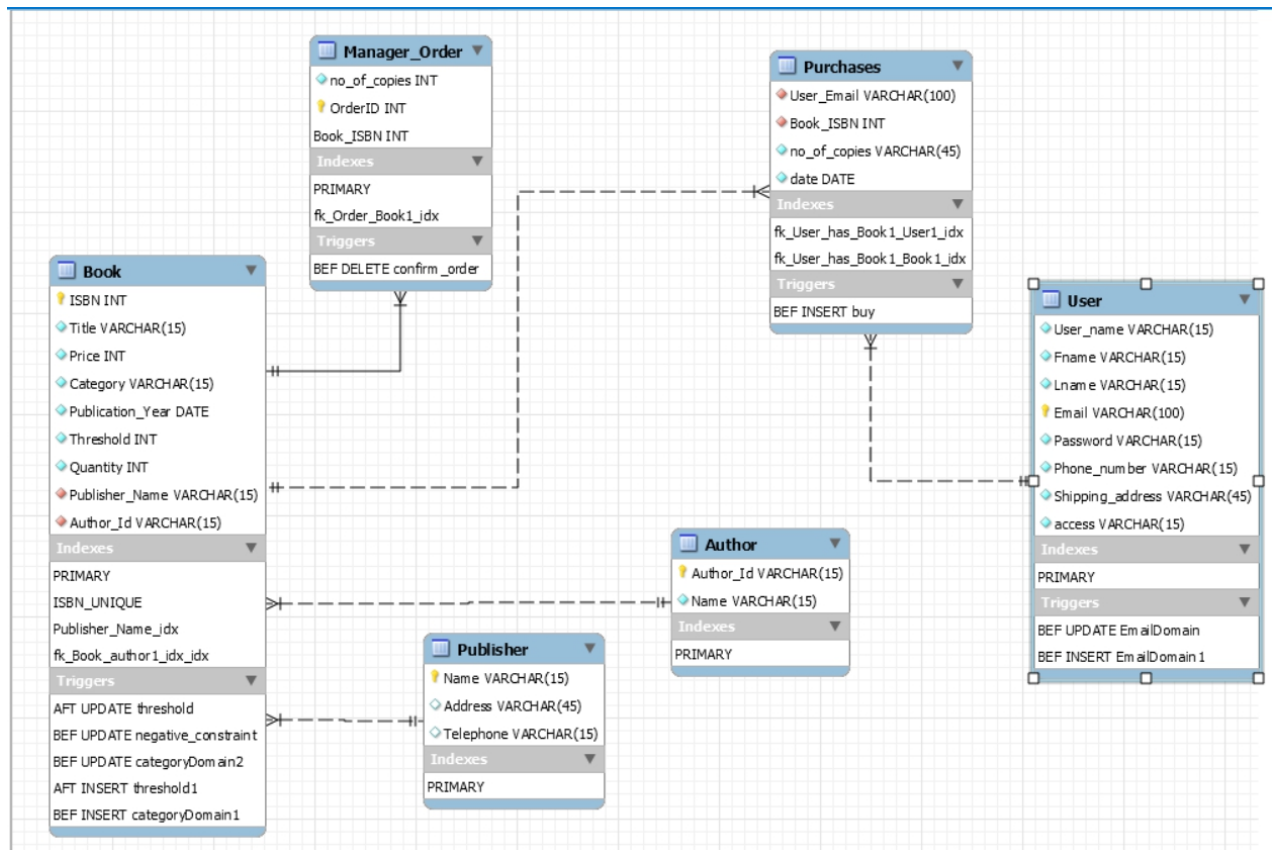


Project
Order Processing System

Members:

- أحمد حمدى ابراهيم رضوان -5
عبد الرحمن ابراهيم على ابراهيم هجرس -35
عبد الرحمن محمود كمال محمود نور -39
يوسف رءوف وديع تادرس صدقى -72

❖ The database ERD diagram:



❖ Analysis of triggers:

- 1- Checks Before updating the quantity of book is still positive so that if customer try to buy a greater quantity than the available the new quantity would be negative which isn't allowed.

```
-- checks negativity of book count
DELIMITER $$
CREATE TRIGGER negative_constraint BEFORE update ON Book
FOR EACH ROW BEGIN
    IF (NEW.Quantity < 0) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'there is no enough books remaining';
    END IF;
END$$
DELIMITER ;
```

2- Checks After updating the book quantity if it less than the threshold and then make an order for more of this book to reach the threshold quantity.

```
DELIMITER $$
-- checks if quantity is less than a threshold if it is then make an order for this book with a count = threshold
CREATE TRIGGER threshold AFTER update ON Book
FOR EACH ROW BEGIN
    IF (NEW.Quantity < OLD.Threshold) THEN
        insert into book_store.Manager_Order (no_of_copies , Book_ISBN)
        values(OLD.Threshold , OLD.ISBN);
    END IF;
END$$
DELIMITER ;
```

3- Checks After inserting if quantity is less than a threshold if it is then make an order for this book with a count = threshold.

```
DELIMITER $$
-- checks if quantity is less than a threshold if it is then make an order for this book with a count = threshold
CREATE TRIGGER threshold1 AFTER insert ON Book
FOR EACH ROW BEGIN
    IF (NEW.Quantity < new.Threshold) THEN
        insert into book_store.Manager_Order (no_of_copies , Book_ISBN)
        values(NEW.Threshold, new.ISBN);
    END IF;
END$$
DELIMITER ;
```

4- Checks before inserting if the category of the book being inserted is from our list of categories.

```
DELIMITER $$
-- checks if the category of the book being inserted is from our list of categories
CREATE TRIGGER categoryDomain1 BEFORE insert ON Book
FOR EACH ROW BEGIN
    IF (NEW.Category != 'Science' and NEW.Category != 'Art' and NEW.Category != 'Religion' and
        NEW.Category != 'History' and NEW.Category != 'Geography') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'only categories of Science,Art,Religion,History,Geography';
    END IF;
END$$
DELIMITER ;
```

5- checks before update if the category of the book being updated is from our list of categories.

```
DELIMITER $$
-- checks if the category of the book being updated is from our list of categories
CREATE TRIGGER categoryDomain2 BEFORE UPDATE ON Book
) FOR EACH ROW BEGIN
    IF (NEW.Category != 'Science' and NEW.Category != 'Art' and NEW.Category != 'Religion' and
        NEW.Category != 'History' and NEW.Category != 'Geography') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'only categories of Science,Art,Religion,History,Geography';
    END IF;
END$$
DELIMITER ;
```

6- Decrease the quantity of the book when buying.

```
DELIMITER $$
-- decrease the quantity of the book when buying
CREATE TRIGGER buy BEFORE insert ON Purchases
) FOR EACH ROW BEGIN
    update Book set Quantity = Book.Quantity - NEW.no_of_copies
    where ISBN = NEW.Book_ISBN;
END$$
DELIMITER ;
```

7- Checks if the email in the required shape before updating.

```
CREATE TRIGGER EmailDomain BEFORE UPDATE ON User

) FOR EACH ROW BEGIN
    IF (NEW.Email NOT LIKE '%_@_%._%' ) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'not valid email';
    END IF;
END$$
DELIMITER ;
```

8- Checks if the email in the required shape before inserting a new user.

