Library Manager Web Application

Filip Bucher

Alicja Banaszewska

Sara Sobstyl

Szymon Poterejko

Bartłomiej Szkodny

Bartłomiej Pałucki

List of contents

1. Project Description

The goal of the project was to design and implement a desktop application for managing a personal book collection. The system was created for individuals who read and collect books and want an easy way to keep track of their library. The application allows users to register all the books they own, along with basic bibliographic information such as title, author, and year of publication. Additionally, for each book, users can specify its physical location — for example, the shelf and bookcase — which makes it easier to locate a specific copy later on.

An important feature of the system is the ability to lend books to other people — friends, family members, or co-users of the system. The application records information about who a particular book was lent to and allows it to be later marked as returned. This way, the user has full insight into which books are currently in their possession and which have been lent out.

The system also supports searching for books not only within the user's own collection but also in a shared database containing all users' collections. Thanks to the use of a cloud-based database, data is synchronized in real time, allowing access to a shared, community-driven book collection. In this way, users can easily check who owns a particular book and, if needed, contact the owner to borrow it.

The application was designed as a practical and user-friendly tool to support the daily management of a personal home library, while also enabling users to share their resources with other literature enthusiasts.

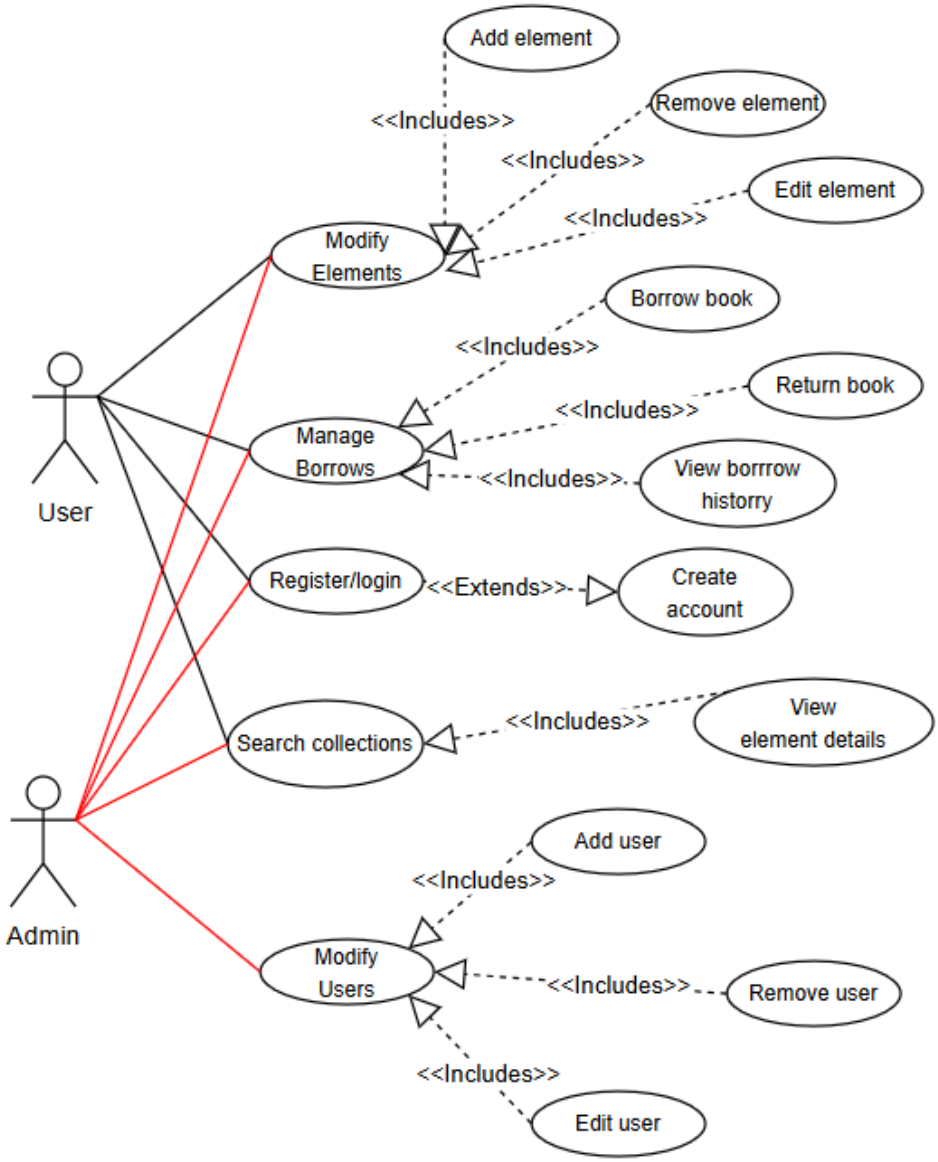
2. Analysis and Design Phase

The analysis and design phase was a key stage in the application development process. Its goal was to thoroughly understand the end user's requirements, identify the system's functionalities, and design its logical architecture and data structure. During this phase, the main UML models were also created, which helped visualize the system’s structure and behavior before implementation began.

2.1 Use Case Diagram

The use case diagram presents the interactions between users (actors) and the system. It shows which functions the application offers and which of them are available to different types of users.

The diagram below illustrates the interactions of users with the system:



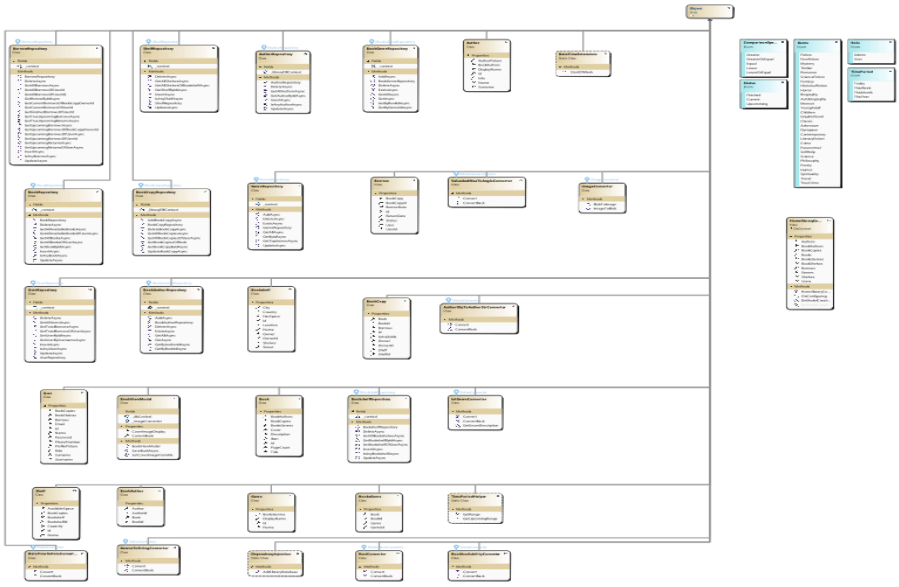
Use case diagram

- **User** has access to functionalities such as managing loans, modifying items (e.g., books), registration and login, searching the collection, and viewing item details.
- T**he Administrator** (Admin) has extended privileges — they can manage users (add, edit, and delete) and also have access to all the functions available to a regular user.

