Look Inna Book

Bonita Hout
101188038

Carleton University
Computer Science
COMP 3005 Database Management Systems
Prof. Ahmed El-Roby and Abdelghny Orogat
December 11th, 2022

Table of Contents

Problem Statement

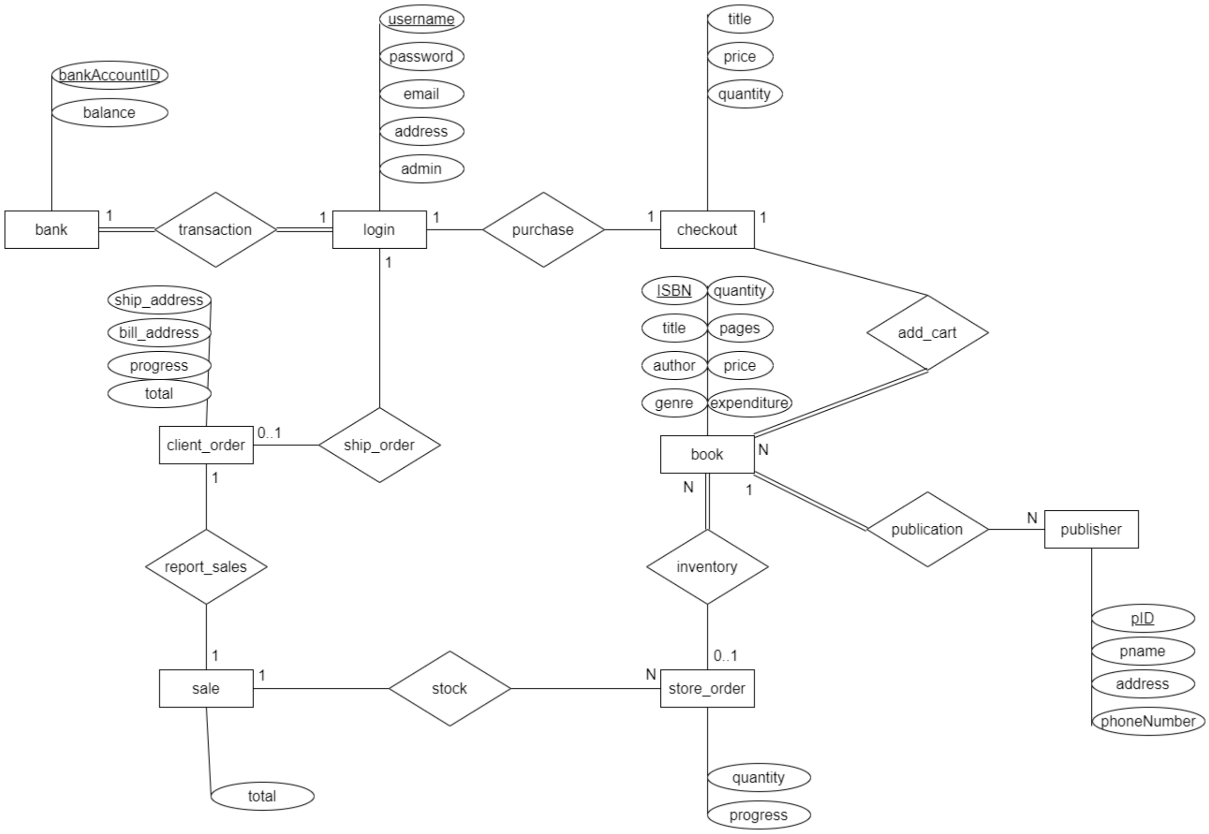
Design and implement an application for an online bookstore, Look Inna Book. This application lets users browse a collection of books that are available in the bookstore. A user can search the bookstore by book name, author name, ISBN, genre, etc.. When a book is selected, information on the author(s), genre(s), publisher, number of pages, price, etc. can be viewed. A user can select as many books as she likes to be added to the checkout basket. A user needs to be registered in the bookstore to be able to checkout. When checking out, the user inserts billing and shipping information (can be different than those used in registration), and completes the order. The bookstore has the feature of tracking an order via an order number. A user can use this order number to track where the order is currently. Although shipping is carried out by a third-party shipping service, the online bookstore should have the tracking information available for when the user inquires about an order using the order number. Assume all books are shipped from only one warehouse (no multiple order numbers for multiple books shipped from multiple warehouses). The bookstore owners can add new books to their collections, or remove books from their store. They also need to store information on the publishers of books such as name, address, email address, phone number(s), banking account, etc.. The banking account for publishers is used to transfer a percentage of the sales of books published by these publishers. This percentage is variable and changes from one book to another. The owners should have access to reports that show sales vs. expenditures, sales per genres, sales per author, etc.. The application should also be able to automatically place orders for new books if the remaining quantity is less than a given threshold (e.g., 10 books). This is done by sending an email to the publisher of the limited books to order a number of books equal to how many books were sold in the previous month (you do not have to implement the email sending component).

