

# Big books

Documentation and Contribution

# Introduction

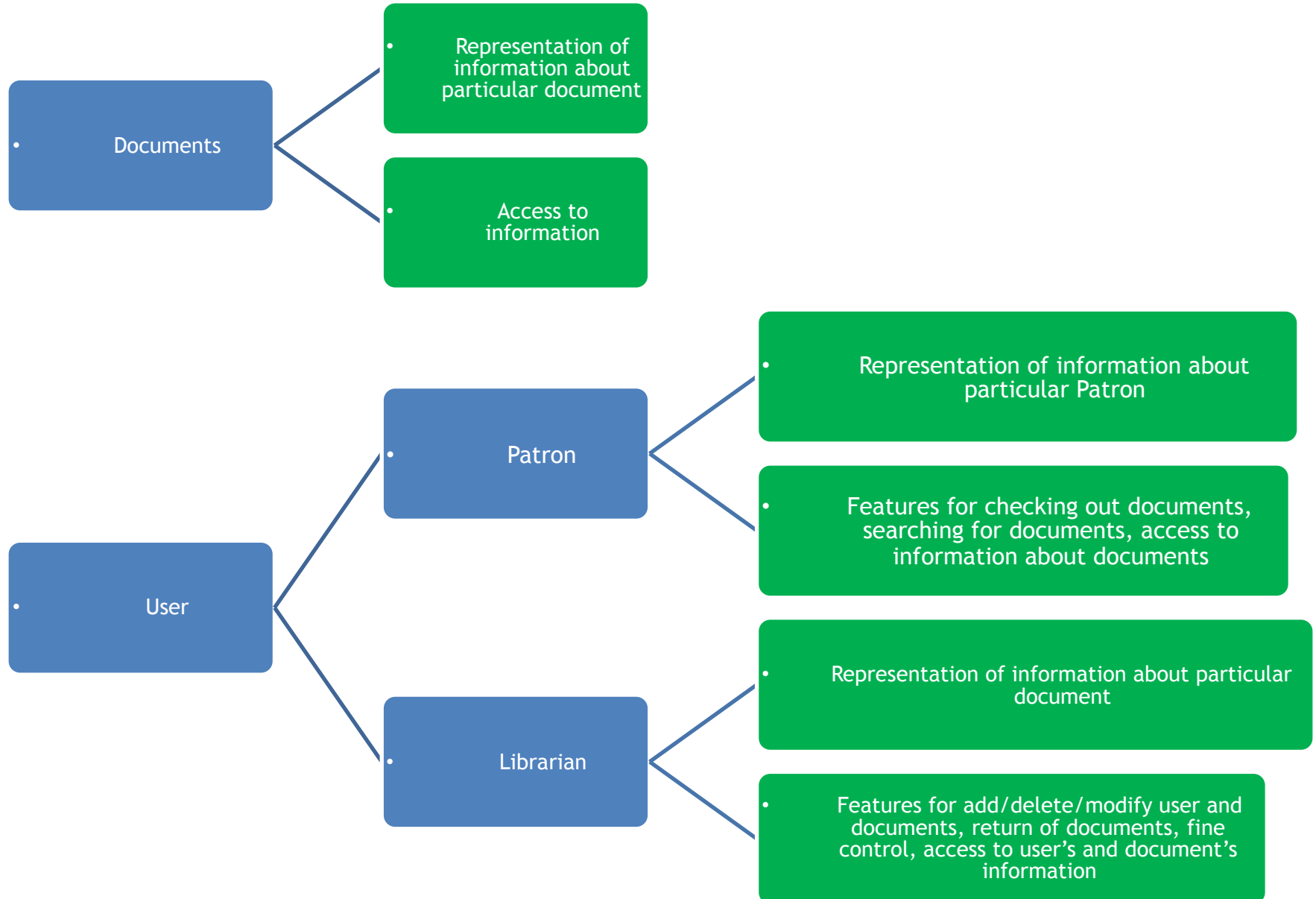
Our system includes in 3 segments

- Database interaction(Oleg Andriyanchenko)
- Friendly representation of data, understandable interface of all system's part(Artur Akhmetshin)
- GUI, interface for users(Denis Pimenov)

# Friendly representation of data, understandable interface of all system's part

- This part of system gives GUI programmer logical and corresponding to the system requirements representation of data.
- It uses data base interface, provided by data base interaction programmer. And represent it in understandable and usable way. Also it edits and deletes information in data base, by data base interaction interface sure.
- All information about data base is hidden for GUI programmer, since all corresponding features are available there.

# Structure of system interface classes



There are two methods in classes `checkOut` and `checkOutTest`, they are the same except only that `checkOut` automatically set current date in data base, but for tests we need operate with another date, so that `checkOutTest` accepts parameter `dateS` and store it into data base

# Documents

- Class includes different features that take data from data base and after that give opportunity to interact with all types of documents
- name;  
type; // Book, JA or AV material  
docID; //ID of proper document  
author;  
copies;  
reference;  
bestseller;  
cost;  
publisher;  
edition;  
year;  
counter;

# Librarian

Class implements Librarian's methods, gives abstraction to GUI developer, that allows not to user data base straight, just use implemented set of methods.

- add/modify/delete users/documents
- public void outstandingRequest
- Auto notifications of users
- public int checkOut/Documents doc)
- public Patron getPatronInfo
- public ResultSet checkedOut
- public boolean payFine
- public boolean returnDoc
- public boolean deleteUser(int userID)
- public ResultSet copiesOfDocument(int docID)
- etc.