The use cases have been logically grouped using the <<include>> and <<extend>> relationships.

2.2 Class Diagram

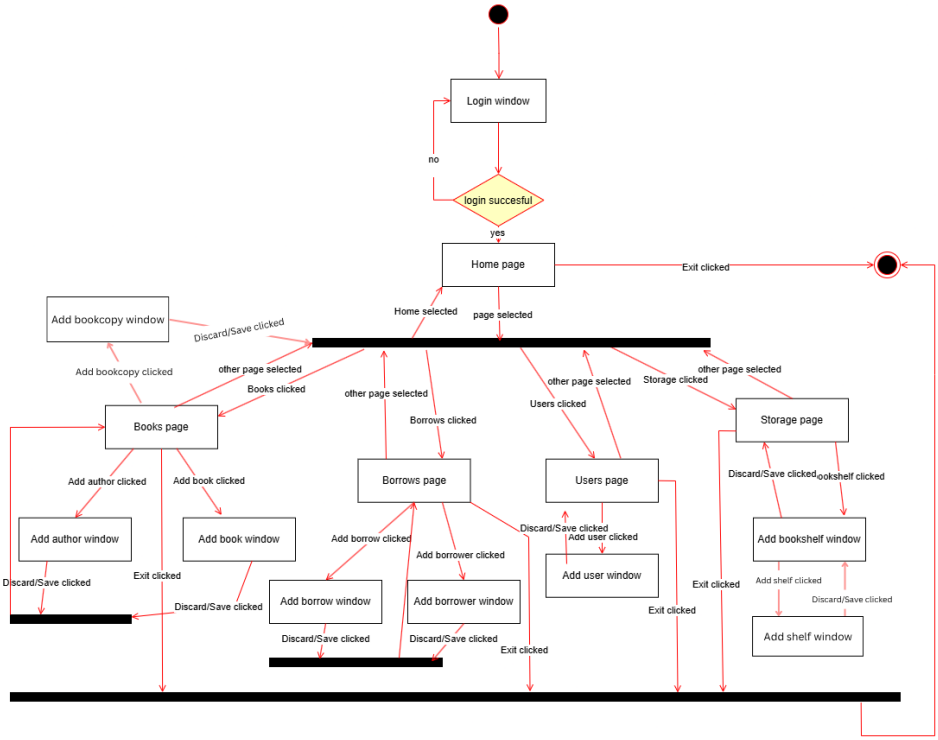
The class diagram illustrates the static structure of the system by showing its classes, their attributes, methods, and the relationships between them.



2.3 State diagram

The state diagram illustrates the lifecycle of the user interface in the application — from the moment the program starts until it is closed. The individual states correspond to different views and windows, and transitions between them are triggered by user actions such as button clicks or saving/rejecting data.

- The start point leads to the login window.
- In the case of a successful login, the user is redirected to the main page, from which they can navigate to various subpages: books, loans, users, and storage.
- Each subpage allows opening dedicated windows for data manipulation (e.g., authors, books, users).
- The data manipulation process ends with clicking the "Save" or "Discard" button.
- The application can be closed from any state by clicking the "Exit" button.



State diagram

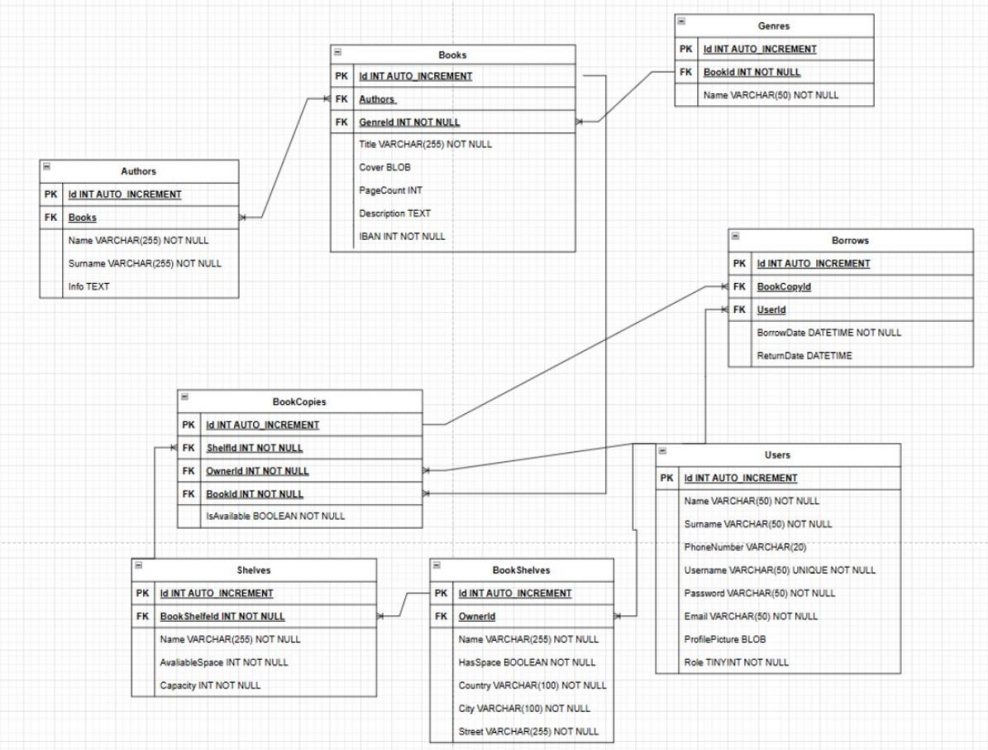
2.4 Database Diagram

The database structure was designed in a relational manner, adhering to normalization principles. Individual tables represent entities present in the application, such as books, authors, users, loans, and shelves. The relationships between tables enable consistent data storage and easy querying of the collections.

Main tables and relationships:

- **Books** – contains information about books. It is linked to the Authors table (multiple authors can be assigned to books) as well as to Genres.
- **Authors** – stores author data; linked to books (one author can have many books).
- **Genres** – literary type assigned to a book.
- **BookCopies** – physical copies of books, linked to shelves (**Shelves**).
- **Borrows** – manages the borrowing history, containing references to **BookCopies**, **Users**, **Books**, and **Owners**.
- **Users** – user data, including login information and role (e.g., admin or regular user).
- **Shelves** – store physical information about bookcases, linked to locations (**BookShelves**).