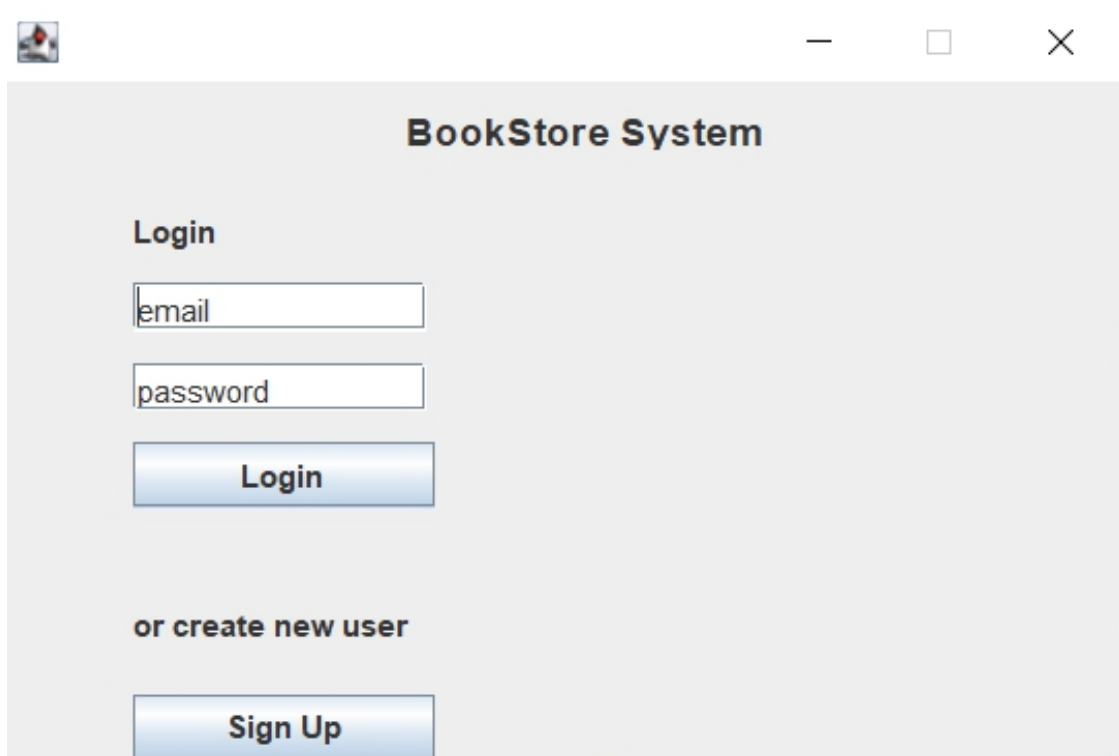
- `CREATE TRIGGER EmailDomain1 BEFORE insert ON User`

```
FOR EACH ROW BEGIN
  IF (NEW.Email NOT LIKE '%_@%._%' ) THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'not valid email';
  END IF;
END$$
DELIMITER ;
```

9- Before deleting the manager order do the update on the book table with the new book quantity.

```
DELIMITER $$
CREATE TRIGGER confirm_order BEFORE DELETE ON Manager_Order
) FOR EACH ROW BEGIN
  update Book set Quantity = OLD.no_of_copies + Quantity
  where ISBN = OLD.Book_ISBN;
- END$$
DELIMITER ;
```

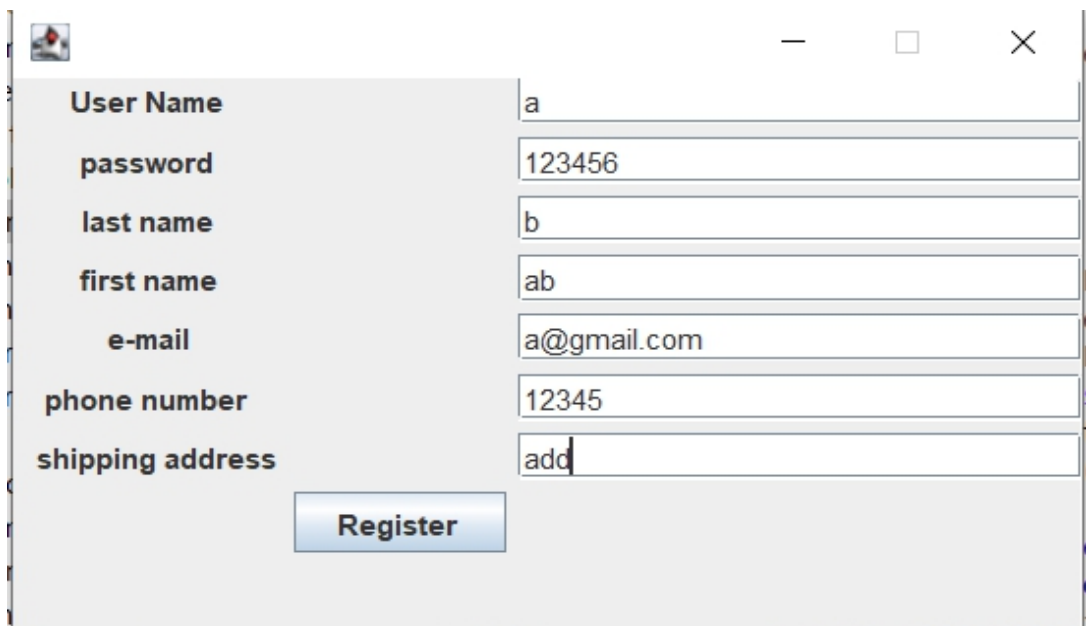
❖ Program Flow



A screenshot of a web application window titled "BookStore System". The window has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area is light gray and contains a "Login" section with two input fields labeled "email" and "password", followed by a blue "Login" button. Below this is the text "or create new user" and a blue "Sign Up" button.