Conceptual Design

- Bank
  - keeps track of amount spent by user, or amount received by publisher or store owner(s)
- Login
  - keeps track who has registered, allows registration, keeps track of who is a client/customer or owner/admin
- Publisher
  - information about the publisher
- Book
  - information about a book
- Stock
  - keeps track of the number of 'book' in the store
- Checkout
  - keeps track of the 'book' the customer would like to order
- Client_order
  - ships the customers' order to the shipping address, changes the balance amount for customer, owner, and publisher
- Store_order
  - when there is less than 10 of a certain 'book', order an additional set from the publisher
- Sales
  - keeps track of the sales from the order's from customers

Assumptions:

- There is a total participation relation of book to publisher
  - Every book only has one publisher
  - Every publisher may have published multiple books
- One checkout transaction per client/customer/user
- Total participation between book and checkout
- Bookstore is one location/inventory



Reduction to Relation Schema

bank (bankAccountID, balance)

login (username, password, email, address, admin, bankAccountID)

publisher (pID, name, address, email, phoneNumber)

book (ISBN, title, author, genre, quantity, publisher, pages, price, expenditure)

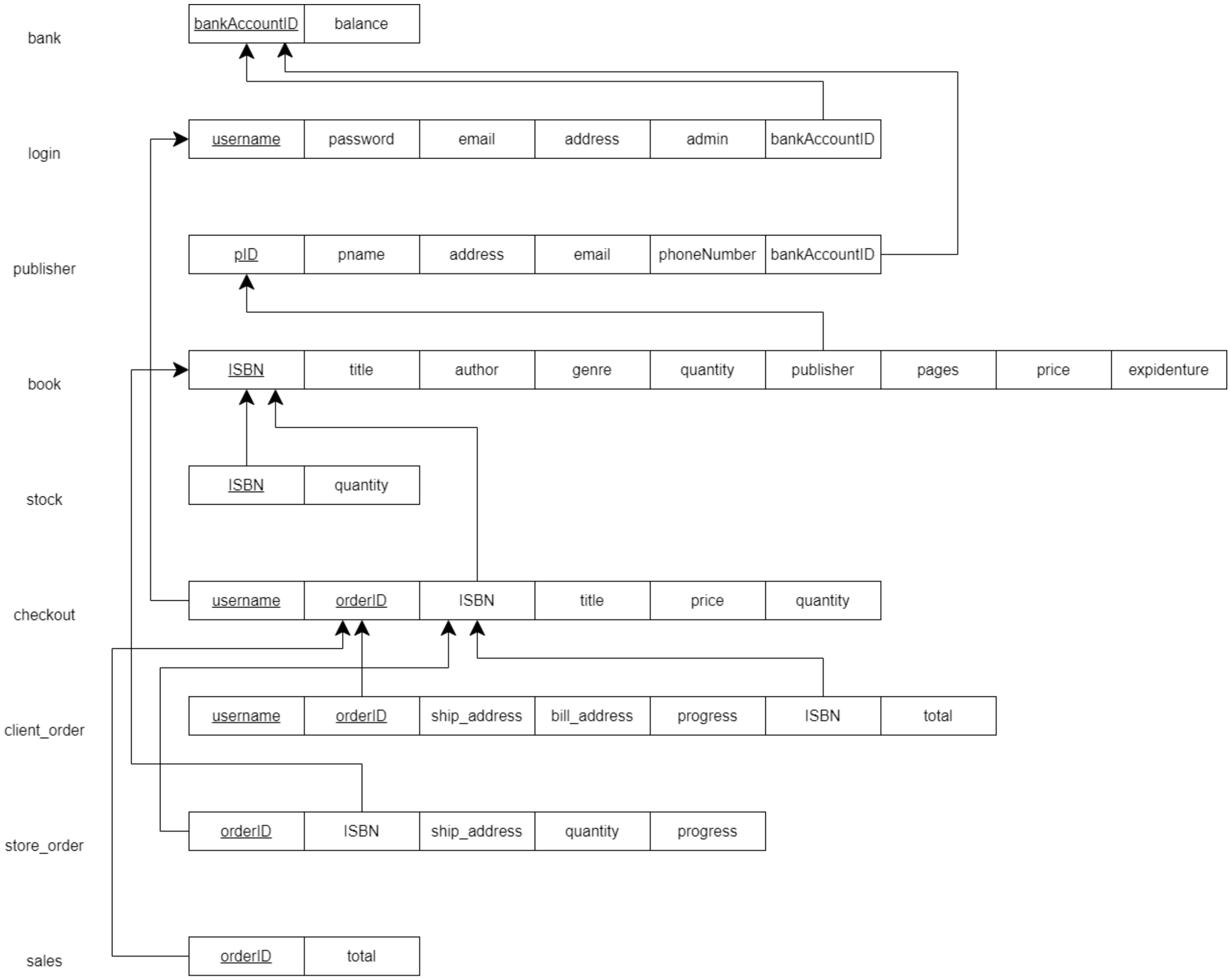
stock (ISBN, quantity)