- **BookShelves** – represents the physical location of shelves (e.g., country, city, street).



Database Diagram

3. Technologies Used and System Architecture

3.1 Technology Selection

Technologies enabling the creation of modern and responsive desktop applications for the Windows environment were chosen for the project implementation. The main technologies and libraries used in the project are:

- Programming language: C#
- Framework: .NET 7 (using Windows Presentation Foundation – WPF)
- Data access: Entity Framework Core (ORM), **Database First** approach
- Design pattern: Repository Pattern
- Database: MySQL hosted in the cloud via the Aiven service
- External Libraries:
    - Clippy – a library that supports user communication through an animated assistant.

The Database First approach allowed generating entity classes and the database context directly based on the existing MySQL database structure hosted in the Aiven cloud. This

enabled quick mapping of the database schema into the application code and ensured consistency of the data model.

WPF was chosen as the frontend technology due to its capability to create complex graphical user interfaces, and interaction with UI elements was implemented using code-behind (logic directly attached to events in the .xaml.cs files), which allowed for simpler and more direct binding between the interface and the application's logic.

In the final stage of the project, an application installer was also prepared, enabling easy distribution and installation on users' computers. The installer automatically copies all necessary files, creates shortcuts, and can perform additional actions (e.g., registering libraries, creating folders for application data).

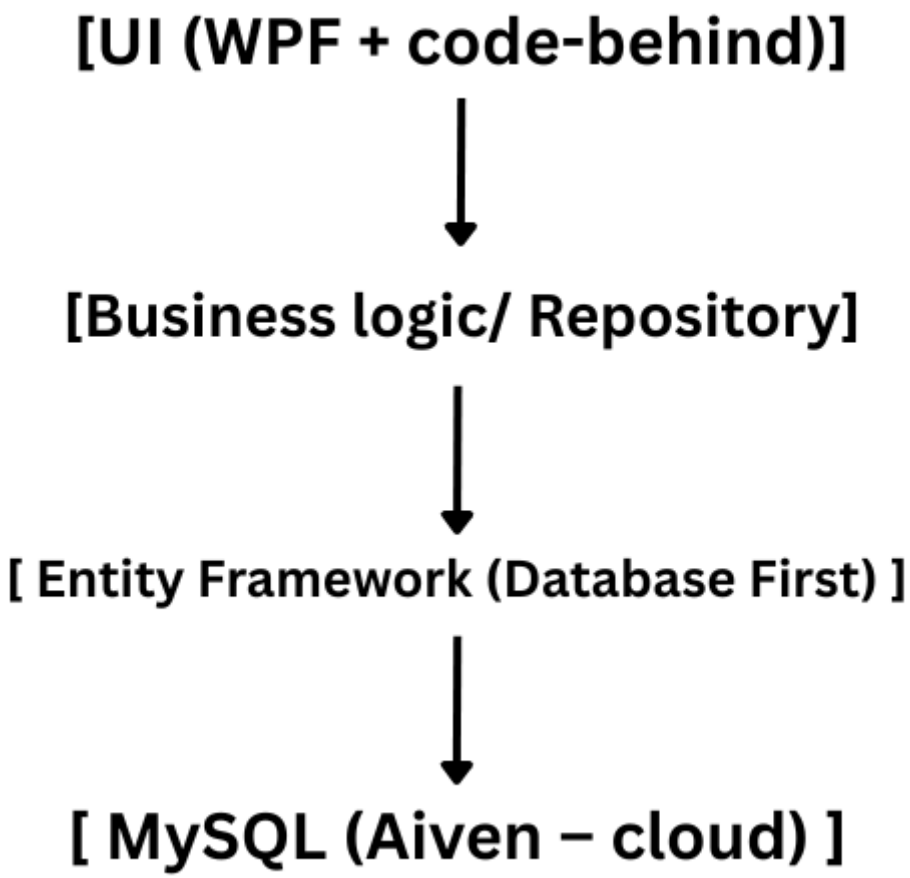
3.2 System Architecture

The system was designed as a desktop application operating in a client-server architecture. The user application (client) connects directly to the cloud-hosted MySQL database maintained by the Aiven service, enabling data centralization and remote access.

The application architecture is based on a layered approach:

- **Presentation Layer (UI)** – built using WPF. User interface logic is handled using the code-behind mechanism, without employing the MVVM pattern.
- **Business Logic Layer** – implements the application's operating rules and processes data according to functional requirements. The Repository pattern was applied here to separate the application logic from direct data access.
- **Data Access Layer** – utilizes Entity Framework Core with the Database First approach. The database structure was automatically mapped in the source code through generated classes and context.

The architecture can be schematically represented in a simplified way as follows:



Such an architecture allows for relatively fast application development, easy modification of the interface, and separation of business logic from implementation details related to the database. Additionally, thanks to the prepared installer, easy installation and distribution of the application in end-user environments is possible.

4. External specification

List of contents

4.1 Register/Login

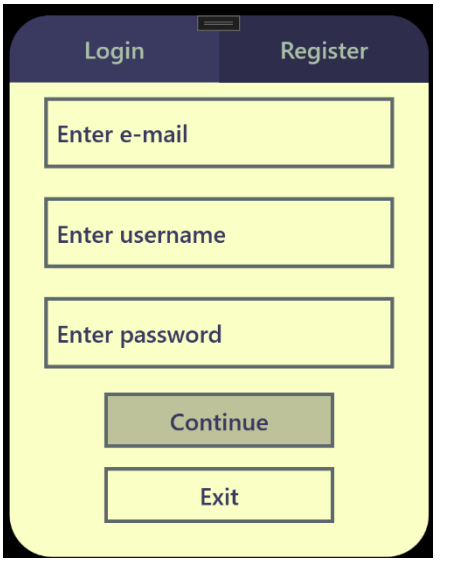
4.1.1Chosing action

You can choose between two actions. You can Login if you already has an account, or create new one if you want to have your own account.

4.1.2 Register

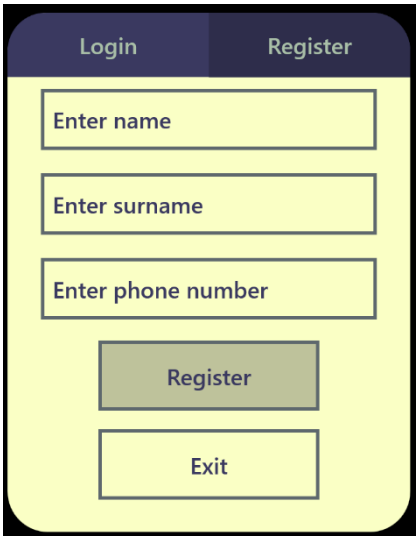
Step 1:

- Go to the Library Manager registration page.
- Enter the following information:
  - Email address – type your valid email (e.g., example@email.com).
  - Username – choose a username
  - Password – create a strong password
- Click Continue.



