**DB System Implementation**

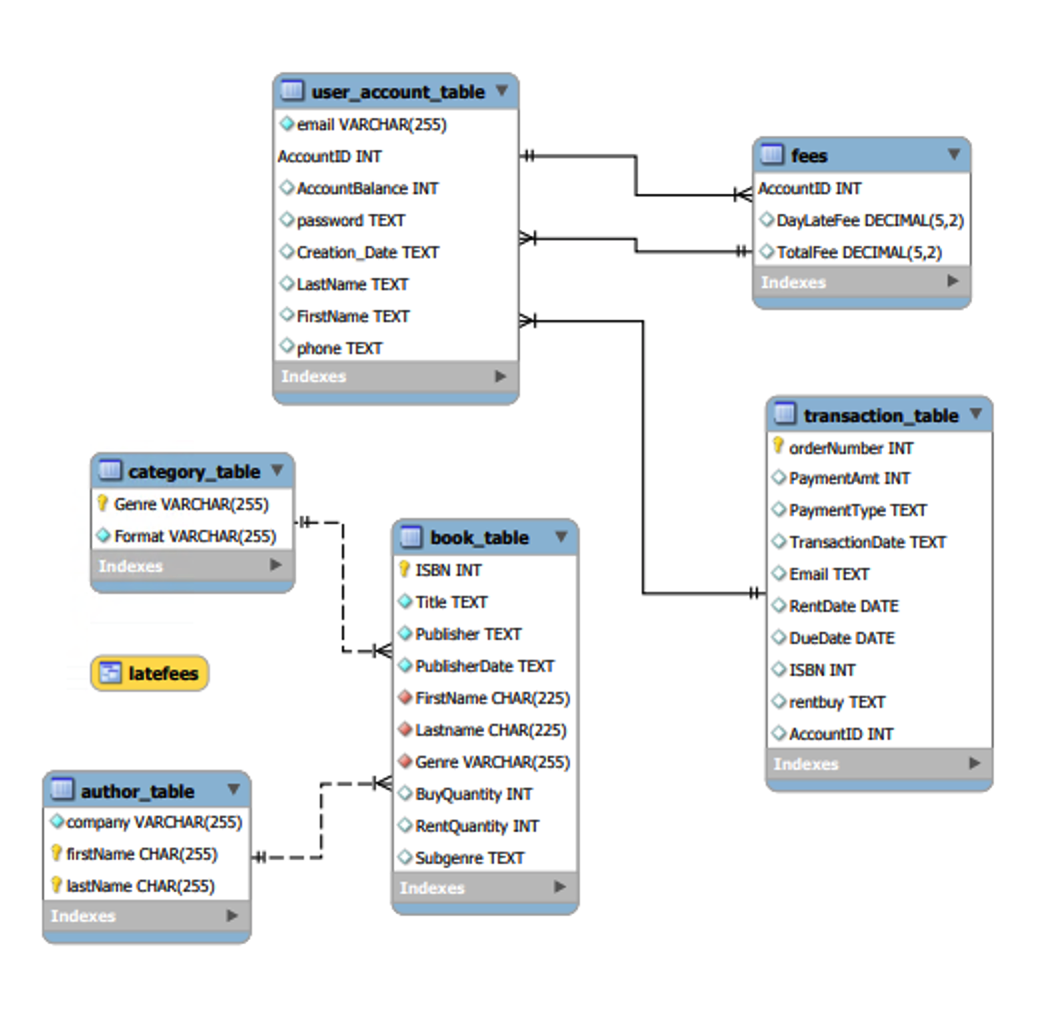
Team OpenDB

  
  
  
*Created by Cameron McLemore, Jon Transfiguracion, Tabitha Singh, Truc Tran, and Daniel Johnson*

