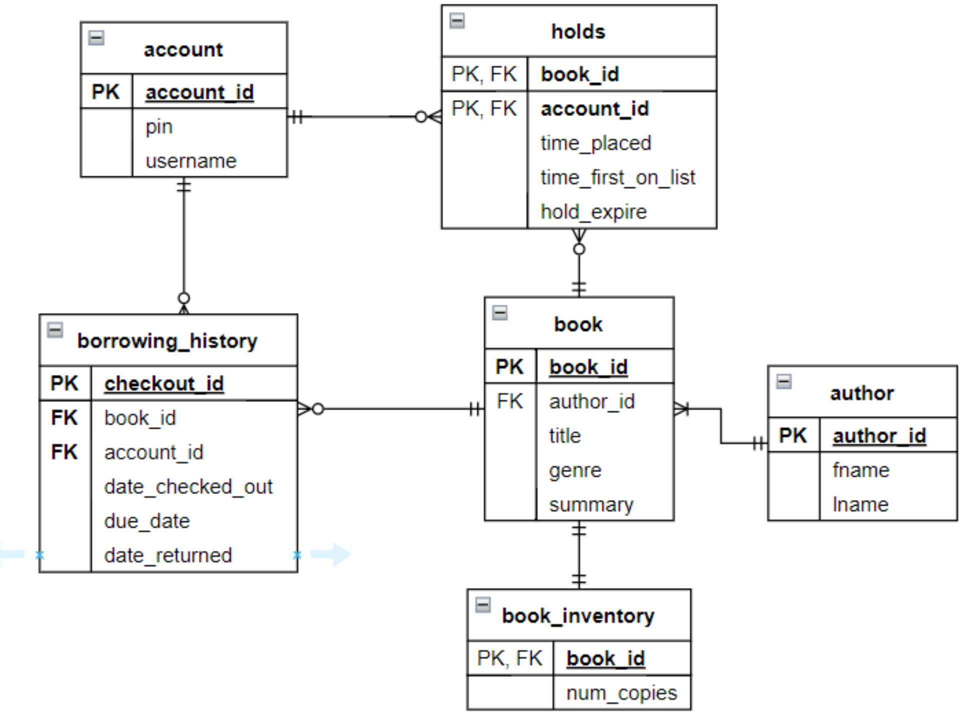
CSCI 366 Final Project

Audrey Tracy

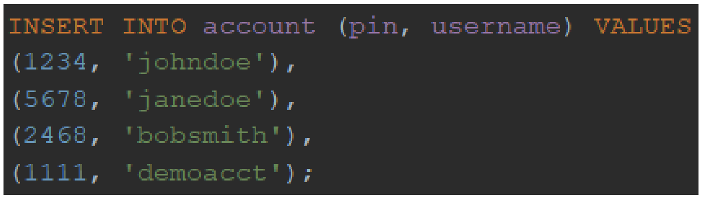
I made a library application with the ability for users to browse books, filter search by title, author last name, or genre, checkout books, place holds if the book is not available, check out the book when it becomes available, and return books / cancel holds.