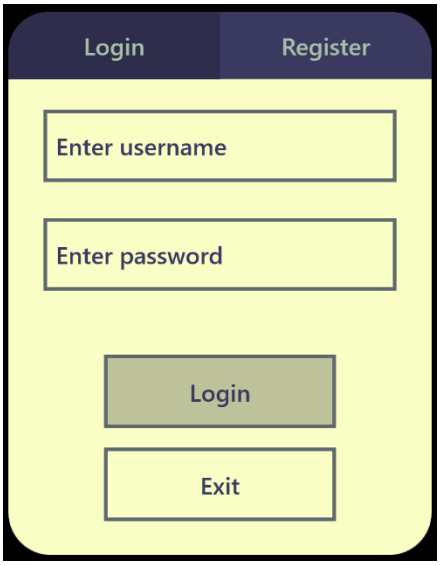
Step 2:

- Fill in the second registration form
- Enter your personal details:
  - First Name – type your real first name (e.g., John).
  - Last Name – type your surname (e.g., Doe).
  - Phone number – provide a valid phone number
- Click the Register button to complete the registration.



4.1.3 Login

- Open the Library Manager login page.
- Enter your login credentials:
  - Username – type the username you used during registration.
  - Password – type your password.
- Click the Login.
- If the username and password are correct, you will be logged into your Library Manager account.



4.2 Menu

4.2.1 Info User

- On every site there is Info User. If we hover over it, it will highlight.

- If we click, a window with information will appear.

4.2.2 Home

- The main page shows us statistics. Each statistic corresponds to its caption.
- In “Total Borrows” we can choose how we want to see it. We can choose:
  - Year
  - Month
  - Day
- In “Upcoming” we can choose how we want to see it. We can choose:
  - Year
  - Month
  - Day

4.2.3 Books

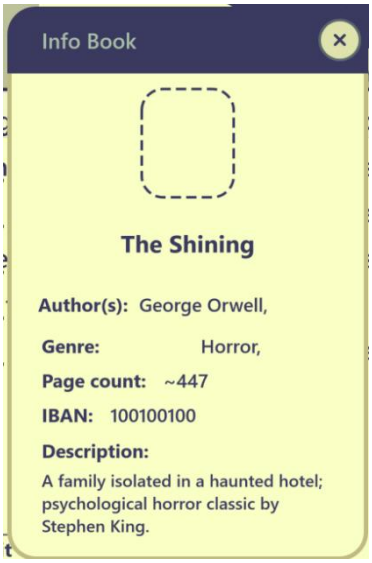
- On this subpage we can see all information about the books that are available in our library. There are two pages: “All books” and “Book copies”.

4.2.3.1 All books

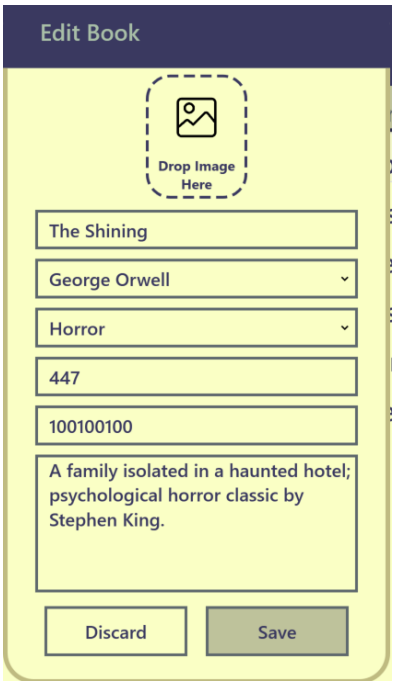
This Page shows all books inserted into the system.

- There are columns:
  - Title
  - Author
  - Genre
  - Page Count
  - Available

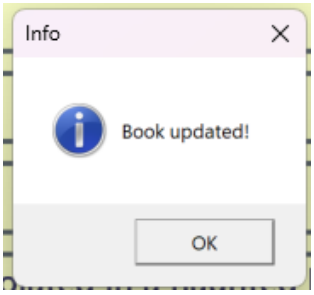
- If you click “Info”, there will appear box with extended informations about the book.



- If you click “Edit”, there will appear box in which you can change every data that refers to this book. You can choose if you want to “Discard” or “Save”

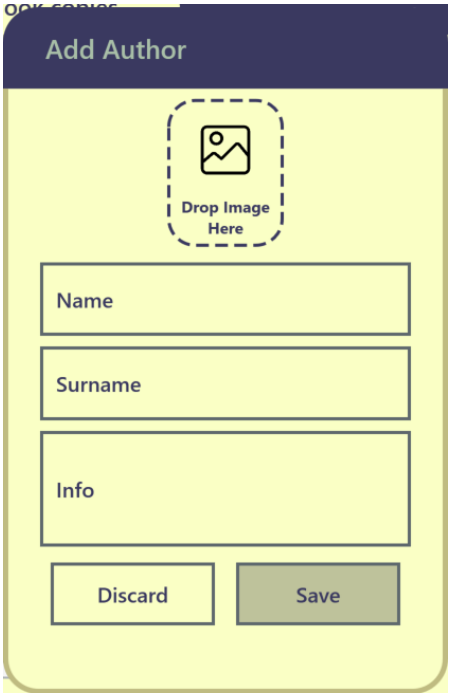


- If you click "Discard”, none of the changes will be saved.
- If you want to save, new box will appear that will inform you that the informations were updated.