**Business Rules**

1. Customers cannot rent more than two books at a time.
2. DueDate cannot be prior to Transaction\_Date
3. DueDate is 4 days after TransactionDate if current date is less than or equal to 2 month from PublishDate of BOOK.
4. DueDate is 5 days after TransactionDate if current date is more than 2 months from PublishDate of BOOK.
5. If the current date is more than 2 months from PublishDate of BOOK and Type of TRANSACTION\_LINE equals ‘rent’, PaymentAmt is $4.50.
6. If the current date is less than or equal to 2 months from PublishDate of BOOK and Type of TRANSACTION\_LINE equals ‘rent’, PaymentAmt is $3.00.
7. $2.00 is added to the AccountBalance of USER\_ACCOUNT per day past DueDate.
8. DueDate is null if Type of TRANSACTION\_LINE is ‘buy’.
9. If the current date is 15 days from TransactionDate, AccountBalance equals AccountBalance plus Price of BOOK.
10. Customers cannot have a new transaction if AccountBalance does not equal $0.00
11. Each book has a genre but not every genre has a book in stock.We dop this by show how many books are in stock.
12. we did this by having a condition where if dateslate > 15 then the latefees will be added to a users account for $15
13. This can be done by pressing 1 under admin tab to show the stores inventory and all the information necessary.
14. the admin can generate reports of all information and apply lates fees in section 3 and 4 in the admin tab.
15. The admin can insert to any table and request a book for a user by using the insert update and delete tab in the admin menu

**Database Design Diagram (Crows-Foot Model)**



**Application’s Major Classes**

* Class UserOptions
  + The UserOptions class serves as the main entry point of the application. This class provides users with the options to navigate through the user friendly application and retrieve information from the database. The class let the user navigate through two user options, and ask whether the user is an admin or a normal user. As a user, they will be able to search for titles from the database, buy or rent a book, return books and check their balances as well as pay their balance. As an admin, they will be able to find titles from the database, update inventory of books, check user’s balance and apply late fee, and generate report listings.
* class Connection

The Connection class works with the UserOptions to form a connection between the application and the database and is made by calling the getConnection method.

* class Book

The Book class is used to store the actual representation of items from the database, each book has fields that correspond to a column from the book table in the database, the way the Book class does this is through an ArrayList<String>.

The Book class connects to the database and gets the records as String data, then places them into the ArrayList<String>.

The Book class is also responsible for the cart method which uses an ArrayList<Object> to make a cart.