Library App Database ERD

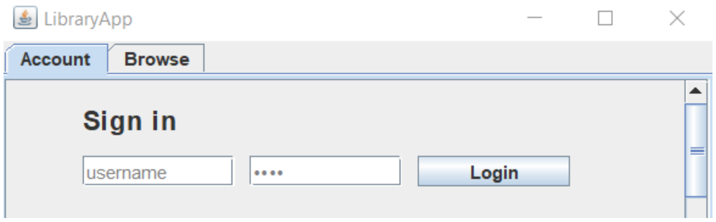


Requirements

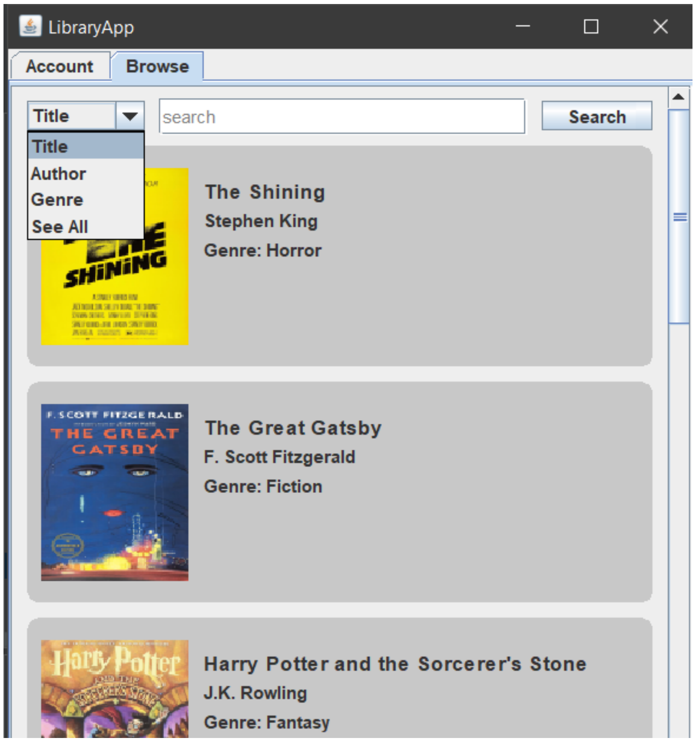
- User will be able to log in
  Note: user cannot make an account (currently). There are four provided accounts for testing:



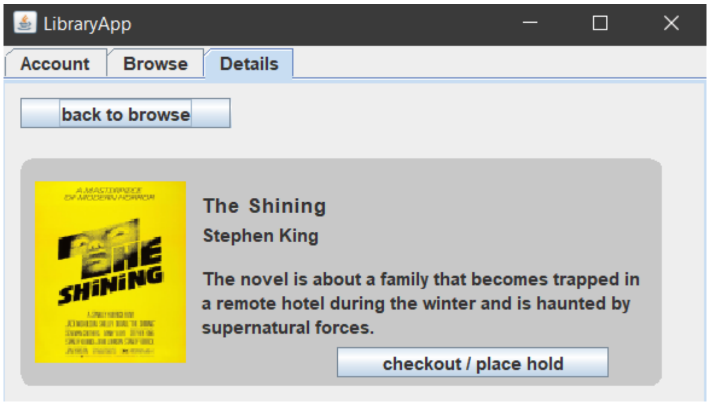
```
INSERT INTO account (pin, username) VALUES
(1234, 'johndoe'),
(5678, 'janedoe'),
(2468, 'bobsmith'),
(1111, 'demoacct');
```



- User will be able to browse books in the library
  Does not require login to browse books
  Filter by all books, search by title, author last name, and genre



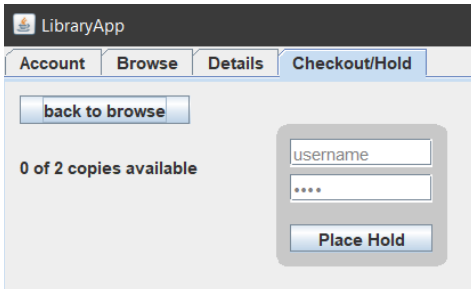
- User will be able to click on a book to see more details (the summary of the book) and to see the option to check the book out



The novel is about a family that becomes trapped in a remote hotel during the winter and is haunted by supernatural forces.

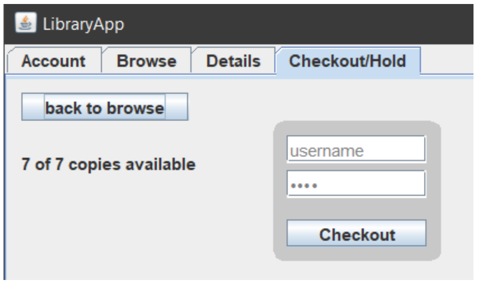
- When the user clicks ‘checkout / place hold’ they will be taken to a screen that allows them to checkout the book (if it is available) or lets them place a hold (if they don’t already have a hold on that book & if they don’t currently have all of the copies checked out)