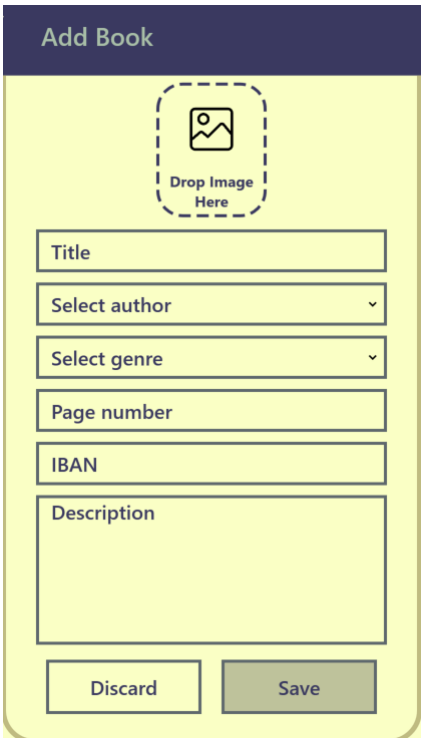


- There is small “+” on right lower corner. If you click on it there are two options:
  - Add Author
  - Add Book

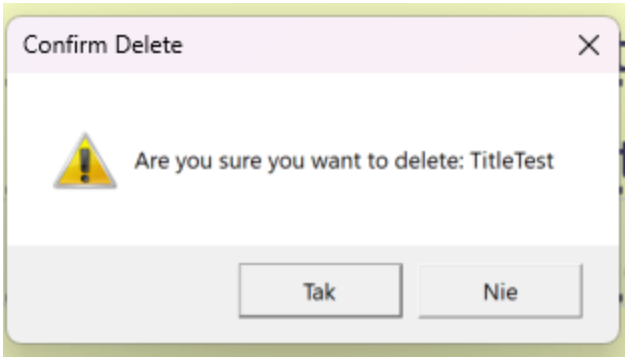
- If you click on “Add Author”, new window will appear in which you can add new Author and some information about him.
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, Author will be added to the list of Authors.



- If you click on “Add Book”, new window will appear in which you can add new Book and some information about it. You can select author that was previously added.
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, Book will be added to the list of Books.

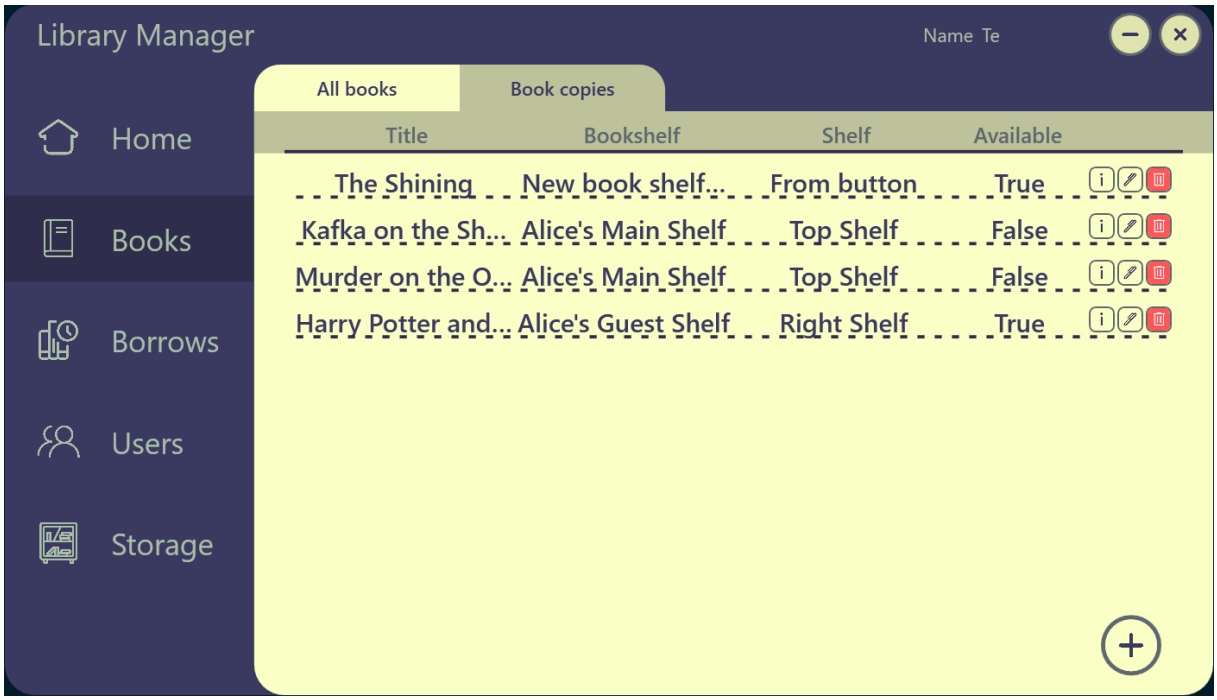


- If you choose book and than click “Delete”, new window will appear asking if we really want to delete it.
- If we choose “Tak”, the book will be deleted.
- If we choose “Nie”, the book won’t be deleted.



4.2.3.2 Book copies

- On this page you can view, edit and delete all the book copies of currently logged in user. If you click on “+” and choose “Add Book Copy” than new window will appear.



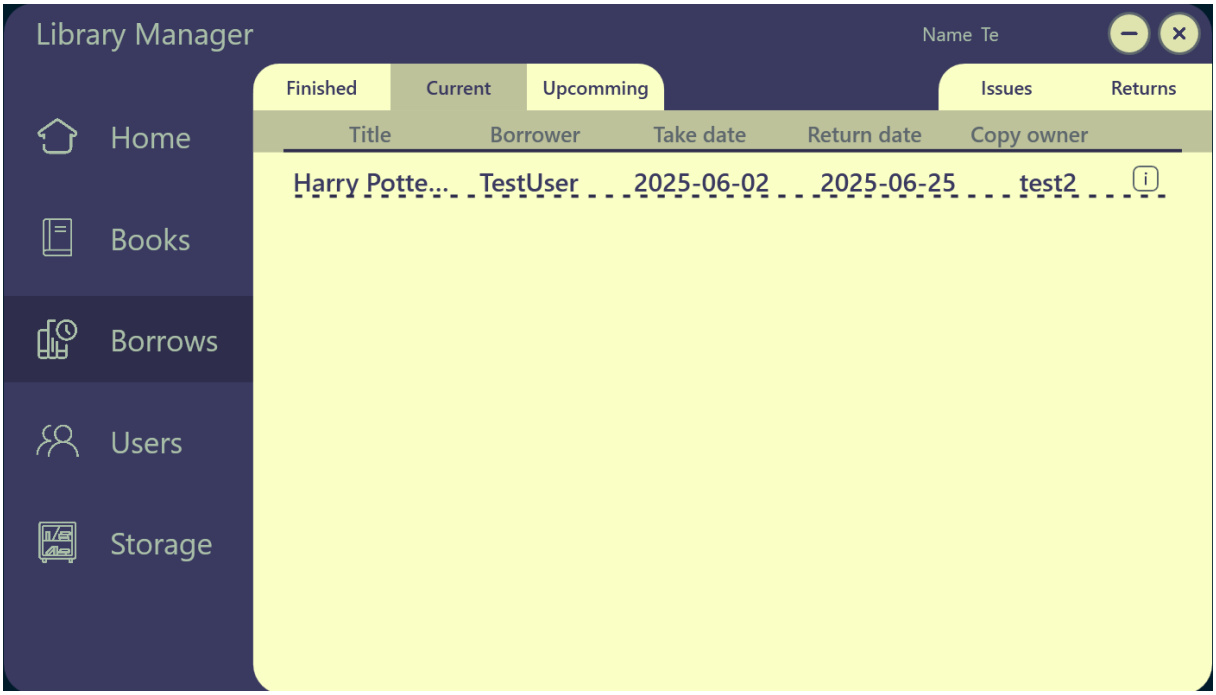
4.2.3.3 Add new book copy

- Here you can add a copy and set if it is available or not.
- If you click "Discard", none of the changes will be saved.
- If you click "Save", new copy will be added.

