DATABASE

# Bookstore Android App

## Developed by:

**Shereen Elkordi (33)**

**Nancy Abd Elkarem (70)**

**Yomna Eldawy (72)**

## Bookstore Schema

```
create schema bookstoreschema;

use bookstoreschema;

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';


CREATE SCHEMA IF NOT EXISTS `bookstoreschema` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;

USE `bookstoreschema` ;



-- -----------------------------------------------------

-- Table `bookstoreschema`.`Category`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bookstoreschema`.`Category` (

  `category_id` INT NOT NULL,

  `category_name` VARCHAR(45) NOT NULL,

  PRIMARY KEY (`category_id`))

ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `bookstoreschema`.`Publisher`CALL `company`.`Count_Emp`(<{in Dnumber int}>, <{out cnt int}>);

-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bookstoreschema`.`Publisher` (
  `publisher_idcategory` INT NOT NULL,
  `publisher_name` VARCHAR(45) NOT NULL,
  `address` VARCHAR(120) NOT NULL,
  `phone_number` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`publisher_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `bookstoreschema`.`Book`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bookstoreschema`.`Book` (
  `Book_ISBN` varchar(13) NOT NULL,
  `title` VARCHAR(45) NOT NULL,
  `publication_year` INT NOT NULL,

  `price` DOUBLE NOT NULL,

  `no_of_copies` INT NOT NULL,

  `threshold` INT NOT NULL,

  `Category_category_id` INT NOT NULL,

  `Publisher_publisher_id` INT NOT NULL,

  PRIMARY KEY (`Book_ISBN`),

  INDEX `fk_Book_Category1_idx` (`Category_category_id` ASC),

  INDEX `fk_Book_Publisher1_idx` (`Publisher_publisher_id` ASC),

  CONSTRAINT `fk_Book_Category1`

    FOREIGN KEY (`Category_category_id`)

    REFERENCES `bookstoreschema`.`Category` (`category_id`)

    ON DELETE RESTRICT

    ON UPDATE CASCADE,

  CONSTRAINT `fk_Book_Publisher2`

    FOREIGN KEY (`Publisher_publisher_id`)

    REFERENCES `bookstoreschema`.`Publisher` (`publisher_id`)

    ON DELETE CASCADE

    ON UPDATE CASCADE)

ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `bookstoreschema`.`Author`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bookstoreschema`.`Author` (
  `author_id` INT NOT NULL,
  `name` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`author_id`))
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `bookstoreschema`.`Book_has_Author`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bookstoreschema`.`Book_has_Author` (
  `Book_ISBN` CHAR(13) NOT NULL,
  `Author_author_id` INT NOT NULL,
  PRIMARY KEY (`Book_ISBN`, `Author_author_id`),
  INDEX `fk_Book_has_Author_Author1_idx` (`Author_author_id` ASC),
  INDEX `fk_Book_has_Author_Book_idx` (`Book_ISBN` ASC),
  CONSTRAINT `fk_Book_has_Author_Book`
    FOREIGN KEY (`Book_ISBN`)
    REFERENCES `bookstoreschema`.`Book` (`Book_ISBN`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Book_has_Author_Author1`
    FOREIGN KEY (`Author_author_id`)
    REFERENCES `bookstoreschema`.`Author` (`author_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `bookstoreschema`.`User`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bookstoreschema`.`User` (
  `username` VARCHAR(20) NOT NULL,
  `password` VARCHAR(20) NOT NULL,
  `first_name` VARCHAR(20) NOT NULL,
  `last_name` VARCHAR(20) NOT NULL,
  `email` VARCHAR(50) NOT NULL,

  `phone_number` VARCHAR(20) NOT NULL,

  `country` VARCHAR(20) NOT NULL,

  `city` VARCHAR(20) NOT NULL,

  `street_name` VARCHAR(50) NOT NULL,

  `street_number` INT NULL,

  `is_manager` INt NOT NULL,

  PRIMARY KEY (`username`))

ENGINE = InnoDB;




-- -----------------------------------------------------

-- Table `bookstoreschema`.`Manager_Order`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `bookstoreschema`.`Manager_Order` (

  `Book_ISBN` CHAR(13) NOT NULL,

  `order_id` INT NOT NULL AUTO_INCREMENT,

  `quantity` INT NOT NULL,

  `confirmed` INT NOT NULL,

  PRIMARY KEY (`order_id`, `Book_ISBN`),

  CONSTRAINT `fk_Manager_Order_Book1`
```

```
    FOREIGN KEY (`Book_ISBN`)
    REFERENCES `bookstoreschema`.`Book` (`Book_ISBN`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `bookstoreschema`.`Customer_Order`
-- -----------------------------------------------------
CREATE TABLE IF NOT EXISTS `bookstoreschema`.`Customer_Order` (
  `quantity` INT NOT NULL,
  `checkout` INT NOT NULL,
  `Book_ISBN` CHAR(13) NOT NULL,
  `User_username` VARCHAR(20) NOT NULL,
  `date` date NOT NULL,
  PRIMARY KEY (`Book_ISBN`, `User_username`),
  INDEX `fk_Customer_Order_Book1_idx` (`Book_ISBN` ASC),
  INDEX `fk_Customer_Order_User1_idx` (`User_username` ASC),
  CONSTRAINT `fk_Customer_Order_Book1`
    FOREIGN KEY (`Book_ISBN`)
    REFERENCES `bookstoreschema`.`Book` (`Book_ISBN`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Customer_Order_User1`
    FOREIGN KEY (`User_username`)
    REFERENCES `bookstoreschema`.`User` (`username`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

delimiter $$

CREATE TRIGGER quantityInsertCheck before INSERT ON Customer_Order
for each row begin
if (new.quantity > (select no_of_copies from Book where Book.Book_ISBN = new.Book_ISBN)) then
	signal sqlstate '02000' set message_text = 'Not available';
```

```
END if;
end$$

delimiter ;

delimiter $$

CREATE TRIGGER bookNumber before update ON Book
for each row begin
if (new.no_of_copies < 0) then
        signal sqlstate '02000' set message_text = 'Illegal number of copies';
END if;
end$$

delimiter ;

delimiter $$

CREATE TRIGGER bookNumber1 before insert ON Book
for each row begin
if (new.no_of_copies < 0) then
```

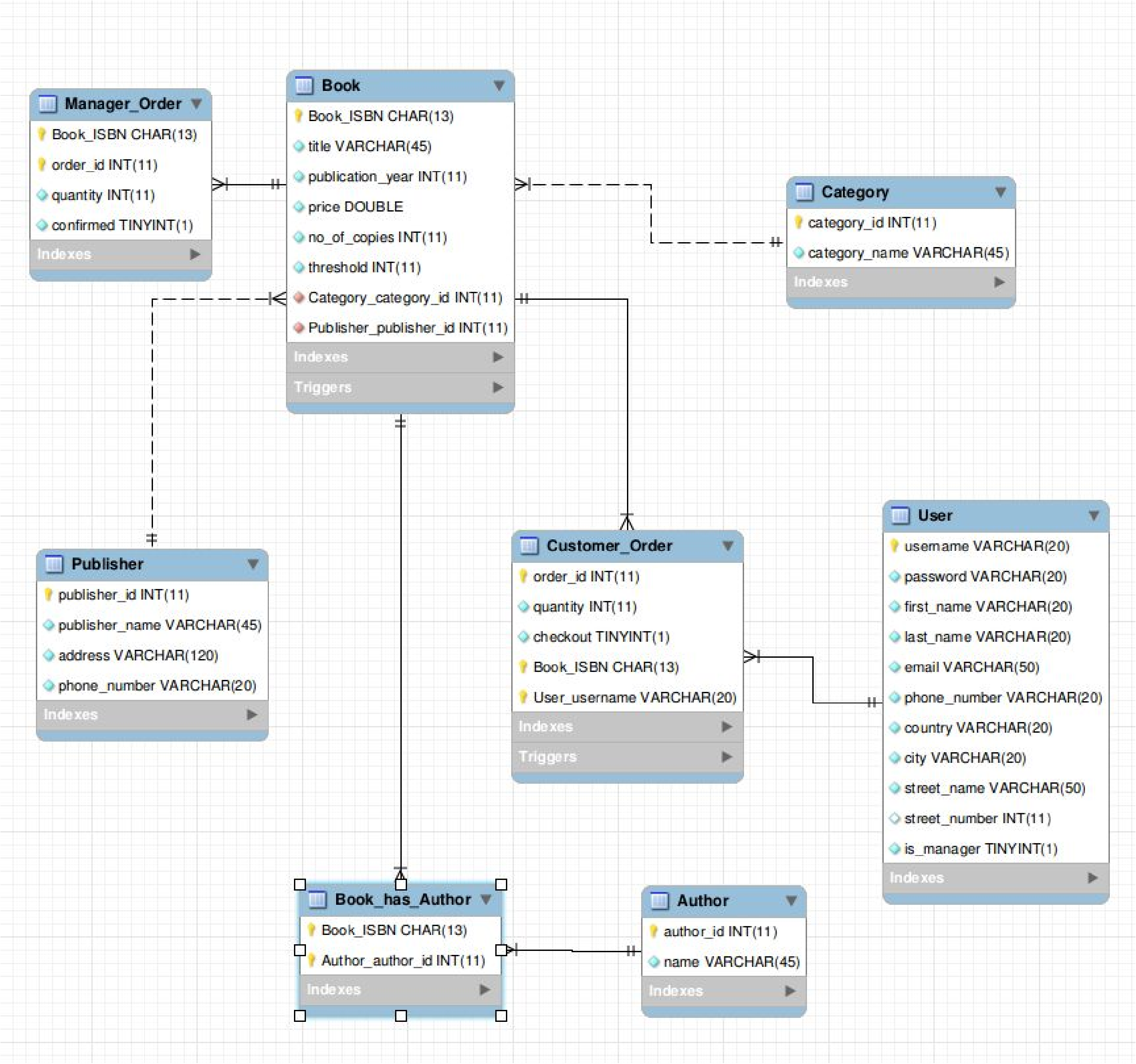
```
        signal sqlstate '02000' set message_text = 'Illegal number of copies';
END if;
end$$


delimiter ;




delimiter $$
CREATE TRIGGER quantityUpdateCheck before update ON Customer_Order
for each row begin
if (new.quantity > (select no_of_copies from Book where Book.Book_ISBN =
new.Book_ISBN)) then
        signal sqlstate '02000' set message_text = 'Quantity not available';
END if;
end$$


delimiter ;


delimiter $$
create trigger checkout after update on Customer_Order
for each row begin
```

```
if (new.checkout = 1) then
        update Book set no_of_copies = no_of_copies - new.quantity
        where Book_ISBN = new.Book_ISBN;
end if;
end$$
delimiter ;

delimiter $$
create trigger confirm after update on Manager_Order
for each row begin
if (new.confirmed = 1) then
        update Book set no_of_copies = no_of_copies + new.quantity
        where Book_ISBN = new.Book_ISBN;
end if;
end$$
delimiter ;

create view ordered_copies as
select Book.Book_ISBN, sum(Manager_Order.quantity) as total
from Manager_Order, Book
where Manager_Order.Book_ISBN = Book.Book_ISBN and Manager_Order.confirmed = 0
```

```
group by Book.Book_ISBN;



delimiter $$

create trigger checkThreshold after update on Book

for each row begin

if (new.no_of_copies + (select total from ordered_copies where Book_ISBN =
new.Book_ISBN) < new.threshold) then

        insert into Manager_Order(Book_ISBN, quantity, confirmed) values(new.Book_ISBN,
new.threshold - (new.no_of_copies +

        (select total from ordered_copies where Book_ISBN = new.Book_ISBN)), 0);

end if;

end$$



delimiter ;



delimiter $$

 create trigger deleteOrder before delete on Manager_Order

 for each row begin

 update Book set no_of_copies = no_of_copies + old.quantity
```

```
        where Book_ISBN = old.Book_ISBN;

end $$

delimiter ;


create index TITLE_INDEX on Book (title ASC);

create index publisher_INDEX on Publisher(publisher_name ASC);

create index publisher_id_INDEX on Book(Publisher_publisher_id ASC);


create index i_category on Book (Category_category_id ASC);

create index i_author_name on Author (name ASC);

create index i_username on Customer_Order (User_username ASC);

create index i_book_isbn on Customer_Order (Book_ISBN ASC );


create index manager_INDEX on User (is_manager ASC);

create index m_book_isbn on Manager_Order (Book_ISBN ASC );
```

## ER Diagram


Manager_Order
Book_ISBN CHAR(13)
order_id INT(11)
quantity INT(11)
confirmed TINYINT(1)
Indexes
Book
Book_ISBN CHAR(13)
title VARCHAR(45)
publication_year INT(11)
price DOUBLE
no_of_copies INT(11)
threshold INT(11)
Category_category_id INT(11)
Publisher_publisher_id INT(11)
Indexes
Triggers
Category
category_id INT(11)
category_name VARCHAR(45)
Indexes
Publisher
publisher_id INT(11)
publisher_name VARCHAR(45)
address VARCHAR(120)
phone_number VARCHAR(20)
Indexes
Customer_Order
order_id INT(11)
quantity INT(11)
checkout TINYINT(1)
Book_ISBN CHAR(13)
User_username VARCHAR(20)
Indexes
Triggers
User
username VARCHAR(20)
password VARCHAR(20)
first_name VARCHAR(20)
last_name VARCHAR(20)
email VARCHAR(50)
phone_number VARCHAR(20)
country VARCHAR(20)
city VARCHAR(20)
street_name VARCHAR(50)
street_number INT(11)
is_manager TINYINT(1)
Indexes
Book_has_Author
Book_ISBN CHAR(13)
Author_author_id INT(11)
Indexes
Author
author_id INT(11)
name VARCHAR(45)
Indexes

## Triggers

1. Before insertion on customer order table there is a trigger that checks whether or not the quantity of the book ordered is available which displays a message to notify the user in case the quantity is not available .
2. Before modifying a book quantity a trigger checks if the quantity entered is negative and issues a message that it is not allowed.
3. Like the previous trigger, there is another one to check the quantity upon the insertion of a new book.
4. A trigger is made upon updating the quantity ordered by the user of a book that is already found in its shopping cart to check whether or not the new quantity is available.
5. A trigger is made to modify the quantity of books when the customer decides to checkout (checkout flag is updated to 1), the new book quantity = the existing quantity - the ordered quantity.
6. A trigger is issued once the confirm flag is set in the manager order table meaning the manager has confirmed receiving the books from the publisher, the no of copies of the book is then increased by the new quantity.
7. When a manager deletes an order it is considered as confirmed as well, so there is a trigger that updates the book number of copies by the quantity ordered.
8. Another Trigger after updating the book quantity checks whether it is still above the threshold or not, if it is not then an ordered is placed by the name of a default manager to get books from the publisher to cover the lack and reach the threshold (it is increased by the value threshold - available - already ordered).

