Wilson Quilli

Professor Yang

CMPSC 132

February 24th, 2024

***Wilson’s Library System Report***

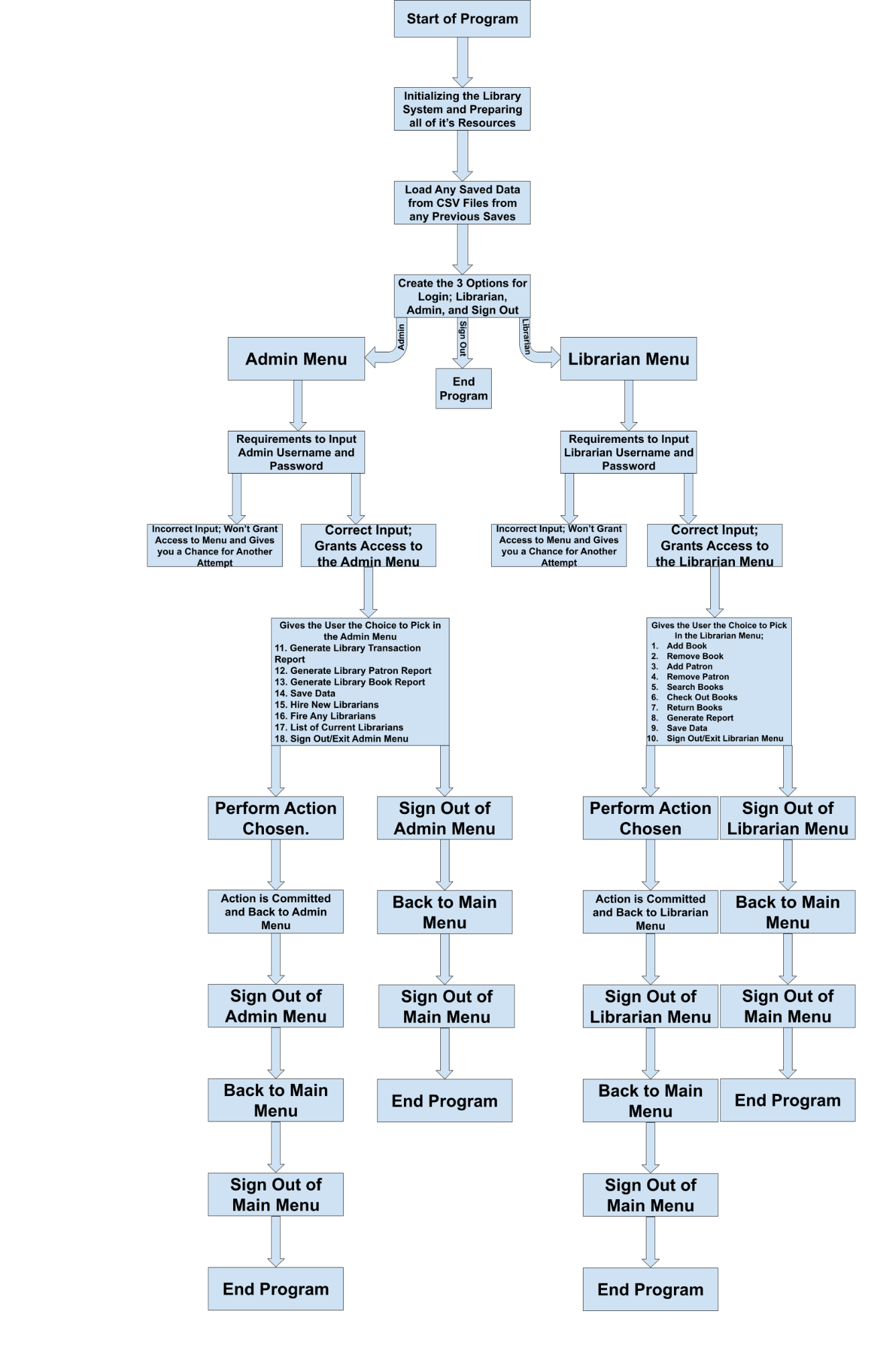
***Description of Project:***

My Library Management System is a program designed to help libraries manage every aspect of their business. The application has three menus: an admin menu meant for administrators, a librarian menu exclusively for librarians, and a main menu accessible by all staff members; both admins and librarians. There are different things to do from each of the three choices. In my system, the login credentials are User: librarian and Password: librarian456 for the librarian and User: admin and Password: admin123 for the admin. The user can also choose to exit the program and it will completely shut it down. For example, the main menu gives the user the option to select between librarian and admin. After that, you will need to authenticate yourself by using a password and username. Furthermore, librarians can add and remove books on the librarian menu. They can also search for any available books, check out and return books for patrons, generate book reports, and save any data. In addition, the admin menu enables administrators to see a list of the current librarians, fire any librarian, hire any new librarians, create usernames and passwords for new librarians to access the system upon hiring, generate reports on library transactions, and patrons, and books, and save your data. The system itself isn't too difficult to use either, since you simply need to enter the action number you want to access when selecting an option from a menu. For example, in the Admin Menu, to hire a new librarian, you simply need to enter the number 15. Overall, my library management system is a user-friendly, effective program that libraries can utilize to manage their businesses.

***Structure of the Code:***

I've created my library management system using a variety of classes. For example, I've made a book, patron, transaction, librarian, administrator, and library. Books' titles, authors, ISBNs, and quantities represent the attributes that make up the Book Class. A patron's name, ID, and contact details are among the attributes that make up the Patron Class. A fine for any past-due books, as well as the checkout and return dates, are all included in the Transaction Class. A username and password are among the properties of the Librarian Class. Similar to the Librarian Class, the Admin Class has login-related attributes including a username and password. Books, transactions, patrons, methods for managing transactions, and user logins are all included in the Library Class. In addition, I also defined functions to generate and show the admin menu, librarian menu, main menu, and all of the actions contained inside those menus. I also used error handling, such as the try and except method, for any incorrect login information and to avoid an error in the whole program. Ultimately, I created my Library Management System Program using several different types of classes, all with their attributes, functions to display and generate the three menus in my system and the actions in them, and error handling such as try and except to avoid any errors in my Program when something is inputted wrong.

***Diagram/Flow Chart of the Code:***



***Comments on the Flow Chart/Diagram:***

In the Flow Chart, you first have access to the main menu upon program launch, following initialization and loading any saved data. Three options are available on the main menu: librarian login, admin login, and sign out, which simply exits the application. To access the admin menu after selecting admin, you must provide the admin username and password. To access the librarian menu, select Librarian. You will then need to enter the librarian username and password. Different actions are available on both of these menus exclusively for the administrator or the librarian. You can commit an action or sign out when you first access these options. You can commit an action or sign out when you first access these options. Once an action is committed, it will be completed and you will return to the menu of the option you selected. You can end the program by signing out completely from the main menu, which will end the program.