checkout (username, orderID, ISBN, title, price, quantity)

client_order (username, oderID, ship_address, bill_address, progress, total, ISBN)

store_order (orderID, quantity, progress, ISBN)

sales (orderID, total, ISBN)



bank	bankAccountID	balance							
login	username	password	email	address	admin	bankAccountID			
publisher	pID	pname	address	email	phoneNumber	bankAccountID			
book	ISBN	title	author	genre	quantity	publisher	pages	price	expidenture
stock	ISBN	quantity							
checkout	username	orderID	ISBN	title	price	quantity			
client_order	username	orderID	ship_address	bill_address	progress	ISBN	total		
store_order	orderID	ISBN	ship_address	quantity	progress				
sales	orderID	total							

—---

Normalization of Relation Schemas

bank (<u>bankAccountID</u>, balance)

Let A stand for bankAccountID, let B stand for balance

R = {A, B}

F = {A -> B}

$\{A\}^{+}$= {B}

- Satisfies both BCNF and 3NF

login (<u>username</u>, password, email, address, admin, <u>bankAccountID</u>)

Let A stand for *username*, let B stand for *password*, …, let E stand for *admin*

R = {A, B, C, D, E, F}

F = {

A -> BCDEF

}

$\{A\}^{+}$= {B, C, D, E, F}

- Satisfies both BCNF and 3NF

publisher (<u>pID</u>, name, address, email, phoneNumber)

Let A stand for *pID*, let B stand for *name*, … , let E stand for *phone*

R = {A, B, C, D, E}

F = {

A -> BCDE

}

$\{A\}^{+}$= {B, C, D, E}

- Satisfies both BCNF and 3NF

book (<u>ISBN</u>, title, author, genre, quantity, publisher, pages, price, expenditure)

Let A stand for *ISBN*, let B stand for *author*, … , let I stand for *expenditure*

R = {A, B, C, D, E, F, G, H, I}

F = {

A -> BCDFG,

EH -> I

}

$\{A\}^{+}$= {B, C, D, E, F, G}

$\{EH\}^{+}$= {I}

- Both satisfies both BCNF and 3NF

stock (<u>ISBN</u>, quantity)

Let A stand for *ISBN,* let B stand for *quantity*

R = {A, B}

F = {

A -> B

}

$\{A\}^{+}$= {B}

- Satisfies both BCNF and 3NF

checkout (username, orderID, ISBN, title, price, quantity)

Let A stand for *username*, let B stand for *orderID*, … , let F stand for *quantity*