## Views

A view is made for the managers to be able to see the orders made to the publishers and its details including the book isbn, and the total quantity ordered by all managers.

## Indexes

1. An index is made on the book title in the book table as the average number of search is very high and the size of the book table is big.
2. An index is made on the category id in the book table as the average number of search is very high and the size of the book table is big.
3. An index is made on the author name because the size of the author table is very big and it would be time consuming if a book is searched by the author name.
4. The same applies for the search of the book by its publisher because the size of the book table is big. An index is then made on the publisher id.
5. An index is made on the username in the customer order table as it is required to get a report about the top users in the system in the last three months, and there are 50 thousand purchases per a day so in a month the table size would reach about 500,000
6. For the same previous reason there is an index on the book ISBN in the customer table as it is required to get the highest selling books and the total number of books sold in the last three months.
7. An index is made on the is_manager flag in the user table because .........
8. An index is made on the book ISBN in the manager order because .........

## Transactions

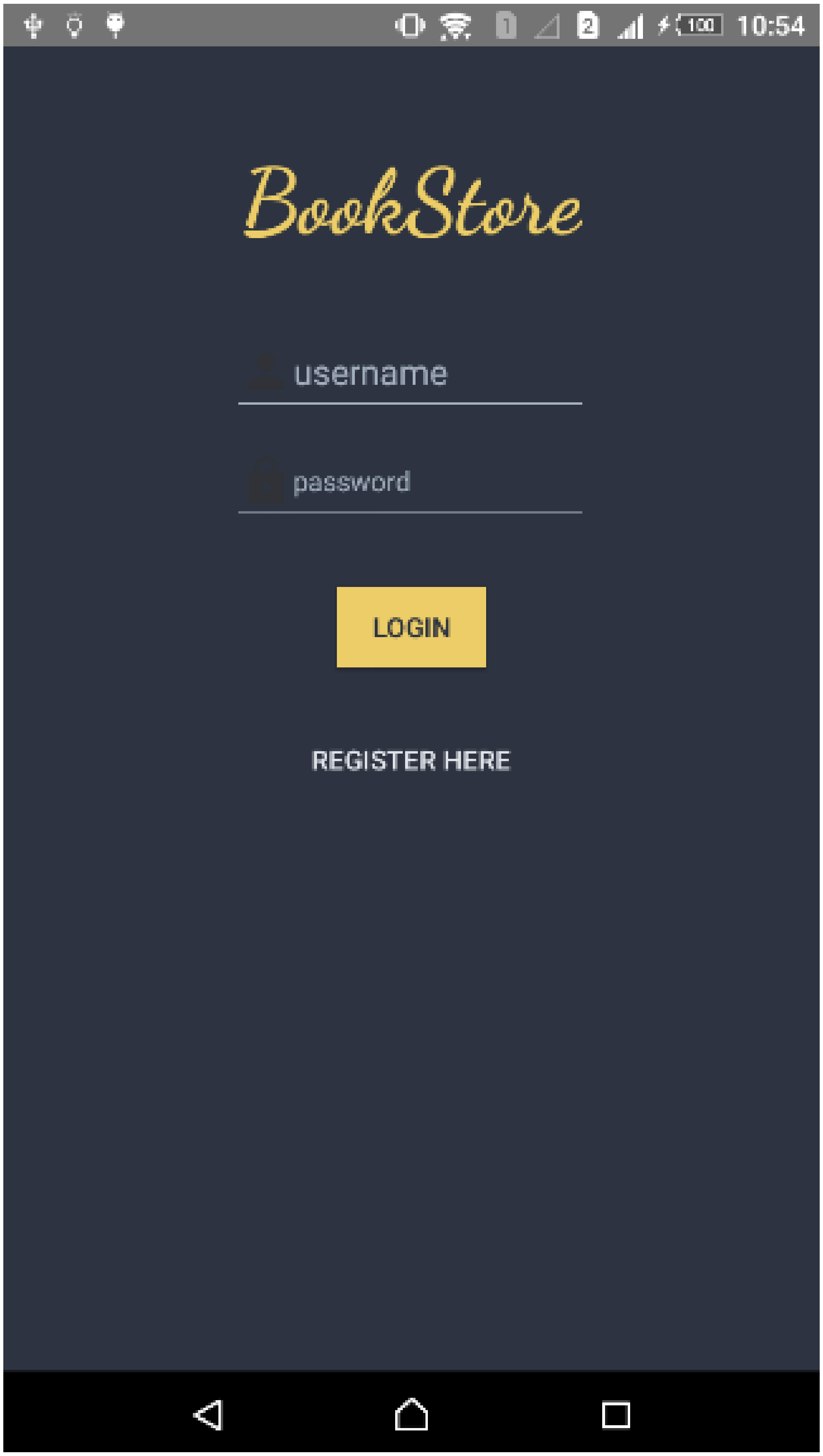
Only one transaction is made where there are separate queries depending on each other when a new book is added the publisher table need to be checked to confirm its existence and gets its id and the author table needs to be checked to ensure its existence or else add the authors in the table of authors as well as inserting the author name along with the newly added book isbn to the table representing the relationship many to many between the book table and the author table (Book_has_author). Transaction is used to ensure no table is altered by another user at the same time.

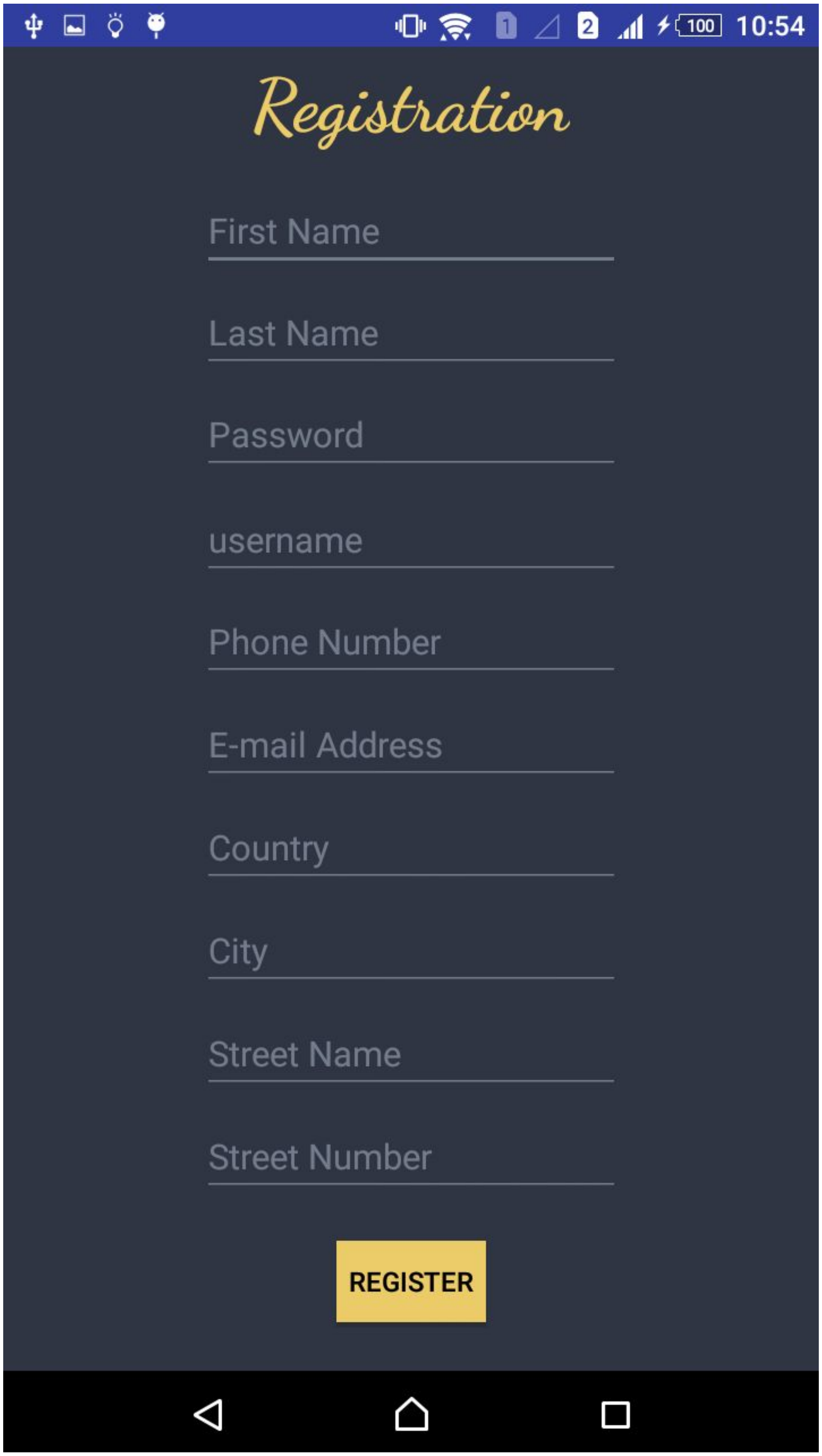
All other functions are executed in one single query statement so no transaction or concurrency control is needed as the sql statement is atomic.

## Database Tuning

1. A category table is made instead of inserting a string value in the book table which makes the size bigger. An id (small number) is inserted instead.
2. The same concept is applied on the publisher name.
3. Authors are kept in a separate table to allow a many to many relationship with the book table.
4. The book ISBN is a char(13) variable to ensure that no user would use a wrong number of digits while entering the ISBN.
5. The user address is split into fields (street name, street number, city, country) because it is a composite attribute.