# Patron

Class provides feature to interact with Patron information and to action in library. It includes following features








- `public IntAndInt checkFineTest`
- `public int checkFine()`
- `public int [] getWaitingList`
- `public Documents[] bookedDocuments`
- `public ResultSet checkedOut(int userID)`
- `public IntAndString renew`
- `public IntAndString checkOut`
- `public String getDateToReturn`
- `public Documents [] getNotifications`
- etc.



# Database structure

- The database we choose is SQLite. It is easy to interact with and it is quite powerful. To interact with it we use JDBC driver.
- The database consists of 4 tables:
  - Users (stores information about users)
  - Documents (stores information about documents)
  - Copies (stores information about every copy of the document)
  - Booking (represents a queue for people who wait for the certain book)
- To interact with database there exists FcukBase.java class. It allows to add, change and delete information from the database.
- Information about methods is stored in FcukBaseInterface.java. It describes the main methods of our java class

# Users

	 id	 name	 phoneNumber	 address	 status	 password	 fine
1	1	Steve Wozniak	89227654321	New York	Librarian	qwerty	0







- ID: identification number of the user
- (Name, Phone Number, Address): personal information of the user
- Status: information about status (Student, Professor, TA, etc)
- Password: password
- Fine: current fine

# Documents

	id	name	author	type	publisher	year	edition	counter	cost	reference	bestseller
1	1	Introduc...	Thomas H...	document	MIT Press	2009	Third Edition	4	100	F	F





- Id: identification number of the document
- (Name, Author, Publisher, Year, Edition): information about the document
- Type: type of the document (AV, book, article, etc)
- Counter: the # of documents in the library
- Cost: cost of the document
- (Reference, bestseller): boolean value

# Copies

	 copyID	 commonID	 availability	 userID	 date	 renew
1	1	1	T	0	<null>	F
2	2	1	F	1	2018-05-26	T
3	3	1	T	0	<null>	F

- CopyID: identification number of the copy
- CommonID: identification number of the document
- Availability: boolean value (“T” for available, “F” for not available)
- UserID: identification number of the user who took the copy (default: 0)
- Date: date of checking out of the certain copy
- Renew: Boolean value (“T” for was once renewed, “F” for wasn’t renewed)

# Booking

 bookID	 userID	 priority	 date
--	--	---	--

- BookID: identification number of the booked copy
- UserID: identification number of the user
- Priority: priority number according to the type of the user
- Date: date of latest possibility to check out

