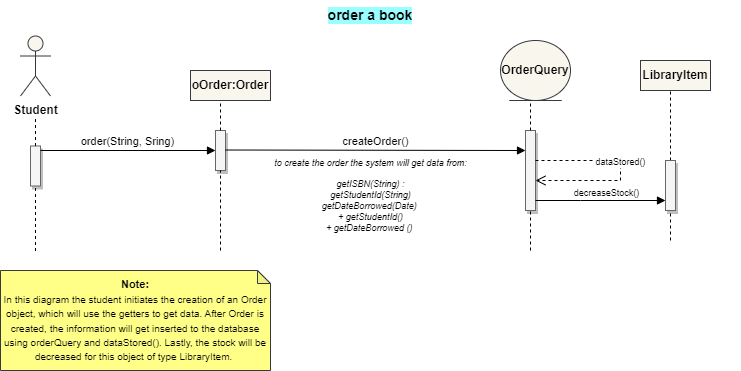
***4.10 Sequence Diagrams***

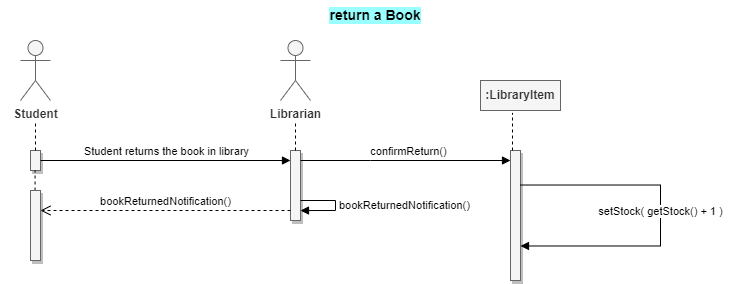


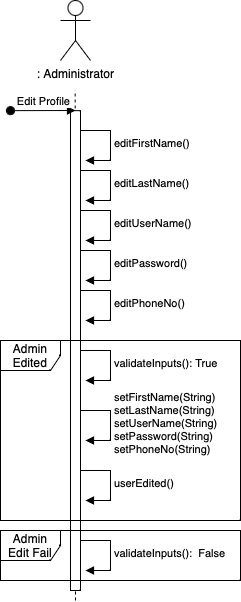