## Screenshots





Registration

First Name

Last Name

Password

username

Phone Number

E-mail Address

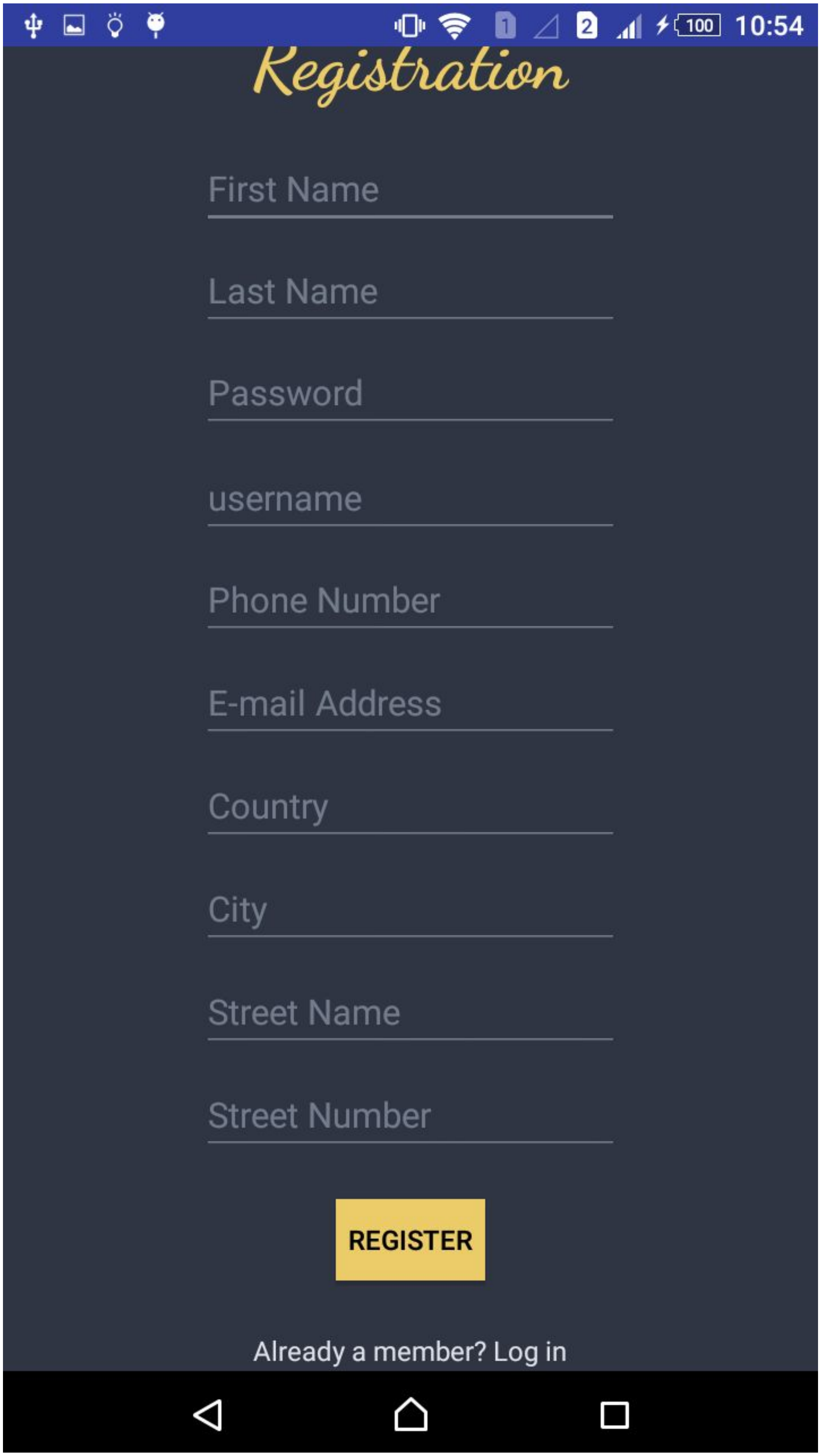
Country

City

Street Name

Street Number

REGISTER



Registration

First Name

Last Name

Password

username

Phone Number

E-mail Address

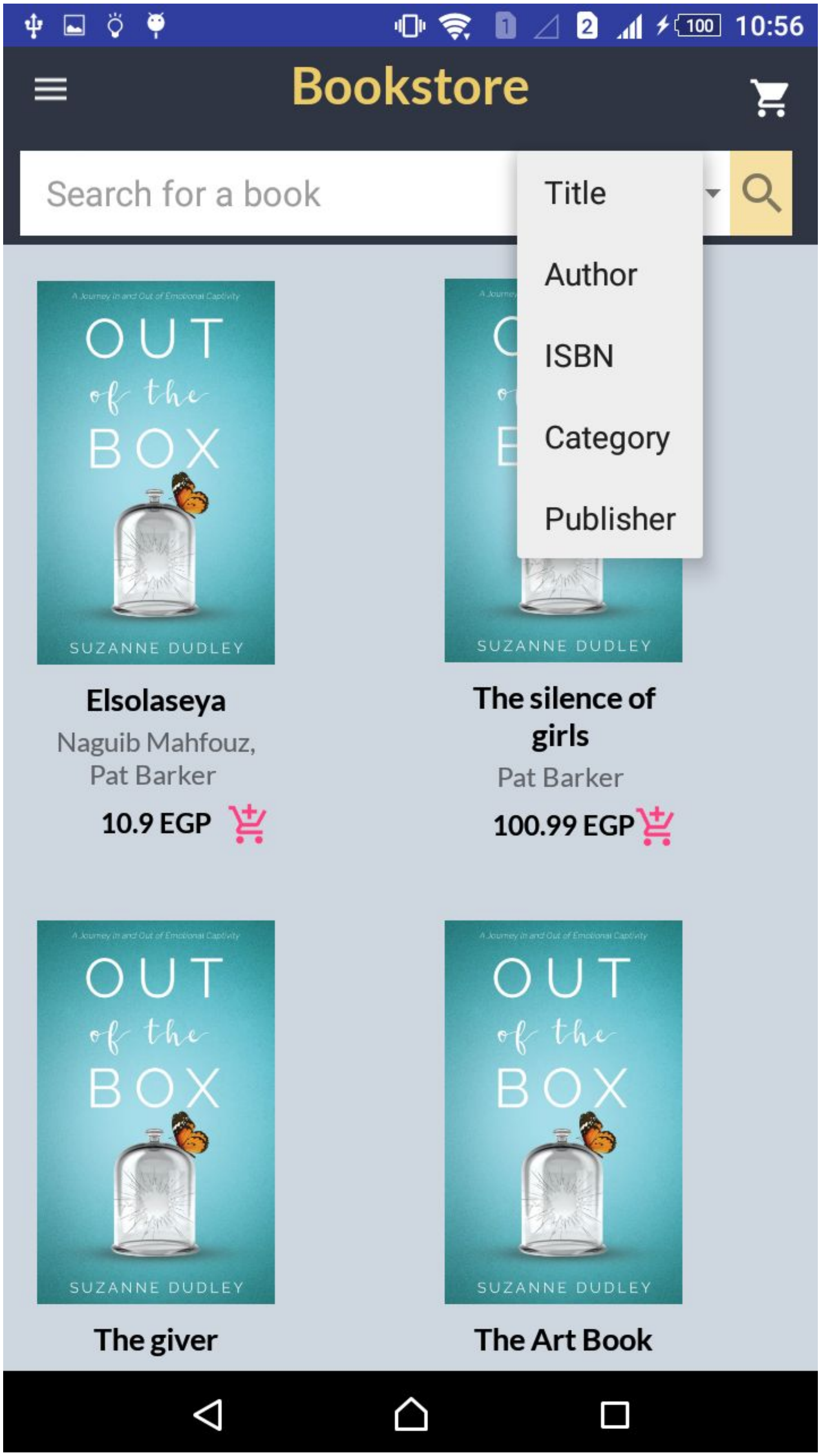
Country

City

Street Name

Street Number

Already a member? Log in

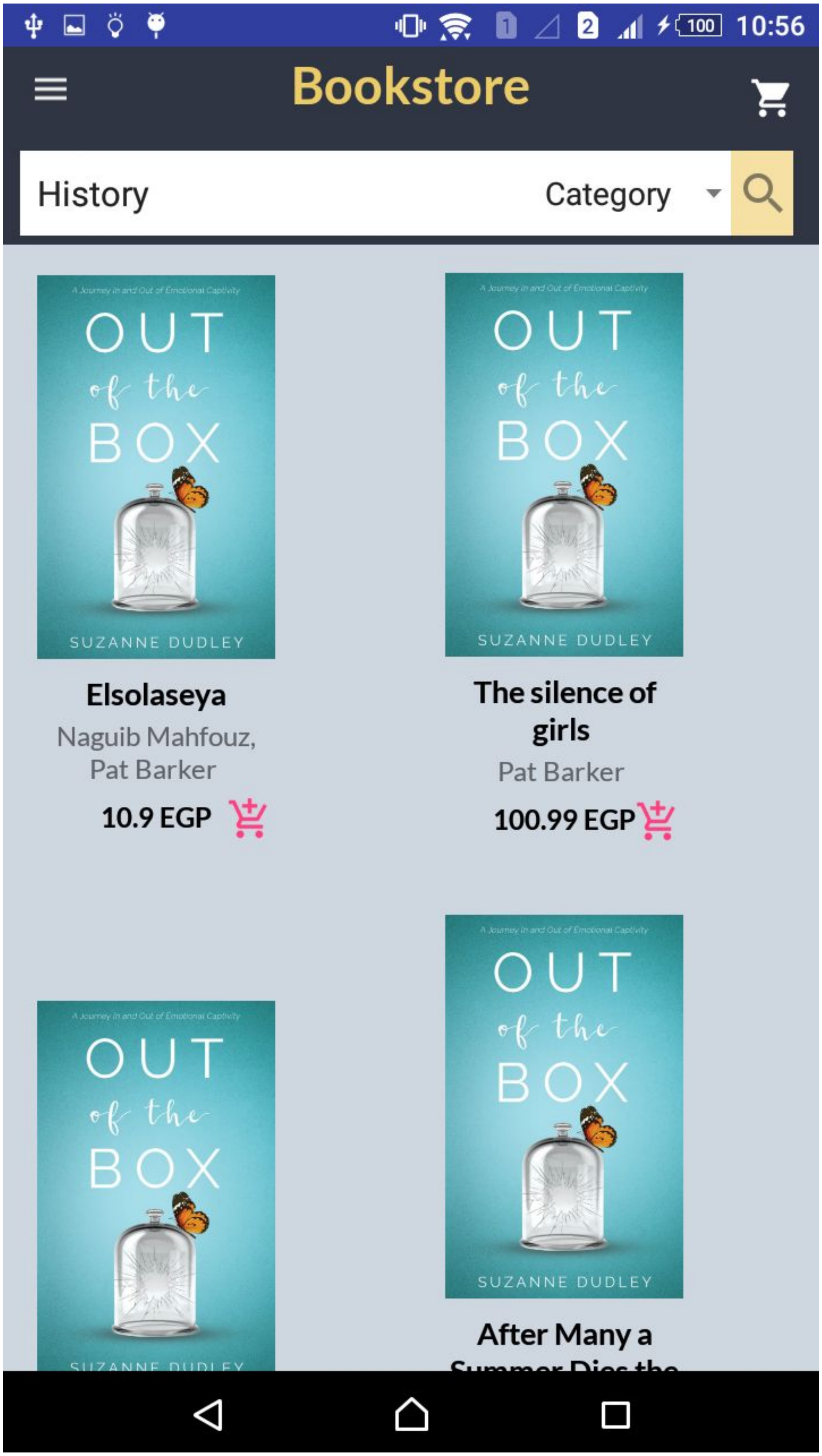


Elsolaseya
Naguib Mahfouz, Pat Barker
10.9 EGP

The silence of girls
Pat Barker
100.99 EGP

The giver

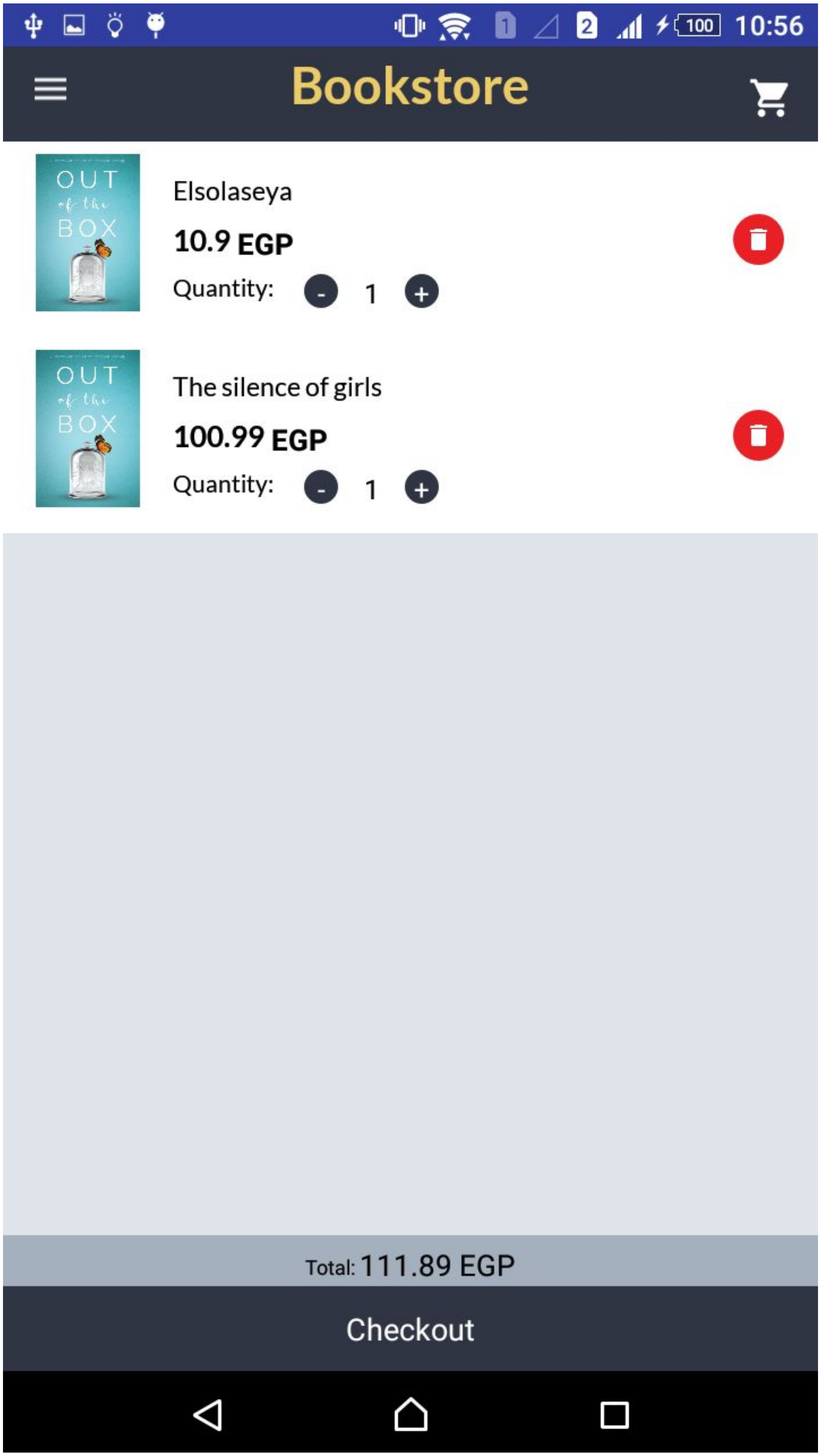
The Art Book



**Elsolaseya**
Naguib Mahfouz,
Pat Barker