- Main screen has two options sign up to create a new user or log in if user has already an account if user press signup

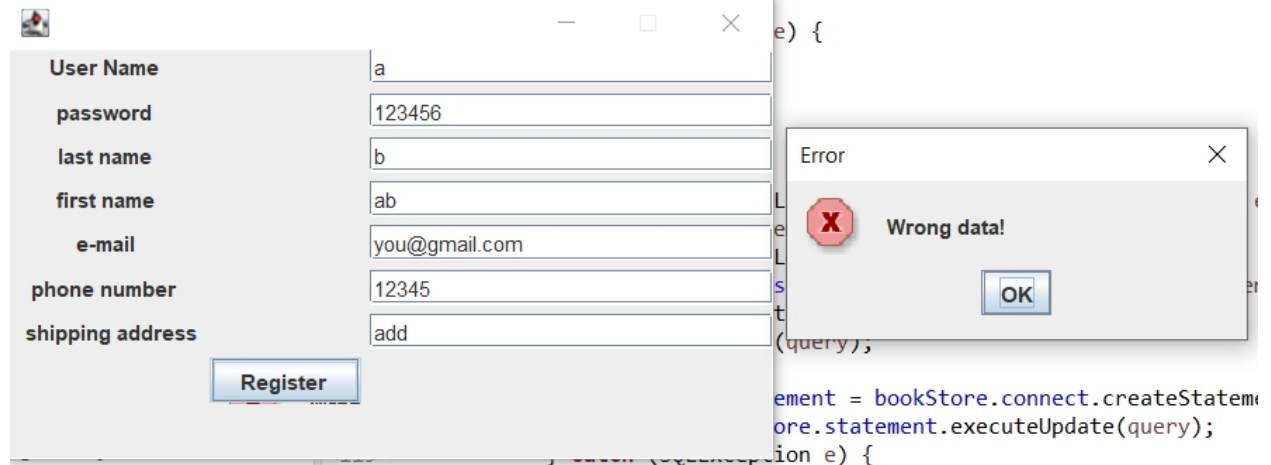
This screen shows.



A screenshot of a web application window titled "BookStore System" showing the registration form. The form has a light gray background and a white border. It contains several input fields with labels on the left: "User Name" (value: a), "password" (value: 123456), "last name" (value: b), "first name" (value: ab), "e-mail" (value: a@gmail.com), "phone number" (value: 12345), and "shipping address" (value: add). A blue "Register" button is located at the bottom right of the form.

- User has to enter his personal information and choose an email which is unique and press register.

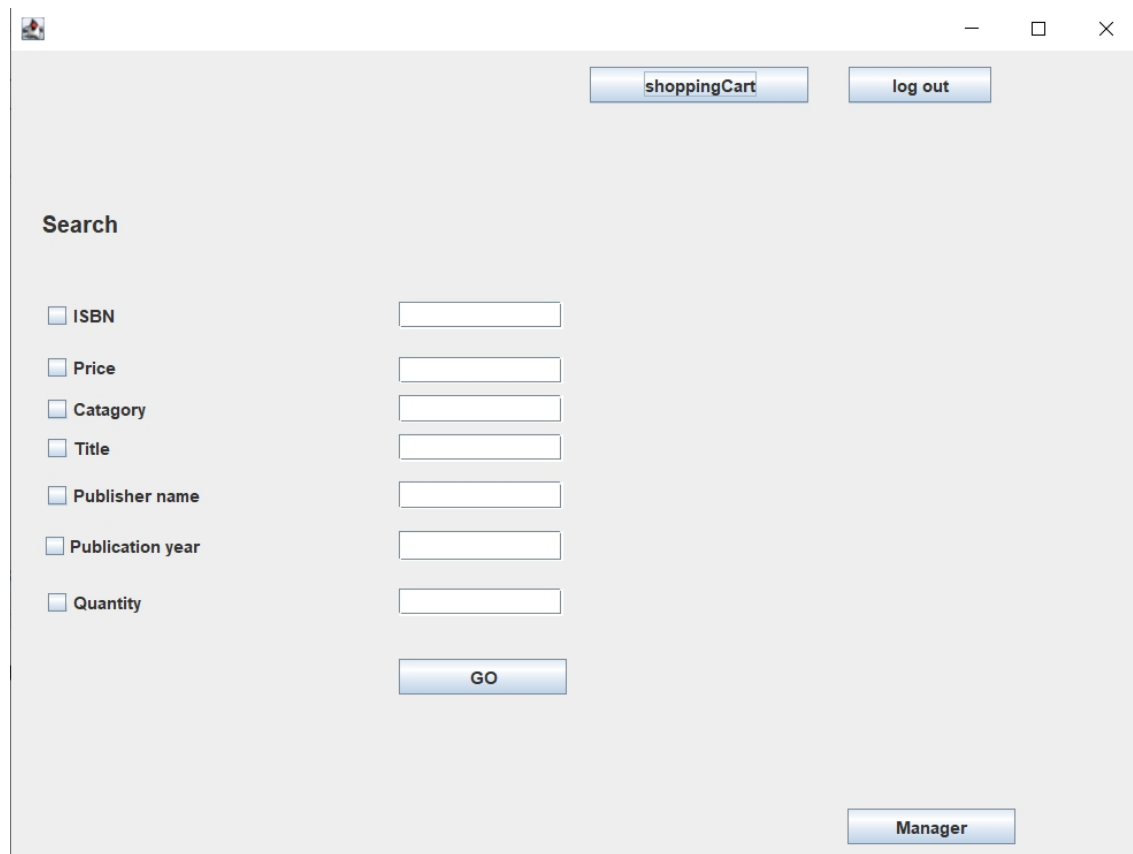
If the email is already taken this message shows.



The image shows a registration form with the following fields: User Name (a), password (123456), last name (b), first name (ab), e-mail (you@gmail.com), phone number (12345), and shipping address (add). A 'Register' button is at the bottom. An 'Error' dialog box is overlaid on the form, displaying a red 'X' icon and the text 'Wrong data!' with an 'OK' button. In the background, some code is visible:

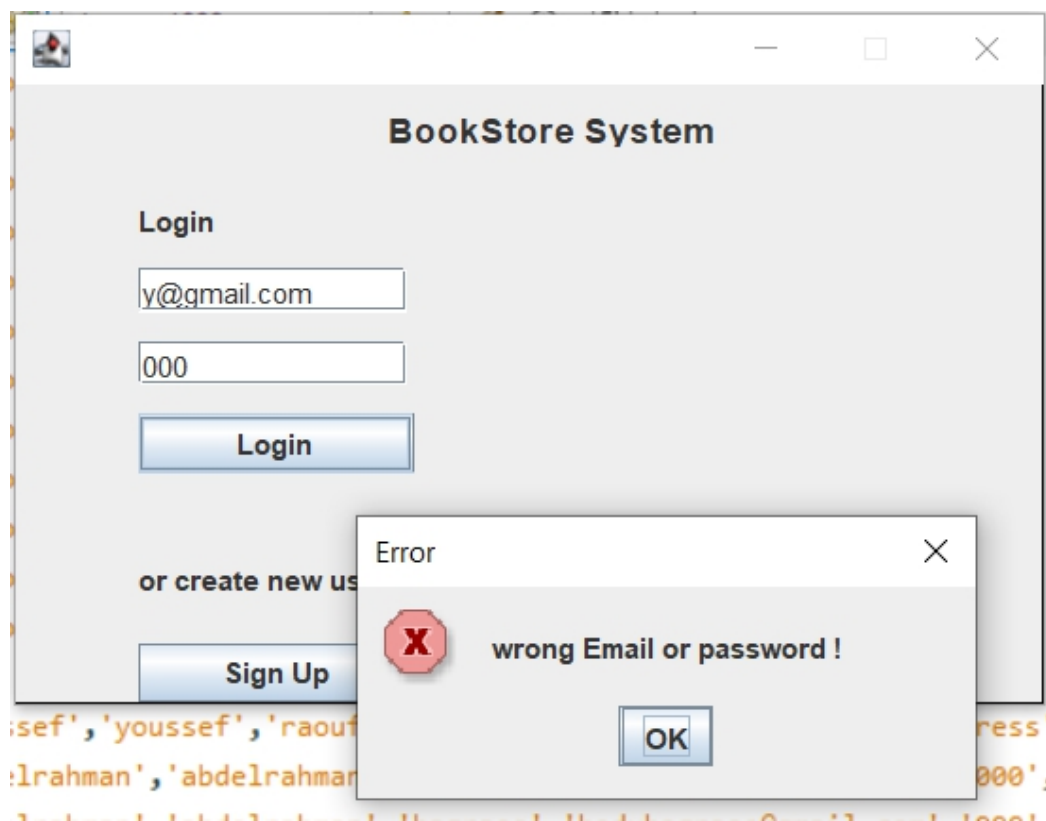
```
e) {  
L  
e  
L  
S  
t  
(query);  
ement = bookstore.connect.createStatement;  
ore.statement.executeUpdate(query);  
ion e) {
```

- If email not taken then user successfully enters his account.



The image shows a web application interface. At the top right, there are 'shoppingCart' and 'log out' buttons. Below these is a 'Search' section with a list of search criteria: ISBN, Price, Catagory, Title, Publisher name, Publication year, and Quantity. Each criterion has a checkbox and a corresponding text input field. A 'GO' button is located below the input fields. At the bottom right, there is a 'Manager' button.

- If user try to log in with wrong information this message shows



- If user successfully enters his account, then he can search for books with ISBN, Titles, price, Category, ... etc and press Go to see the matched books with his search.