4.2.4 Borrows

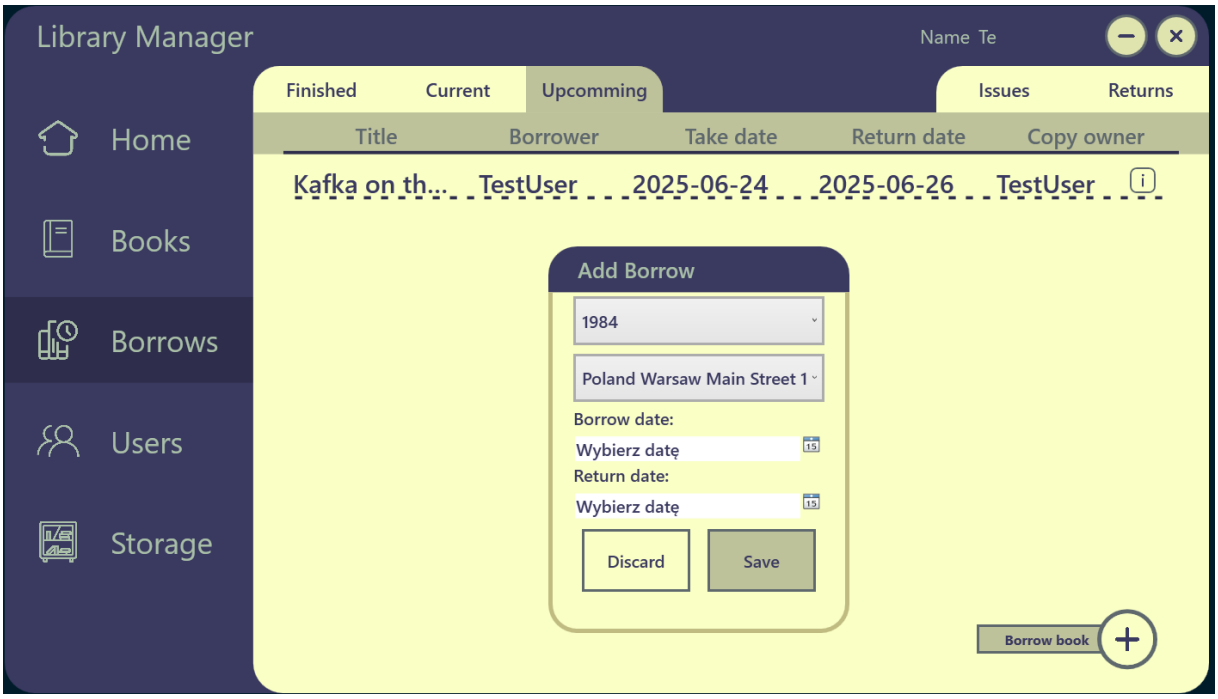
4.2.4.1 Finished & Current

- In those you can see the informations about Borrows such as:
  - Title
  - Borrower
  - Take date
  - Return date
  - Return date
  - Copy owner



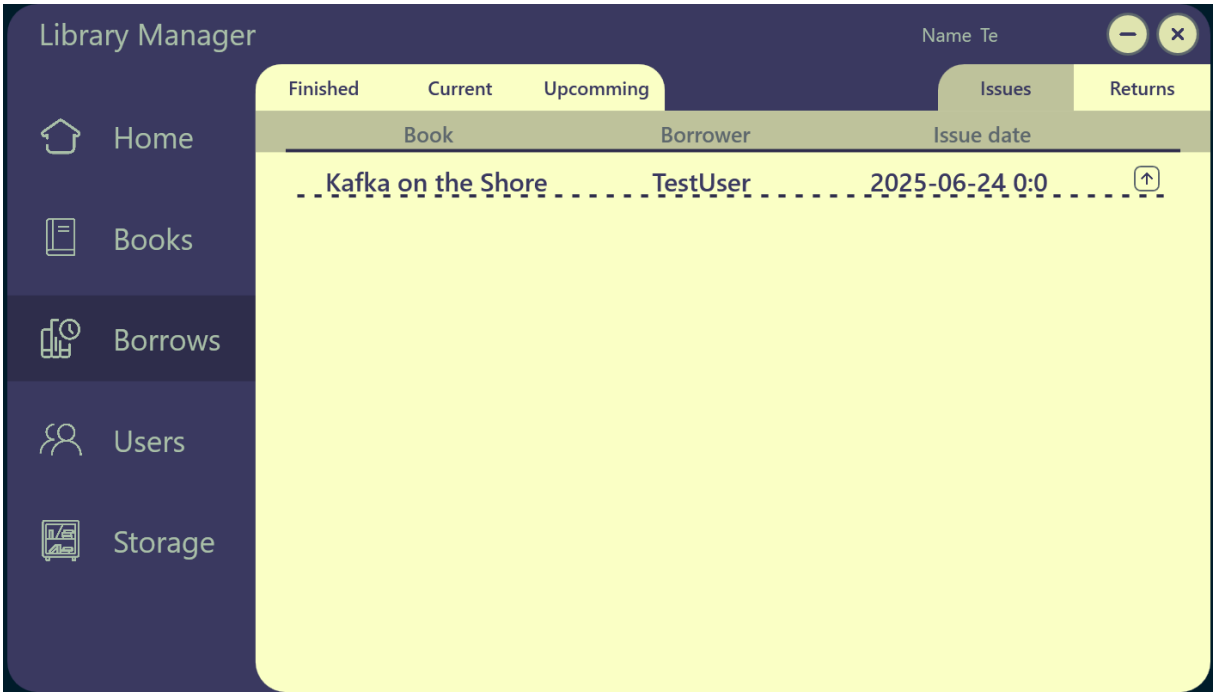
4.2.4.2 Upcoming

- In those you can see the informations about Borrows such as:
  - Title
  - Borrower
  - Take date
  - Return date
  - Return date
  - Copy owner
- You can click on “+” and choose “Borrow book”. New window will appear in which you can choose the book that you want to borrow, from which location you want to borrow it, and you can choose the Borrow date and Return date.
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, Borrowing will be saved.