R = {A, B, C, D, E, F}

F = {

AB -> CDEF

}

$\{AB\}^+$= {C, D, E, F}

- Satisfies both BCNF and 3NF

client_order (username, oderID, ship_address, bill_address, progress, total, ISBN)

Let A stand for *username*, let B stand for *orderID*, … , let G stand for *ISBN*

R = {A, B, C, D, E, F, G}

F = {

A -> CD

B -> EFG

}

$\{A\}^+$= {C, D}

$\{B\}^+$= {E, F, G}

- Both satisfies both BCNF and 3NF

store_order (orderID, quantity, progress, ISBN)

Let A stand for *ISBN*, let B stand for *quantity*, let D stand for *ISBN*

R = {A, B, C, D}

F = {

A -> CD

B -> D

}

$\{A\}^+$= {C, D}

$\{B\}^+$= {D}

- Both satisfies both BCNF and 3NF

sales (orderID, total, ISBN)

Let A stand for *orderID*, let B stand for *total*, let C stand for *ISBN*

R = {A, B, C}

F = {

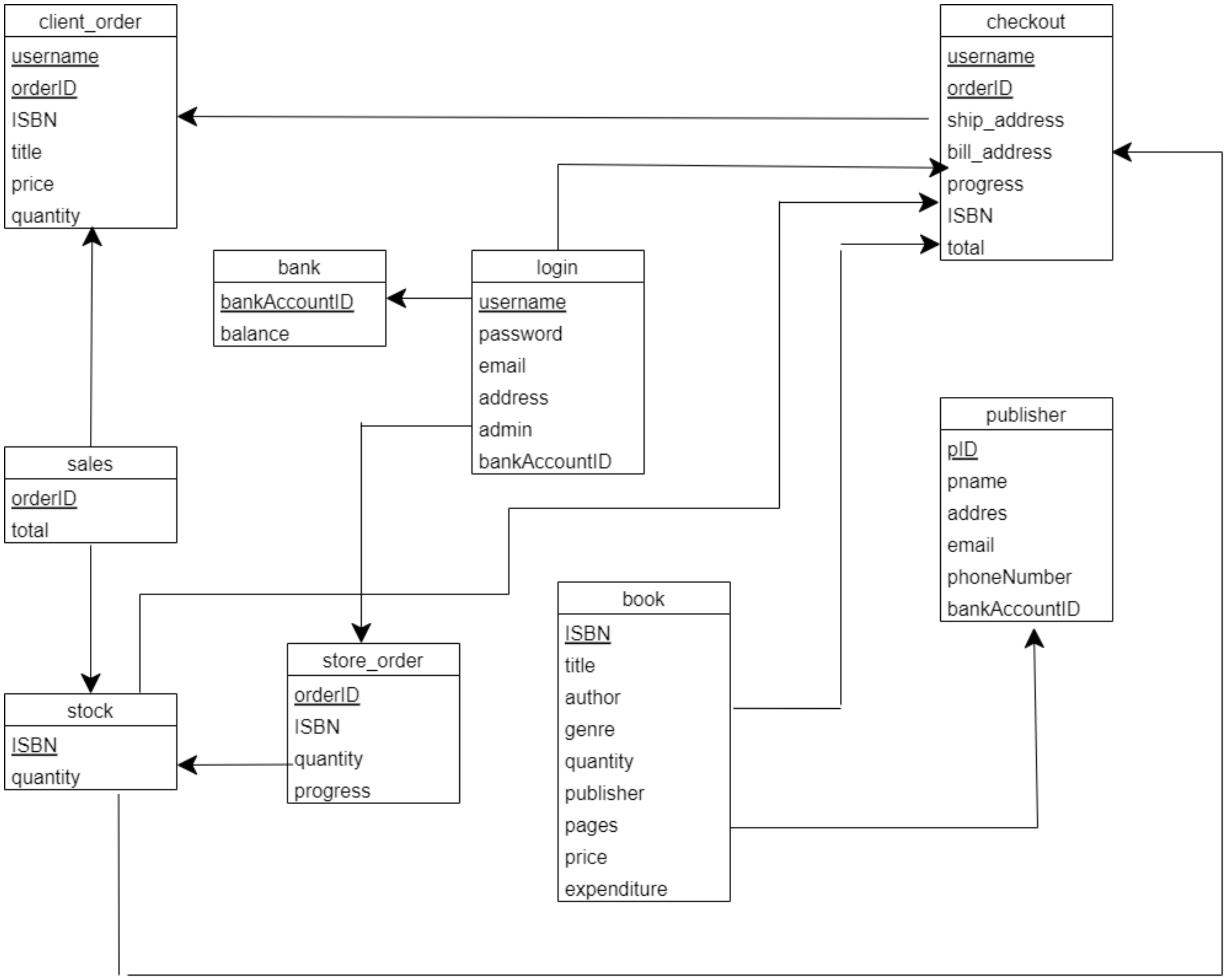
A-> BC

}

$\{A\}^{+}$ = {B, C}

- Satisfies both BCNF and 3NF

Database Schema Diagram



Implementation

This program uses Python version 3.10.2 with psycopg2 version 2.9.5. Users will first login into the program, and can register if needed. As a customer, users will be able to search for their desired book, add the book into their cart, and submit for checkout and wait for shipment. As the owner, the user is able to view all registered users in the bookstore and view all publishers in the book store. This application is incomplete. Ideally, the owner would also be able to view the sales vs expenditures and add or remove a book from their store. As this application is incomplete, some other features may not be present and input checking is limited (program assumes correct input).