***Instructions on How to Use:***

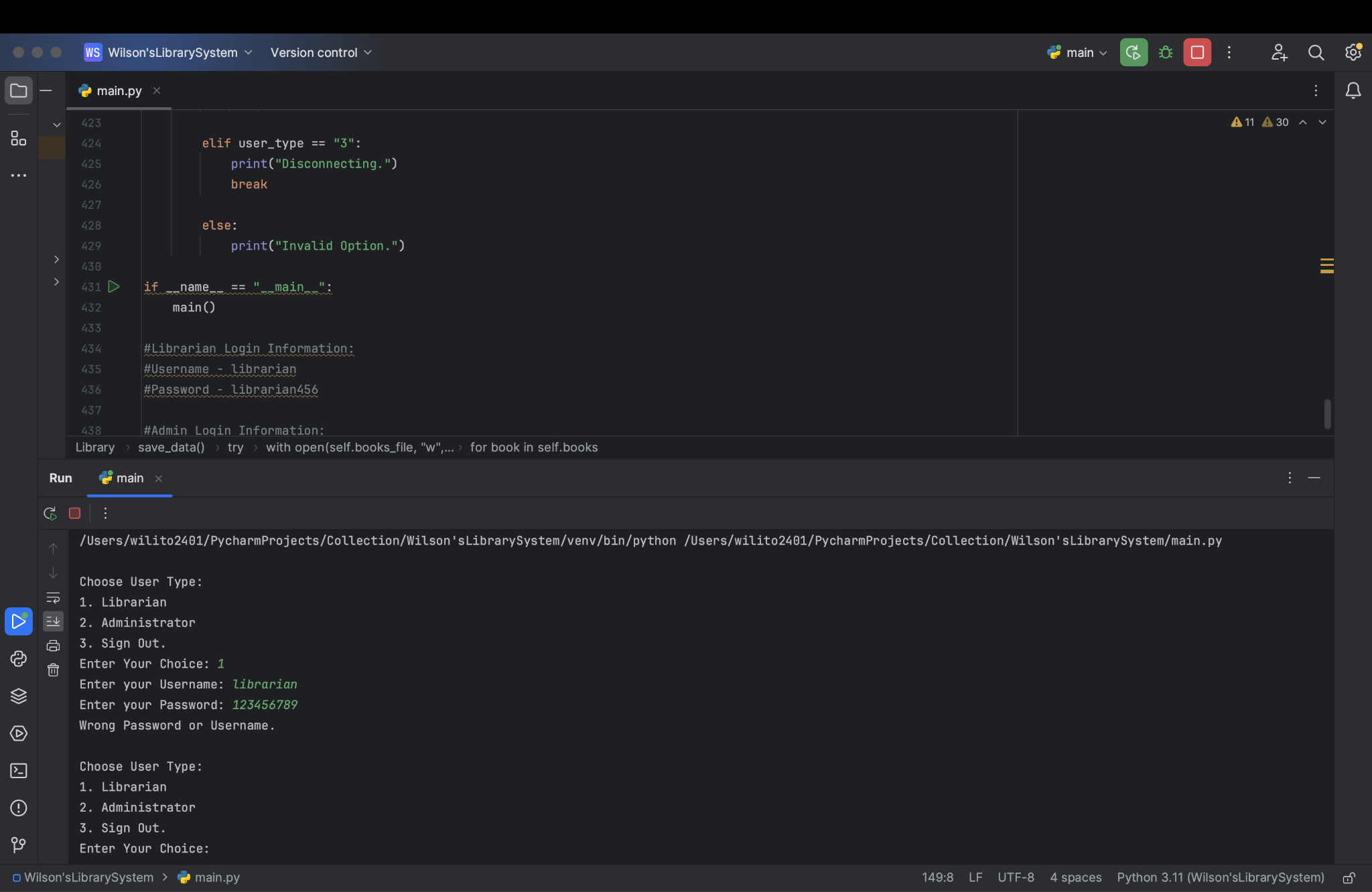
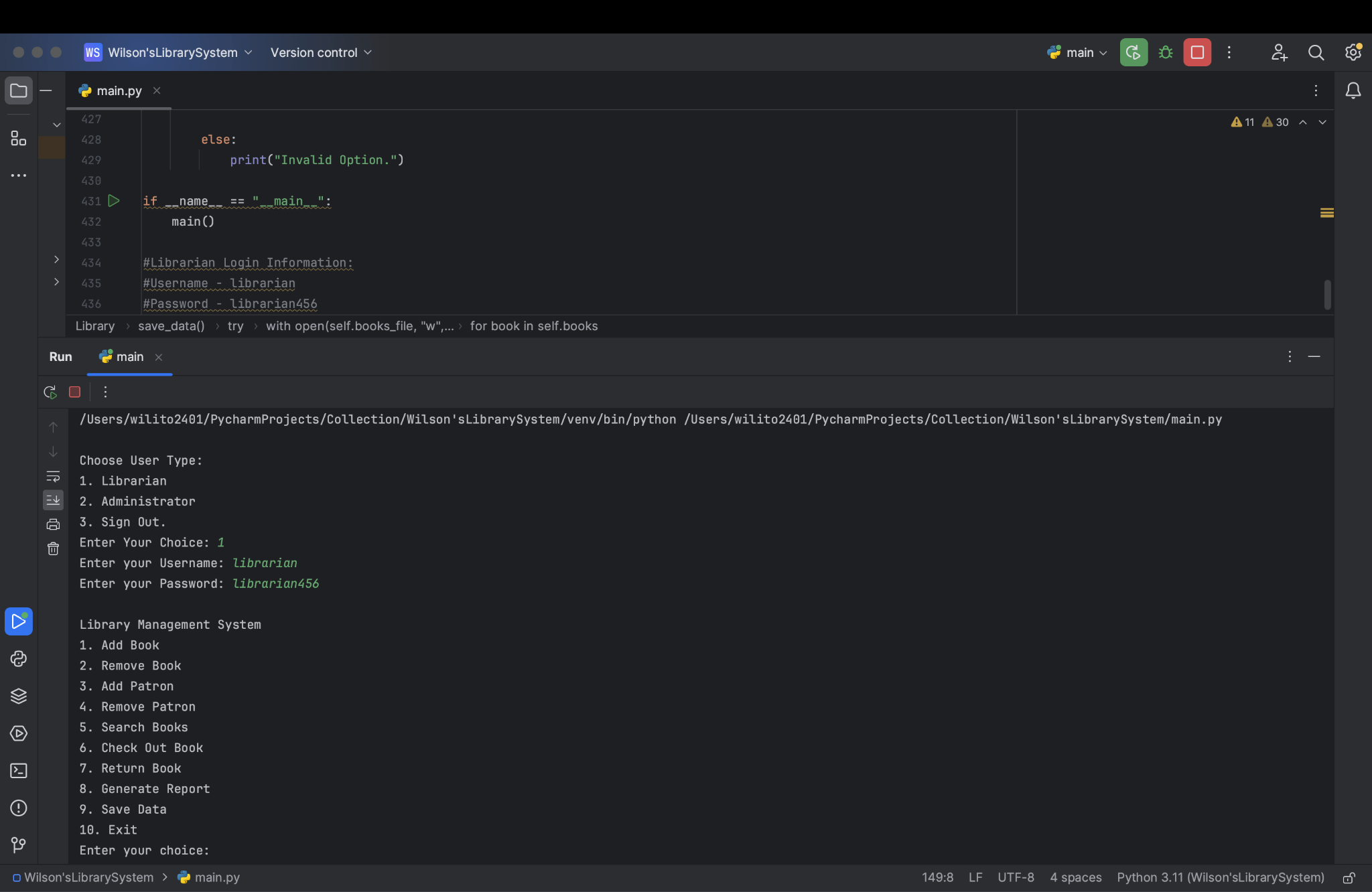
1. Run the Code
2. Choose Between Librarian, Admin, or Sign Out using the numbers 1-3
3. Enter the Information (username and password) to access the menu
4. Follow and Input the Action you wish to make, in the menu, by simply inputting the number shown on the menu of the action you want to commit such as generating reports
5. (If you want to, save any data inputted for future uses)
6. Sign Out of either menu when done, and you will be taken back to the main menu
7. Sign Out/Exit in the Main Menu to close/shut down the program