Search

☐ ISBN

☐ Price

☒ Catagory

☐ Title

☐ Publisher name

☐ Publication year

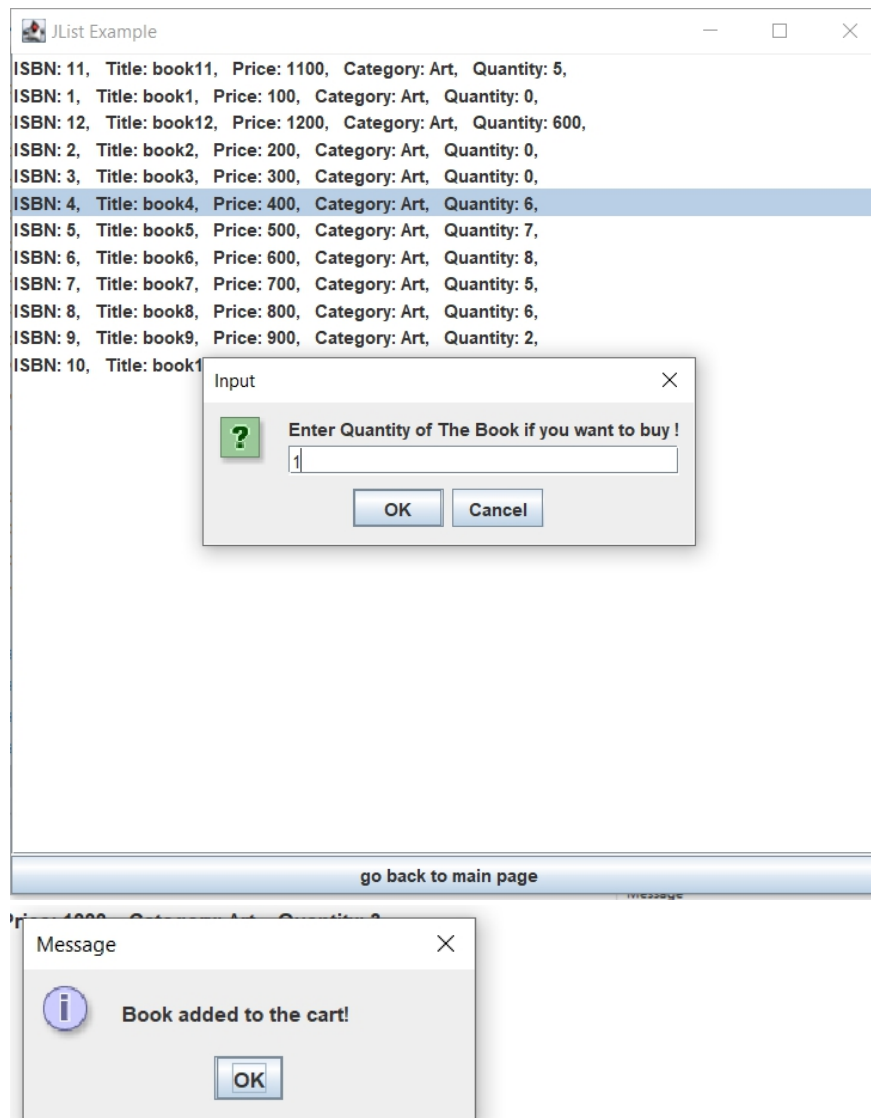
☐ Quantity

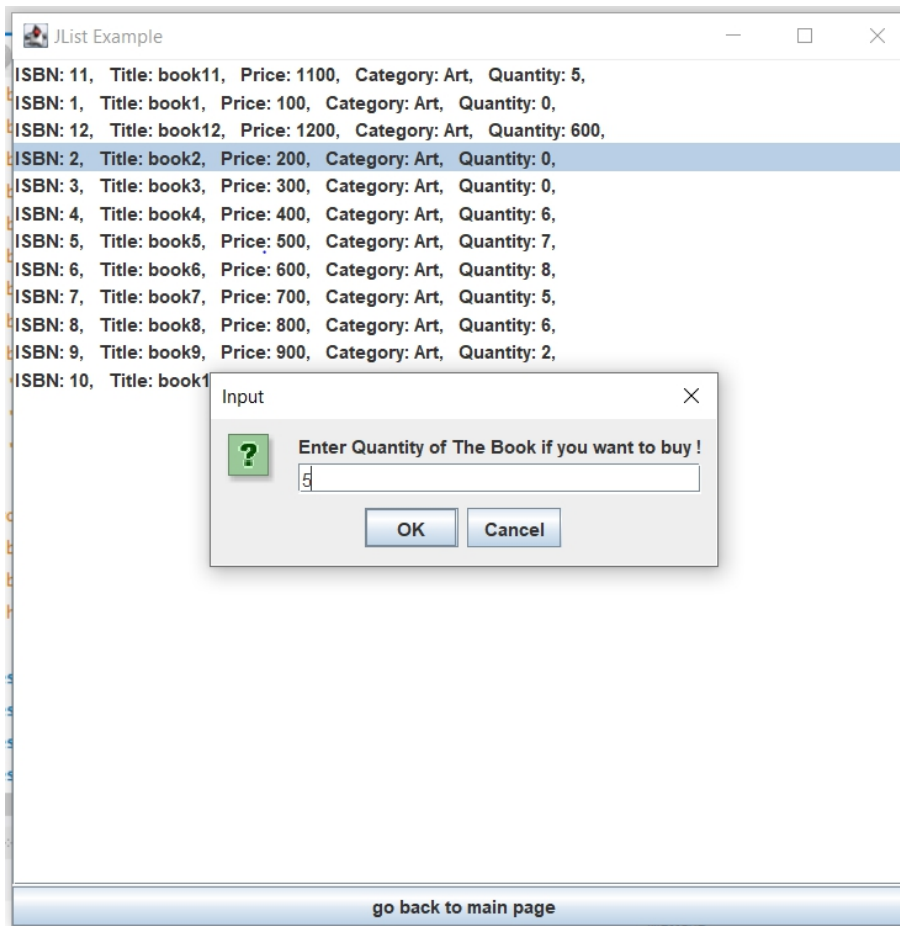
JList Example

ISBN: 11, Title: book11, Price: 1100, Category: Art, Quantity: 5,
ISBN: 1, Title: book1, Price: 100, Category: Art, Quantity: 0,
ISBN: 12, Title: book12, Price: 1200, Category: Art, Quantity: 600,
ISBN: 2, Title: book2, Price: 200, Category: Art, Quantity: 0,
ISBN: 3, Title: book3, Price: 300, Category: Art, Quantity: 0,
ISBN: 4, Title: book4, Price: 400, Category: Art, Quantity: 6,
ISBN: 5, Title: book5, Price: 500, Category: Art, Quantity: 7,
ISBN: 6, Title: book6, Price: 600, Category: Art, Quantity: 8,
ISBN: 7, Title: book7, Price: 700, Category: Art, Quantity: 5,
ISBN: 8, Title: book8, Price: 800, Category: Art, Quantity: 6,
ISBN: 9, Title: book9, Price: 900, Category: Art, Quantity: 2,
ISBN: 10, Title: book10, Price: 1000, Category: Art, Quantity: 3,

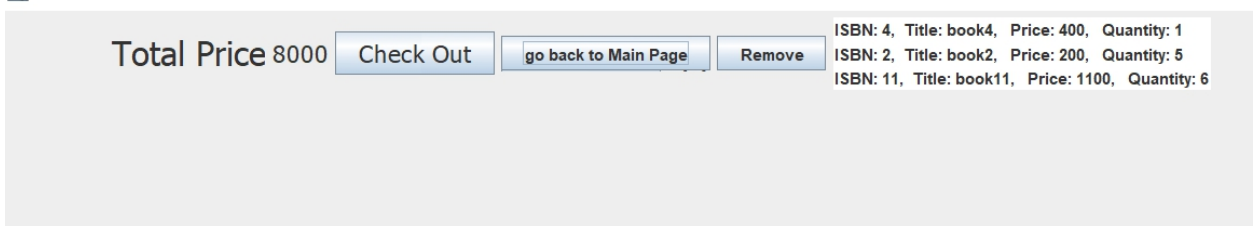
go back to main page

- If user want to buy any book just press it and enter the quantity he wants and dynamically his order added to his cart.