Hold



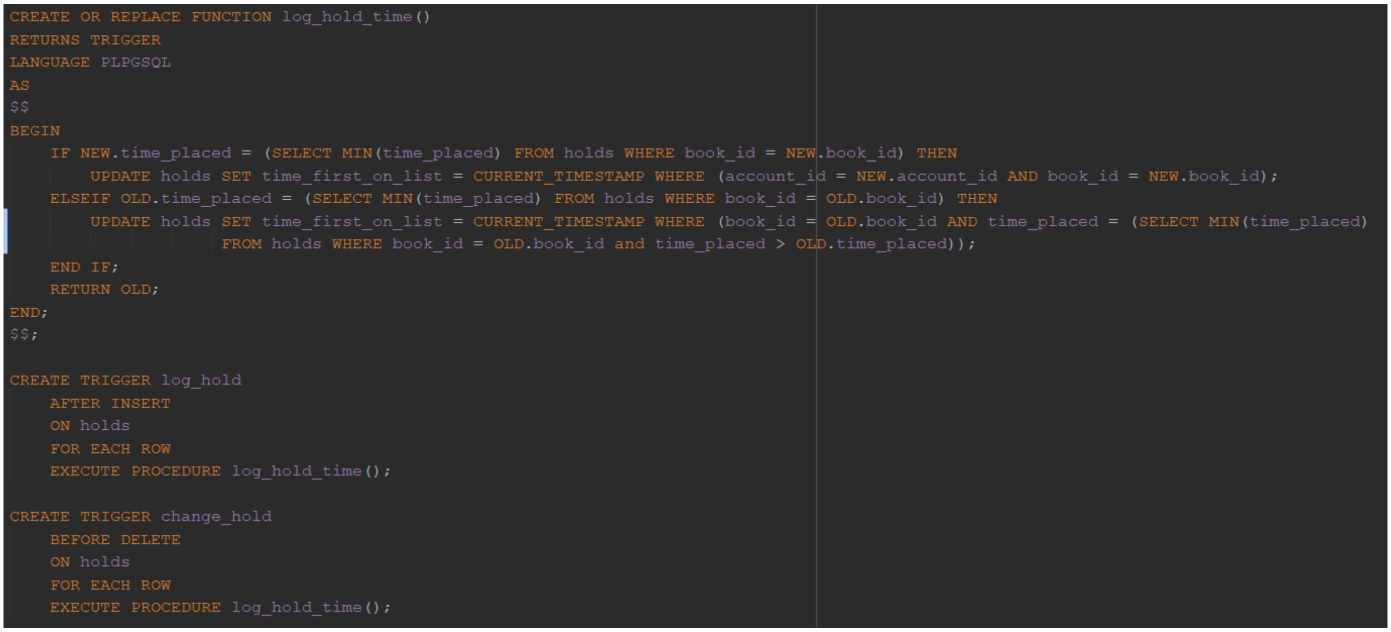
0 of 2 copies available

Checkout



7 of 7 copies available

- When the user reaches first on the list, a timestamp is stored in the database. 10 days later the hold record is deleted (they need to check out the book within 10 days of it becoming available or the next person becomes first on the list)