***Verification of the Code’s Sanity***

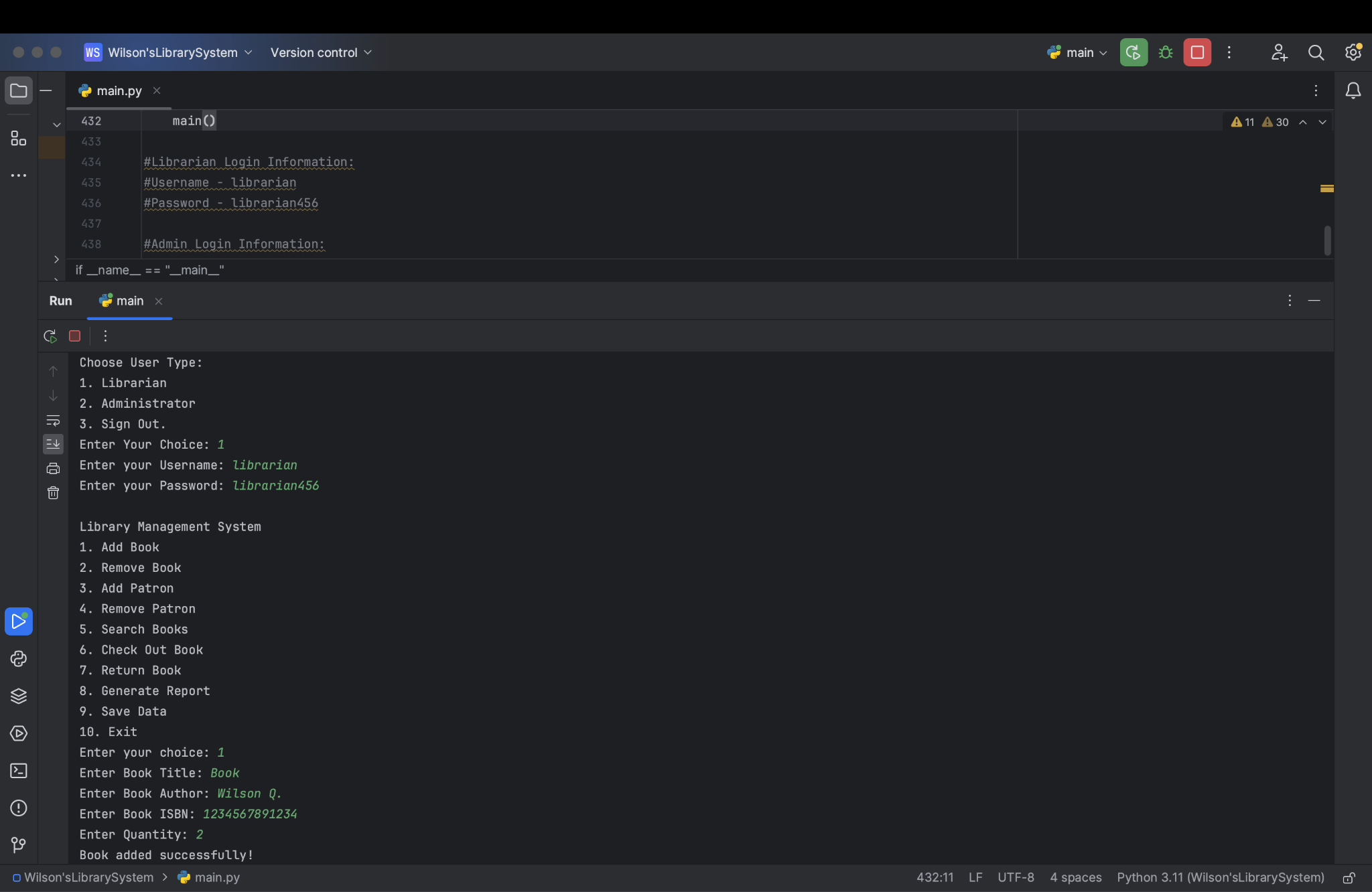
I have to examine every part of the code to make sure it functions properly and confirm its sanity. I'm able to verify that my Library Management System functions effectively and "gets the job done" after reviewing and testing the code. All of the classes, methods, and functions that I have utilized correctly throughout the code do what they're supposed to in the library system. To prevent mistakes in the program when anything has been entered wrong, I implemented error handling functions like the Try and Except Function. I used CSV files correctly on my system to store any data that someone wanted to save. I set in place a functional authentication system that allows administrators and librarians to access their menus by entering their passwords and usernames. In the end, my code seems to function properly after being tested to ensure it is functional.

***Screenshots and Scenarios of the Program in Action:***

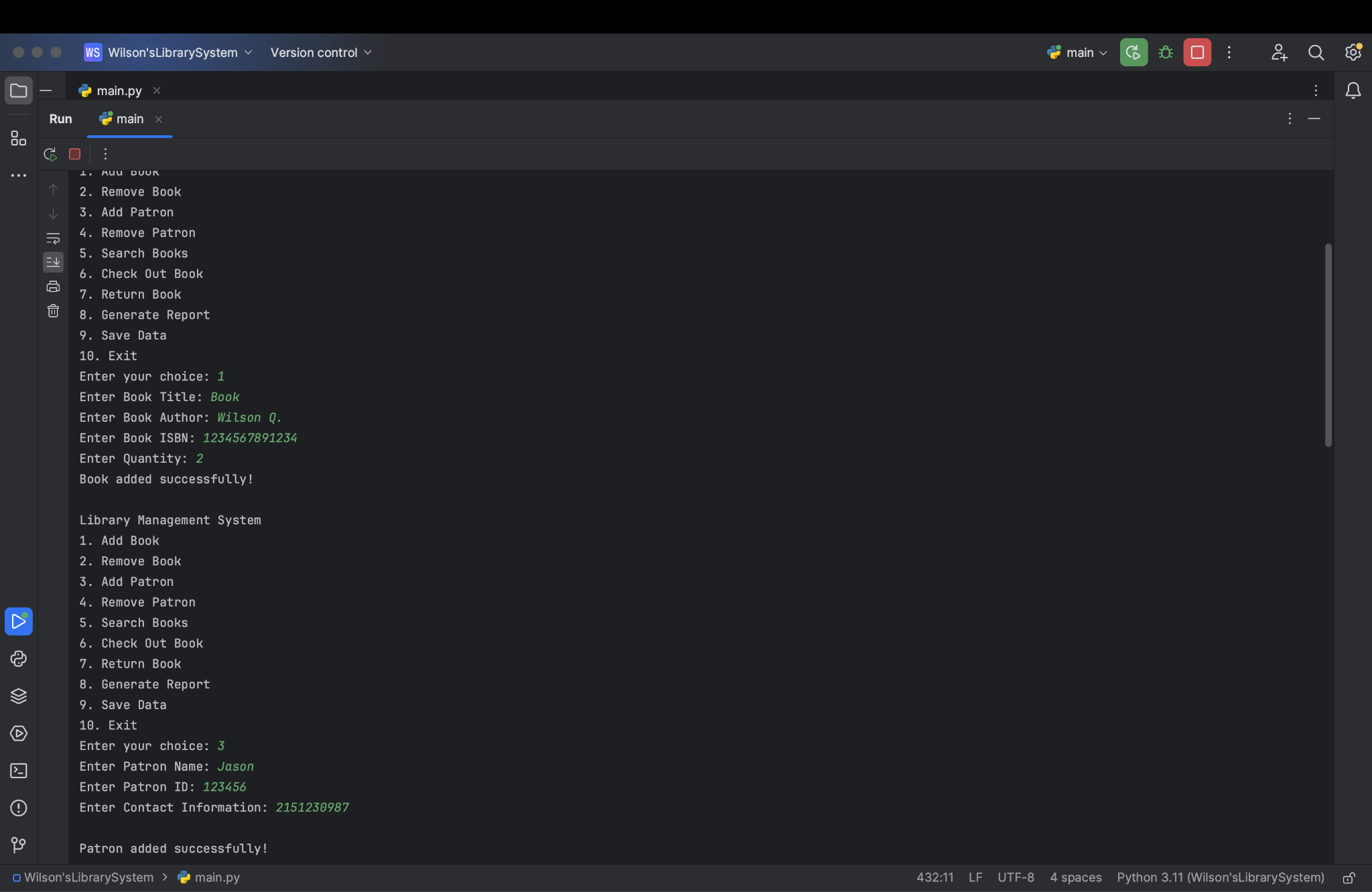
***Scenario #1 - Inputting Correct and Incorrect Login Information: 1st Picture is Correct Login and 2nd Picture is Incorrect Login***