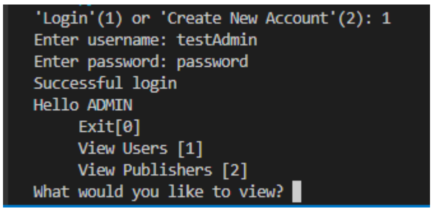
Program files include:

- main.py
- admin.py

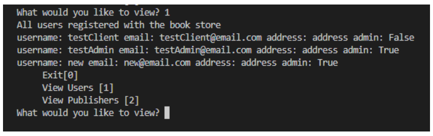
- login.py
- checkout.py
- defs.py

LOGIN

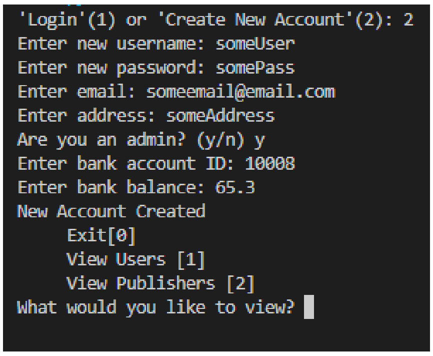
- User will login the program, or create an account if needed



```
 'Login'(1) or 'Create New Account'(2): 1
 Enter username: testAdmin
 Enter password: password
 Successful login
 Hello ADMIN
      Exit[0]
      View Users [1]
      View Publishers [2]
 What would you like to view? █
```



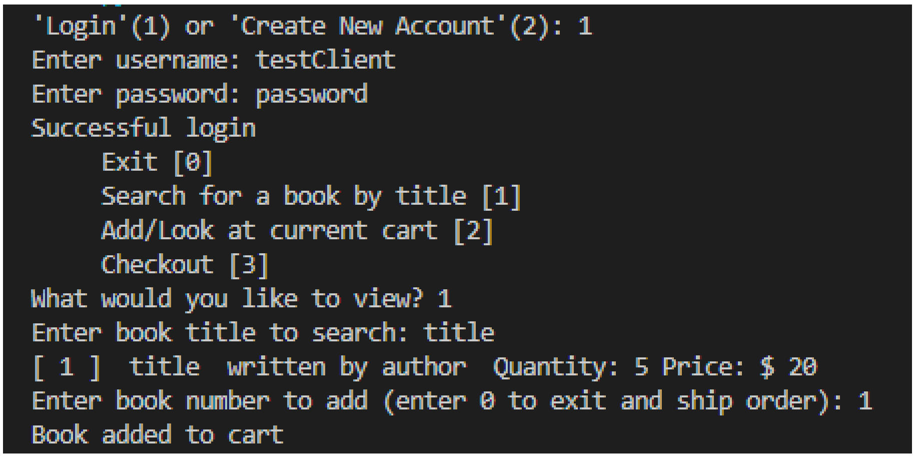
```
 What would you like to view? 1
 All users registered with the book store
 username: testClient email: testClient@email.com address: address admin: False
 username: testAdmin email: testAdmin@email.com address: address admin: True
 username: new email: new@email.com address: address admin: True
      Exit[0]
      View Users [1]
      View Publishers [2]
 What would you like to view? █
```



```
 'Login'(1) or 'Create New Account'(2): 2
 Enter new username: someUser
 Enter new password: somePass
 Enter email: someemail@email.com
 Enter address: someAddress
 Are you an admin? (y/n) y
 Enter bank account ID: 10008
 Enter bank balance: 65.3
 New Account Created
      Exit[0]
      View Users [1]
      View Publishers [2]
 What would you like to view? █
```