# FcukBase.java

```
public class FcukBase implements FcukBaseInterface{
    private static Connection connection = null;
    private static String url = "jdbc:sqlite:databaseTest.sqlite";

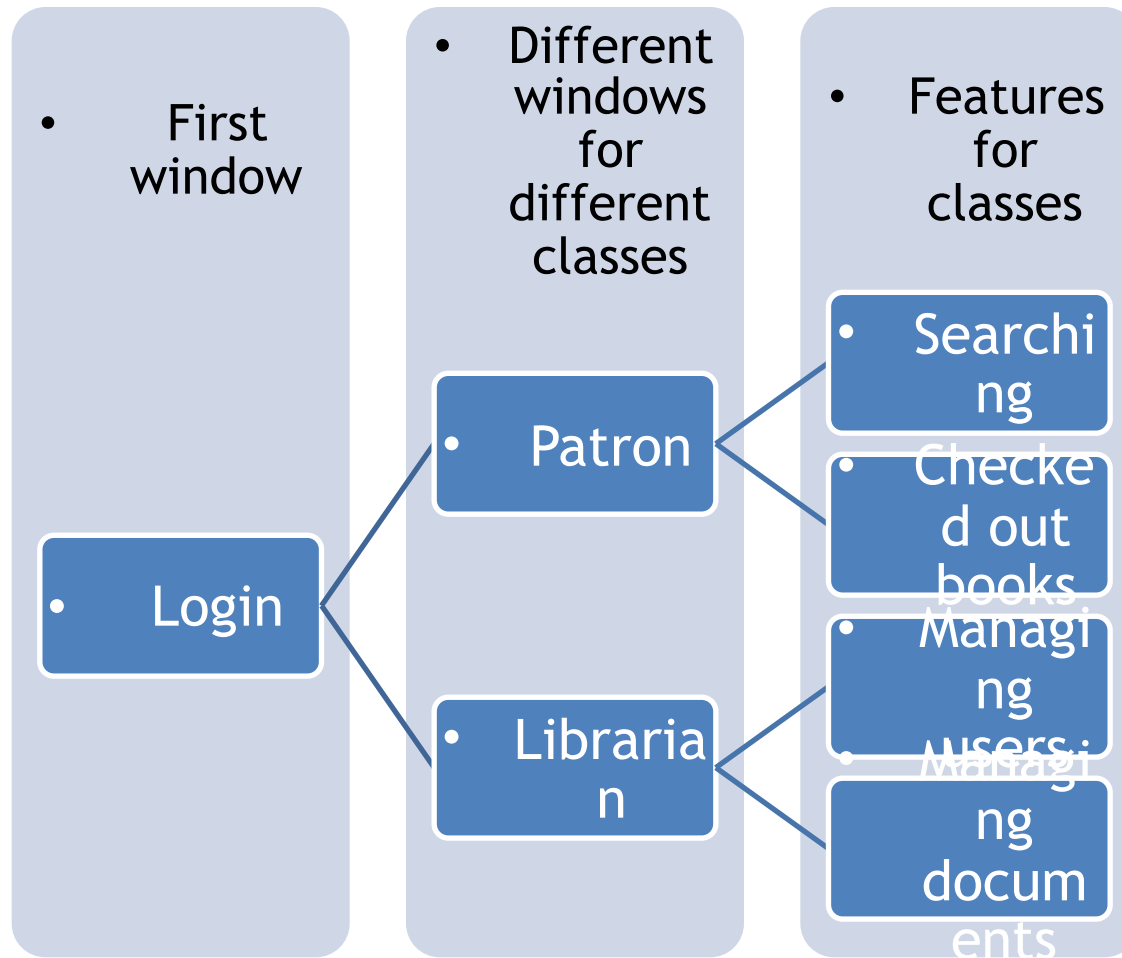
    public FcukBase() {
        if (connection == null) {
            try {
                connection = DriverManager.getConnection(url);
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

- JDBC driver is used to connect to the database. In FcukBase class queries are sent to database to get needed information or to write new information.

# GUI

- Made with JavaFX
- Using Scene Builder

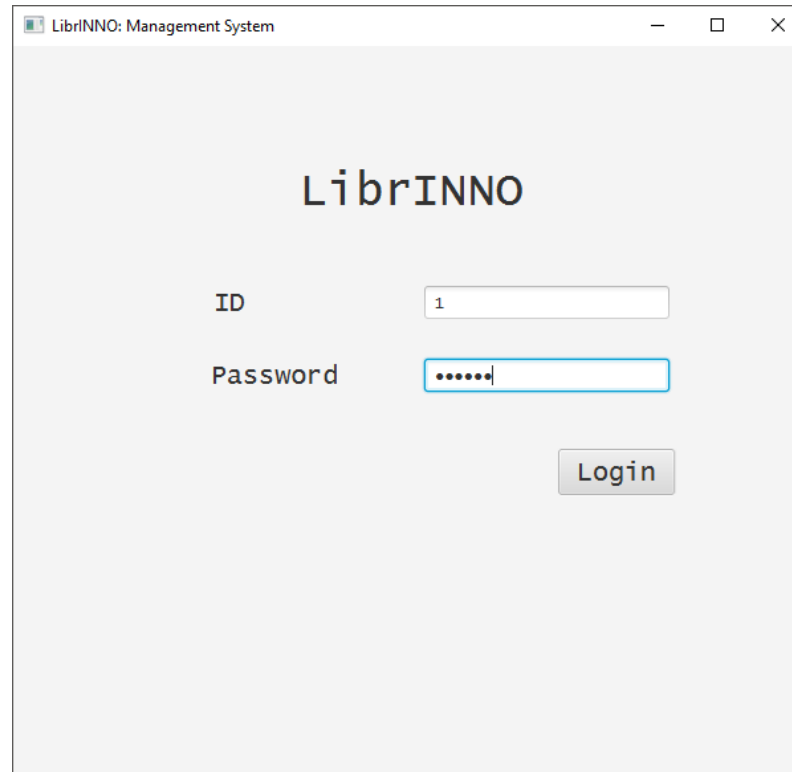
# GUI steps





# Login

- Expecting user ID and password



The screenshot shows a web browser window titled "LibrINNO: Management System". The page has a light gray background. In the center, the text "LibrINNO" is displayed in a large, dark font. Below this, there are two input fields. The