******

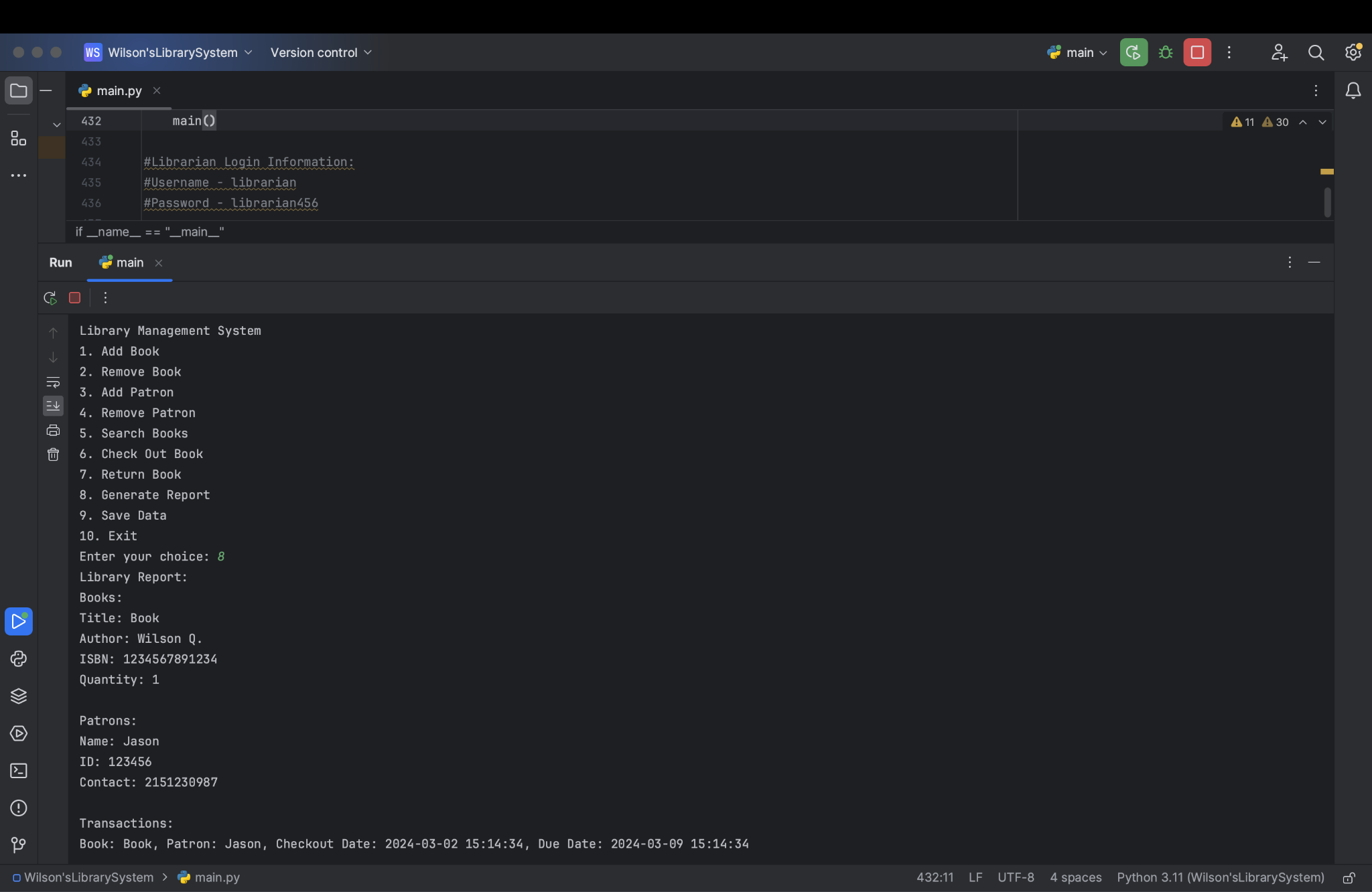
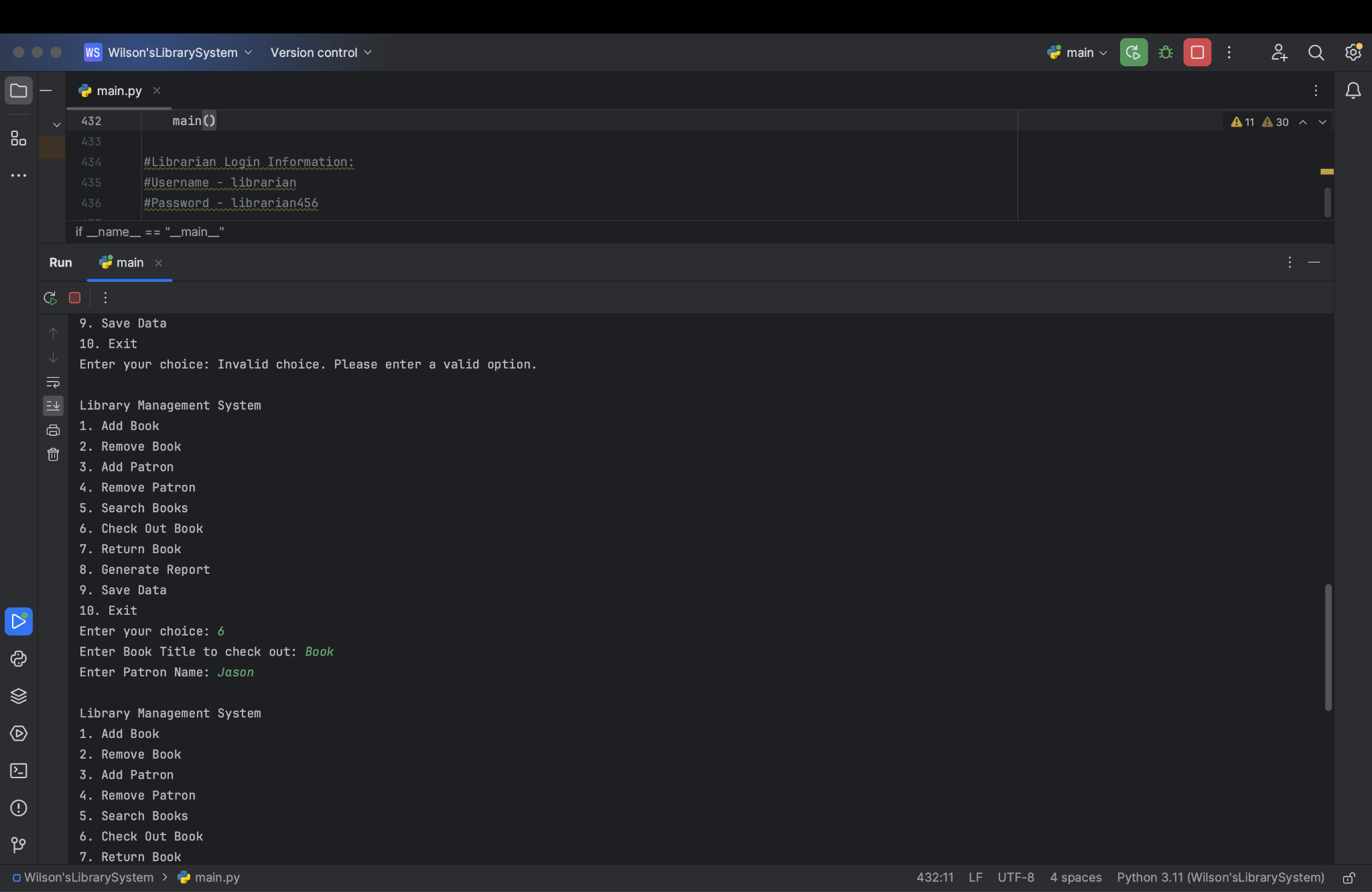
***Scenario #2 - Adding a Book:***