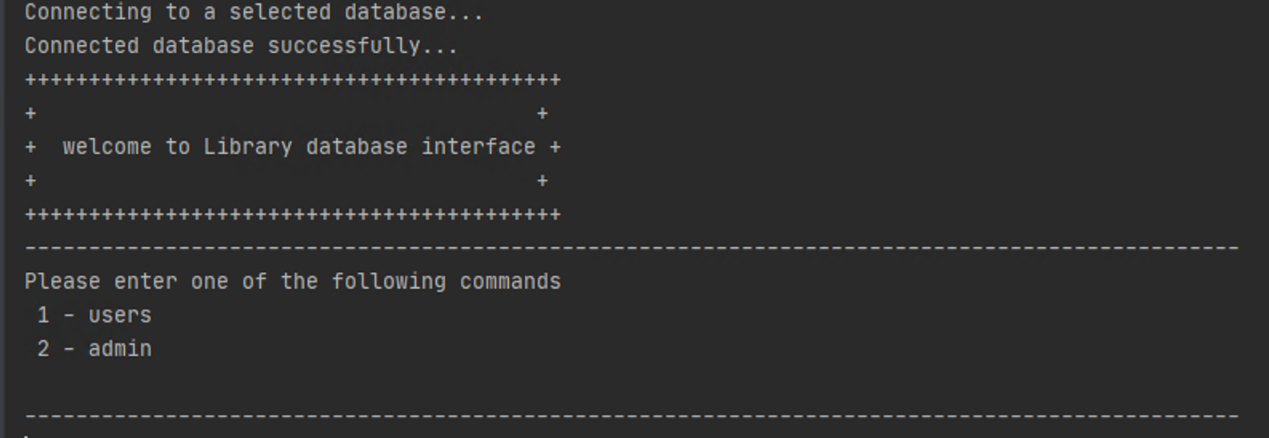
**Application’s Functions**

* getConnection()creates the connection to the DataBase
* receipt()This function displays a receipt to the console with information about the user’s purchase
* CreateDate()This function gets the current date and prints it to the console
* getDueDate()gets the due date and displays the date
* setDueDate()sets the duedate
* rentbook()This function is intended to be used when a book is rented, and updates the rent quantity of a book in the database
* buybook()This function is intended to be used when a book is purchased, and updates the buy quantity of a book in the database
* getUserInfo()This function gets information from the user for use when they purchase a book
* insertdbd()This function inserts a new customer into the database into transaction table
* insertdbdBUY()This function is exactly the same as insertdb but instead of making a due and rent date, it supplements 0000-00-00 due to the fact that there is not date for these.
* UserOptions()This function is one of the two main functions (the other being adminOptions() ). It presents to the user all of the available options, and calls the relevant function to complete the user’s desired task.
* SearchForBook()This function allows the user to search for a book in the database, and add it to their cart to purchase or rent.
* rent\_buy()This function allows the user to either rent or buy a book
* ReturnBook()This function is intended to be used when a book is returned, and updates the rent quantity of a book in the database.
* AccountBalance()This function shows the user their current account balance, and allows them to make a payment on the balance.
* updateQuery()This function updates the book table to add to a rent quantity
* adminOptions()This function is one of the two main functions (the other being userOptions() ). It presents to the admin all of the available options, and calls the relevant function to complete the admin’s desired task.
* ExicuteQuery()This function allows the admin to write an sql statement that will be executed by the database.
* Invetory()This function shows the admin all of the books currently in the database.
* userLogs()This function allows the admin to view all of the users and apply late fees to their account balance.
* ShowAllTables()This function shows the admin all of the table data currently in the database.

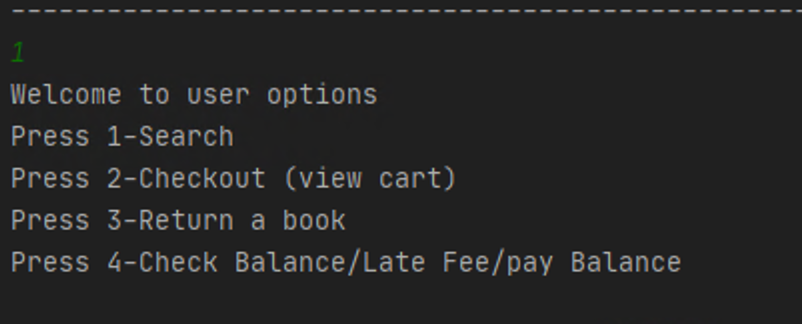
**User’s Manual**

1. Team OpenDB developed a user-friendly application that grants users the ability to navigate the UNF Library as Users or Admins.
   1. When the application is ran, users will be displayed with the following options:

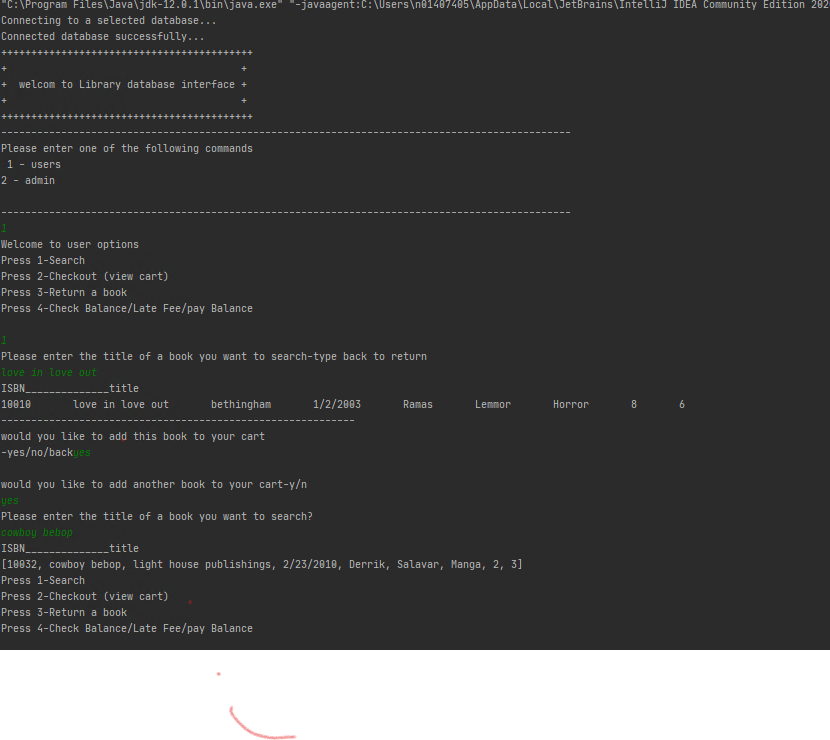
* “1 - users”
* “2  - admin”



1. If the user has selected “1” for user access, they will be directed to the following choices:



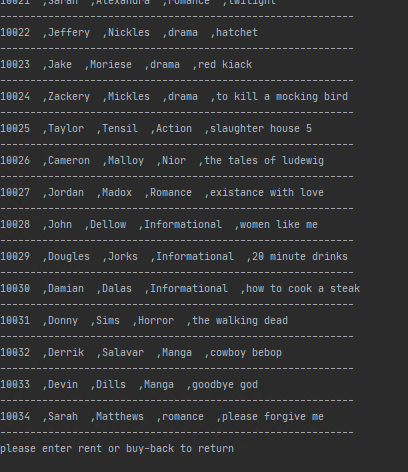
* 1. “press 1-Search”The application will prompt the user for a book to search and will output information for the book entered. The application will also notify the user if the book is in stock. Users will then be given the option to rent or buy the book that was searched