USER: BOOK SEARCH

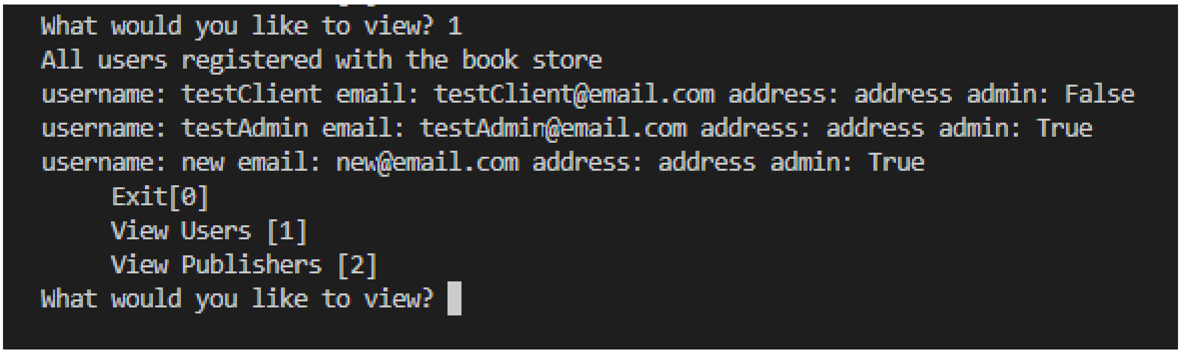
- Customer will be able to search for a book by title



```
 'Login'(1) or 'Create New Account'(2): 1
 Enter username: testClient
 Enter password: password
 Successful login
      Exit [0]
      Search for a book by title [1]
      Add/Look at current cart [2]
      Checkout [3]
 What would you like to view? 1
 Enter book title to search: title
 [ 1 ]  title  written by author  Quantity: 5 Price: $ 20
 Enter book number to add (enter 0 to exit and ship order): 1
 Book added to cart
```

OWNER: ALL USERS

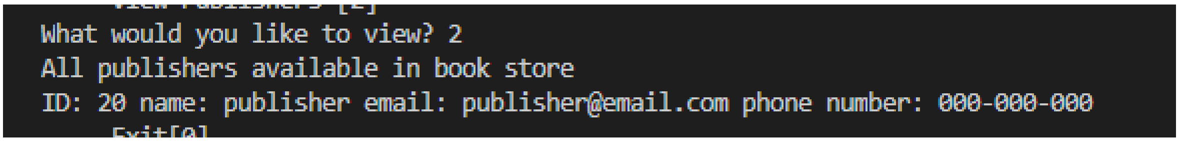
- Output all users registered with the bookstore



```
 What would you like to view? 1
 All users registered with the book store
 username: testClient email: testClient@email.com address: address admin: False
 username: testAdmin email: testAdmin@email.com address: address admin: True
 username: new email: new@email.com address: address admin: True
      Exit[0]
      View Users [1]
      View Publishers [2]
 What would you like to view? █
```

OWNER: ALL PUBLISHERS

- Output all the publishers in the bookstore



```
        View Publishers [2]
  What would you like to view? 2
  All publishers available in book store
  ID: 20 name: publisher email: publisher@email.com phone number: 000-000-000
        Exit[0]
```

Bonus Features

No bonus features were added.

GitHub Repository

https://github.com/bonita-h/COMP-3005.git
Includes a diagram folder, python folder, and SQL folder.

Appendix I

Available 11:00 pm, 11:30 pm, and 12:00 pm on December 12th