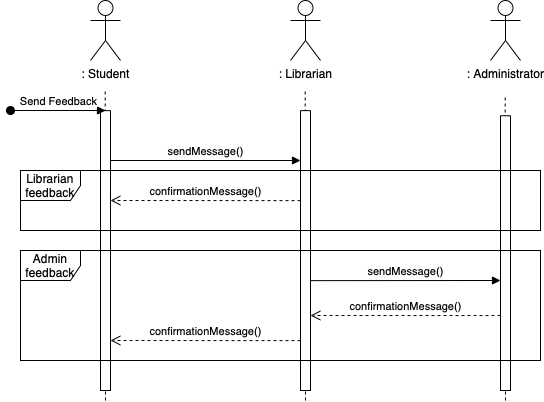


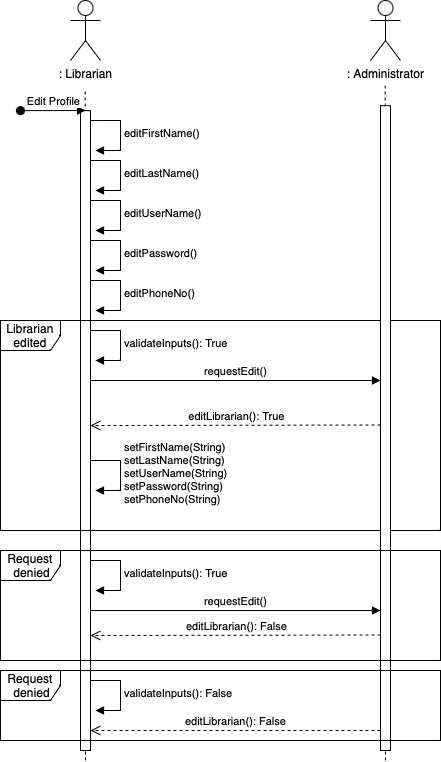


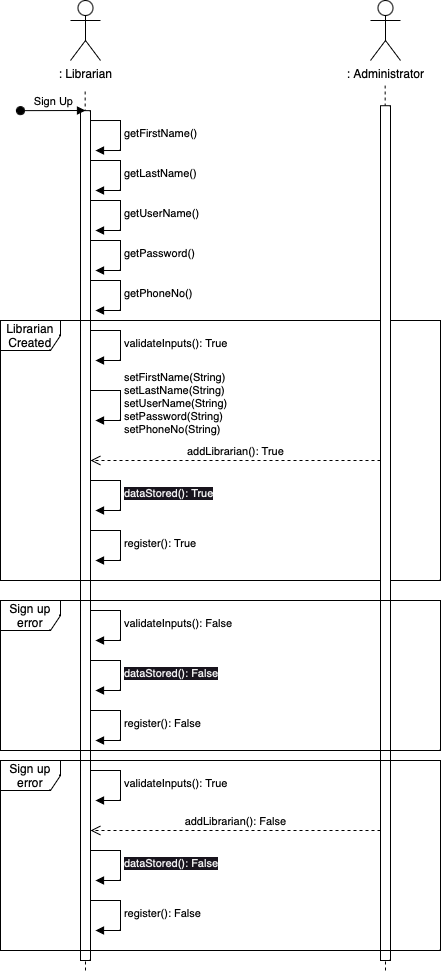
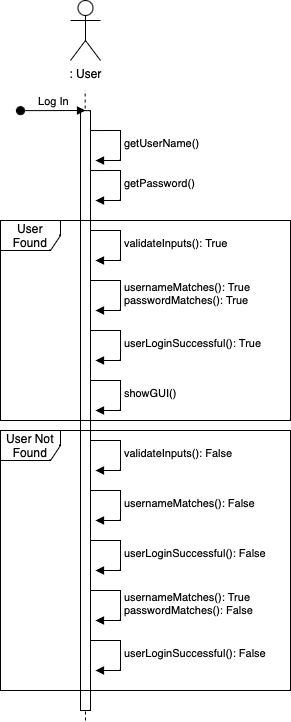


