

**FACULTATEA: Automatică și Calculatoare  
SPECIALIZAREA: Calculatoare Engleză  
DISCIPLINA: Programare Orientata Pe Obiect  
PROIECT: Online Book Shop Application**

**Îndrumător laborator: Studenți:**

**As. Aron Baka-Balint Pîșu Cristina**

**Rățoi Răzvan-Valeriu**

**Gr. 30421**

**Short description**

This application simulates an online book shop, where the user can browse through the products, mark some as favorites, add to cart and finally buy everything they liked. The application also has some features where you can filter the selection of the books presented by the category, or search some specific ones by the author or title. Before the user can access the application, they need to log in with their account or, in the case where they do not have one, they can sign up and create their account

**Some Use-Cases**

A. Buying products

Instructions:

1. Log in with your account.

2. In the products section of your menu bar, select the desired category or all products.

3. Add to cart all the books that suit your wishes.

4. Press the “Go to cart” button either from the product window or from the main page.

4. While in the cart, press the “Buy all” button and you are all set!

B. Searching for a specific book

Instructions:

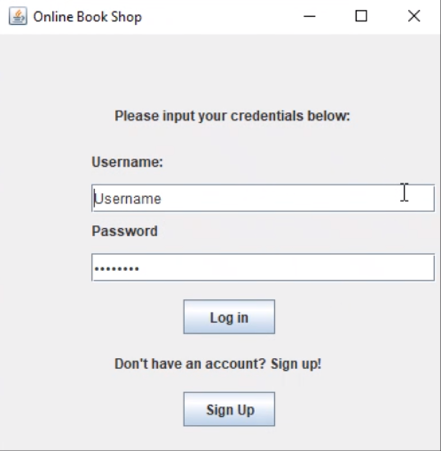
1. While on the main page, select the “Search” item in the menu bar.

2. In the text field, write either the author or the title of that book.

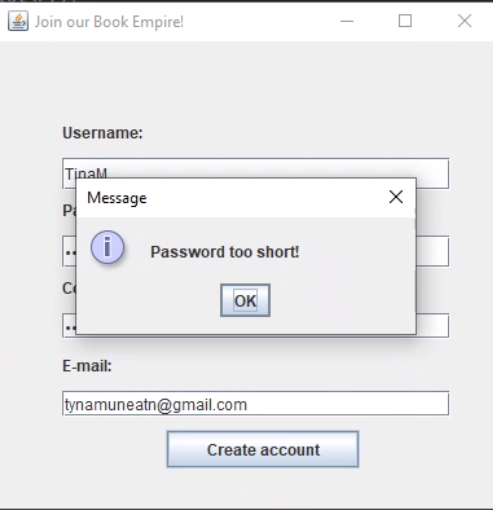
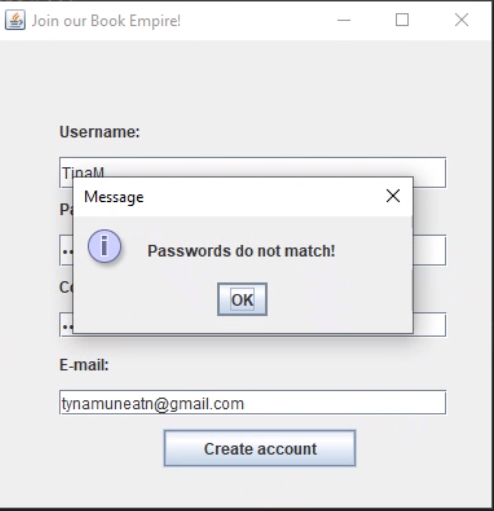
3. Press the “Find” button.

4. A new product page will appear with every book similar to the input.

**The User Interface of the Application**

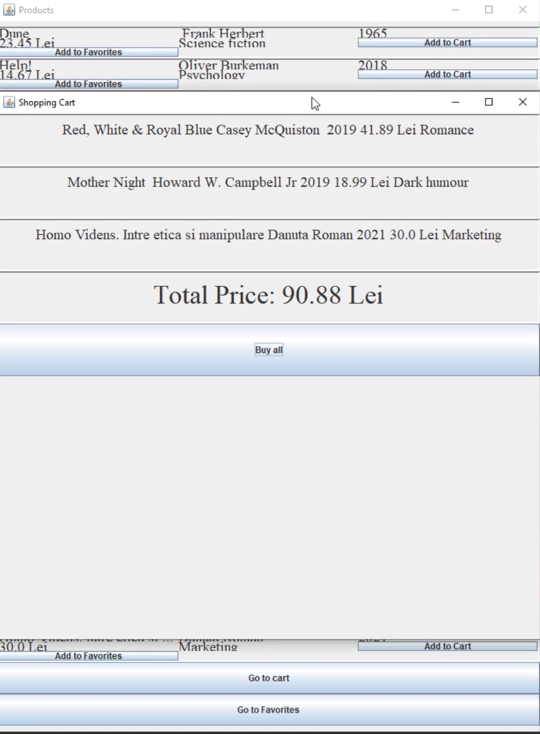
****

The Log In page, in which you can either enter your username and password, or you can sign up and create one.



The error pop-up if the password The error pop-up if the 2 passwords

has less than 8 characters. are not the same.



The Cart page, where you can see the selected books and the option to buy them.