first is labeled "ID" and contains the number "1". The second is labeled "Password" and contains six dots, indicating a masked password. To the right of the password field is a "Login" button with a gray gradient and rounded corners.

LibrINNO: Management System

LibrINNO

ID

Password

Login

# Librarian

- Managing users
- Information on selecting from the list

The screenshot shows the 'LibrINNO: Management System' window with the 'Users' tab selected. The interface includes a 'Back' button, a 'User:' input field, and an 'Add new user' button. A table lists four users: Steve Wozniak (ID 1), Steve Something (ID 2), Artur (ID 3), and Oleg (ID 4). At the bottom, there are buttons for 'Pay Fine', 'Bookings', 'Modify', and 'Delete'.

Name	ID
Steve Wozniak	1
Steve Something	2
Artur	3
Oleg	4

This screenshot shows the same 'LibrINNO: Management System' window, but with the details for the selected user, 'Steve Something', displayed on the right side of the interface. The user's details include a phone number, address, and role. The table on the left remains the same, with 'Steve Something' (ID 2) highlighted in blue.

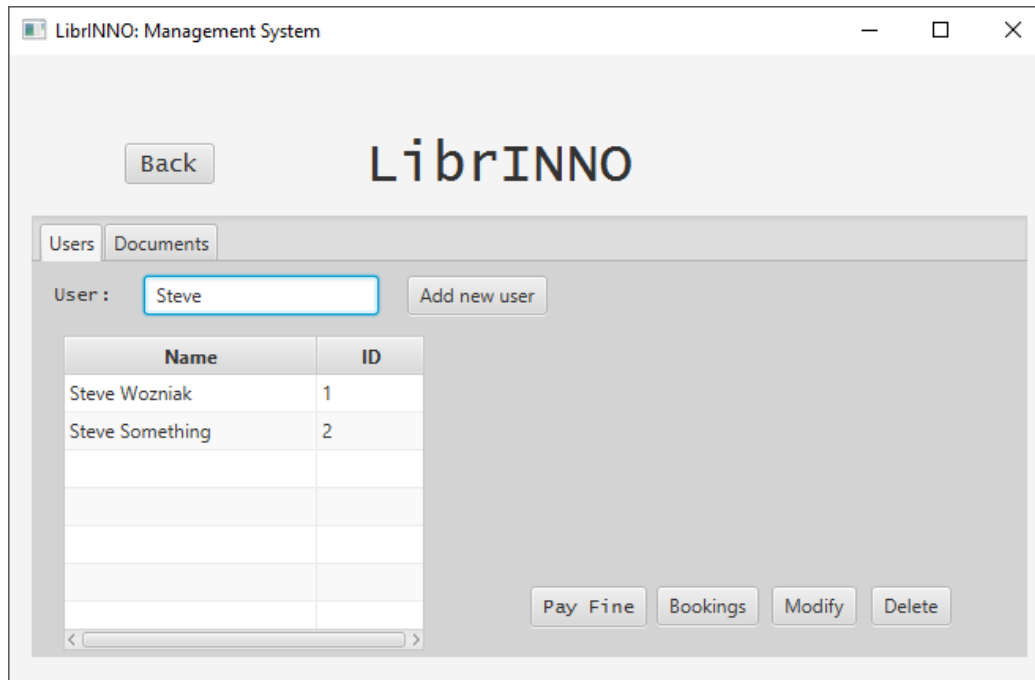
Name	ID
Steve Wozniak	1
Steve Something	2
Artur	3
Oleg	4

**Steve Something**

+789654123  
Ulitsa Street  
FacultyMember

# Librarian

- Searching users by name or ID
- Searching documents by title or author

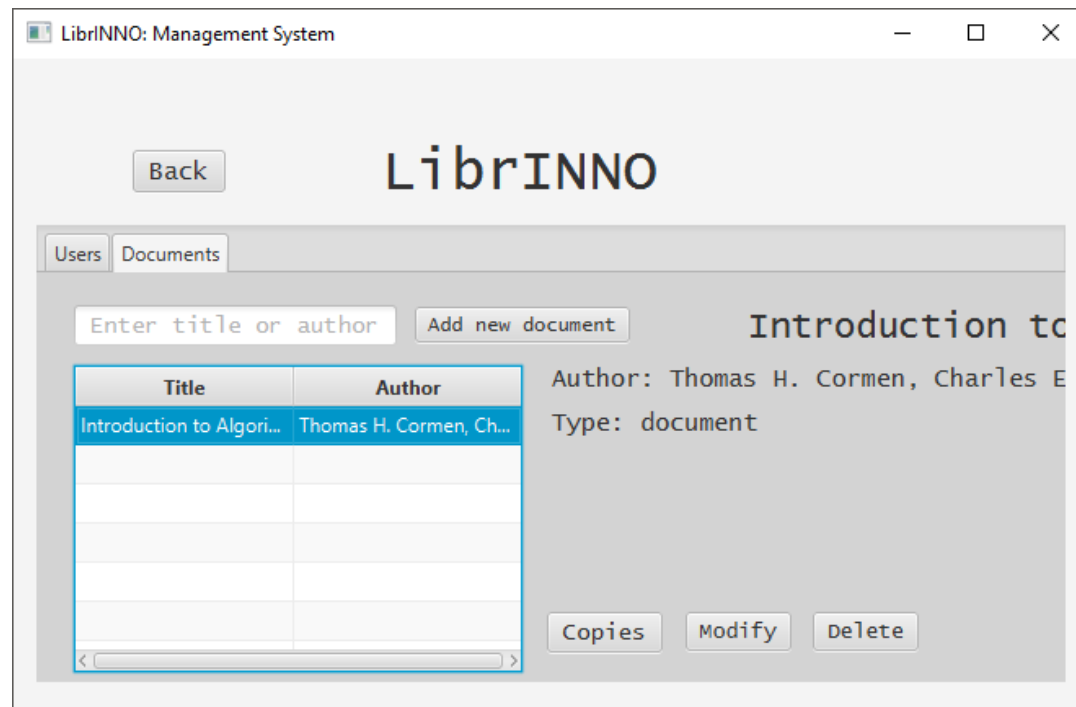


The screenshot displays the LibrINNO Management System interface. At the top, there is a title bar with the text "LibrINNO: Management System" and standard window controls. Below the title bar, a "Back" button is visible on the left, and the "LibrINNO" logo is centered. The main content area features two tabs: "Users" (selected) and "Documents". Under the "Users" tab, there is a search bar labeled "User :" containing the text "Steve", followed by an "Add new user" button. Below the