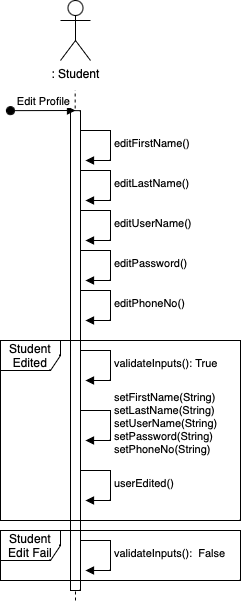


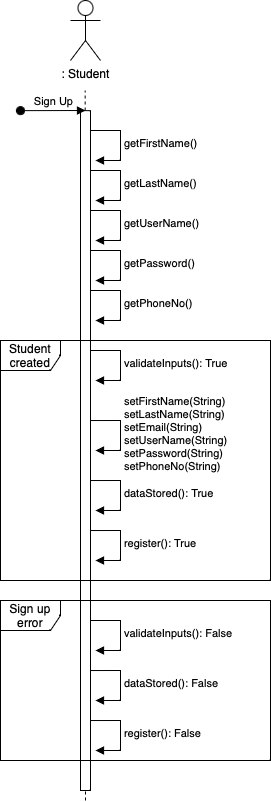




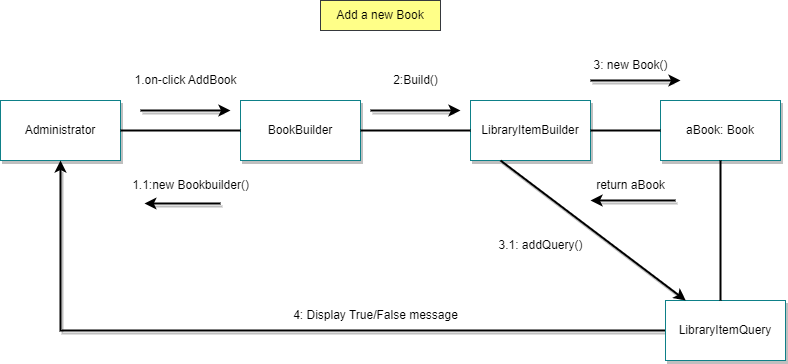


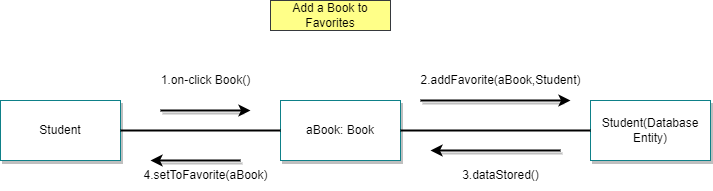
  


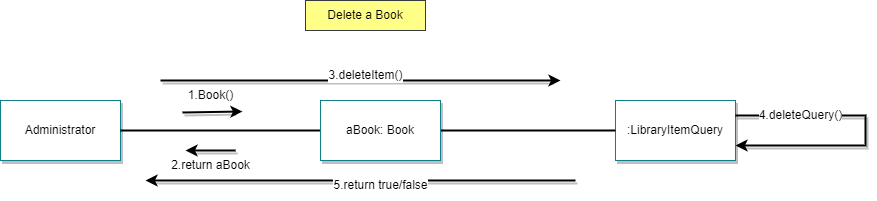


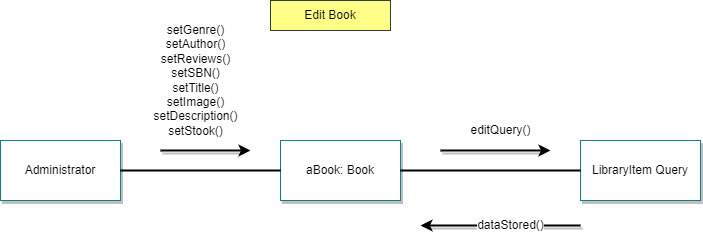


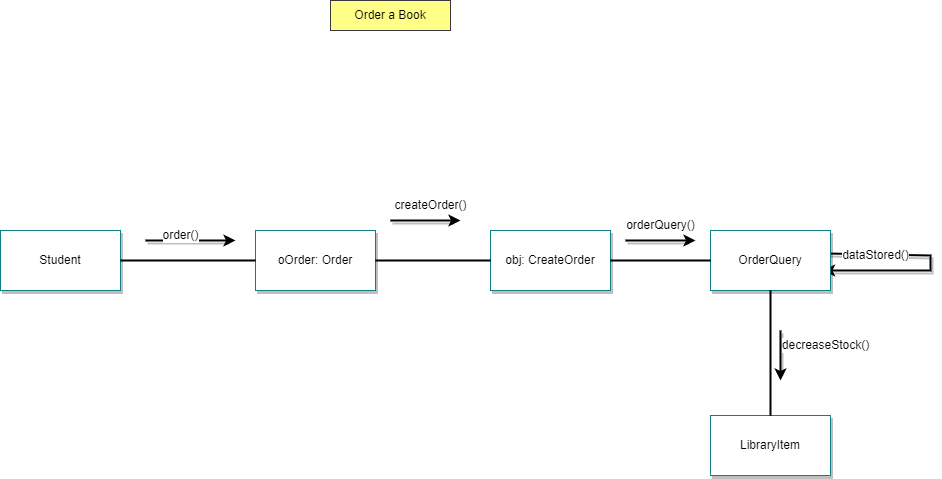
***4.11 Collaboration diagrams***

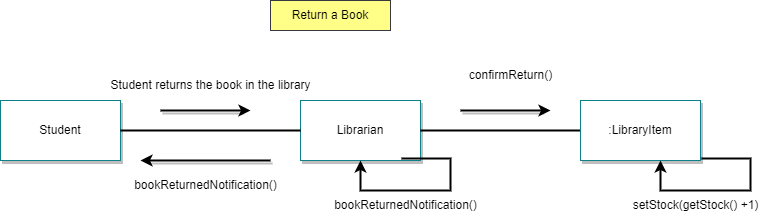






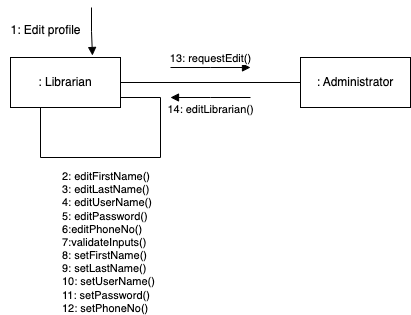