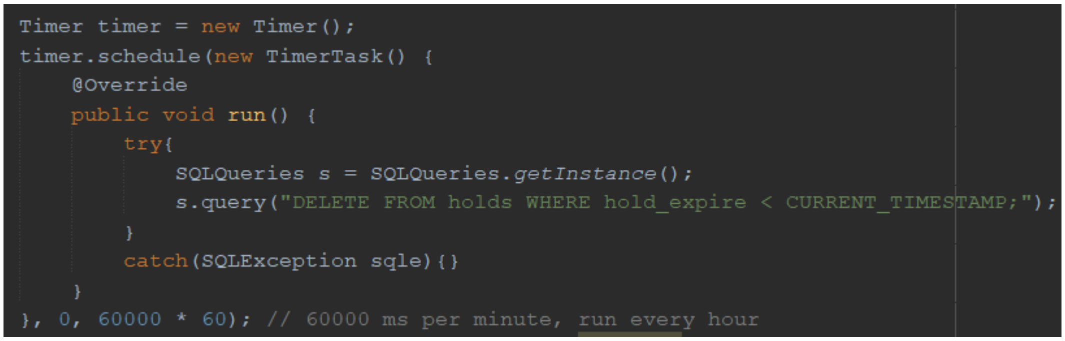


```
CREATE OR REPLACE FUNCTION log_hold_time()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS
$$
BEGIN
    IF NEW.time_placed = (SELECT MIN(time_placed) FROM holds WHERE book_id = NEW.book_id) THEN
        UPDATE holds SET time_first_on_list = CURRENT_TIMESTAMP WHERE (account_id = NEW.account_id AND book_id = NEW.book_id);
    ELSEIF OLD.time_placed = (SELECT MIN(time_placed) FROM holds WHERE book_id = OLD.book_id) THEN
        UPDATE holds SET time_first_on_list = CURRENT_TIMESTAMP WHERE (book_id = OLD.book_id AND time_placed = (SELECT MIN(time_placed)
                     FROM holds WHERE book_id = OLD.book_id and time_placed > OLD.time_placed));
    END IF;
    RETURN OLD;
END;
$$;

CREATE TRIGGER log_hold
    AFTER INSERT
    ON holds
    FOR EACH ROW
    EXECUTE PROCEDURE log_hold_time();

CREATE TRIGGER change_hold
    BEFORE DELETE
    ON holds
    FOR EACH ROW
    EXECUTE PROCEDURE log_hold_time();
```

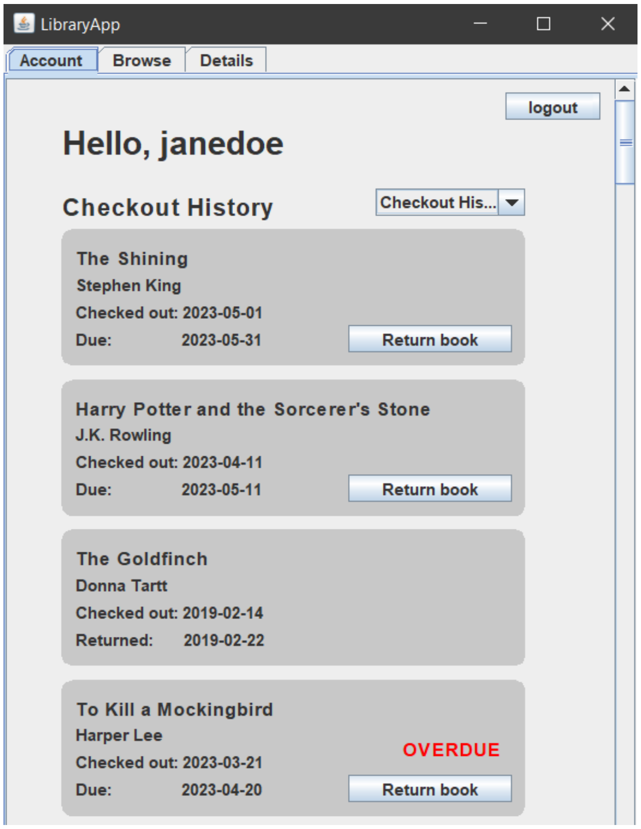
Trigger function storing timestamp for first on holds list



```
Timer timer = new Timer();
timer.schedule(new TimerTask() {
    @Override
    public void run() {
        try{
            SQLQueries s = SQLQueries.getInstance();
            s.query("DELETE FROM holds WHERE hold_expire < CURRENT_TIMESTAMP;");
        }
        catch(SQLException sqle){}
    }
}, 0, 60000 * 60); // 60000 ms per minute, run every hour
```

Java code for ‘cleaning’ holds list

- When the user signs in to their account they will be able to see their holds and checkout history (past checkouts, current checkouts, and overdue books), they will be able to cancel holds, and they will be able to return books.
  - The ability to return books with the click of a button would never actually exist in the real world but it is how my app is modeling a library having books returned.