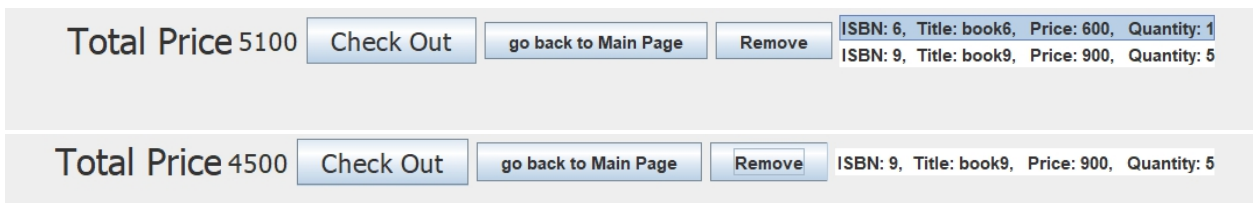




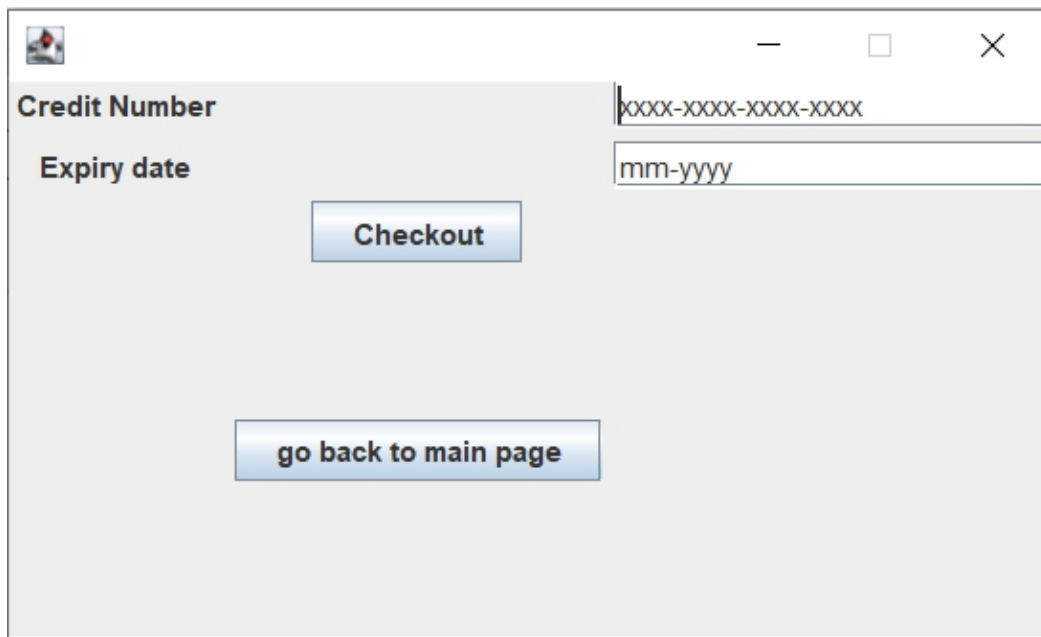
- After choosing the required books press go back button and press the shopping cart button to find your order with required money to pay.



- User can remove from his order by select the book and press remove.



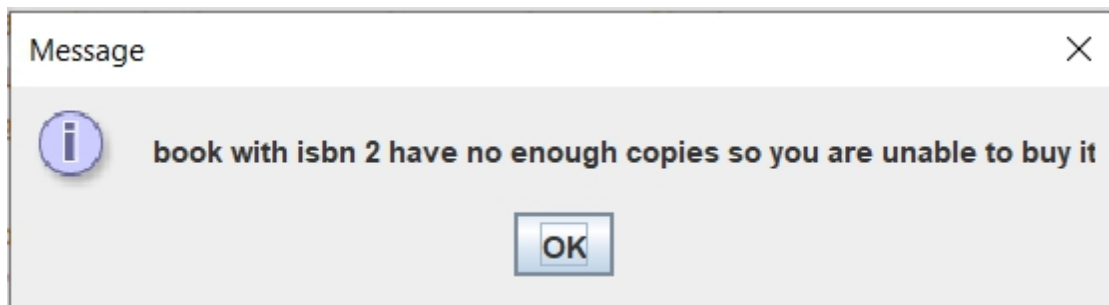
- Or press checkout to confirm his order



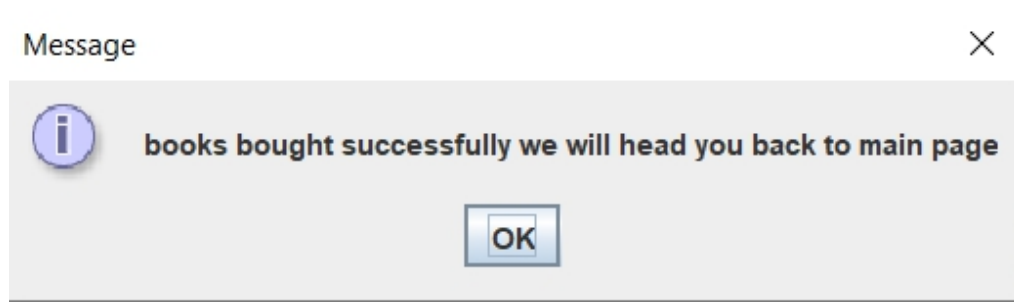
A screenshot of a checkout form window. The window has a title bar with a small icon, a minus sign, a maximize button, and a close button. The form contains two input fields: "Credit Number" with a placeholder "XXXX-XXXX-XXXX-XXXX" and "Expiry date" with a placeholder "mm-yyyy". Below these fields is a "Checkout" button. At the bottom of the form is a "go back to main page" button.

- User require to enter his credit information and press checkout.

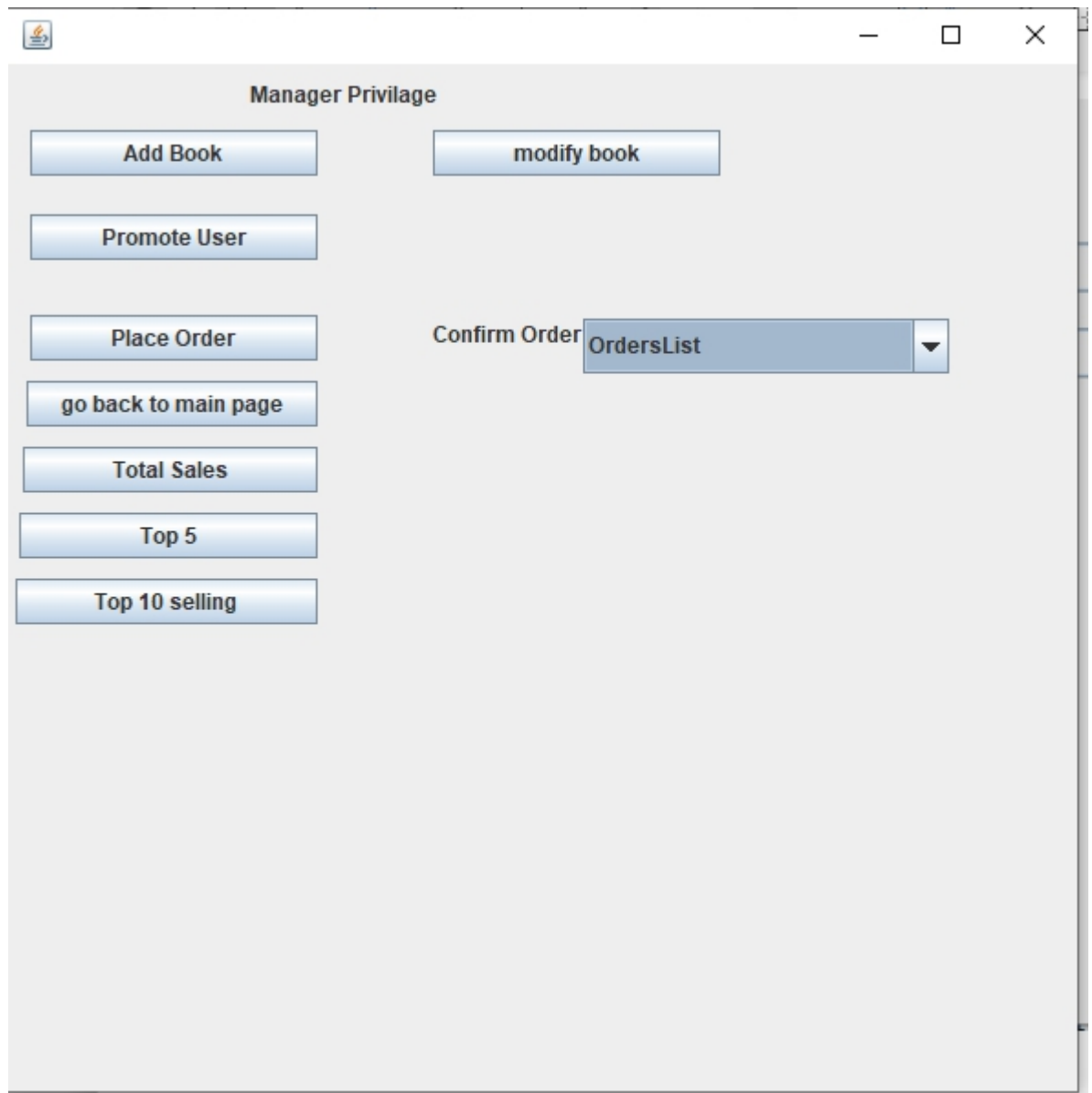
If there was an error of his order like ordering quantity of book which is greater than the available quantity program inform user and exclude this from his order.