search bar is a table with two columns: "Name" and "ID". The table contains two rows of data: "Steve Wozniak" with ID "1", and "Steve Something" with ID "2". At the bottom right of the interface, there are four buttons: "Pay Fine", "Bookings", "Modify", and "Delete".

Name	ID
Steve Wozniak	1
Steve Something	2

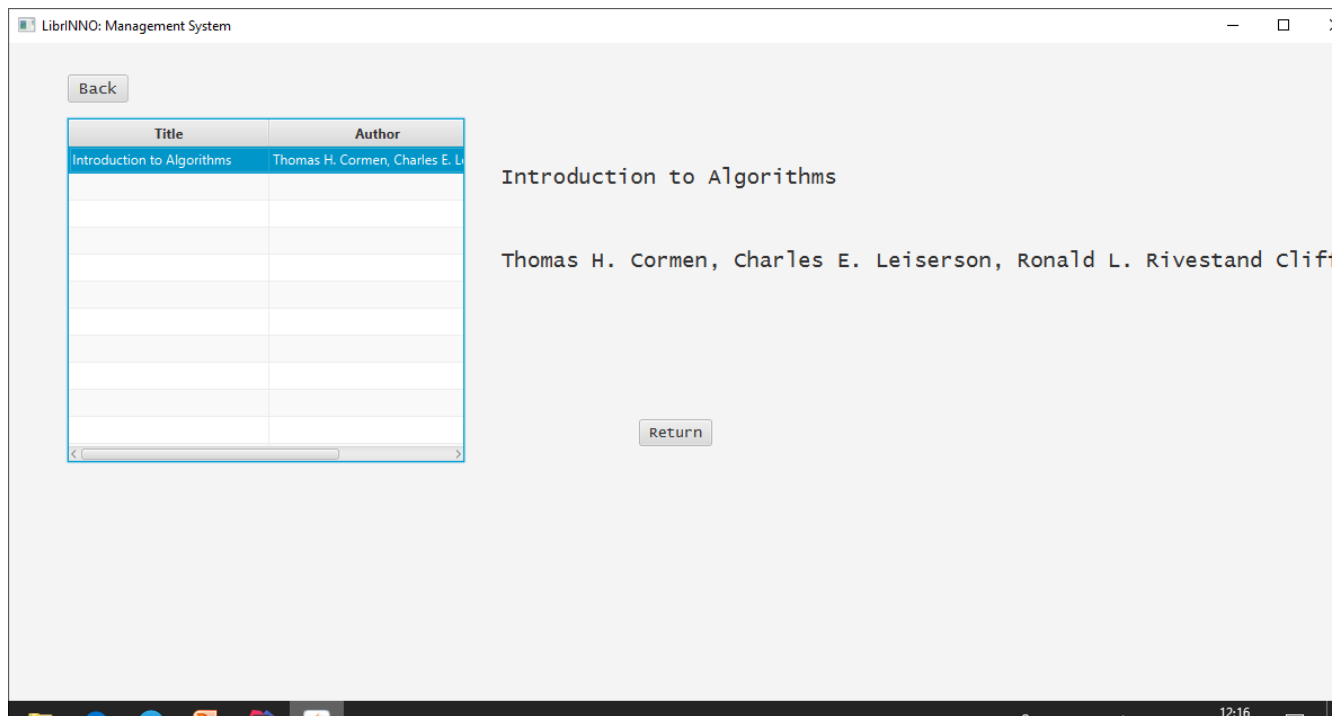
# Librarian

- Managing documents
- Full functional, but developing design



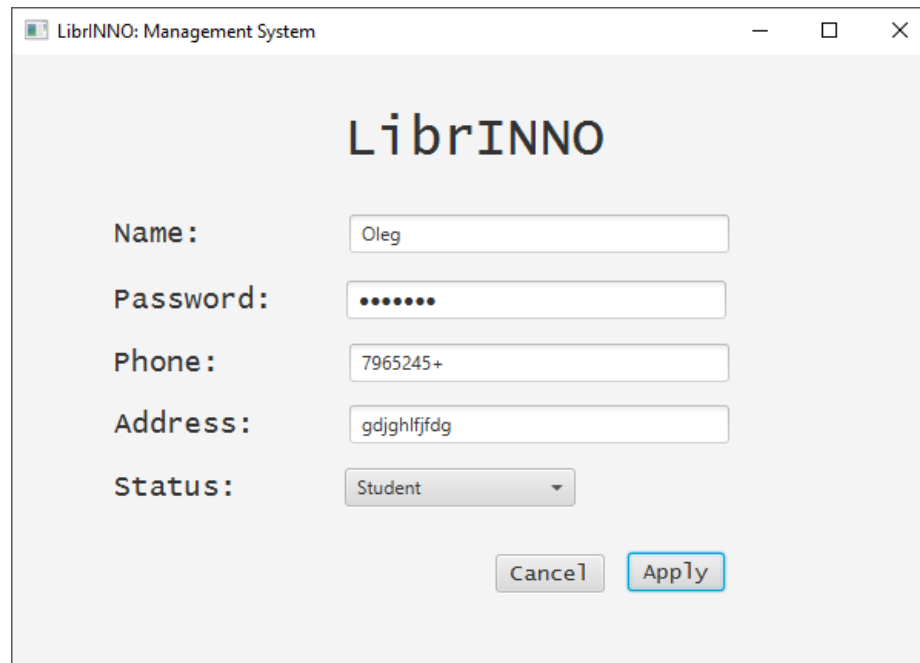
# Librarian

- For particular user you can see checked out documents
- Return only through librarian



# Librarian

- Modify user
- Automatically filled fields
- Empty fields for adding user



The screenshot shows a window titled "LibrINNO: Management System". Inside the window, the "LibrINNO" logo is displayed at the top. Below the logo, there are five labeled input fields: "Name:" with the value "Oleg", "Password:" with masked characters "••••••", "Phone:" with the value "7965245+", "Address:" with the value "gdjghlfjfdg", and "Status:" with a dropdown menu showing "Student". At the bottom right of the form, there are two buttons: "Cancel" and "Apply".

Field	Value
Name:	Oleg
Password:	••••••
Phone:	7965245+
Address:	gdjghlfjfdg
Status:	Student

# Librarian

- Through librarian you can pay fine
- Press “Pay Fine”, enter amount and press OK

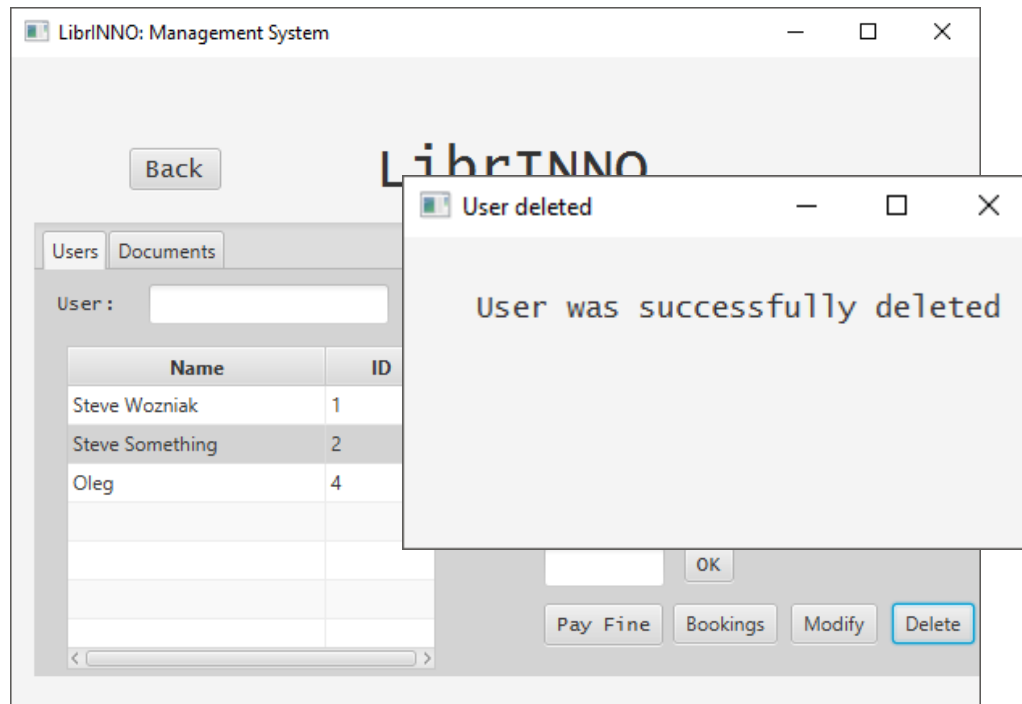
The screenshot shows the 'LibrINNO: Management System' window. At the top, there is a 'Back' button and the title 'LibrINNO'. Below this, there are two tabs: 'Users' and 'Documents'. The 'Users' tab is active, displaying a table of users and a form for user details.

Name	ID
Steve Wozniak	1
Steve Something	2
Artur	3
Oleg	4

Below the table, there is a form for user details. It includes a 'User:' label, an input field, and an 'Add new user' button. To the right of the input field, the name 'Oleg' is displayed. Below the name, there is a text