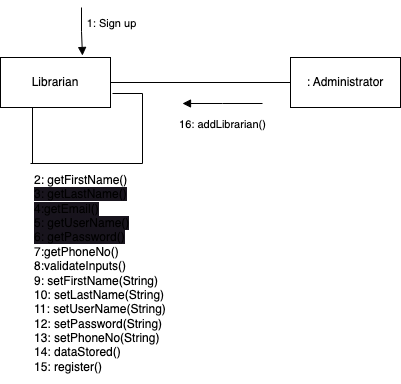


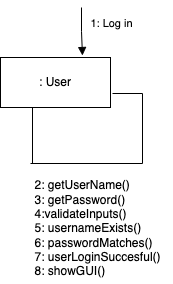








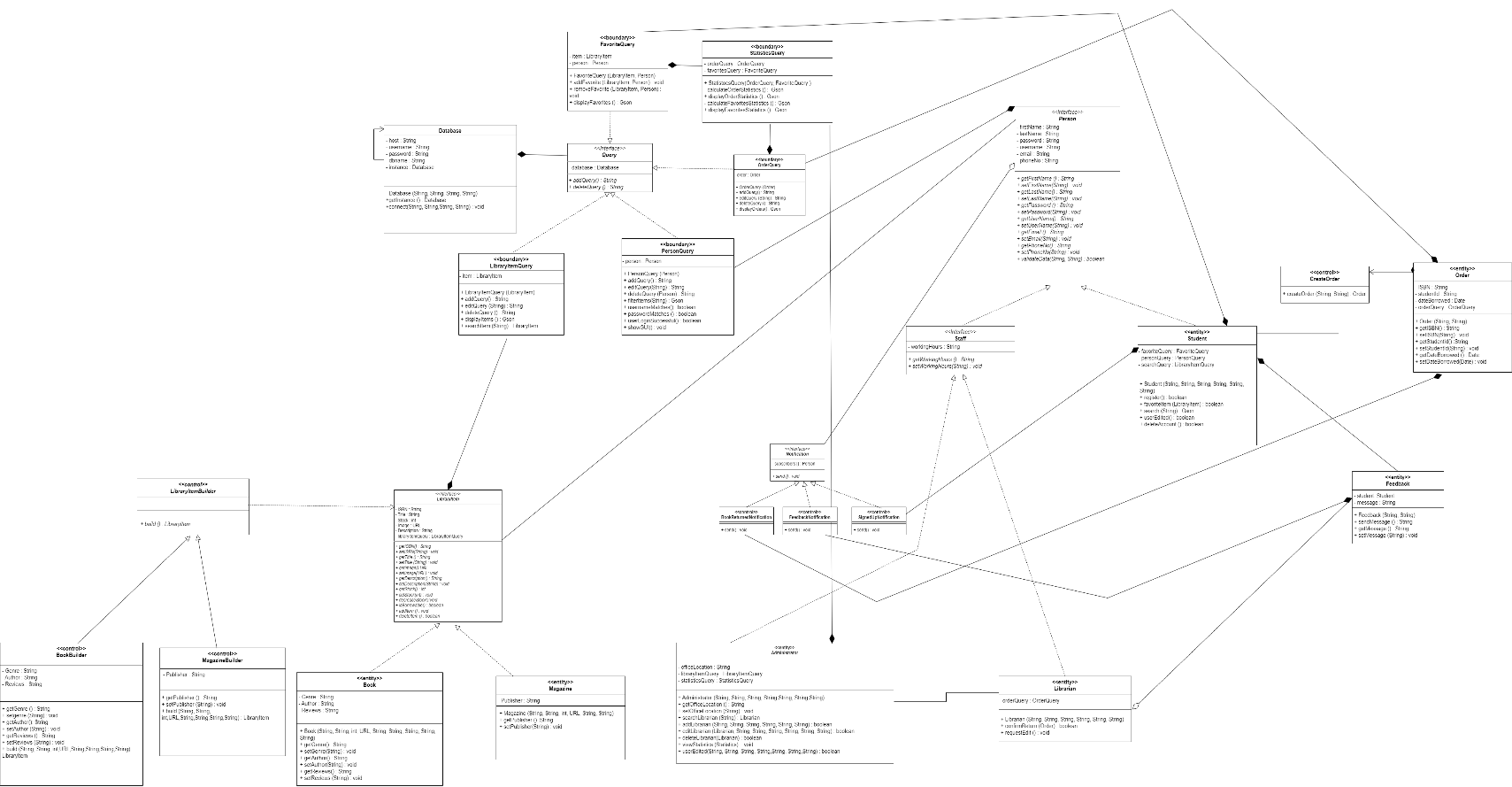








***4.12 Class diagram***

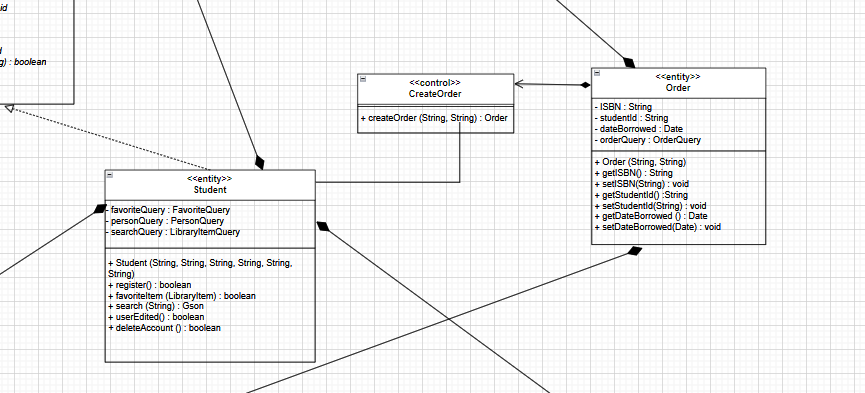


**5. Design patterns**

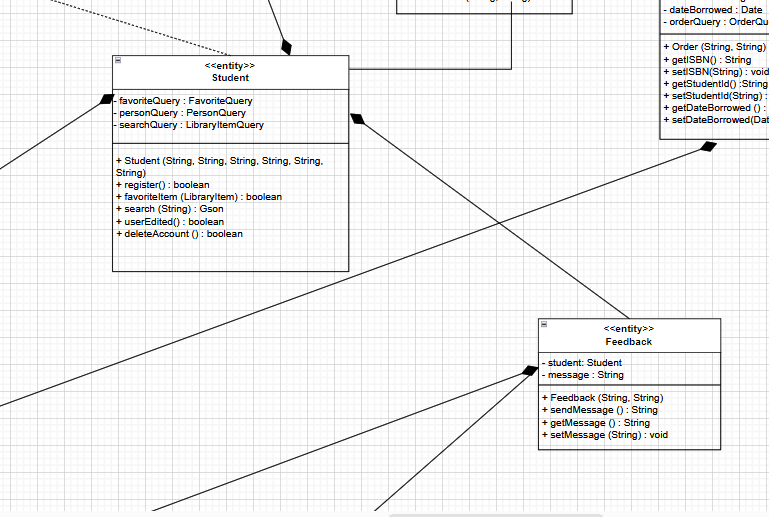
**S.O.L.I.D. Principles**

While planning the general flow of implementation of code for our electronic library system and designing the classes for our project, we managed to put in good use several principles in favour of an efficient product, which is easily adaptable to changes. The principles we have kept in mind are the S.O.L.I.D. ones, which stand for: S – Single Responsibility Principle, O – Open/ Closed Principle, L – Liskov Substitution Principle, I – Interface Segregation Principle, D – Dependency Inversion Principle. To illustrate the usage of them we are providing you with some snippets of our diagrams.