******

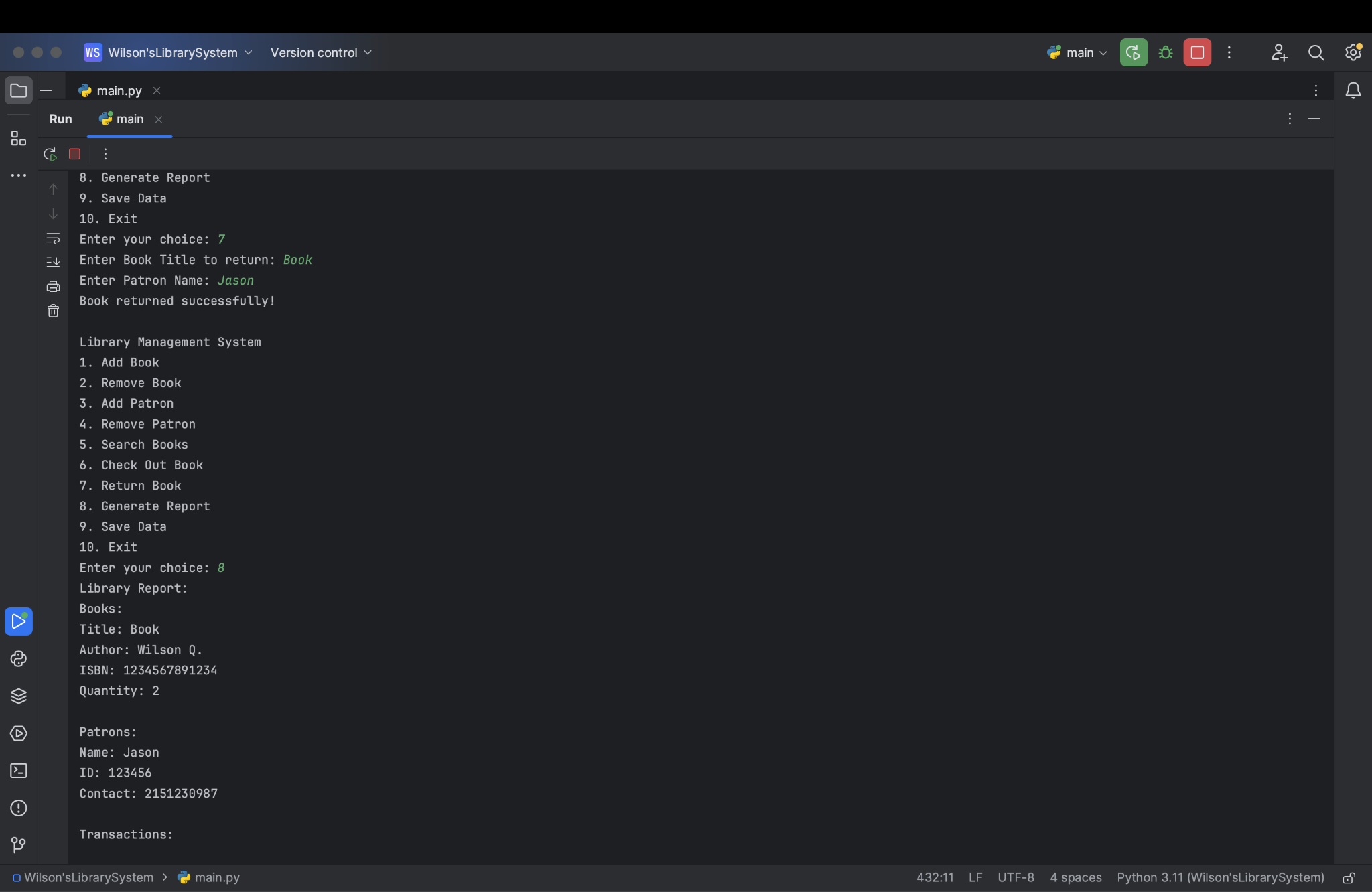
***Scenario #3 - Adding a Patron:***

******

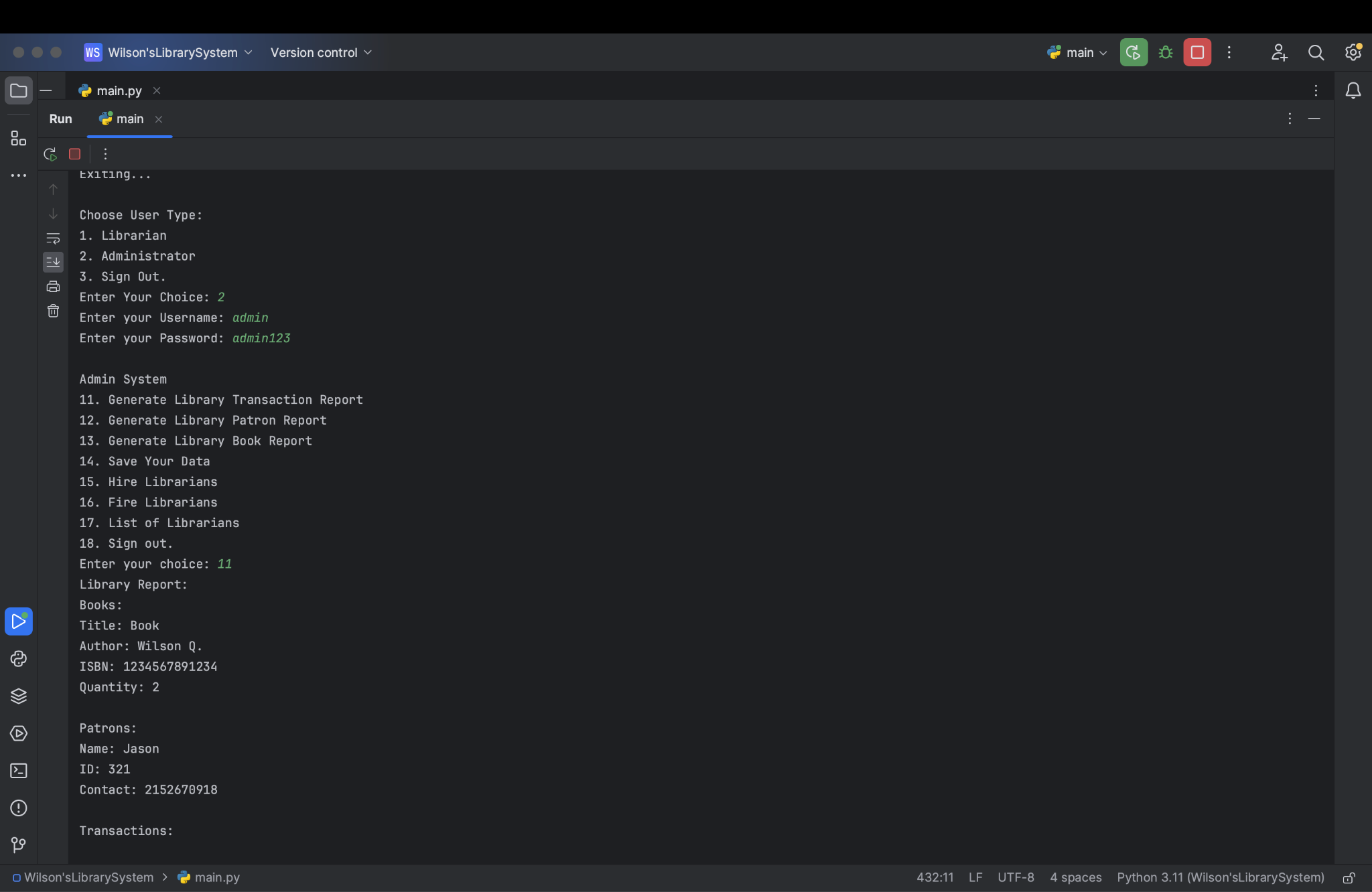
***Scenario #4 - Checking out a Book:***

******

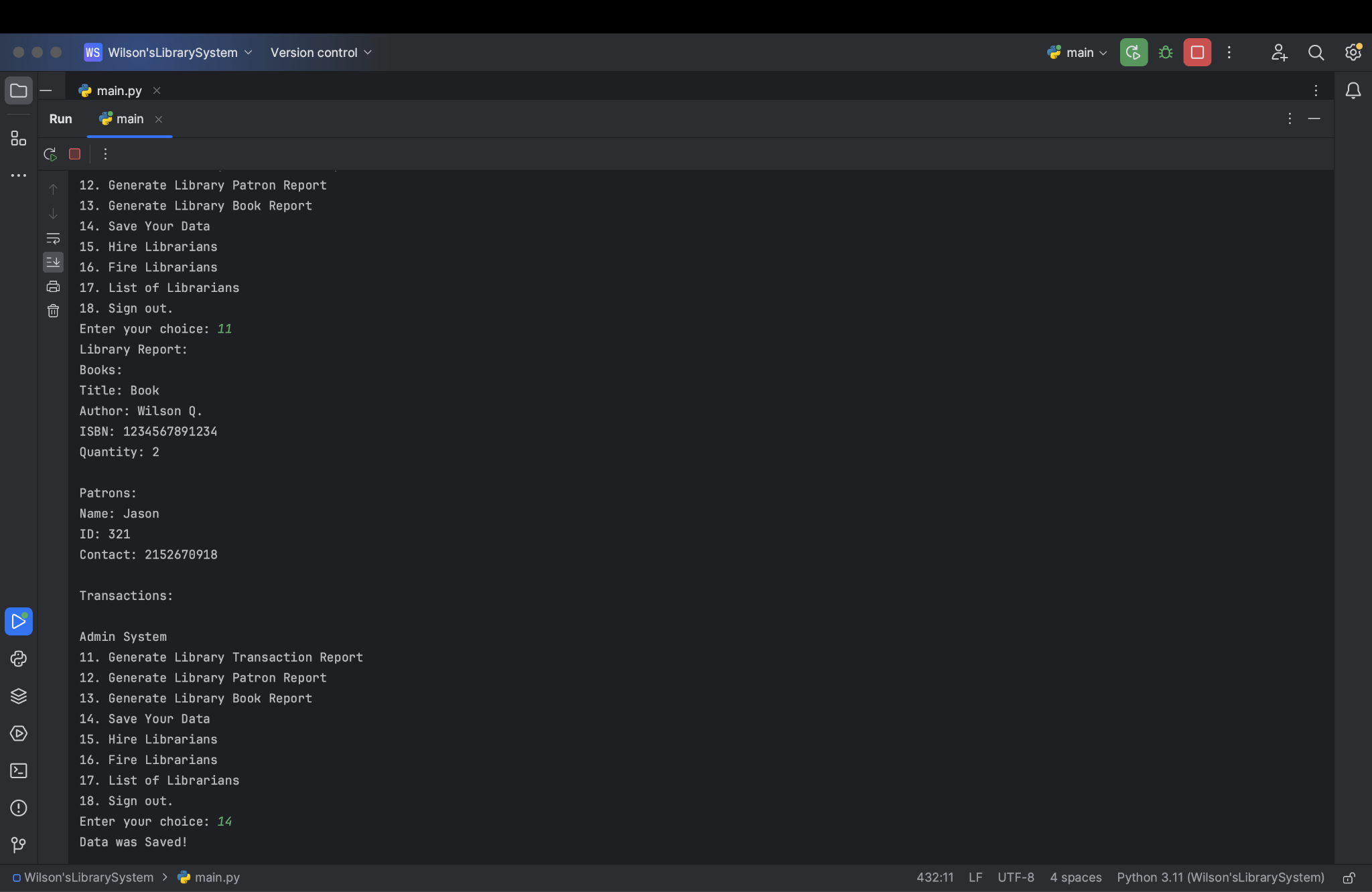
***Scenario #5 - Returning a Book:***

******

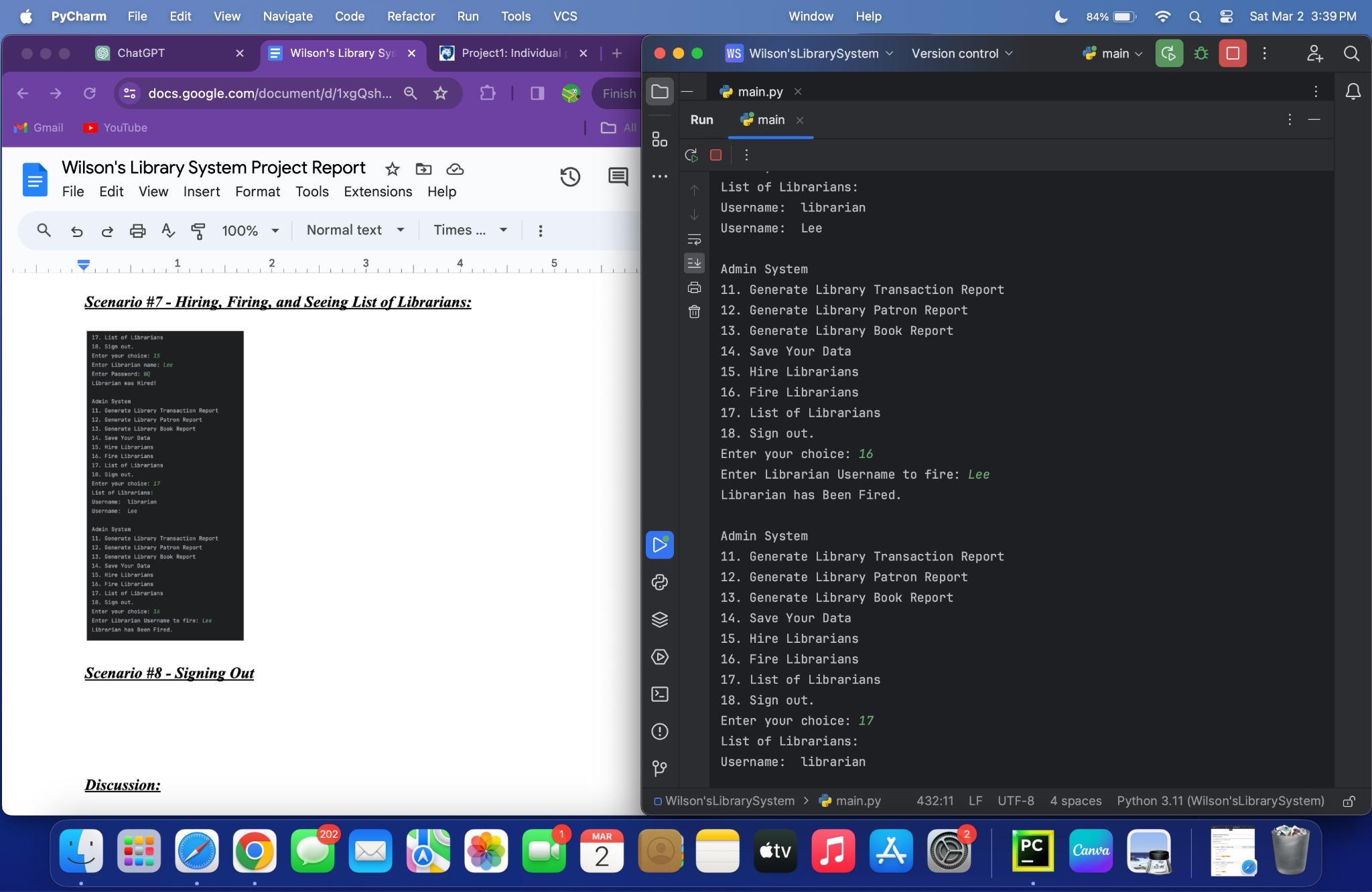
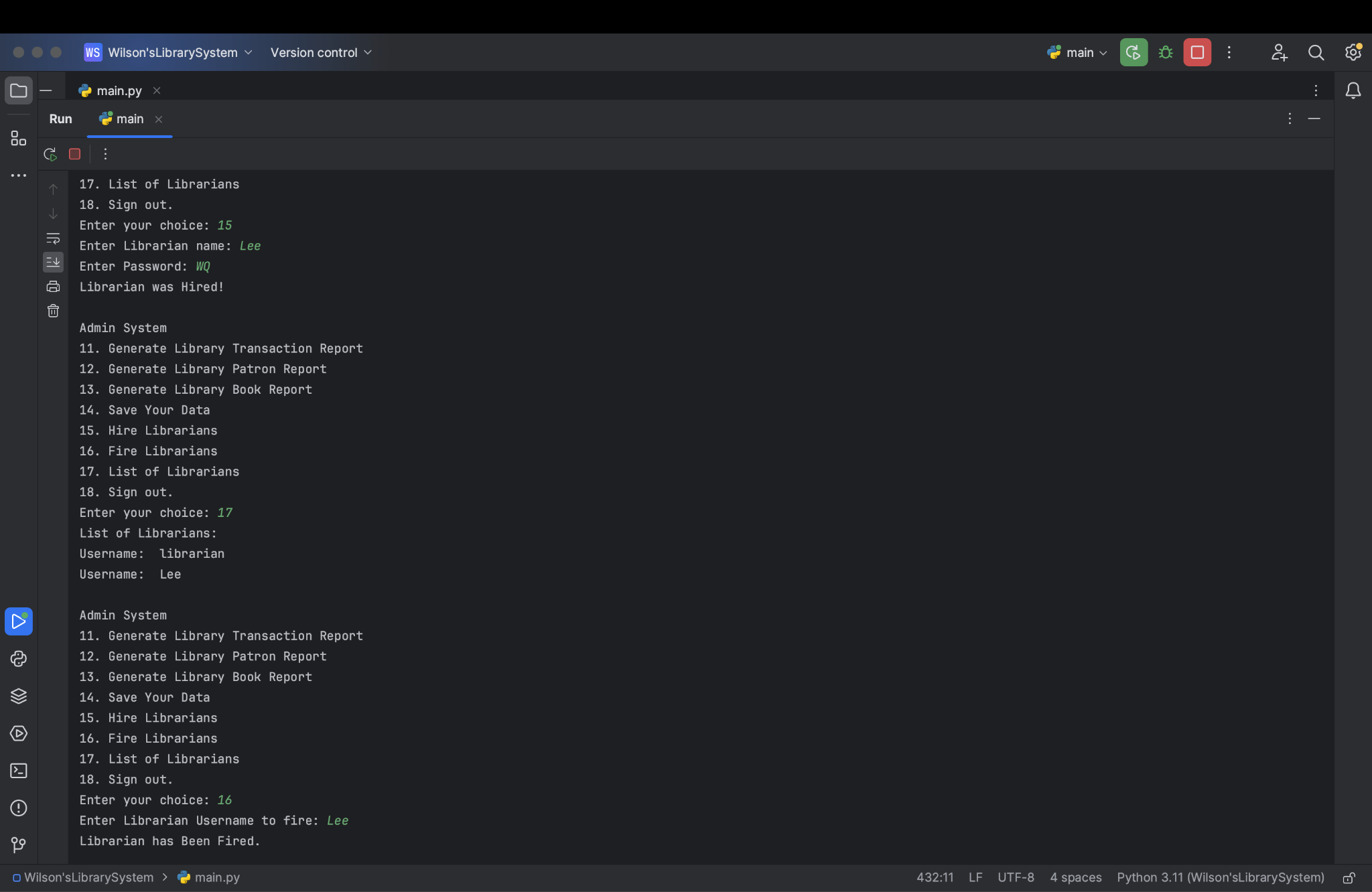
***Scenario #5 - Generating Report; Done in Admin Menu:***

******

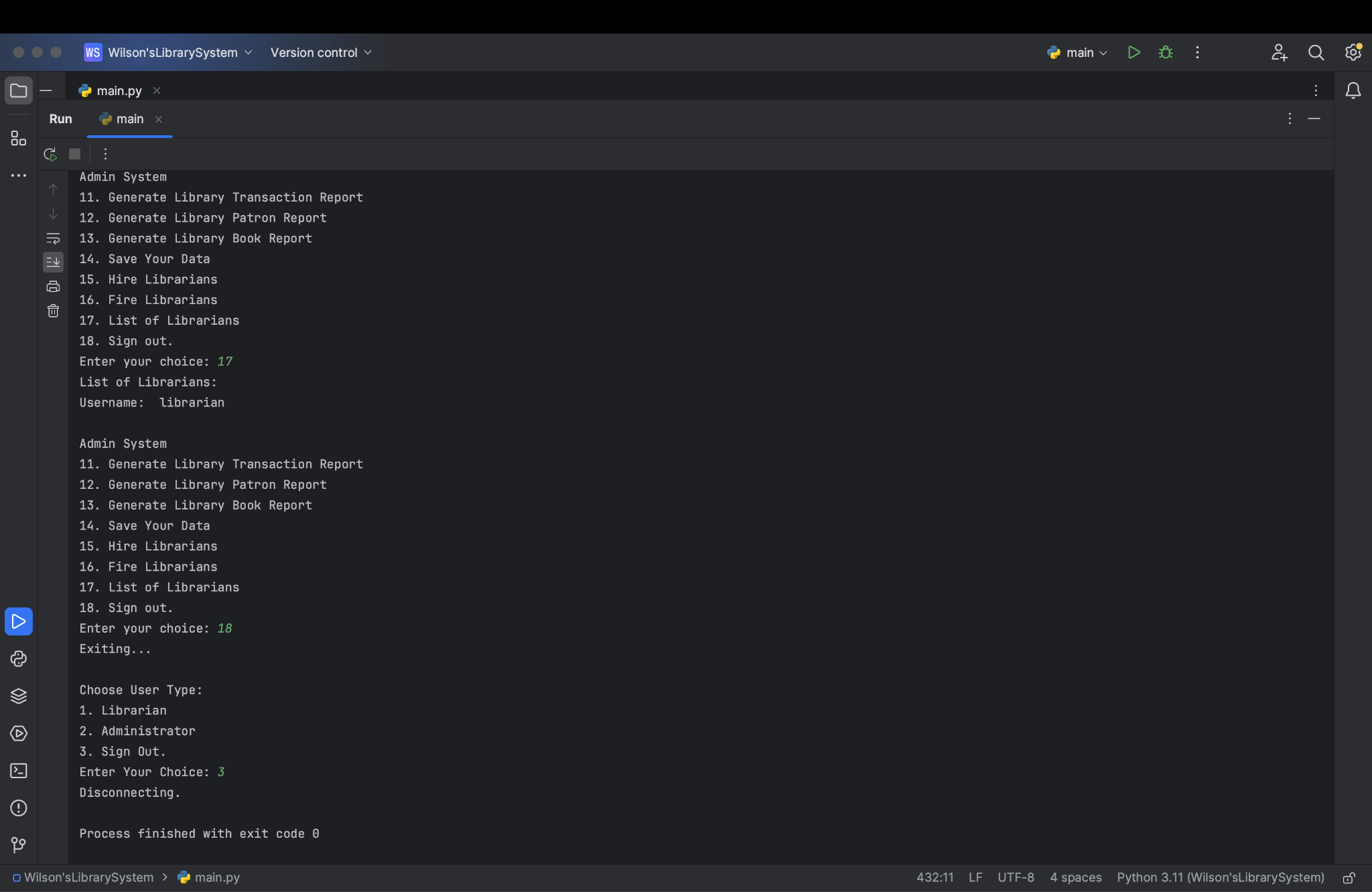
***Scenario #6 - Saving Data:***

******

***Scenario #7 - Hiring, Firing, and Seeing List of Librarians:***

******

***Scenario #8 - Signing Out***

******

***Discussion:***

In my project, I have learned a lot of different things. Such as specific methods on how to do different things on code. I learned more about encapsulation, classes, forms of display, error handling, attributes, and more about CSV. The code of the Library Management System demonstrates a well-organized design, with classes being essential for library operations. The library management system includes many different features from controlling books, users, and transactions to user verification. However, if the input is implemented wrongly, you still have a chance to try again, due to my error handling. In the end, the library management system is very easy to use and can be efficient for libraries.

***Challenges:***

While coding this Library Project, I faced many challenges and obstacles along the way. I have some experience with several aspects of the project, including making CSV files and classes that are utilized throughout. In my CMPSC 131 class, I learned about CSV files. In my CMPSC 132 class, I'm currently learning about classes and all the attributes and methods that go along with them. The hiring and firing of librarians through the admin menu was the most difficult challenge I experienced during the project. Every time I tested the code, I encountered a variety of errors, but after correctly defining the librarian and hiring and firing, it now functions as I wanted. The due date and trying to print it down when someone checks out a book was another difficulty for me. I never learned how to display the due date and its syntax correctly, so I was unfamiliar with it and how to do it. Eventually, after some research on Google and YouTube, I figured out how to utilize "%Y-%m-%d %H:%M:%S." The letters must appear in the exact order for the display to function correctly: Y stands for Year, M for Month, D for Day, H for Hour, M for Minute, and S for Seconds. A week after the book is checked out, this will print the due date along with the year, month, day, etc. I also faced some CSV difficulties, but they weren't too hard for me because I was already familiar with them; I only needed a refresher. In the end, I had several difficulties with creating certain functions, dates and times, CSV files, and some syntax mistakes.

***Areas for Improvement:***

After finishing and revising my project, I realized there were certain areas where I could do better in the future to make things easier and less difficult. I can get better at managing errors and all the try and except functions and methods that come with them. Despite having very little experience with error handling from my CMPSC 131 class, I was able to complete it here with the help of my past error-handling code, and some advice from YouTube, and Google. The login process, which requires a password and username for authentication, is another area that needs work. It was the librarian login that gave me some difficulties there, not the admin login. Even though I entered the password and username correctly, I kept receiving "wrong username or password." I also kept seeing a variety of warnings, including "a function wasn't defined" and "a tuple could not be called." I had to create a completely new function for logged\_in\_librarian in order to make admin and librarian logins work. Although not as much because of my prior experiences, I still had some difficulties with CSV files and the storage system, as I have discussed in previous sections of my project; so I could use more improvement there too. Furthermore, as previously indicated, I had trouble figuring out how to make the checked-out books' due date and display a week later. However, I was able to solve this problem by learning about the %Y-%m-%d syntax and importing datetime and timedelta. In conclusion, I can still work on my error handling, adding more syntax, the login authentication process, CSV files, and functions.

***Tests:***

These are some tests that I ran while trying to test out the code after it was done and some things I added.

* Test #1 - The roles were not implemented correctly and there was no difference between the librarian and admin permissions.
* Test #2 - Testing out the Librarian role and adding books. Result: Used Harry Potter details to add books and the test was successful.
* Test #3 - Testing out the Librarian Role and Removing the Harry Potter Book. Result: Successful test and Harry Potter was removed.
* Test #4 - I came to the realization that the librarians and admin had the same actions in their own menus, so I changed the admin’s actions to give them more importance.
* Test #5 - Testing Admin’s new Actions. Result: Error because I was still getting the same actions.
* Test #6 - I changed the actions for Librarian and Admin in their numbers to input a choice so for example, Add Book is #1 for the librarian and now I changed Library Transaction Report is #11 for the admin; avoiding the same function being #1. Result: Successful
* Test #7 - Adding Error handling Try and Except Function, to avoid any incorrect inputs. Result: Successful and Incorrect Inputs can be inputted
* Test #8 - Error - Due Date Error when Checking out a Book, Importing datetime and timedelta to see if it works, and new method I found after researching. Result: Successful
* Test #9 - Final Test to see if everything works. Result: Everything seems to work as intended!