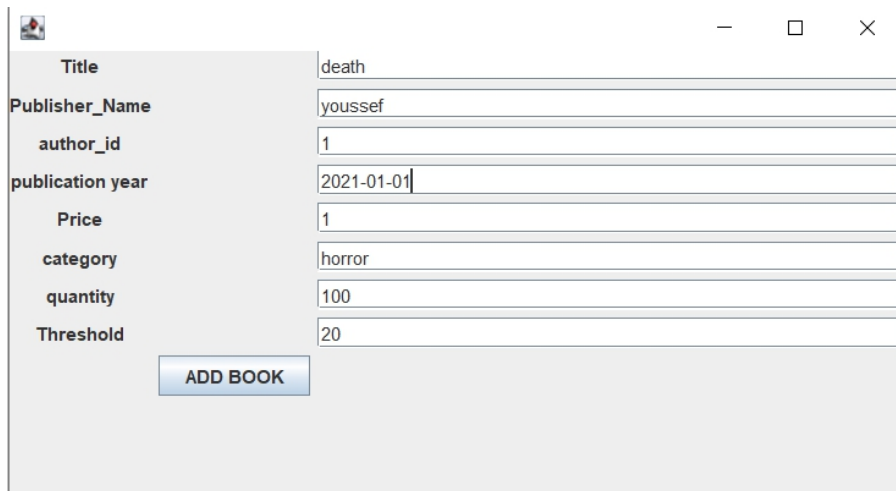
- After process his order totally



- Finally, there is two Types of users a customer and manager, manager has all customer features but with more features like adding new books, modify existing books, place orders on books and confirm or cancel orders.
- So, if the account was for a manager by using the manage button manager can do his magic.



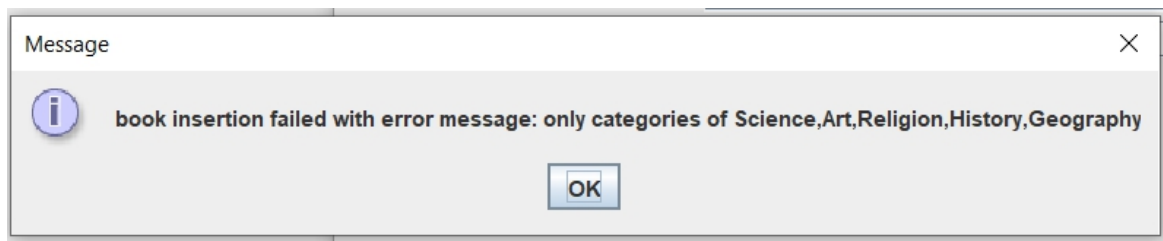
- By Pressing Add Book Button



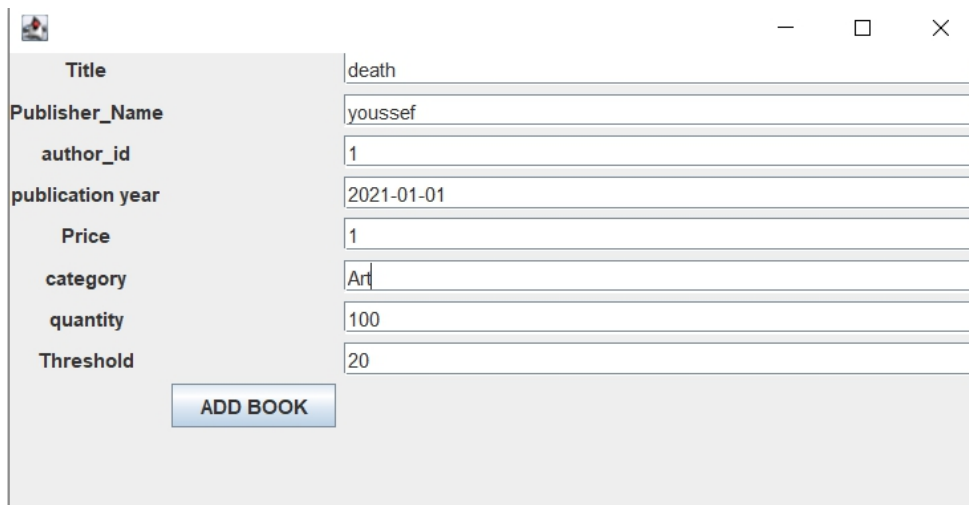
Title	death
Publisher_Name	youssef
author_id	1
publication year	2021-01-01
Price	1
category	horror
quantity	100
Threshold	20

ADD BOOK

Manager has to enters some information about the new book and the new book must match the restrictions so if the category wasn't provided message will be shown.