1. Single Responsibility Principle

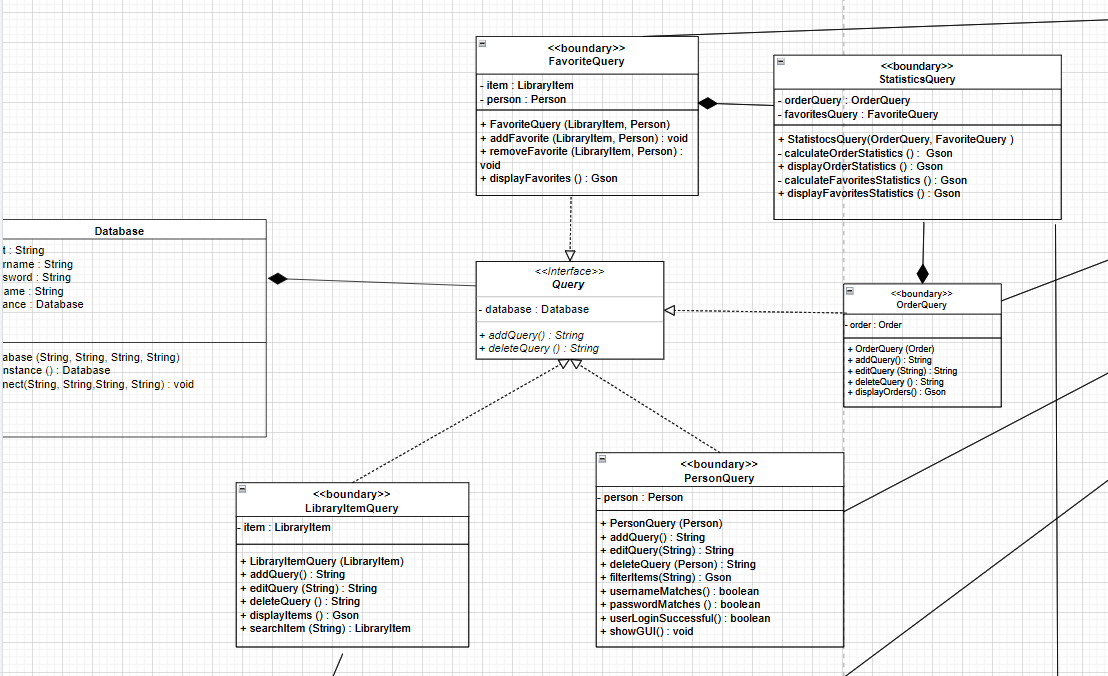


One of the main features provided to students is the ordering of library items. However, this functionality does not define the student, therefore it does not have to belong to the student class. To satisfy this condition, a separate class for orders is created, the objects of which are instantiated by the



As in the previous example, the student’s actual responsibility is not the delivery of feedback. Therefore, this task can be extracted to a separated, single- responsible class, which is the Feedback class.

2. Open/Closed Principle

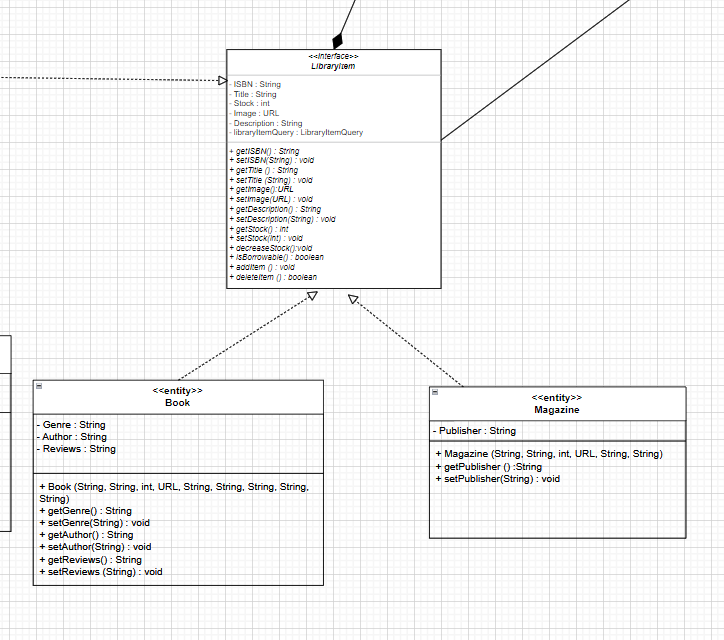


The query interface we have created, perfectly illustrates the second principle, in which whenever new queries need to be created, it will not be necessary to make changes to the query class, which may cause serious problems with different parts of the code. Instead of doing this, using the design model that we have shown in the image above, we can simply create a new class which implements the Query interface, and assign new behaviour to it as needed.