4.2.4.3 Issues & Returns

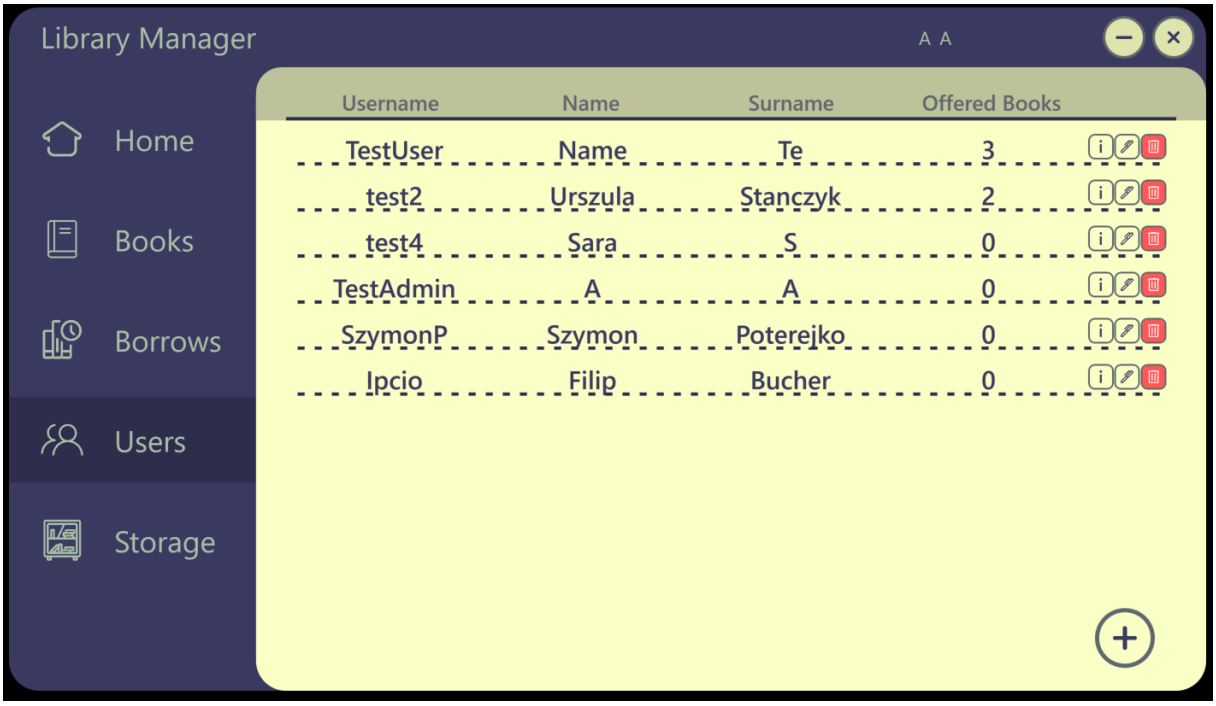
- In those you can see the information about borrow issues and returns such as:
  - Book
  - Borrower
  - Issue date



Next to every issue and return entry there is a button for confirmation that the book was returned or picked up by the borrower.

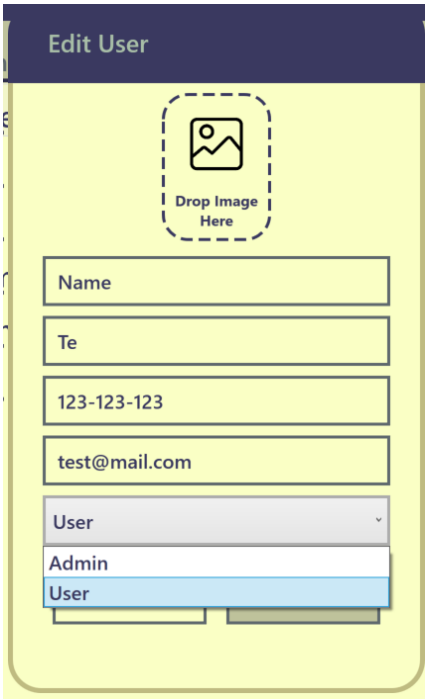
4.2.5 Users

- On this page you can see all the users and get more detailed information as well as edit some information about user. You can also delete an user.

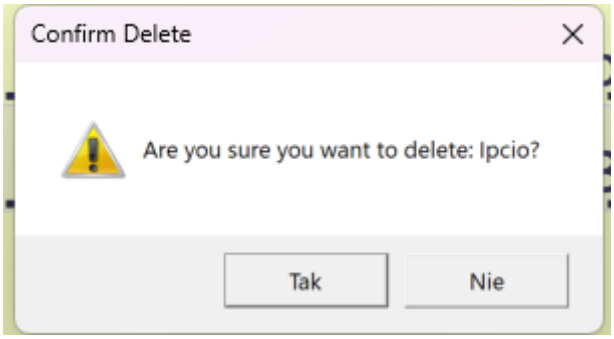


- If you click on “i”, new window will appear in which you get detailed information about specific user.