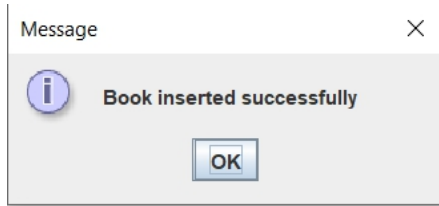


After change it



Title	death
Publisher_Name	youssef
author_id	1
publication year	2021-01-01
Price	1
category	Art
quantity	100
Threshold	20

ADD BOOK



- By Pressing modify book

Title	death
Price	1
category	Art
quantity	100
Threshold	20

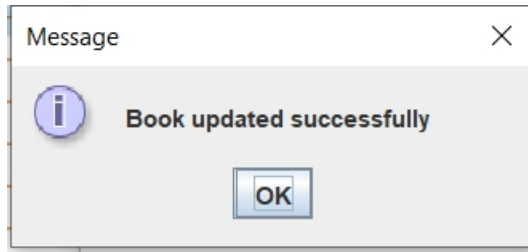
Update

Manager has to enter informations about the modified book and the updates must match the constrains.

```
insert into Book values(1,'book1',100,'Art', '2021-01-01',10,3 , 'youssef',1);
insert into Book values(2,'book2',200,'Art', '2021-01-01',10,4 , 'youssef',1);
insert into Book values(3,'book3',300,'Art', '2021-01-01',10,5 , 'youssef',1);
insert into Book values(4,'book4',400,'Art', '2021-01-01',10,6 , 'youssef',1);
insert into Book values(5,'book5',500,'Art', '2021-01-01',20,7 , 'youssef',1);
insert into Book values(6,'book6',600,'Art', '2021-01-01',20,8 , 'youssef',2);
insert into Book values(7,'book7',700,'Art', '2021-01-01',20,5 , 'youssef',2);
insert into Book values(8,'book8',800,'Art', '2021-01-01',20,6 , 'youssef',2);
insert into Book values(9,'book9',900,'Art', '2021-01-01',20,3 , 'youssef',2);
insert into Book values(10,'book10',1000,'Art', '2021-01-01',20,4 , 'youssef',2);
insert into Book values(11,'book11',1100,'Art', '2021-01-01',20,5 , 'youssef',2);
insert into Book values(12,'book12',1200,'Art', '2021-01-01',20,600,'youssef',2);
```

Title	book1
Price	100
category	Art
quantity	100
Threshold	20

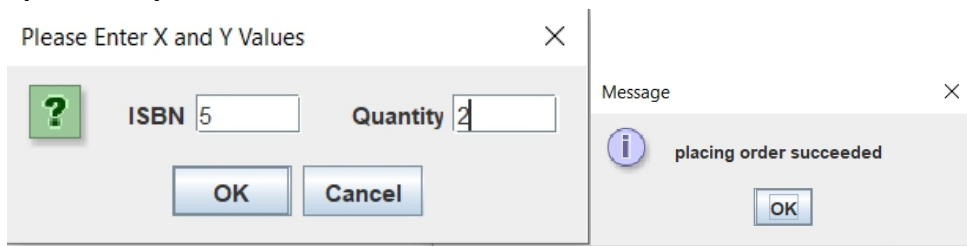
Update



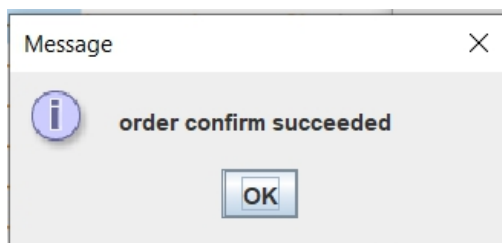
- If manager wants to promote specific user he can by pressing the promote button and enters user email.



- If manager wants to make an order, he can by pressing placing order button and type book ISBN and the quantity.



- Manger can confirm orders by press the bar list and press a specific order.

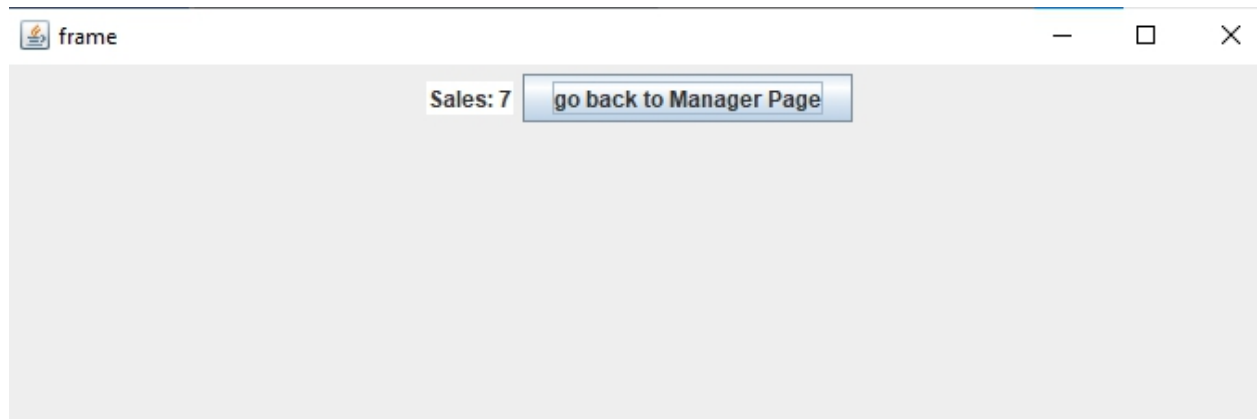


Reports :

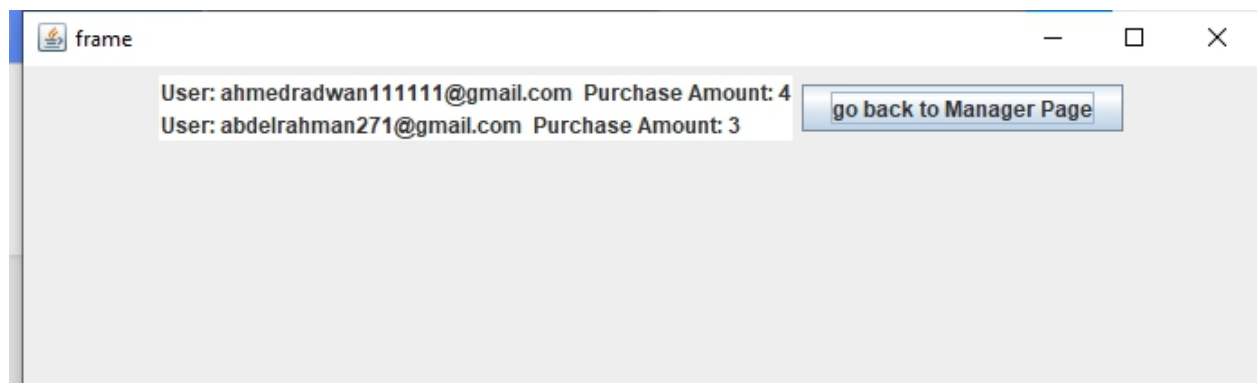
- we made it using swing in manager page you can find three buttons
 - "Total Sales"
 - "Top 5 "
 - "Top 10 Selling "

- when you press one of these buttons we send a query to our database and open a new page with a JList inside it and put the array of results we got from the query .

Total sales :



Top 5 :



Top 10 Selling :

 frame

Book_ISBN: 13 Copies: 6

Book_ISBN: 10 Copies: 1

go back to Manager Page