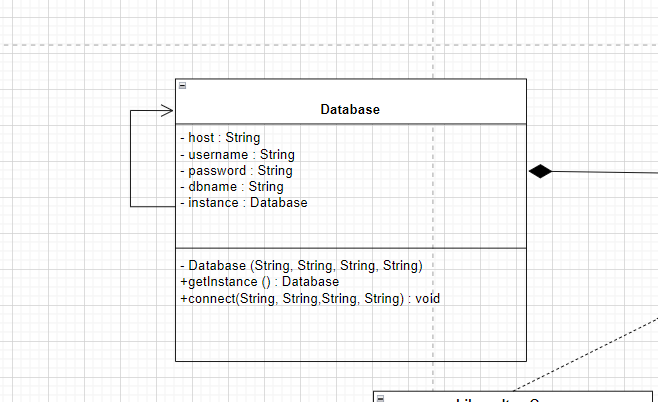
3.Interface Segregation Principle

The above diagram, also serves as an appropriate demonstration of the Interface Segregation Principle. To further explain the logic behind it, the concrete query classes only have to implement the two methods that are part of the query interface, that are two basic operations that each query class must have. More precisely, we removed the editQuery method from the Query interface, since not every class that implements this interface needs to exhibit this behaviour.

4.Dependency Inversion Principle



In this example, we have an interface called LibraryItem that contains several methods that are overridden by the concrete classes. Consequently, these concrete classes can be instantiated and used with the reference type of the Interface, making use of one of the three pillars of Object Oriented Programming. It is important to mention that this is not the only case where we have made use of this S.O.L.I.D. principle.

**Design Patterns**

1. Singleton

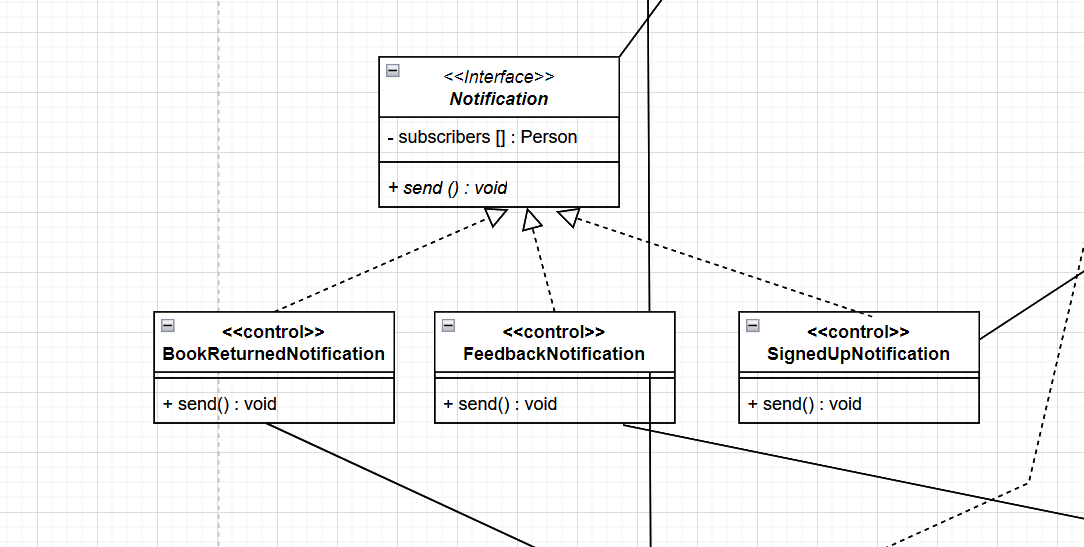
The singleton is a fundamental design pattern, which belongs to the creational design patterns. It is used to prevent the instantiation of more than one object from a particular class. There are several benefits of using this particular design pattern, from which we can mention the added efficiency of sparing the memory from overloading with several instances of the same object. In our case, only one instance of the database needs to be created and used. To achieve this, the Database class contains an object of itself, and the constructor which initiates the database connection is declared as private. Instead of directly calling the constructor from outside of the class, which would lead to the creating of various objects from it, the constructor is only called once inside the getInstance method, only if it has not previously been done.