Hello, janedoe

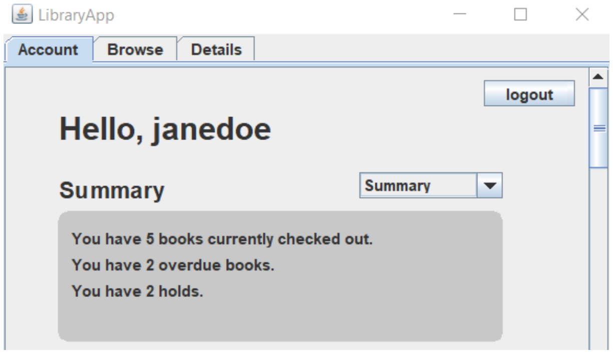
Checkout History

The Shining
Stephen King
Checked out: 2023-05-01
Due: 2023-05-31

Harry Potter and the Sorcerer's Stone
J.K. Rowling
Checked out: 2023-04-11
Due: 2023-05-11

The Goldfinch
Donna Tartt
Checked out: 2019-02-14
Returned: 2019-02-22

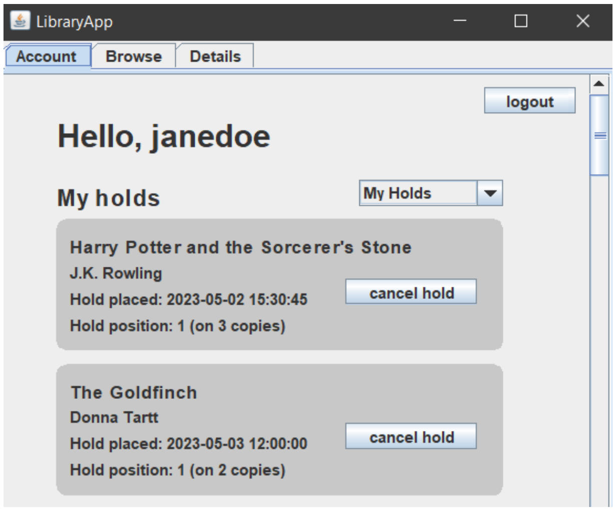
To Kill a Mockingbird
Harper Lee
OVERDUE
Checked out: 2023-03-21
Due: 2023-04-20



Hello, janedoe

Summary

You have 5 books currently checked out.
You have 2 overdue books.
You have 2 holds.



Hello, janedoe

My holds

Harry Potter and the Sorcerer's Stone
J.K. Rowling
Hold placed: 2023-05-02 15:30:45
Hold position: 1 (on 3 copies)

The Goldfinch
Donna Tartt
Hold placed: 2023-05-03 12:00:00
Hold position: 1 (on 2 copies)

- If, for example, there are two users with a hold on a book and two books become available, either user can check out the book. However if there is only one copy available and two holds, only the first user on the holds list can check out the book.

Class Design

I have ten java files:

- AccountPanel.java
  - Extends JScrollPane, providing information about library user's holds & checked out books
- AccountPanelItem.java
  - Extends JPanel, formats information about holds or checked out books on a grey rectangle background, for use in AccountPanel scroll pane
- BookDetails.java
  - Extends JPanel, formats information about book very similarly to bookListItem (but includes summary and checkout/hold button)

- BookListItem.java
    - Extends JPanel, for display on SearchPanel. Takes parameter information about book (info comes from ResultSet acquired in SearchPanel)
- CheckoutHold.java
    - Has LoginTextField for user to enter credentials and then checkout book or place hold (if no copies available). Also lists how many copies are available.
- ImproperHoldPositionException.java
    - Exception class, thrown when the user is not high enough on the holds list to check out the book
- LibraryApp.java
    - Main class. Sets up GUI instance for display, also contains TimerTask for deleting expired holds
- LoginTextField.java
    - Extends JPasswordField, but with the greyed-out placeholder-looking text (like what you'd see in HTML input tags with the placeholder attribute)
- SQLQueries.java
    - Singleton class setting up connection to database and handling queries
    - Since much of the transfer of data is done through button presses and dropdown selections, I use basic Statements. However, for any data involving textboxes, PreparedStatement is used.
- SearchPanel.java
    - Extends JScrollPane, displays the library's books and provides basic search filtering capabilities (search by title, author last name, genre)

For more detailed look at source code, see github: https://www.github.com/audreytracy/library-app.git

SQL file used for setting up database: src/sql/test.sql