**10.9 EGP**

**The silence of girls**
Pat Barker
**100.99 EGP**

**After Many a Summer Dies the**



Bookstore

Elsolaseya

10.9 EGP

Quantity: - 1 +

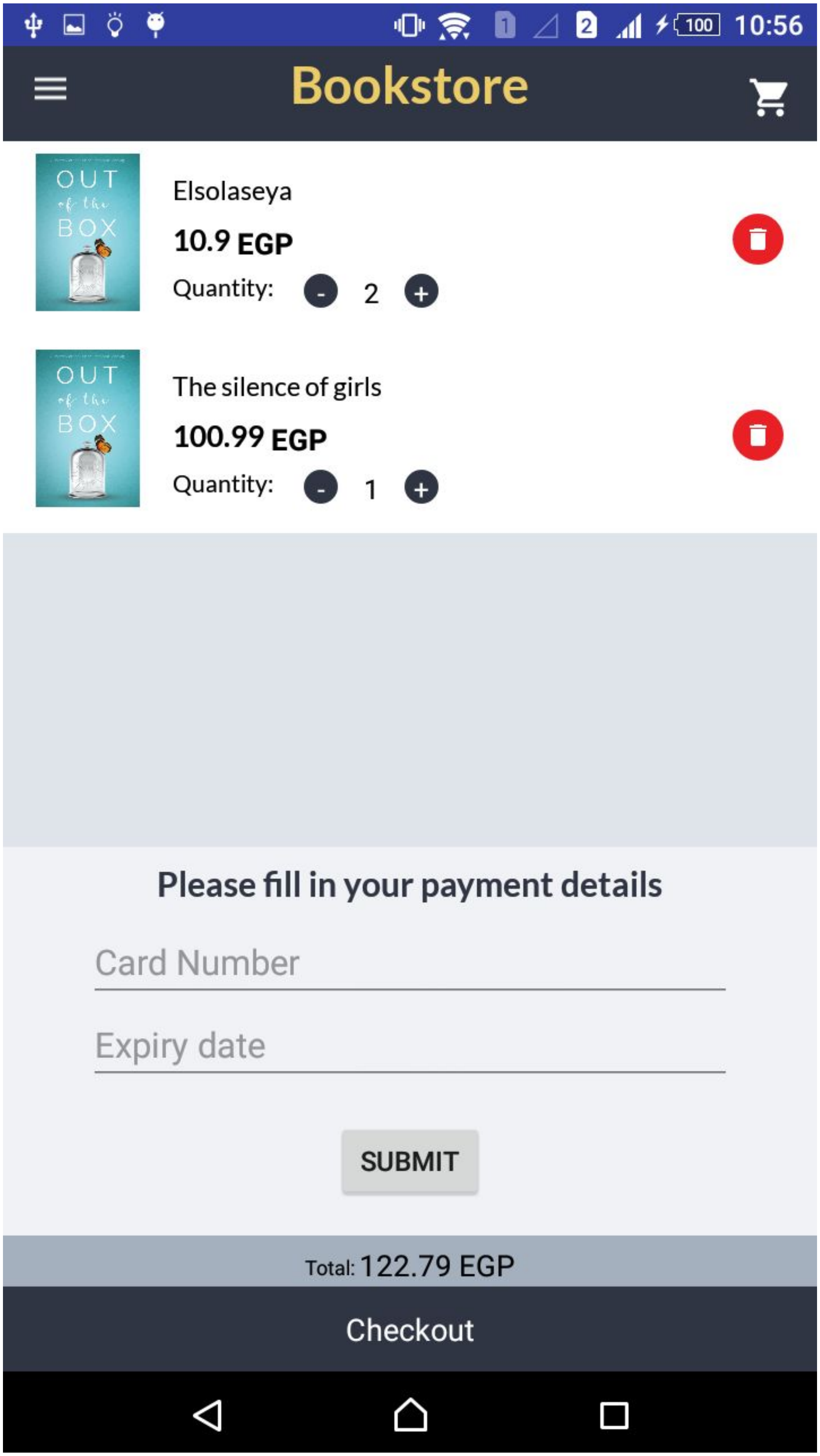
The silence of girls

100.99 EGP

Quantity: - 1 +

Total: 111.89 EGP

Checkout



Elsolaseya

10.9 EGP

Quantity: 2

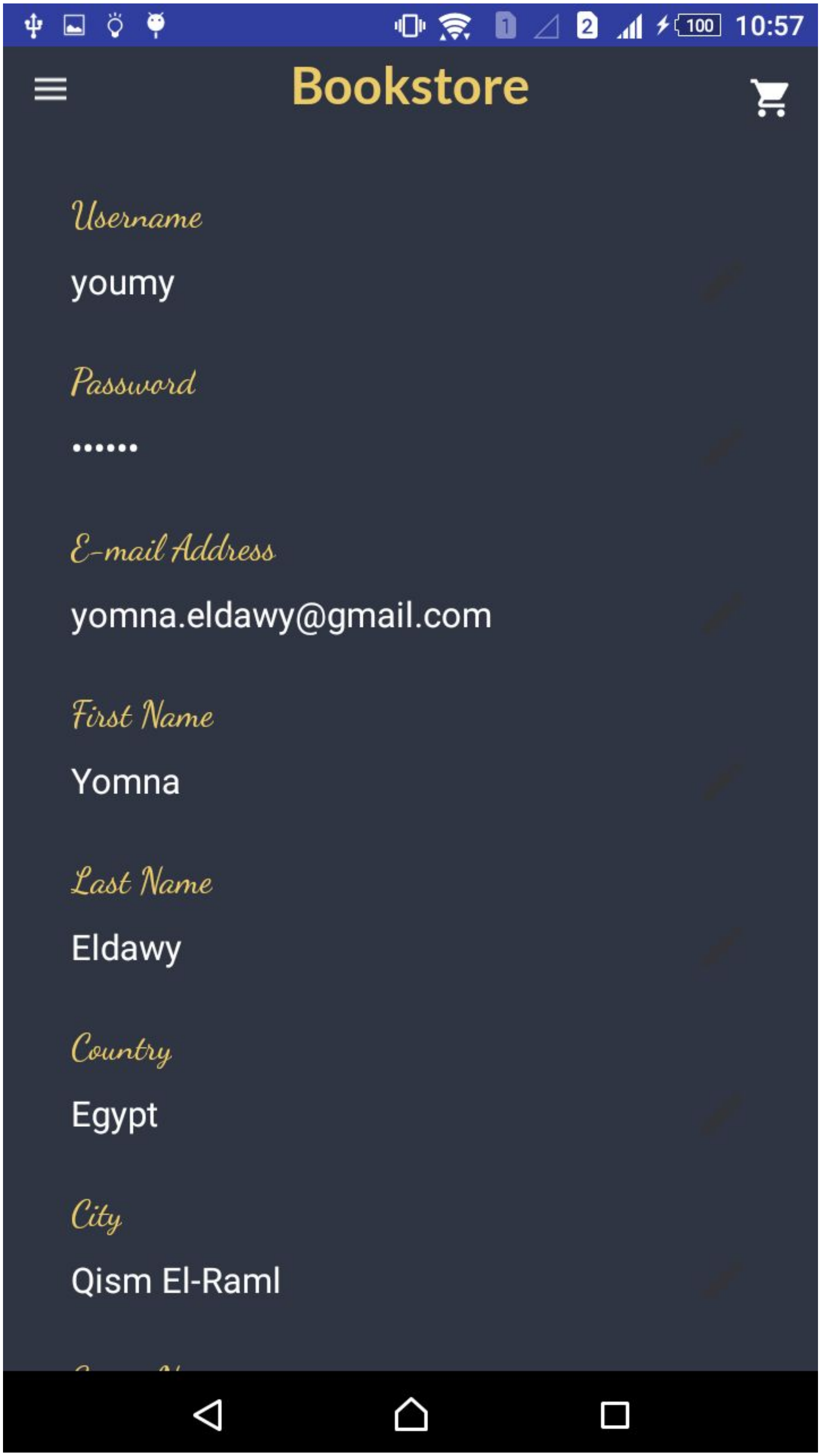
The silence of girls

100.99 EGP

Quantity: 1

Please fill in your payment details

Total: 122.79 EGP



Username

youmy

Password

••••••

E-mail Address

yomna.eldawy@gmail.com

First Name

Yomna

Last Name

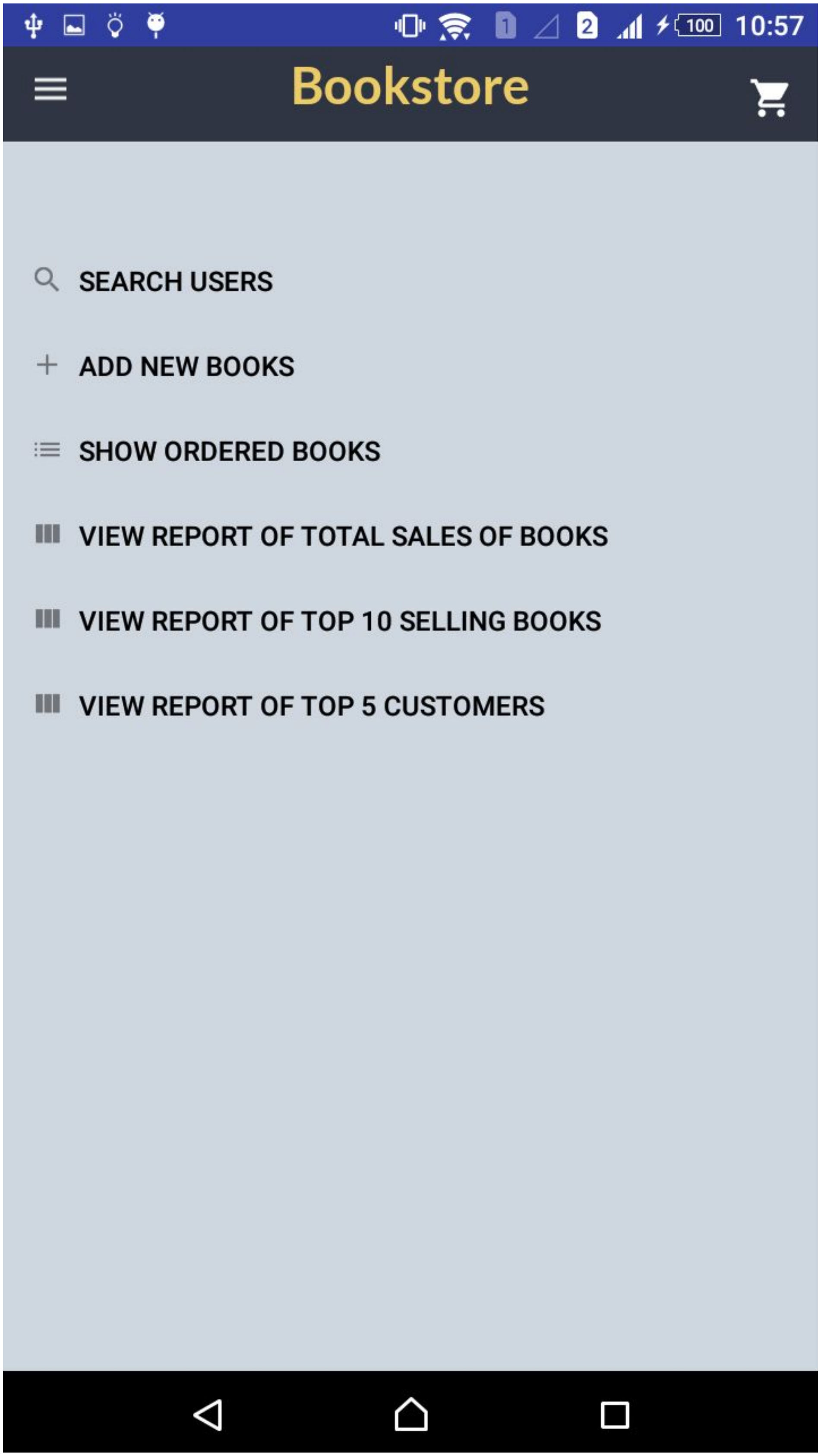
Eldawy

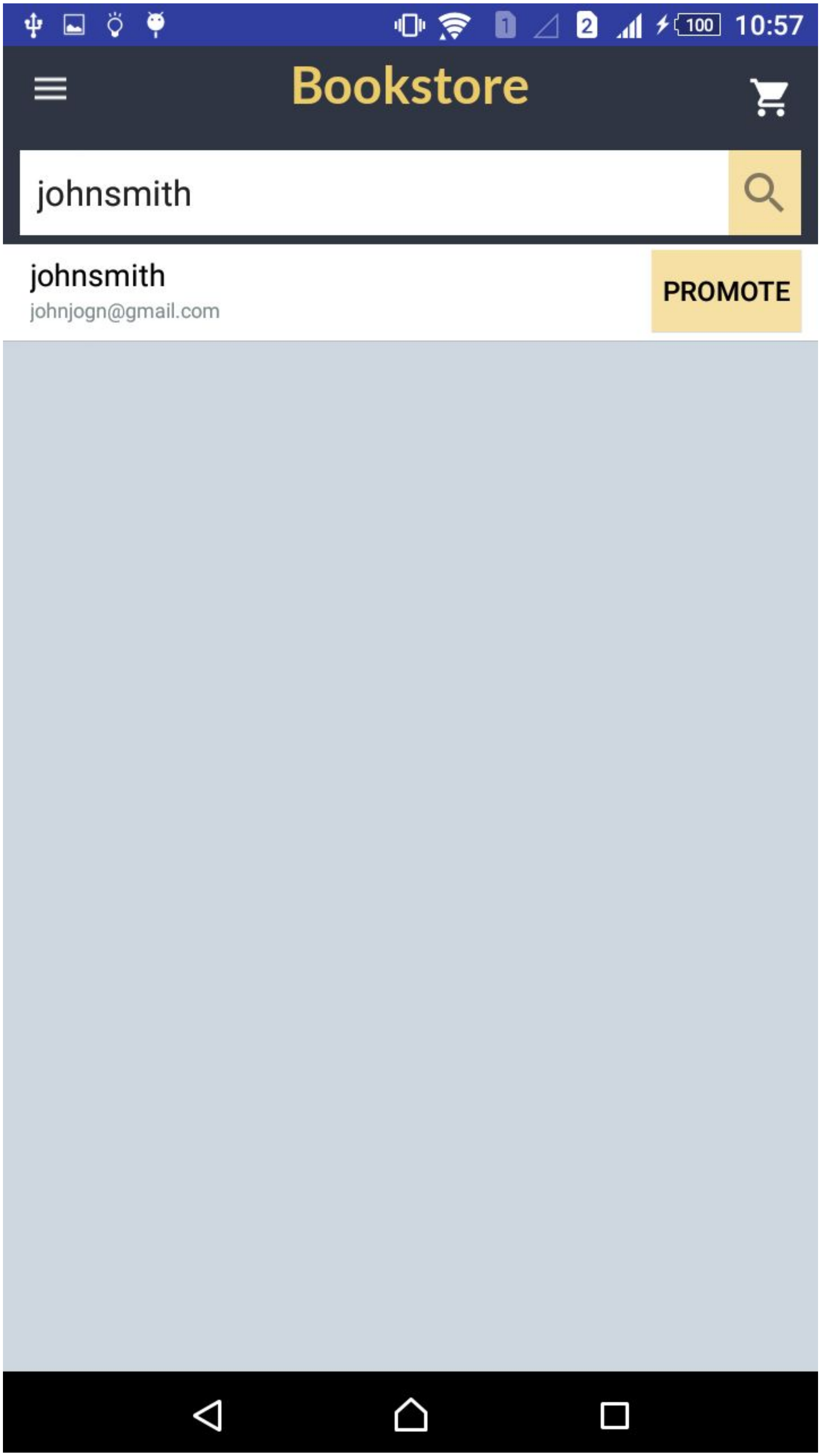
Country

Egypt

City

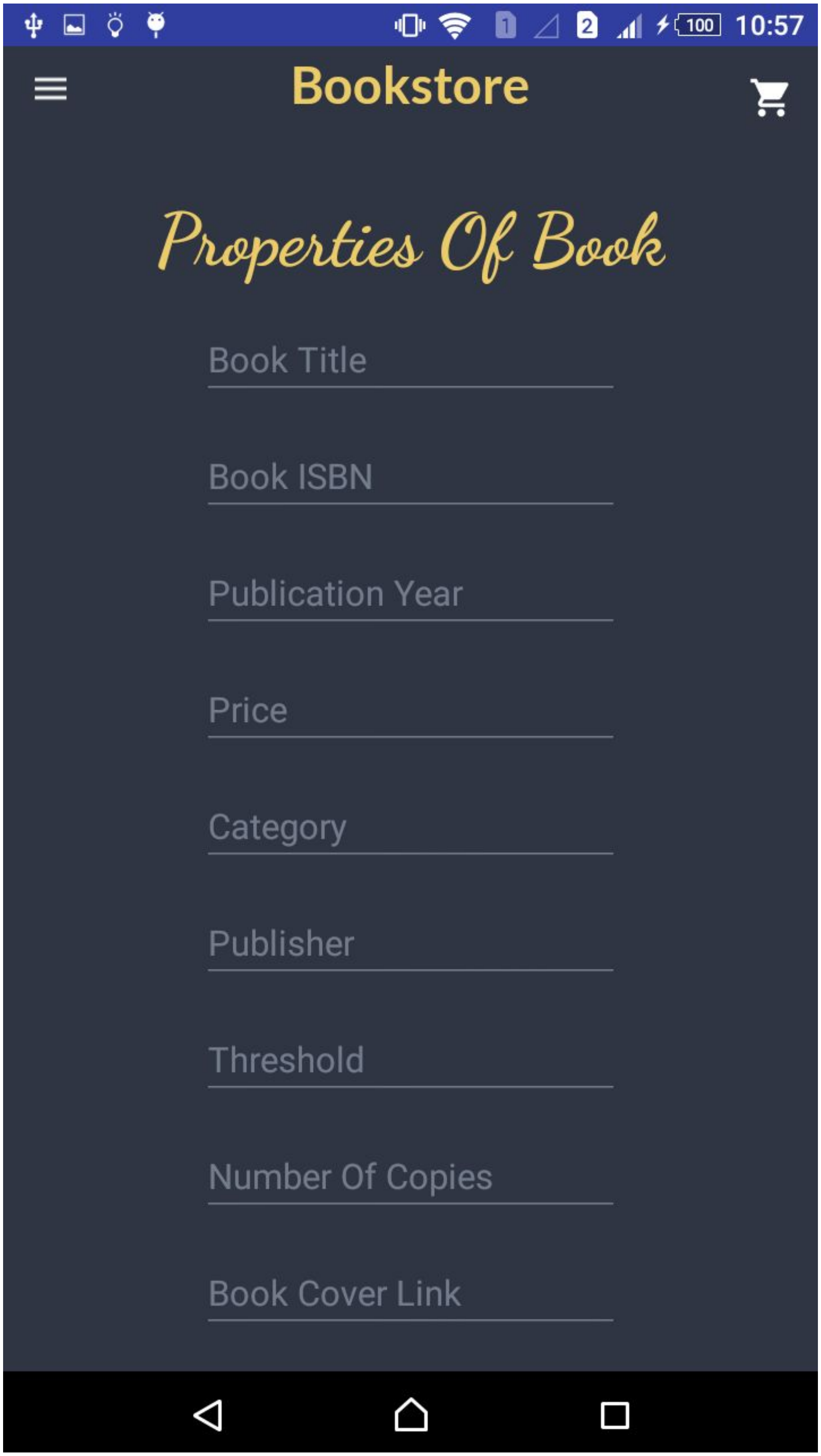
Qism El-Raml

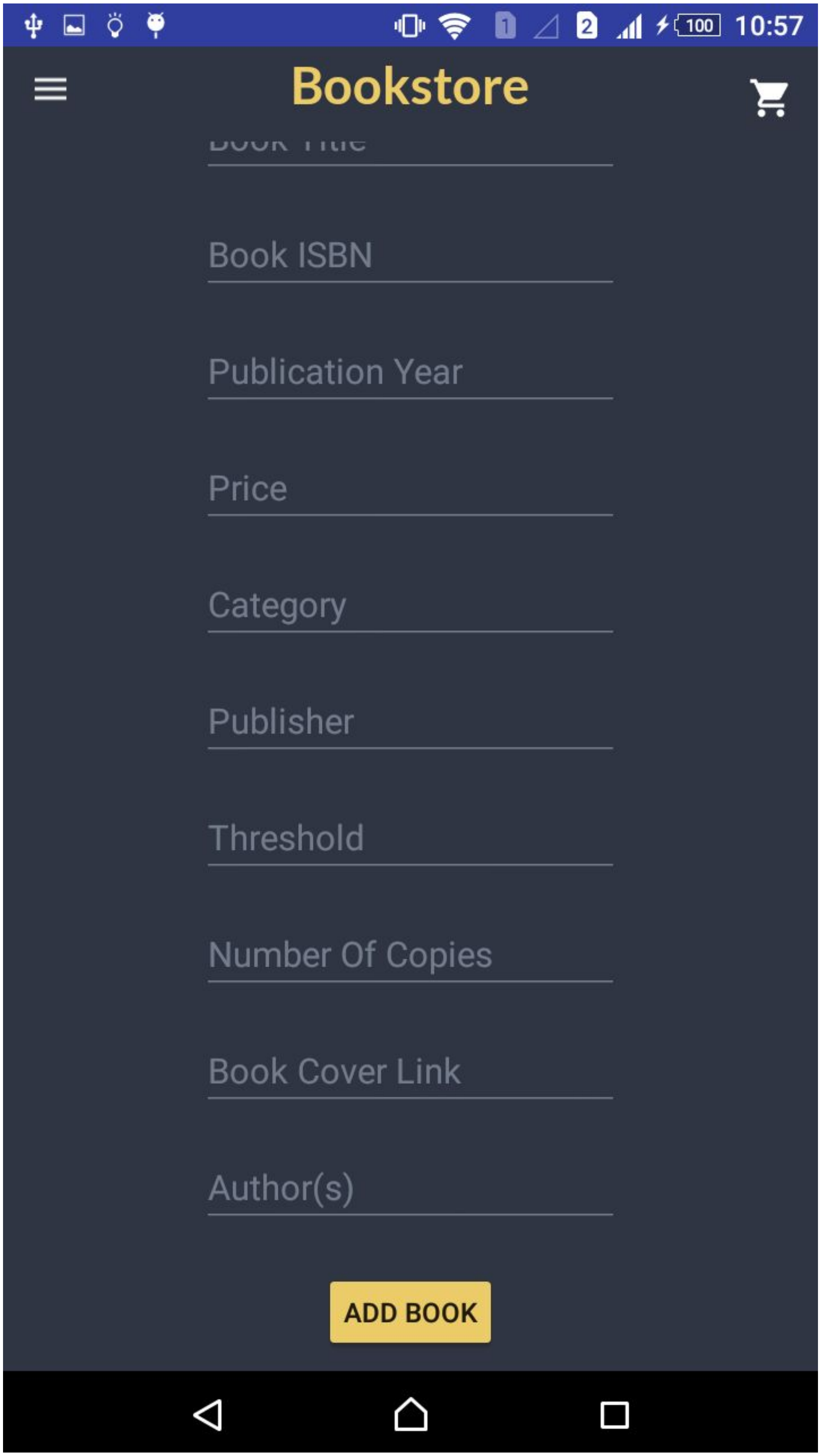

100
10:57
Bookstore
SEARCH USERS
ADD NEW BOOKS
SHOW ORDERED BOOKS
VIEW REPORT OF TOTAL SALES OF BOOKS
VIEW REPORT OF TOP 10 SELLING BOOKS
VIEW REPORT OF TOP 5 CUSTOMERS



johnsmith

johnsmith
johnjogn@gmail.com


10:57
Bookstore
Properties Of Book
Book Title
Book ISBN
Publication Year
Price
Category
Publisher
Threshold
Number Of Copies
Book Cover Link



10:57

Bookstore

Book Title

Book ISBN

Publication Year

Price

Category

Publisher

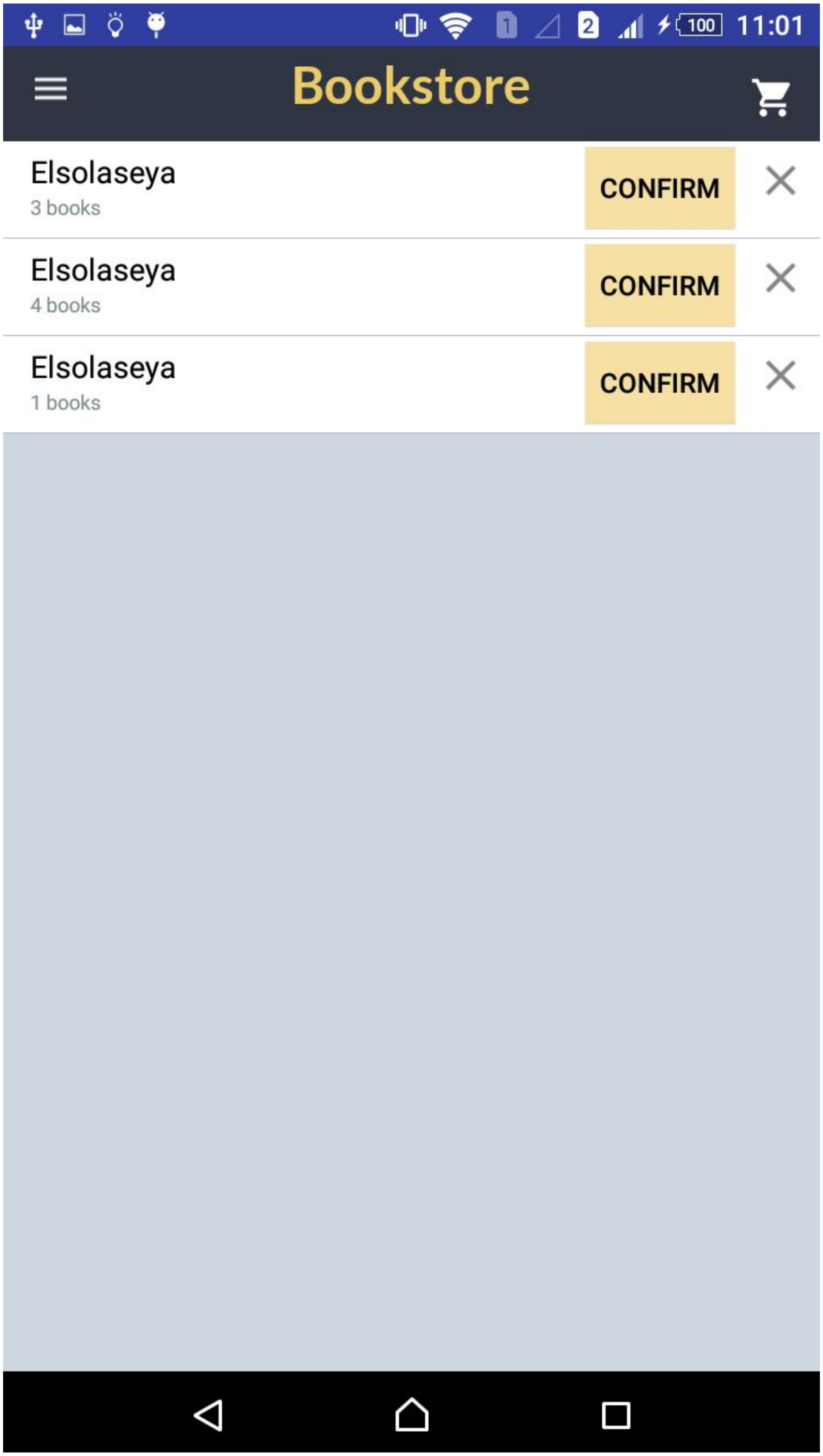
Threshold

Number Of Copies

Book Cover Link

Author(s)

ADD BOOK

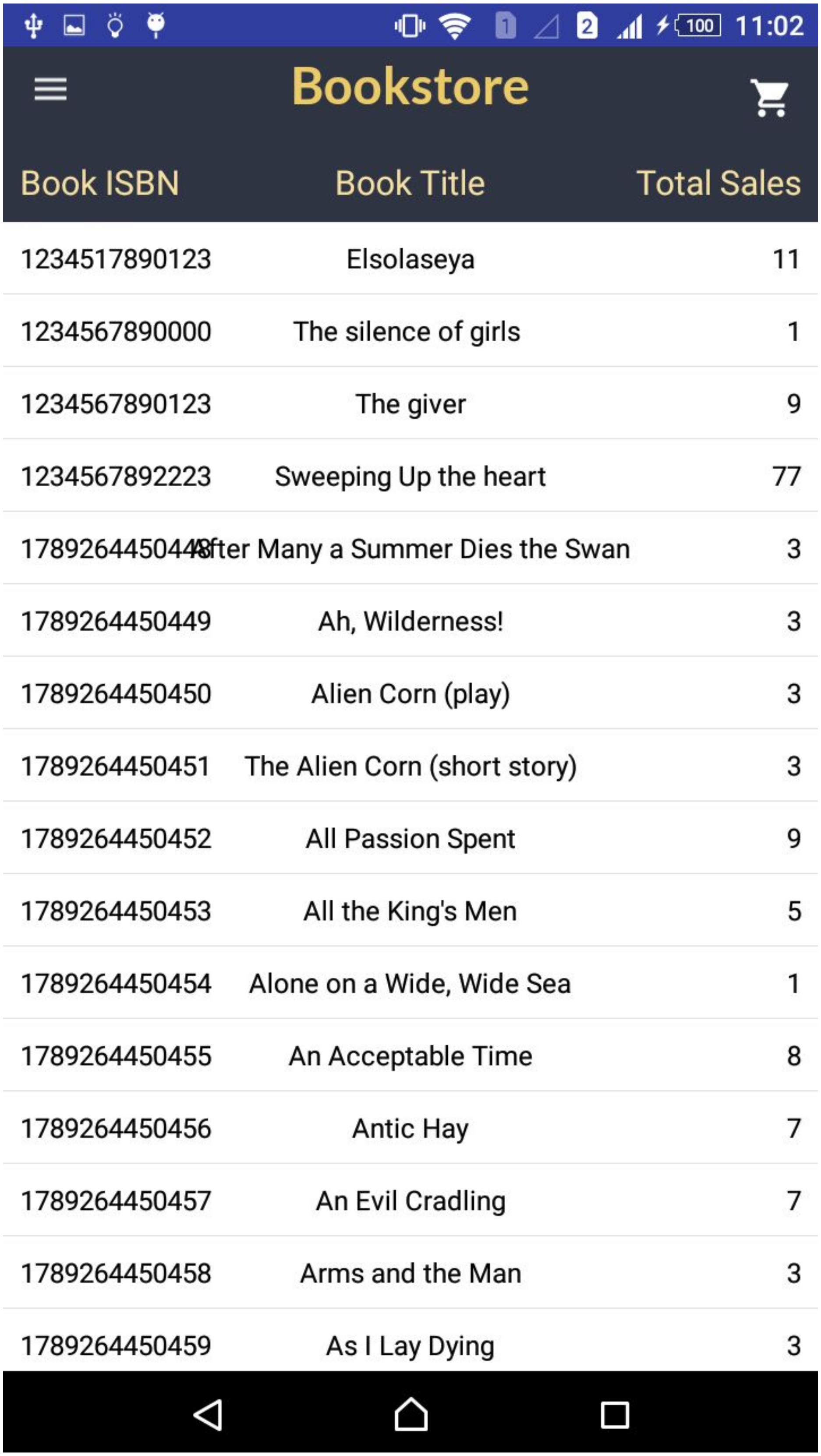


Bookstore

Elsolaseya
3 books

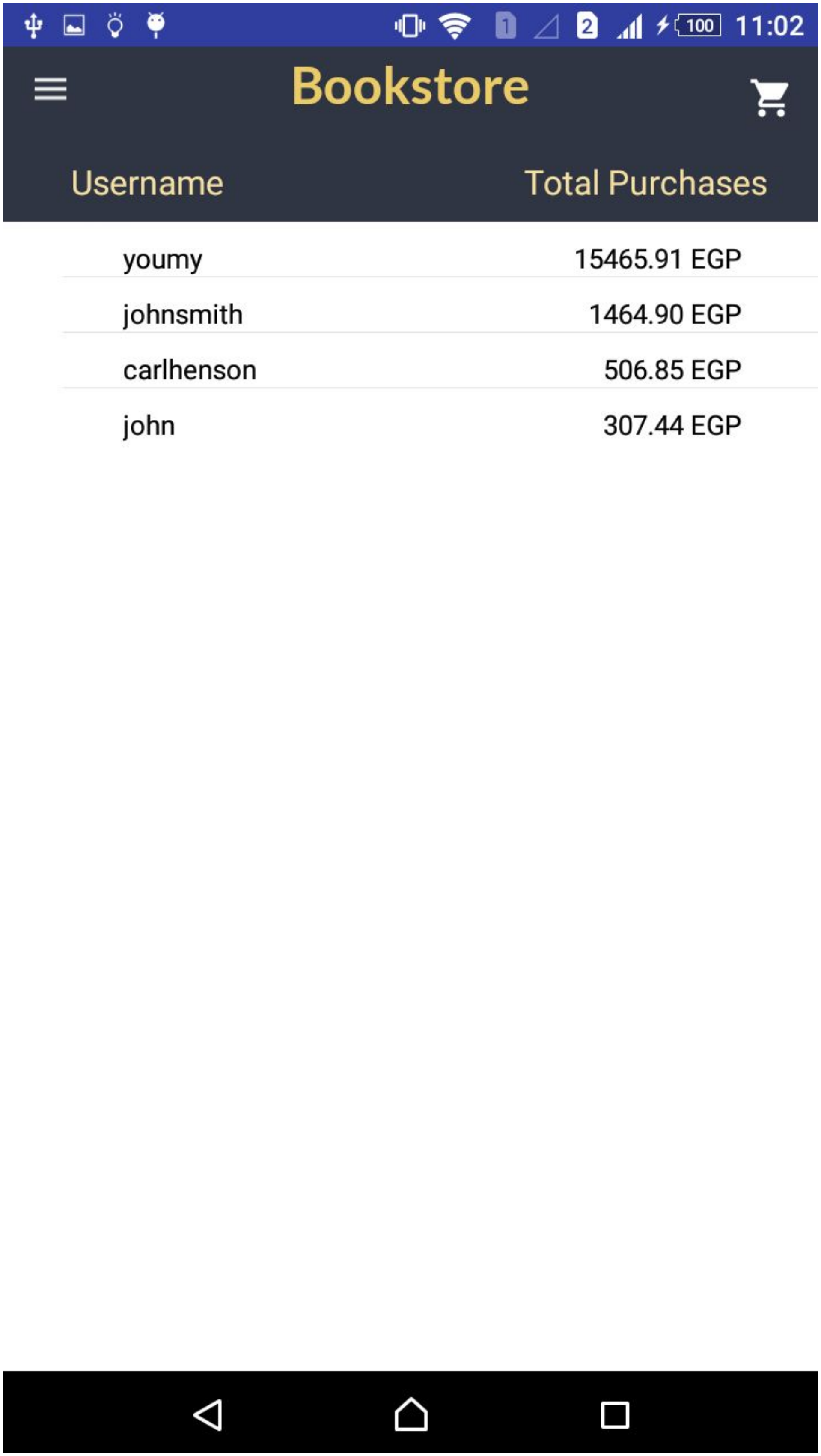
Elsolaseya
4 books

Elsolaseya
1 books

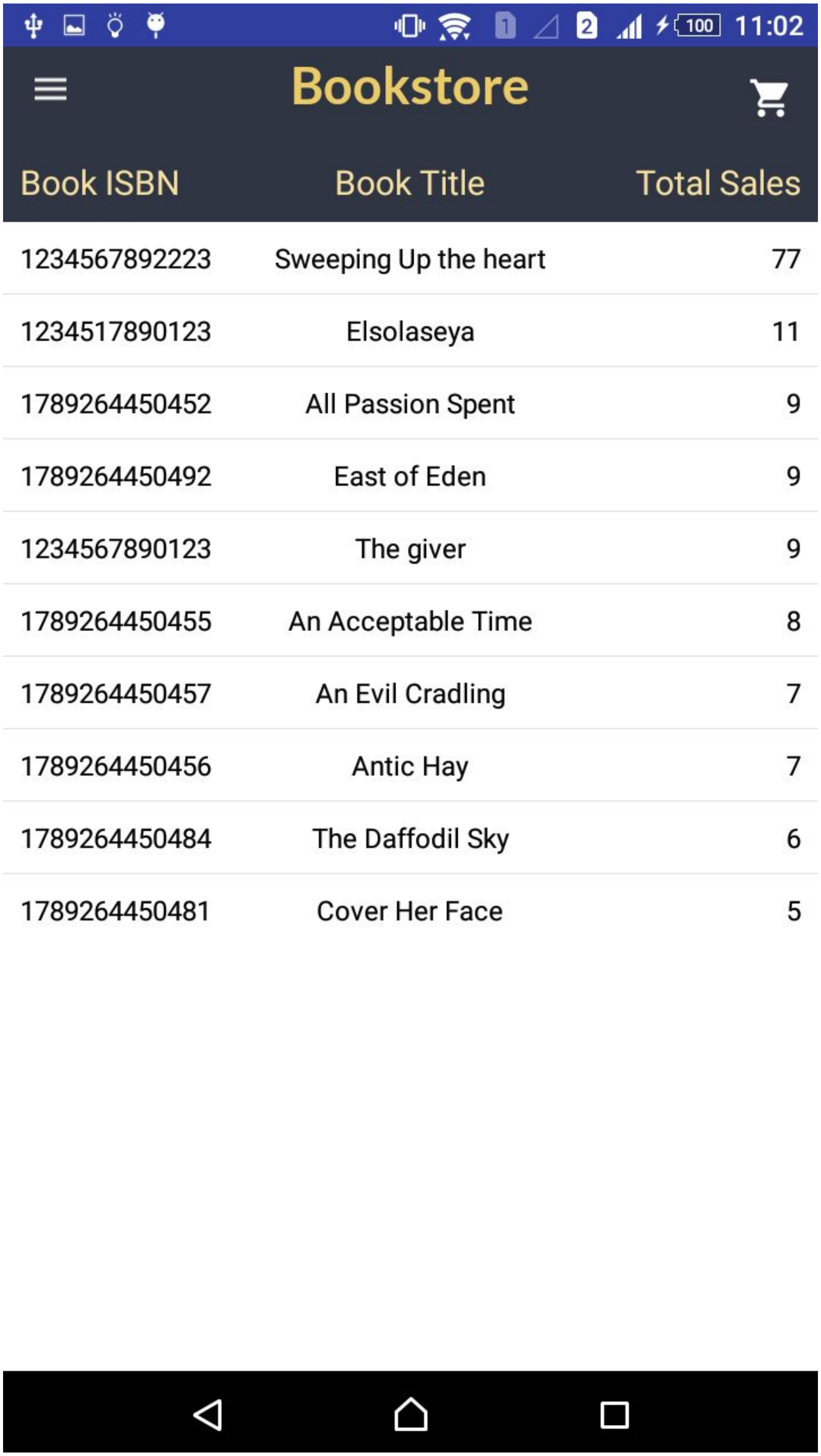


Bookstore

| Book ISBN | Book Title | Total Sales |
|---|---|---|
| 1234517890123 | Elsolaseya | 11 |
| 1234567890000 | The silence of girls | 1 |
| 1234567890123 | The giver | 9 |
| 1234567892223 | Sweeping Up the heart | 77 |
| 1789264450448 | After Many a Summer Dies the Swan | 3 |
| 1789264450449 | Ah, Wilderness! | 3 |
| 1789264450450 | Alien Corn (play) | 3 |
| 1789264450451 | The Alien Corn (short story) | 3 |
| 1789264450452 | All Passion Spent | 9 |
| 1789264450453 | All the King's Men | 5 |
| 1789264450454 | Alone on a Wide, Wide Sea | 1 |
| 1789264450455 | An Acceptable Time | 8 |
| 1789264450456 | Antic Hay | 7 |
| 1789264450457 | An Evil Cradling | 7 |
| 1789264450458 | Arms and the Man | 3 |
| 1789264450459 | As I Lay Dying | 3 |

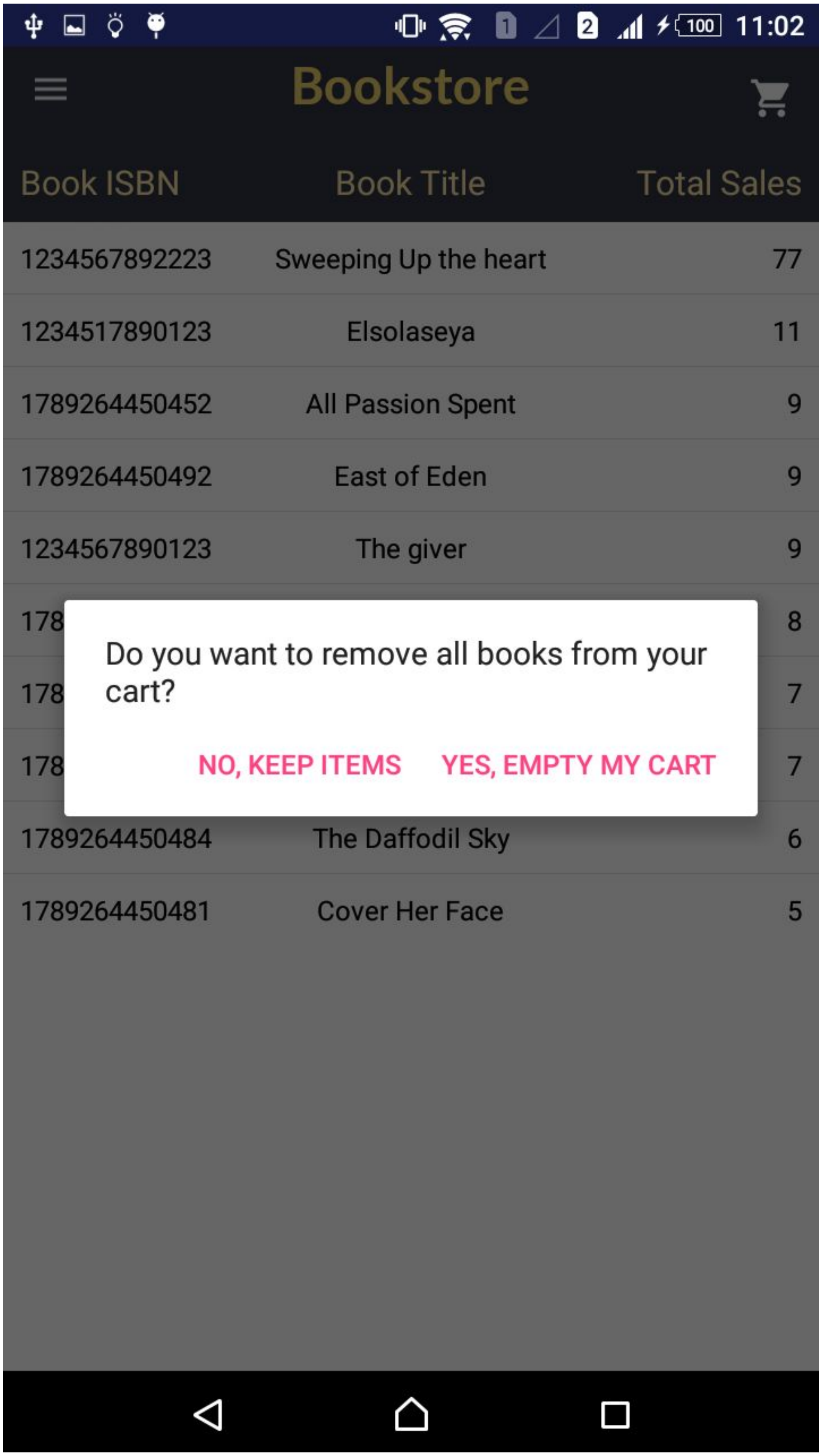


| Username | Total Purchases |
|---|---|
| youmy | 15465.91 EGP |
| johnsmith | 1464.90 EGP |
| carlhenson | 506.85 EGP |
| john | 307.44 EGP |



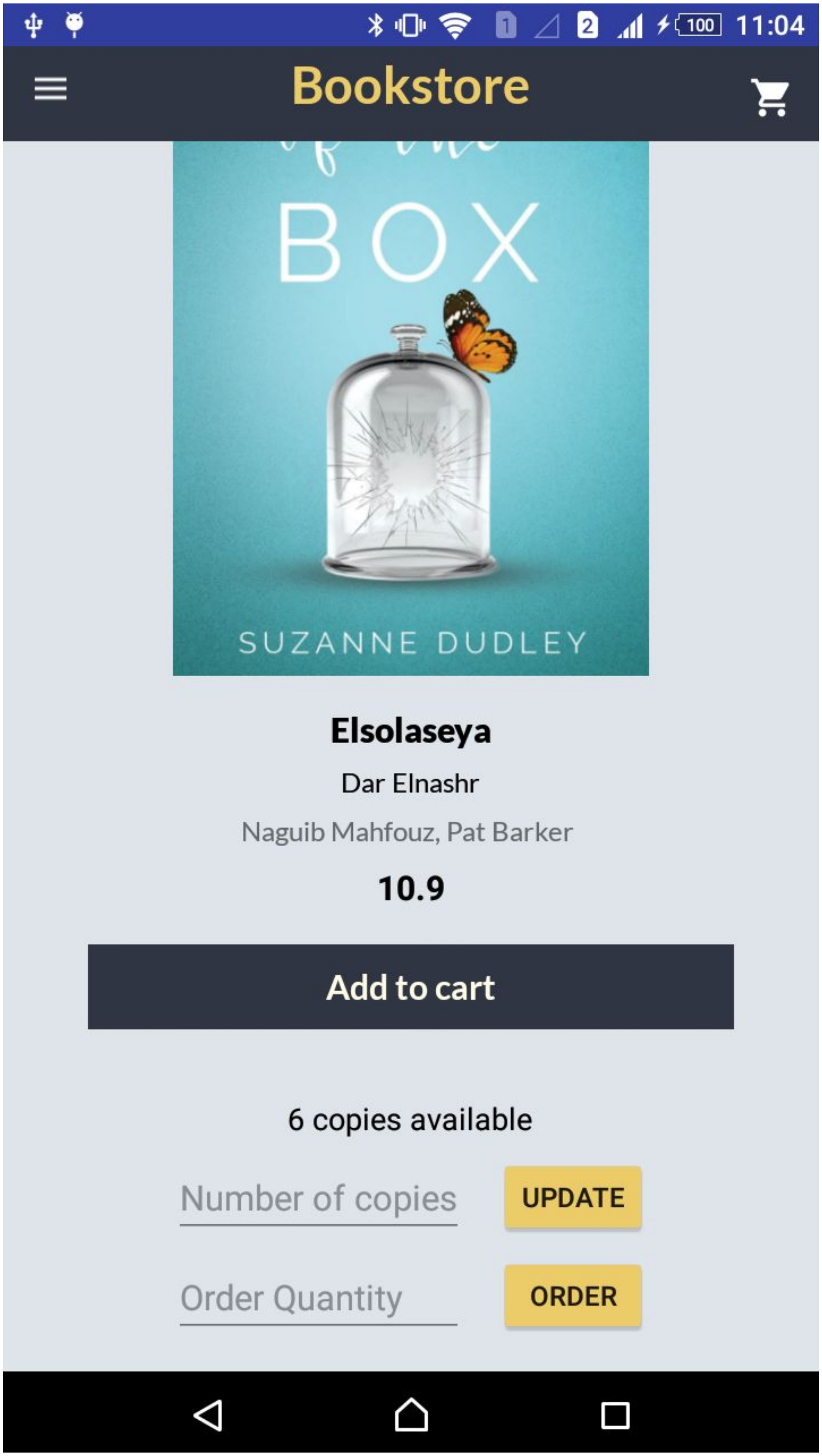
Bookstore

| Book ISBN | Book Title | Total Sales |
|---|---|---|
| 1234567892223 | Sweeping Up the heart | 77 |
| 1234517890123 | Elsolaseya | 11 |
| 1789264450452 | All Passion Spent | 9 |
| 1789264450492 | East of Eden | 9 |
| 1234567890123 | The giver | 9 |
| 1789264450455 | An Acceptable Time | 8 |
| 1789264450457 | An Evil Cradling | 7 |
| 1789264450456 | Antic Hay | 7 |
| 1789264450484 | The Daffodil Sky | 6 |
| 1789264450481 | Cover Her Face | 5 |



| Book ISBN | Book Title | Total Sales |
|---|---|---|
| 1234567892223 | Sweeping Up the heart | 77 |
| 1234517890123 | Elsolaseya | 11 |
| 1789264450452 | All Passion Spent | 9 |
| 1789264450492 | East of Eden | 9 |
| 1234567890123 | The giver | 9 |
| 178 | | 8 |
| 178 | | 7 |
| 178 | | 7 |
| 1789264450484 | The Daffodil Sky | 6 |
| 1789264450481 | Cover Her Face | 5 |

Do you want to remove all books from your cart?



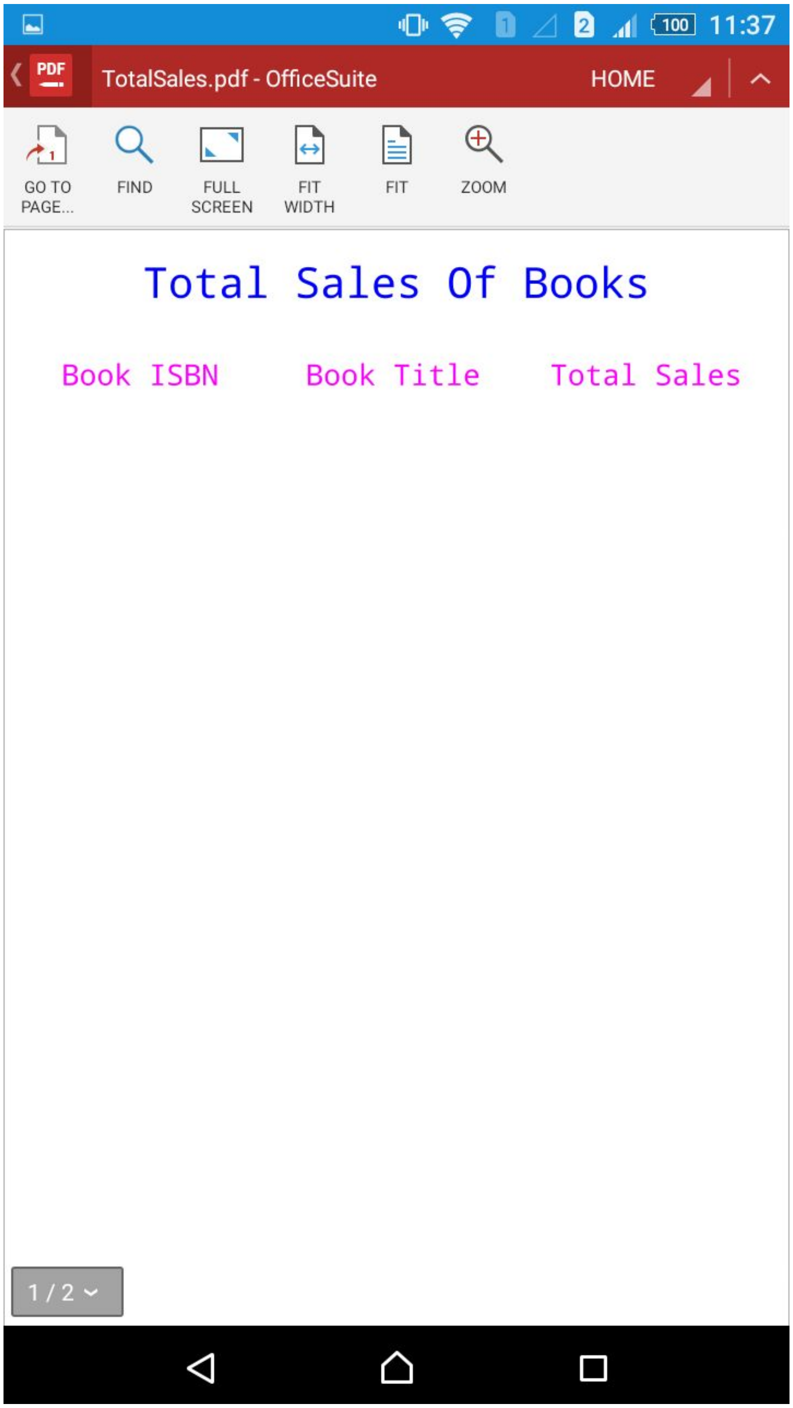
**Elsolaseya**

Dar Elnashr

Naguib Mahfouz, Pat Barker

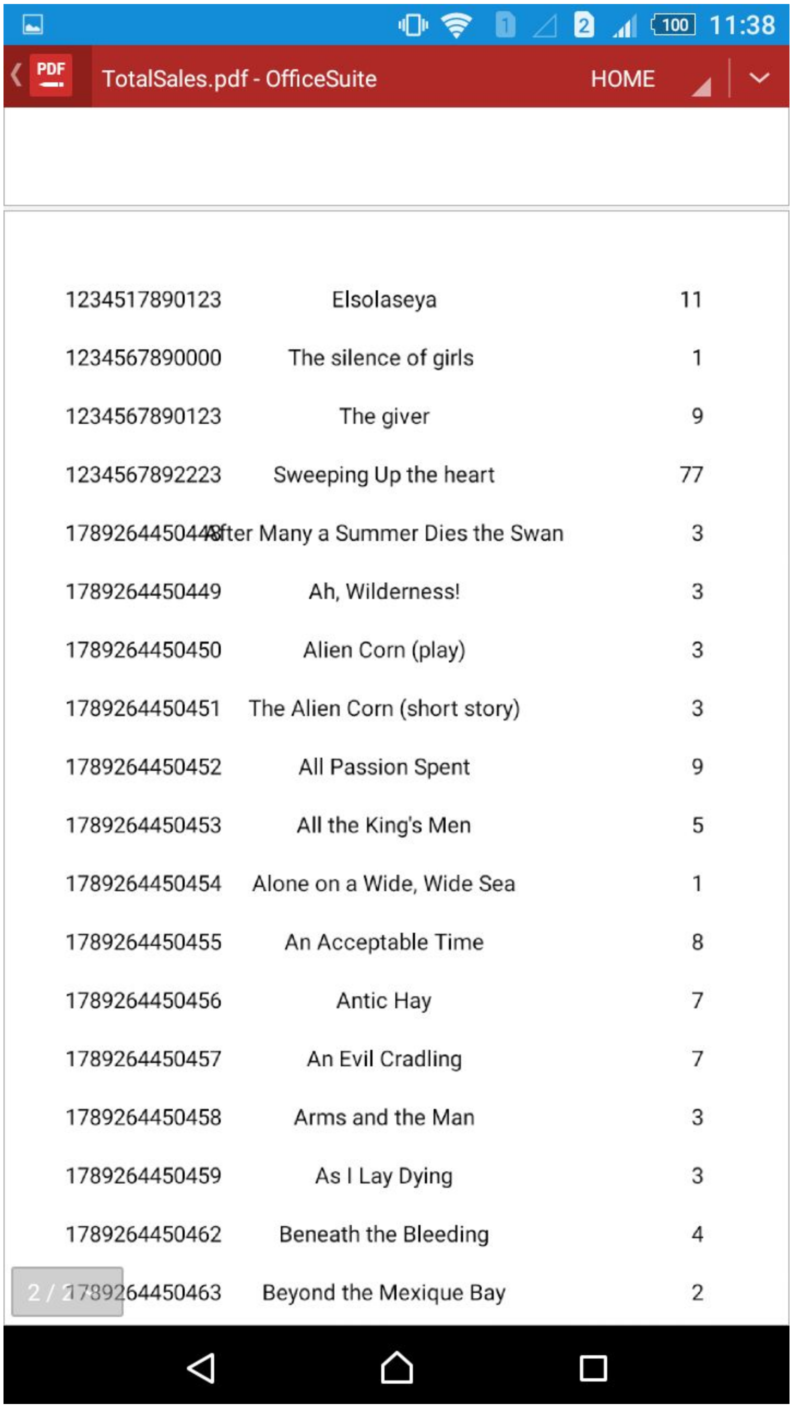
**10.9**

6 copies available

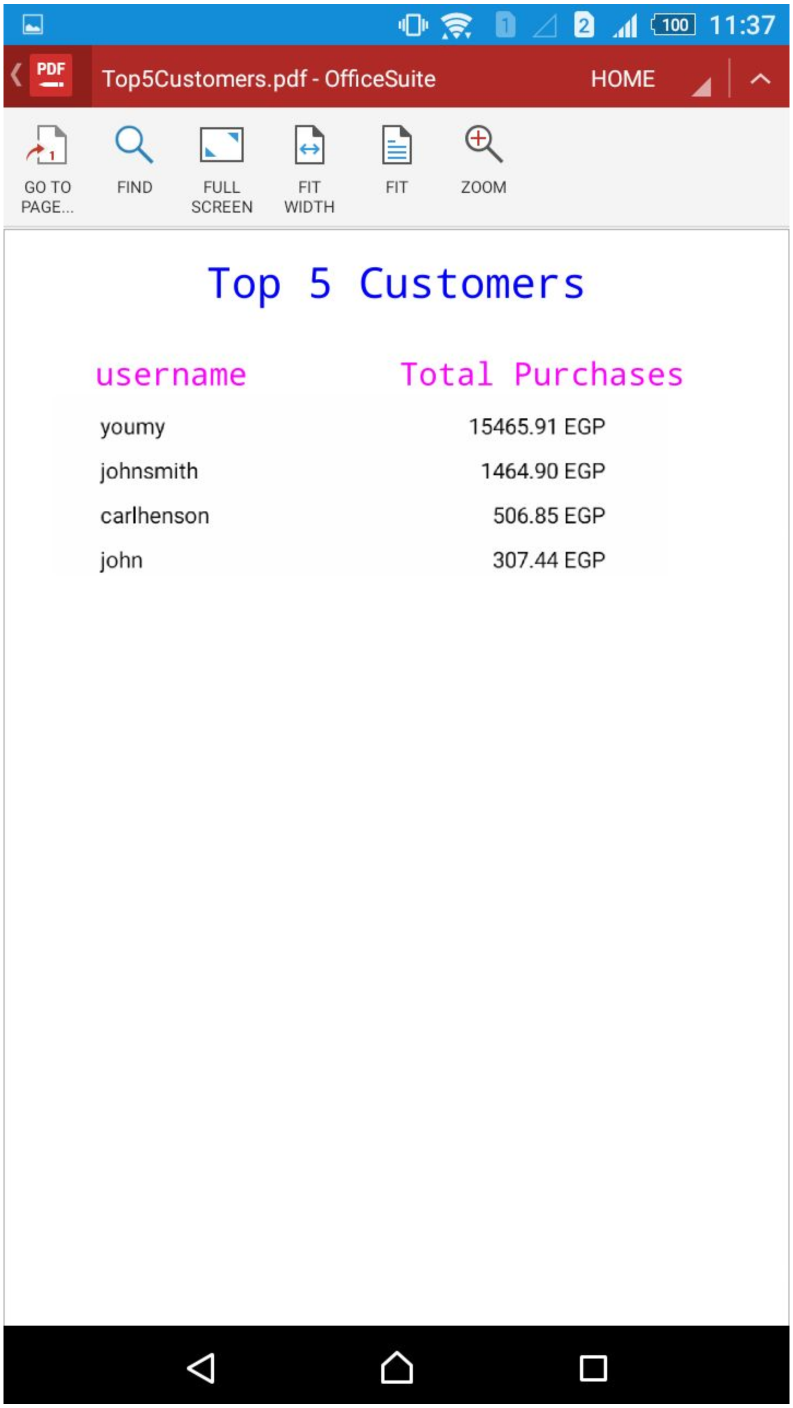


Total Sales Of Books

| Book ISBN | Book Title | Total Sales |
|---|---|---|

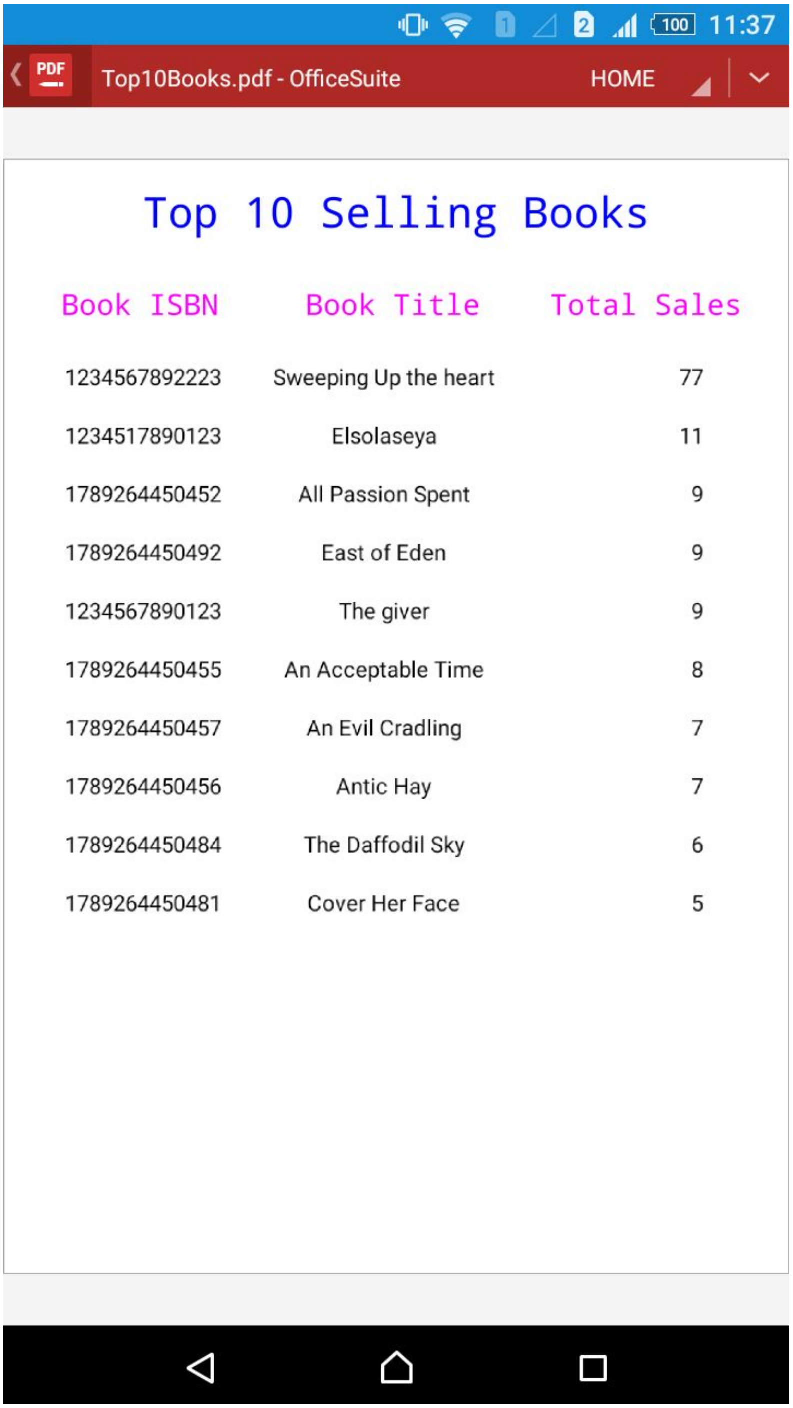


| | | |
|---|---|---|
| 1234517890123 | Elsolaseya | 11 |
| 1234567890000 | The silence of girls | 1 |
| 1234567890123 | The giver | 9 |
| 1234567892223 | Sweeping Up the heart | 77 |
| 1789264450448 | After Many a Summer Dies the Swan | 3 |
| 1789264450449 | Ah, Wilderness! | 3 |
| 1789264450450 | Alien Corn (play) | 3 |
| 1789264450451 | The Alien Corn (short story) | 3 |
| 1789264450452 | All Passion Spent | 9 |
| 1789264450453 | All the King's Men | 5 |
| 1789264450454 | Alone on a Wide, Wide Sea | 1 |
| 1789264450455 | An Acceptable Time | 8 |
| 1789264450456 | Antic Hay | 7 |
| 1789264450457 | An Evil Cradling | 7 |
| 1789264450458 | Arms and the Man | 3 |
| 1789264450459 | As I Lay Dying | 3 |
| 1789264450462 | Beneath the Bleeding | 4 |
| 1789264450463 | Beyond the Mexique Bay | 2 |



Top 5 Customers

| username | Total Purchases |
|---|---|
| youmy | 15465.91 EGP |
| johnsmith | 1464.90 EGP |
| carlhenson | 506.85 EGP |
| john | 307.44 EGP |



Top 10 Selling Books

| Book ISBN | Book Title | Total Sales |
|---|---|---|
| 1234567892223 | Sweeping Up the heart | 77 |
| 1234517890123 | Elsolaseya | 11 |
| 1789264450452 | All Passion Spent | 9 |
| 1789264450492 | East of Eden | 9 |
| 1234567890123 | The giver | 9 |
| 1789264450455 | An Acceptable Time | 8 |
| 1789264450457 | An Evil Cradling | 7 |
| 1789264450456 | Antic Hay | 7 |
| 1789264450484 | The Daffodil Sky | 6 |
| 1789264450481 | Cover Her Face | 5 |