Behind this page the Product page can be seen, displaying the title, author, year of publication, category and price of the title. At the bottom of this page there are 2 buttons: “Go to Cart” and “Go to Favorites”, which they each will open a new window, displaying the books added to the Cart or the ones marked as favorites.

**Solution presentation**

For creating this application, we created several classes that extend JFrame, each of these being used to display a window of some sort. Some of them are PageView – displaying the main page of the application -, ProductView – displaying the page of the products -, CartView – displaying the page of the selected books for purchasing- and so on. These classes are “linked” between each other either directly or through other classes. For example, to get to the Cart, one of the paths the user may go through is Log In 🡪 Main page 🡪 Products 🡪 Cart. Also for the Cart page we used a Singleton Pattern, in order to keep the information in the page while the user is browsing other pages.

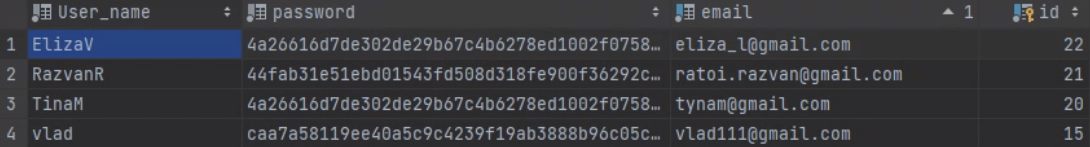
We also used a Book class, in which we kept the model of the book, with its attributes, getters, setters, with which we were able to create ArrayLists to pass the information about the books from class to class.

Finally, we created a BookRepo class, which was used to retrieve data about the books from the storage and return them to the classes that needed to show them.

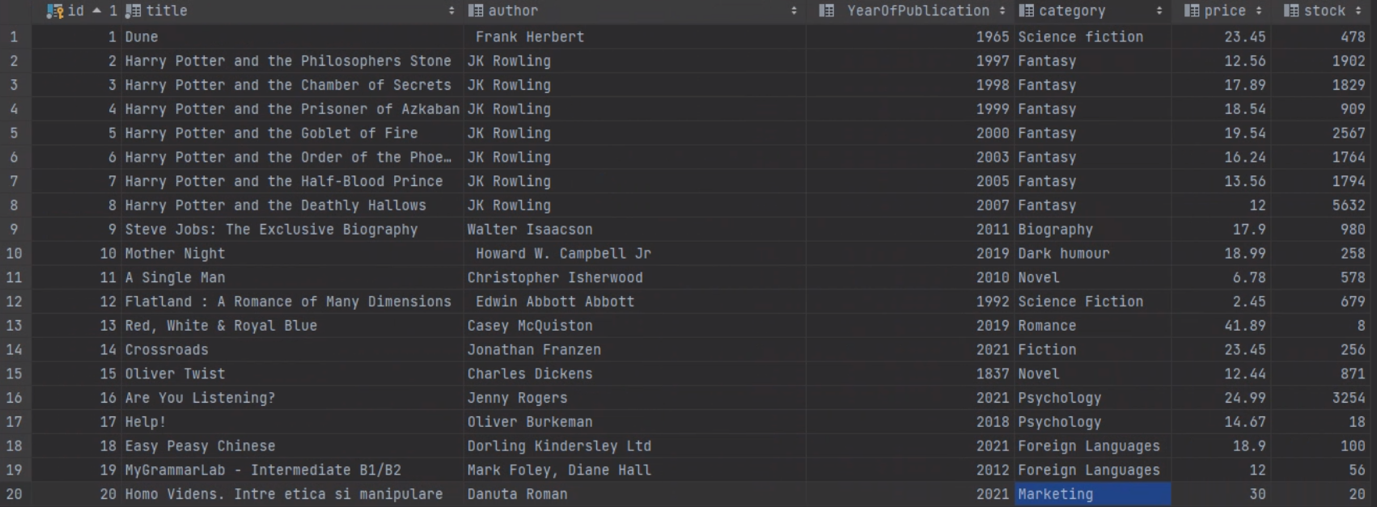
This “storage” of the books we used is represented by a database, which had 2 tables, a “Book” one and an “Users” one. We retrieved the necessary data from the tables using queries (in PostgreSQL) and also to store data into tables. The retrieval of the data was called when the user wanted to see all the products, the products filtered by a certain category or when they searched for the title or author. The storing of the data is represented by the signing up of the user, where the data introduced by them would be placed in the Users table, but not raw. For the password of the users, we firstly encrypted them, then they would be placed in the table.

For the encryption of the passwords, we used a certain class, in which we had the hash functions that returned the passwords in their new form. When the user would log in, the introduced password would be encrypted too and then compared to everything in the Users table.

The next images will show the tables in the database that was connected to the application:



This table represents the users, having stored the username, the encrypted password and the e-mails.



This one represents the books in the shop, showing the title, author, year of publication, category, price and stock. The selection from this table was used with different queries based on the preference of the user.

**Further improvements**

For the improvement of this application, the first thing we would like to do is restructuring it into a cleaner pattern, and separate the executive commands that are done between more classes from the ones that are only used for the respective class (such as JFrame instructions to sculpt the window). Another improvement would be to display whether the selected book is in or out of stock, based on the connections to the tables from the database, and even to simulate the delivery of the order, for which would be used an address of the user. Some final improvements would be represented by the addition of more aesthetic elements to make the application more appealing to the eye, such as photos of the books in the product page, icons assigned to different buttons (“Find” button, “Go to cart”).