field containing '7965245+' and another text field containing 'gdjghlfjfdg'. Below these fields, the role 'Student' is displayed. At the bottom of the form, there is an 'OK' button and a 'Pay Fine' button (highlighted with a blue border). To the right of the 'Pay Fine' button, there are three more buttons: 'Bookings', 'Modify', and 'Delete'.

# Librarian

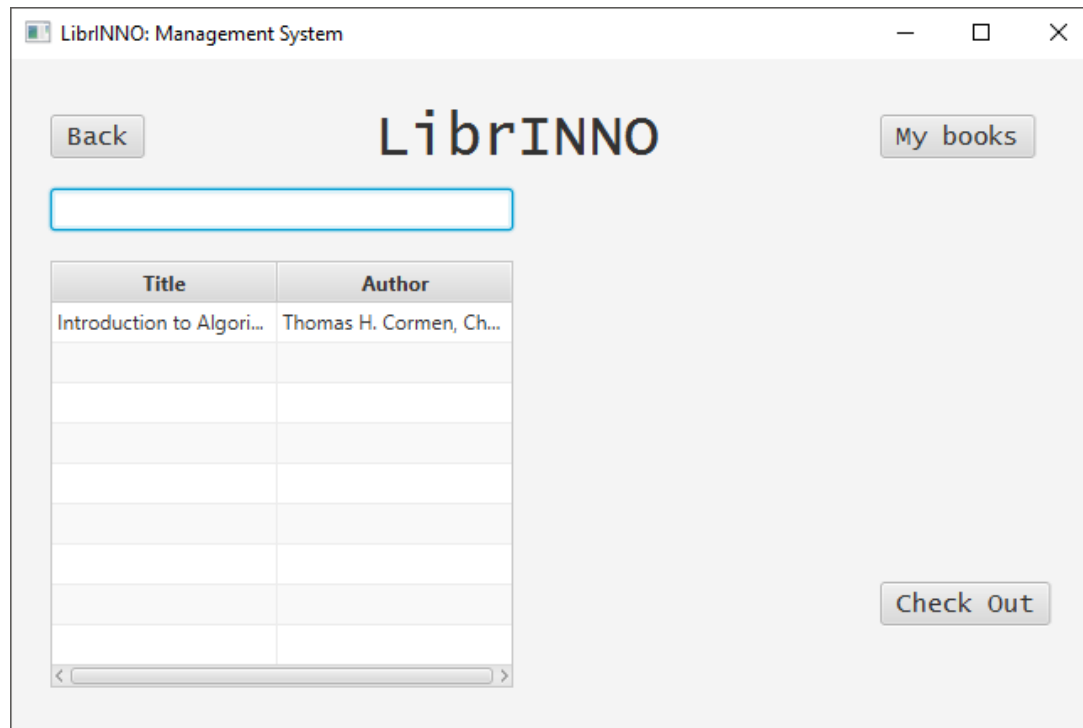
- Deleting user
- Immediately deleted from the list





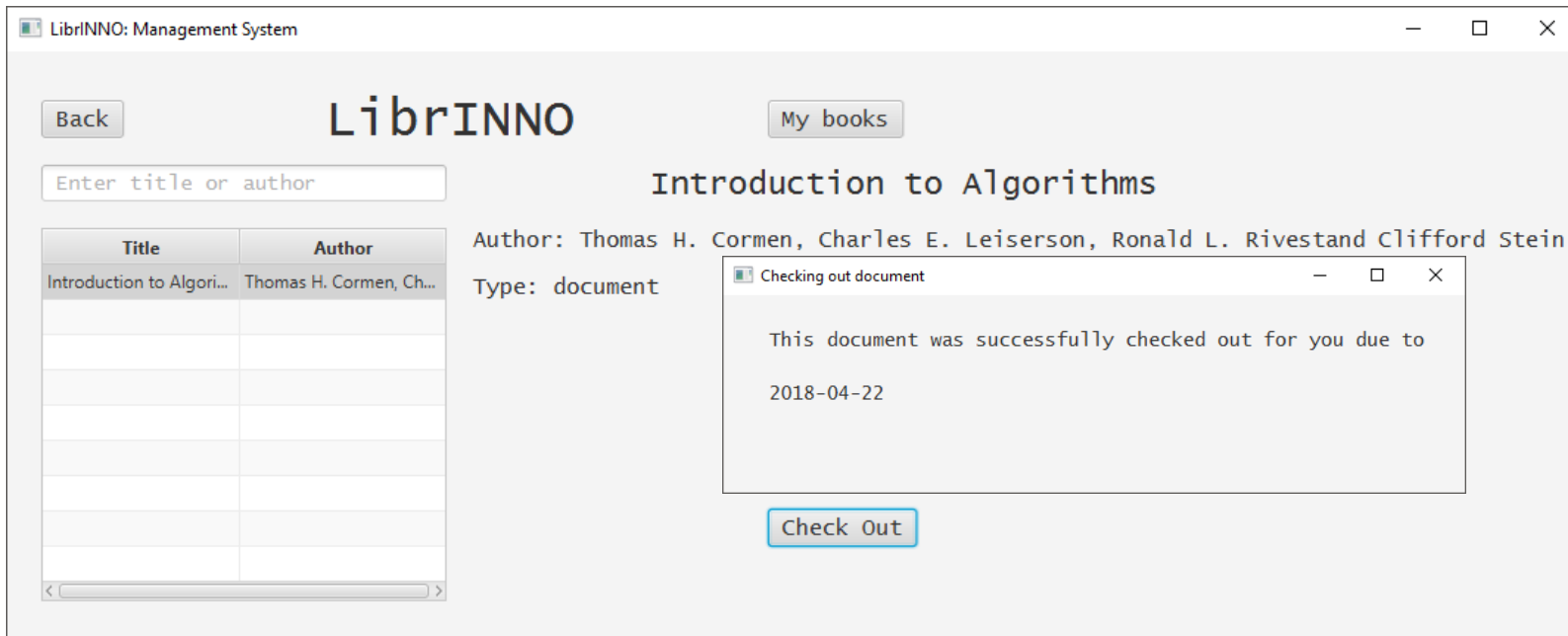
# Patron

- Searching books



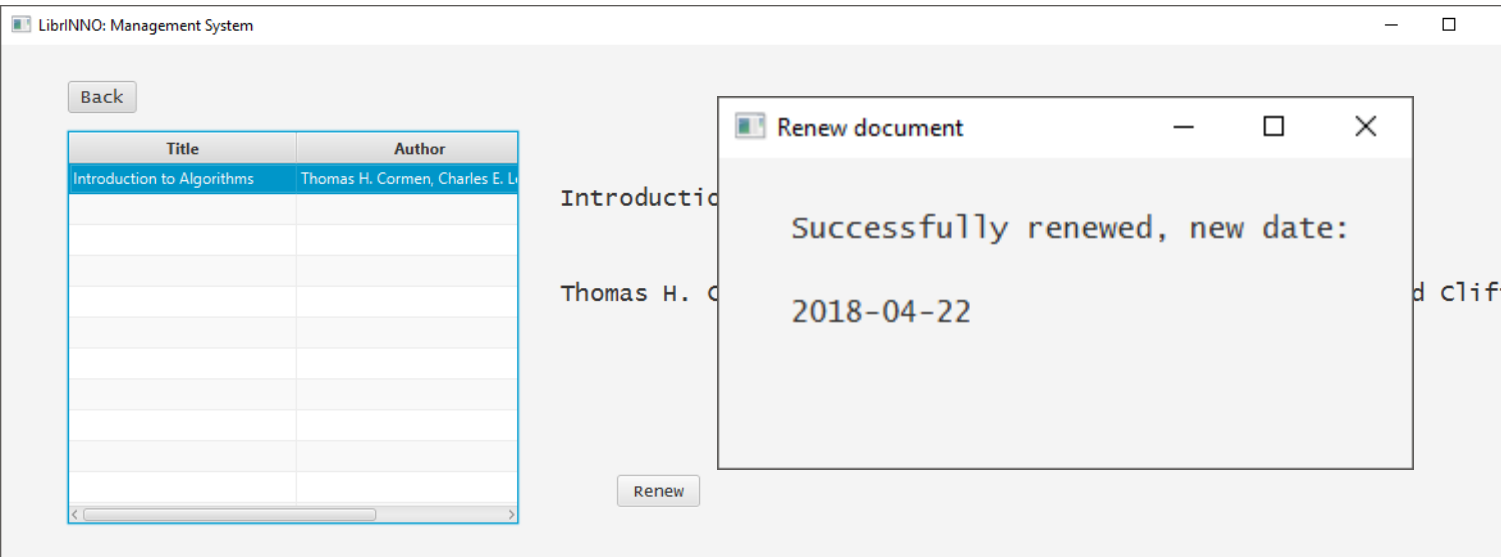
# Patron

- Check out document and informing about date (if it is available now)



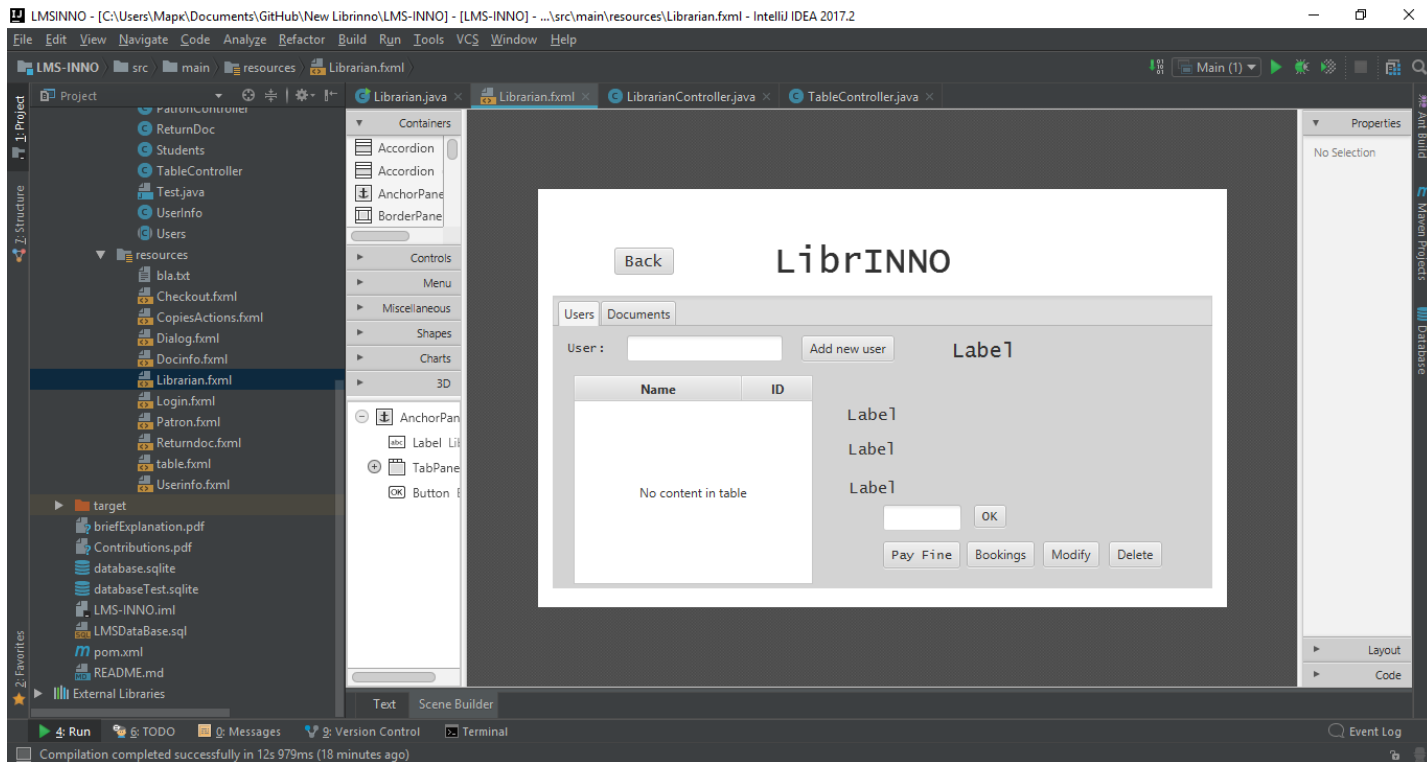
# Patron

- List of checked out books
- Feature to renew selected one



# GUI development

- Working with Scene Builder
- Creating .fxml files for every window



# GUI development

- For each .fxml creating .java controller to interact with elements and actions

