

Project 5-7 Design

Developer: Linnea Jones

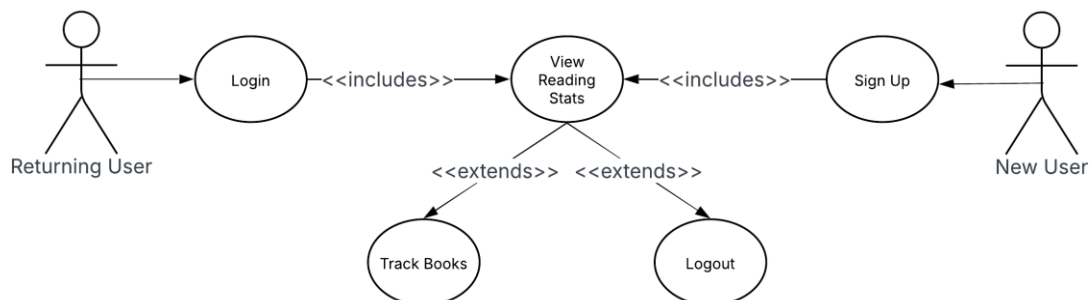
Project Summary: The project is called ReadStat, and it is an app similar to an established and popular app called StoryGraph. Like StoryGraph, ReadStat will allow users to track the books that they read and view charts about common characteristics of the books they read such as page number range, genre, and book format (ex: audio, print, digital). This will be done through an established database of books that the user can interact with but not edit or add to.

Project Requirements:

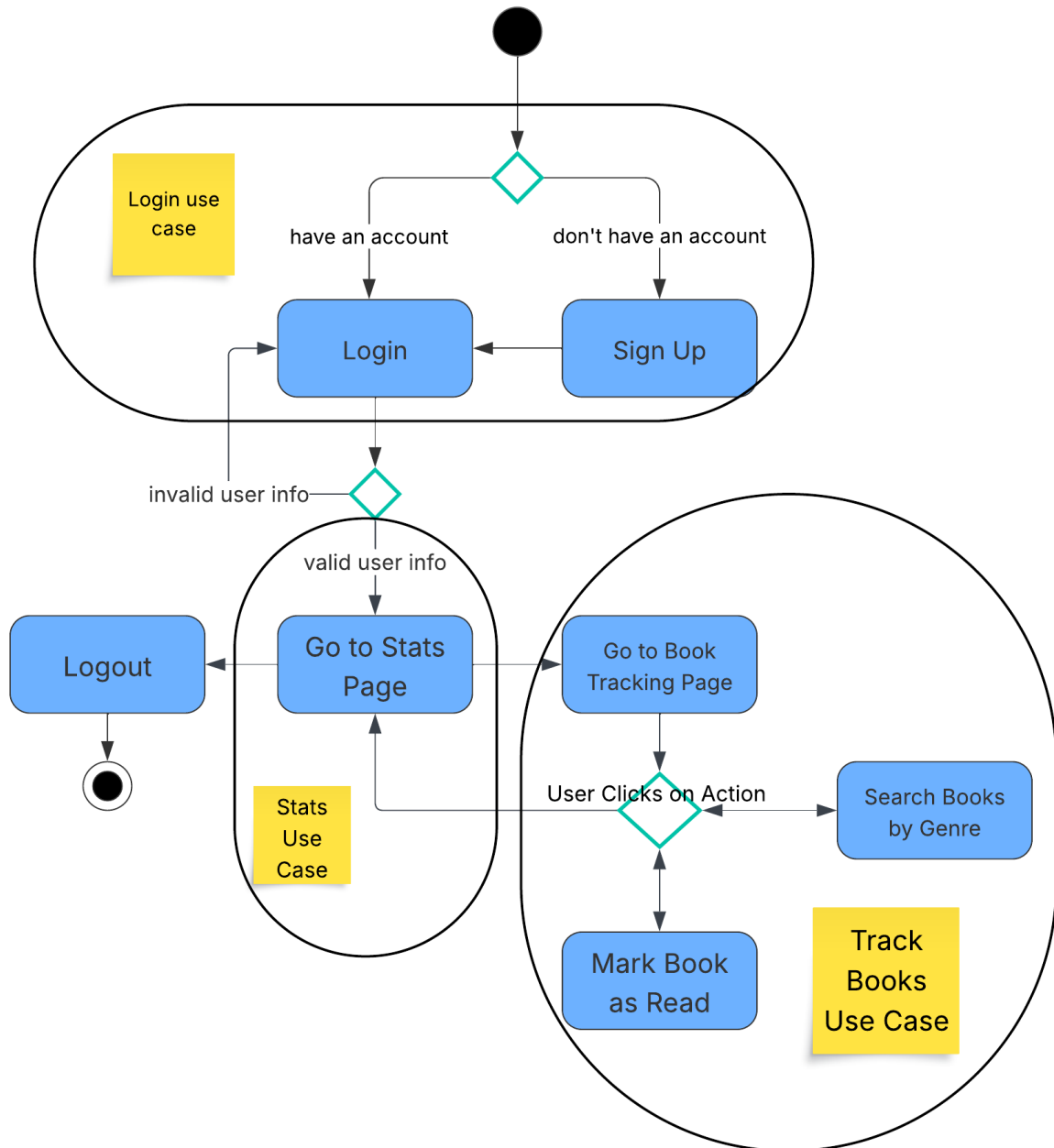
Main goals:

- Allow user(s) to mark all kinds of books as 'read.'
 - Maintain a database of books with different characteristics.
 - Allow user(s) to search books by genre.
 - Use SQL to store book data and associate them with the user.
- Allow user(s) to view charts displaying common statistics of the books they've read.
 - Use JFreeChart to create statistical charts about groups of books.
- Present an interactable interface to the user using Swing.
 - Use SQL to allow user(s) to create accounts or login and retain their data throughout sessions.
 - Have a login/signup page for the user.
 - Have two different pages for statistics and book tracking.

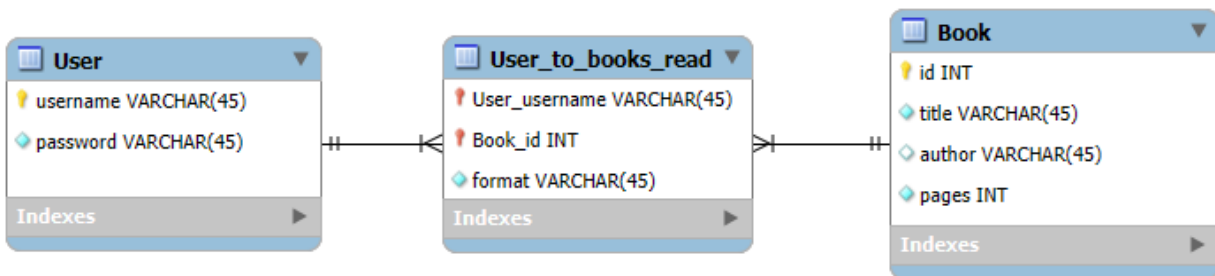
Users and Tasks:



Activity Diagram:



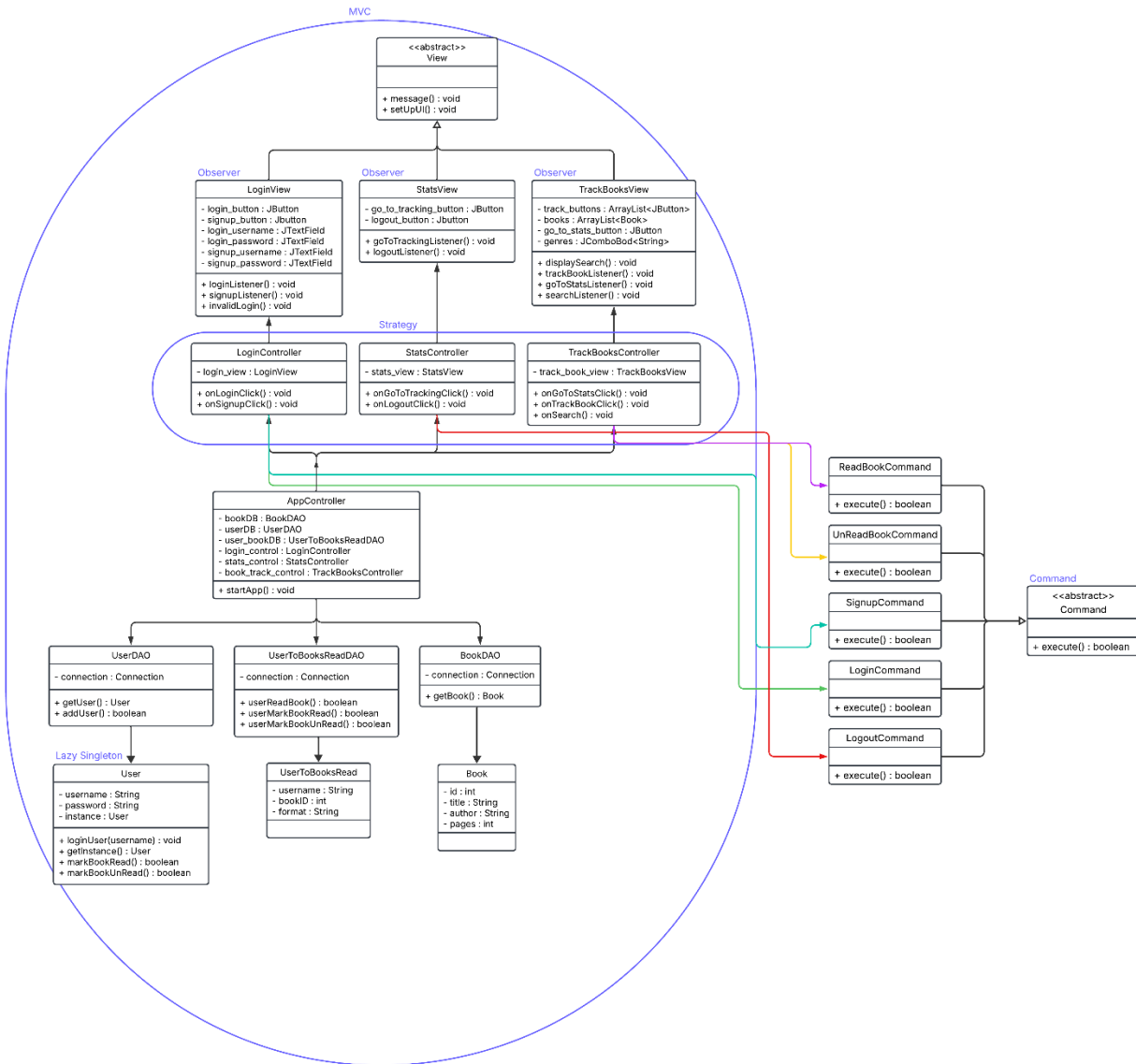
Data Storage: The project will be using MySQL for the database. The data will be stored through MySQL workbench on the local machine. This will be done using the MySQL Connector/J library through Java's JDBC (Java Database Connectivity) API. This will require a SQL connection class, and book and user query classes on top of their regular Java class implementations. The MySQL database structure is displayed in an entity relationship diagram below:



UI Mockups:



UML Class Diagram: The UML class diagram is shown below, with patterns (singleton, MVC, observer, strategy, command) labeled as used. The associations of command classes are color coded since there were so many overlapping association arrows.