1. Builder



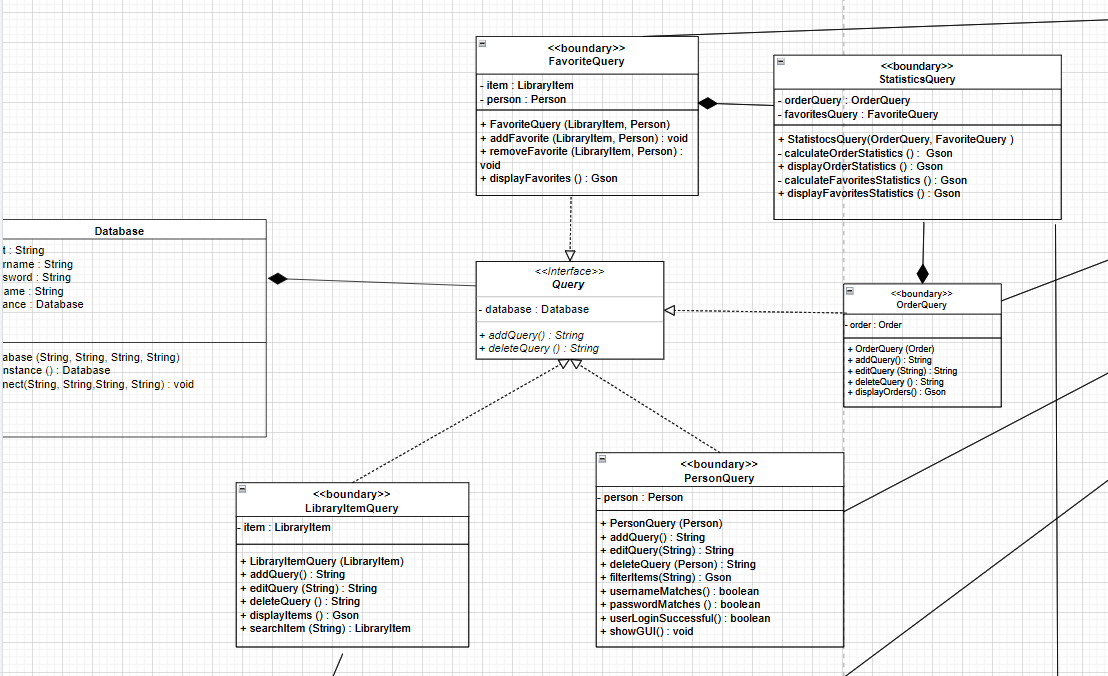
The builder design pattern is a creational design pattern, that is applicable in the instantiation of objects with complex constructors. The object is constructed step by step, until the finished product is made available. The first thing we did was creating an interface called LibraryItem which contains five private attributes that are shared between library items, and their respective accessors and mutators. The next step was creating an abstract class, called LibraryItemBuilder, which would serve as the builder for our items. The steps for their creation are defined by the implementation of the setter methods in each concrete class. Finally, the build method is the one that constructs and returns the object. Both the concrete classes called BookBuilder and MagazineBuilder override the build abstract method inherited from the abstract class LibraryItemBuilder.

1. Observer



The Observer design pattern is a behavioural design pattern that allows for the creation of a subscription mechanism, in which all the items which are observing a certain change to occur, will get notified once this event is fired. We have clearly demonstrated the usage of such pattern, in a process that is directly linked to the awaiting of such signals. Precisely, the notification system, which consists of the process that notifies the successful sign up of the users, and the informative email about the duty of returning the borrowed library item, shows the usage of this principle in our project.

1. Factory



The Factory design pattern is another creational design pattern that serves the purpose of creating concrete subclasses that must perform specific tasks, whose general behaviour is determined in the interface from which they are all implemented from. Since the design model for our product relies on the communication with the database via the queries, it appeared reasonable to create a Query interface, that defines as general behaviour the two most common operations of addition and deletion. Other different queries that implement from it, must override these two methods, and also may include additional features that only belong to that class.