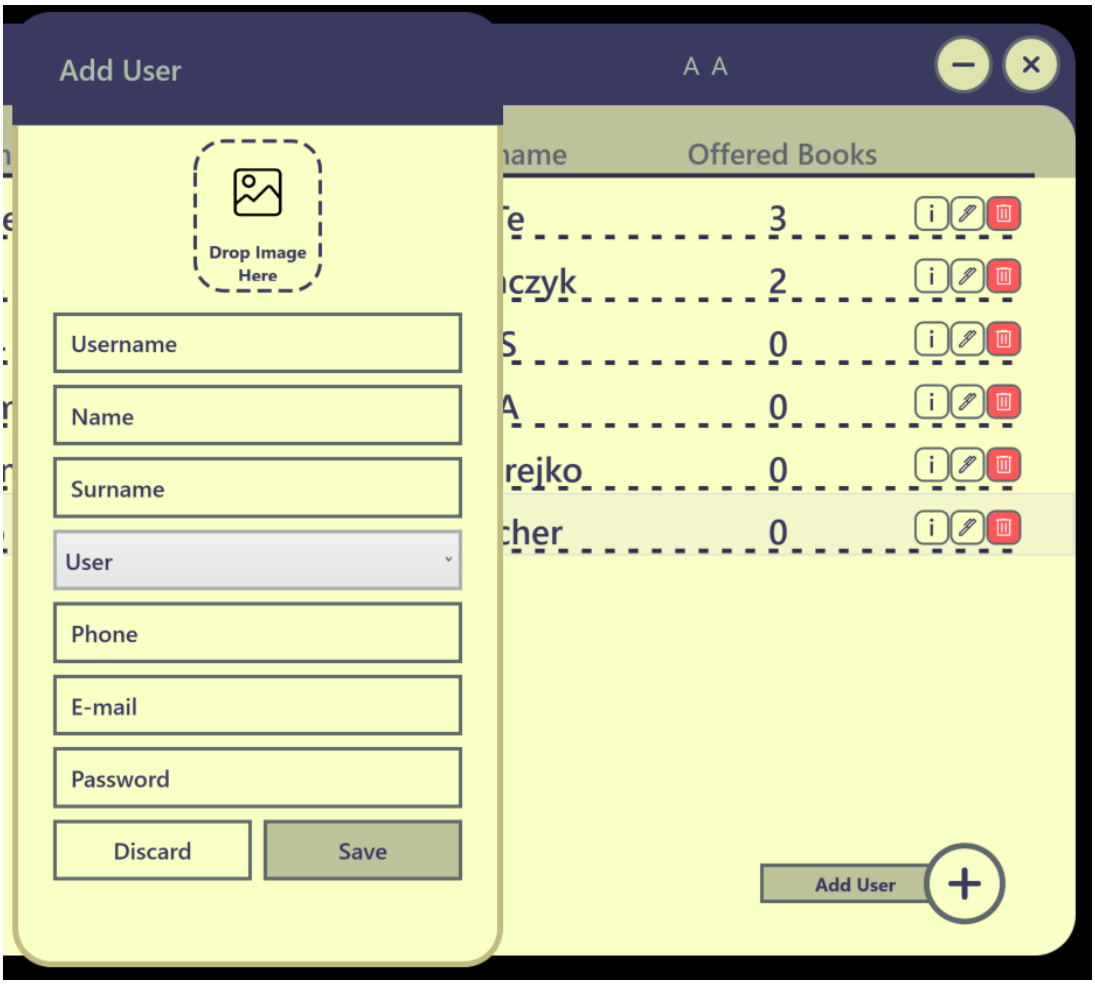
- If you click on edit, new window will appear in which you can change information about specific user.
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, information will be saved.



- If you want to delete you should click on trash icon. Than the window will appear asking you if you want to delete User.
- If we choose “Tak”, the User will be deleted.
- If we choose “Nie”, the User won’t be deleted.

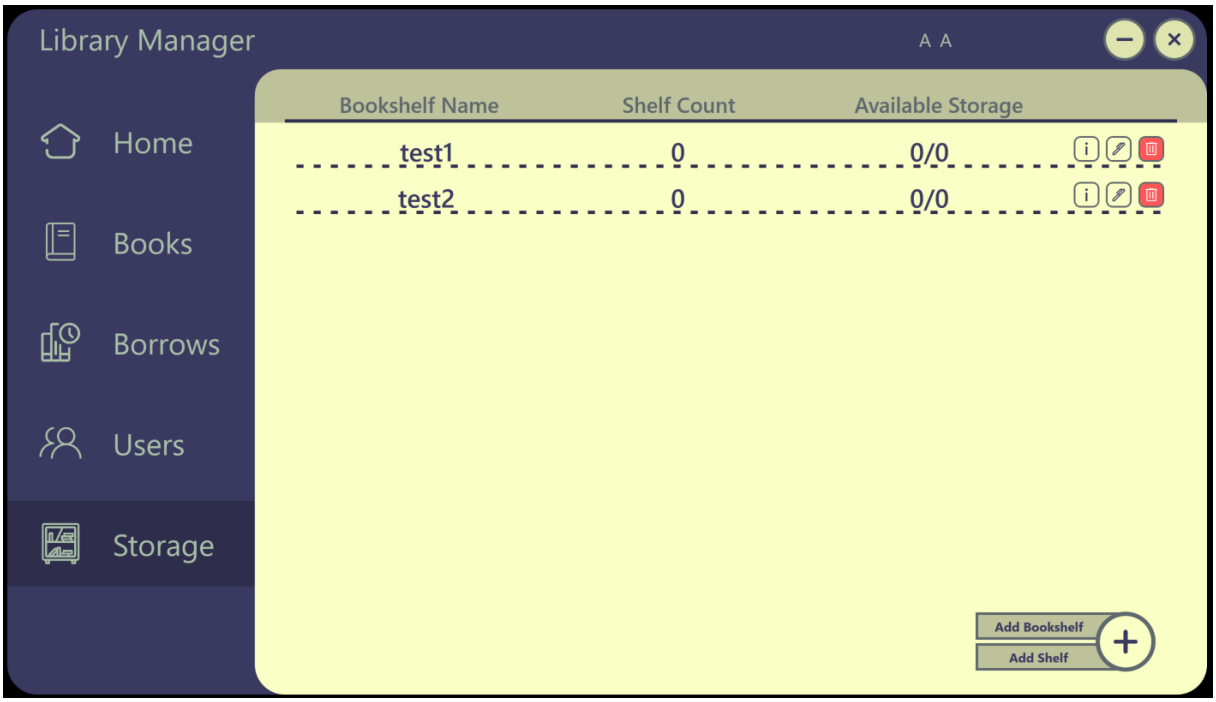


- If you click on “+” button and choose “Add User”, you can add new user. Process of creation is similar to Registration Page, but here you can choose a role if you are Admin.
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, information will be saved.



4.2.6 Storage

- On this page you can see the storage. The information is:
  - Bookshelf Name
  - Shelf Count
  - Available Storage



- If you click on “i”, new window will appear in which you get detailed information about specific Bookshelf.

- If you click on edit, new window will appear in which you can change information about specific Bookshelf. You can also add new shelves.
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, information will be saved.

- If you click on “+”, than new window will appear in which you can add the:
  - Name
  - Capacity
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, information will be saved.

- If you click on “+” and than “Add Shelf”, new window will appear in which you can add the:
  - Name
  - Capacity
  - Bookshelf name
- If you click "Discard”, none of the changes will be saved.
- If you click "Save”, information will be saved.

5. Project conclusions

It was one of the biggest projects we made during these studies. We made it in quite a big group while having to make other big programs, and we used multiple technologies at once, like MySQL and WPF.

Our organization was very good, even better than we expected. We picked one of us for a leader that gave himself and others tasks to do regularly and that worked really well. Every one of us learnt something new and had an opportunity to use technologies they never used before in a quite friendly way. We focused not only on programming itself, but also on making our app look pretty and user-friendly- we even designed an icon.

It was an effort, and it took longer than we expected, but in the end, we are proud of our application. We're sure it helped us become better developers, especially while working in a team.