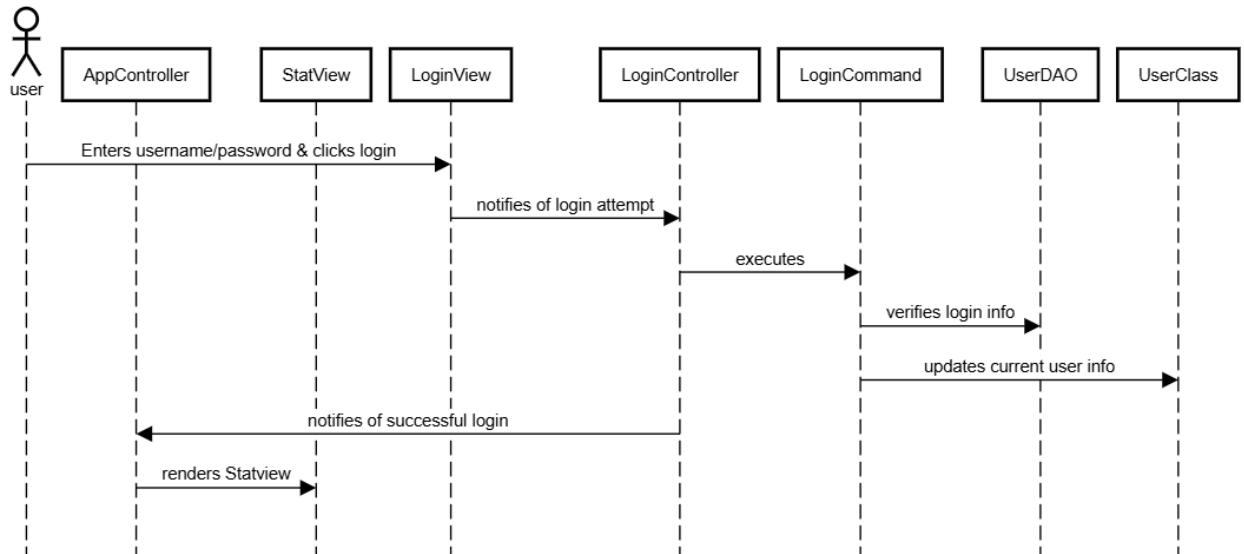


User Interactions:

- **Login:** The first thing a returning user can do on the app is login. They enter their login info and click login. Then, behind the scenes, the app sends that information to the controller who sends it to a command which attempts to log them in. If their information matches a record in the database, they will be logged in successfully

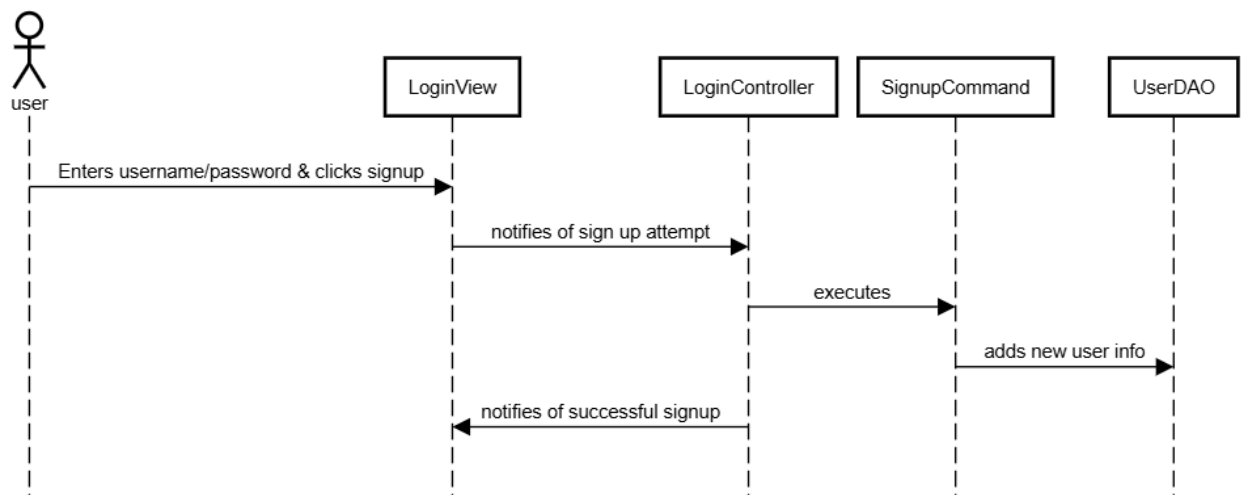
and redirected to the stats page.

Successful Login Sequence Diagram



- **Signup:** If a user is new and has yet to make an account, they will have to signup to interact with the app. As with login, they enter their information, then hit the signup button. From there, behind the scenes, their information is sent to the login controller, who sends it to a signup command which attempts to add their information to the database by making sure the username doesn't already exist. Once that is done successfully, the user is notified so that they know to attempt to login now.

Successful Sign Up Sequence Diagram



- **Track Book:** On the stats page, the user is presented with a list of books, each with a button labeled “add book.” If they click on this button, they are marking it as read. This means that, behind the scenes, the view is notifying the track book controller of this click, in addition to that book’s information. The controller then sends the information to the read book command, which communicates with the user to book DAO in order to add a relationship between the user and that book to the table.

Book Track Sequence Diagram