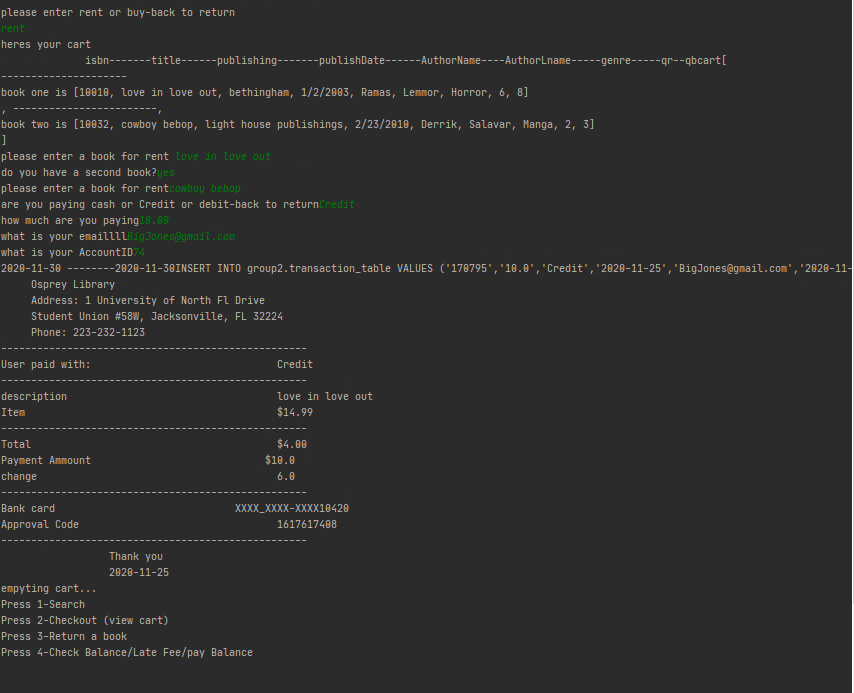


Everybook will be displayed and it will ask if you want to buy or rent a book

The user can decide to rent or buy a book after choosing to Checkout. If the user has chosen to buy, the application will prompt the user for the desired book; the application will ask for the payment type then later display the price and receipt for the book. Likewise, if the customer picks to rent a book, the application will prompt the user for a book and then print the receipt of that book.

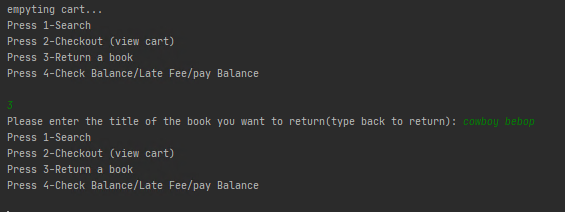
* 1. “press 2-Checkout (rent) / buy”





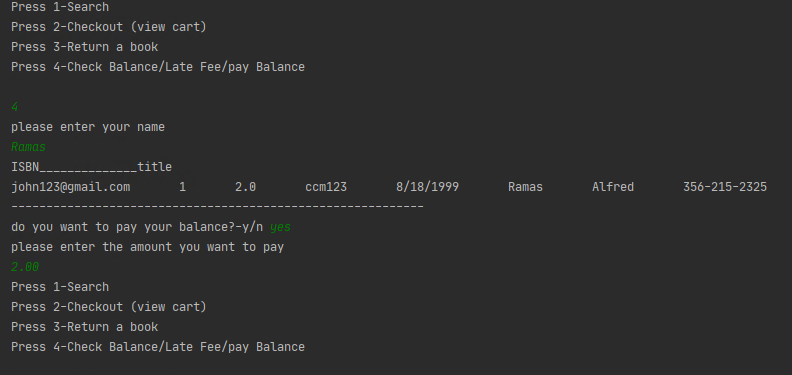
* 1. “press 3-Return”

The application will ask the user for the title of the book the user is returning.

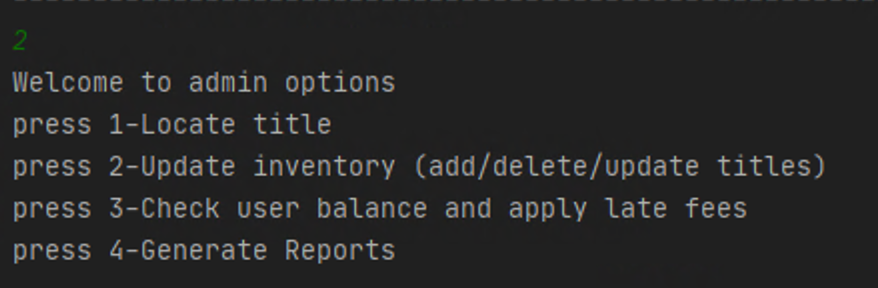


* 1. “press 4-Check Balance/Late Fee/pay Balance”

The application will ask the user for their name then it will print the user account information including the User #ID, the user balance as well as their name. The user will then be given the option to make a payment on their balance.



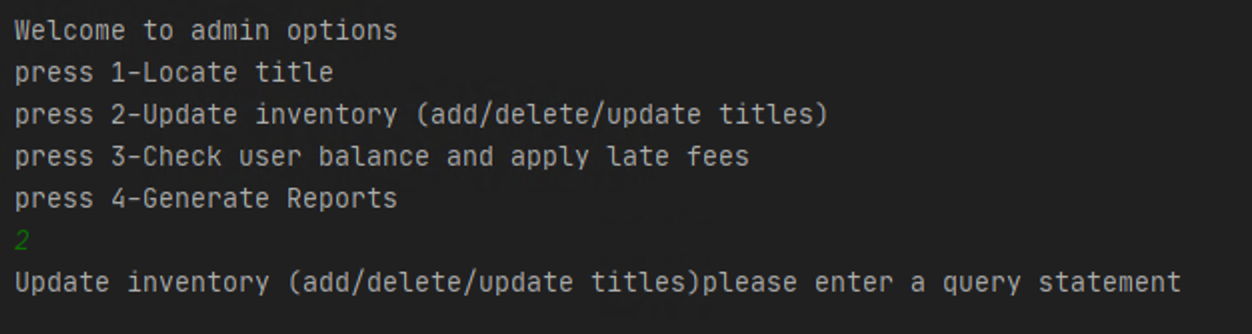
1. If the user has selected “2” for admin access, they will be directed to the following choices:



* 1. “press 1-Locate title”

“Locate title” will generate the title of all books as well as other information such as the ISBN, first and last name of the author, and genre for the admin.

* 1. “press 2-Update inventory (add/delete/update titles)”

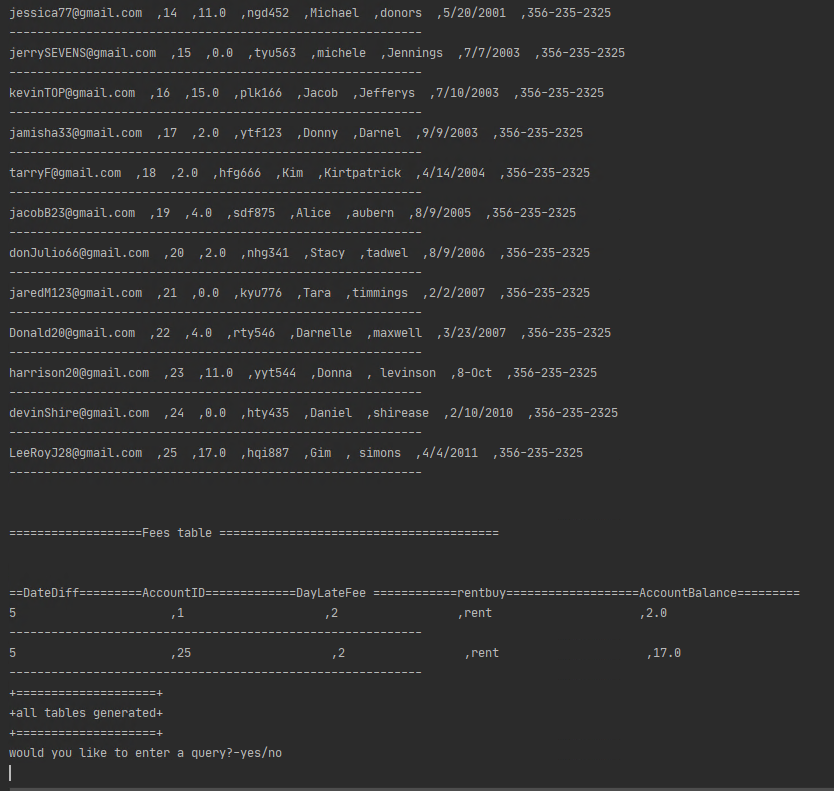


The admin will be able to add, delete, and update information for the inventory within the database.

* 1. “press 3-Check user balance and apply late fees”



After the admin has pressed “3”, the application will grant access to users’ Account IDs, balances, last names, and phone numbers. The admin will then be able to apply any necessary late fees to the account.

* 1. “press 4-Generate Reports”
  2. 

Lastly, the admin is able to generate tables from the UNF Library database. This will display all the tables in the Database and i till create a list of users with an overdue book. it will then go on to allowing the admin to either manually implement or fix an incorrect database entry

*Note: Admins may type “back” in order to quit the program.*

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |