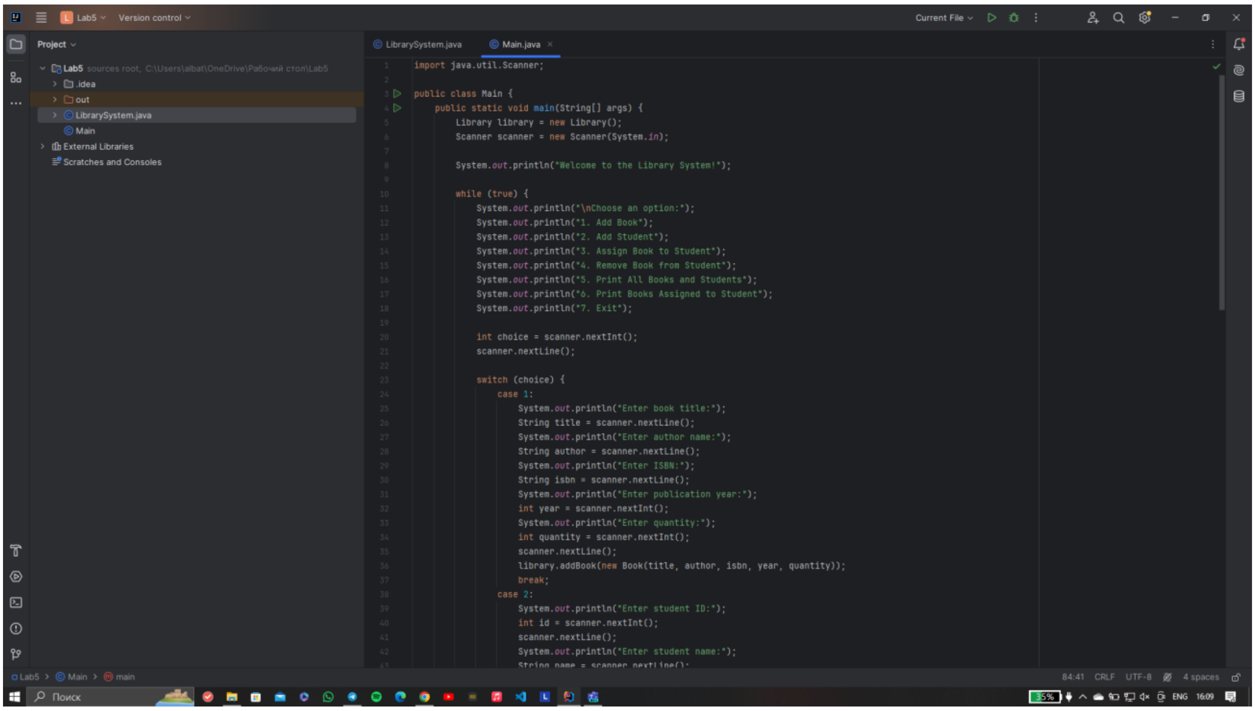
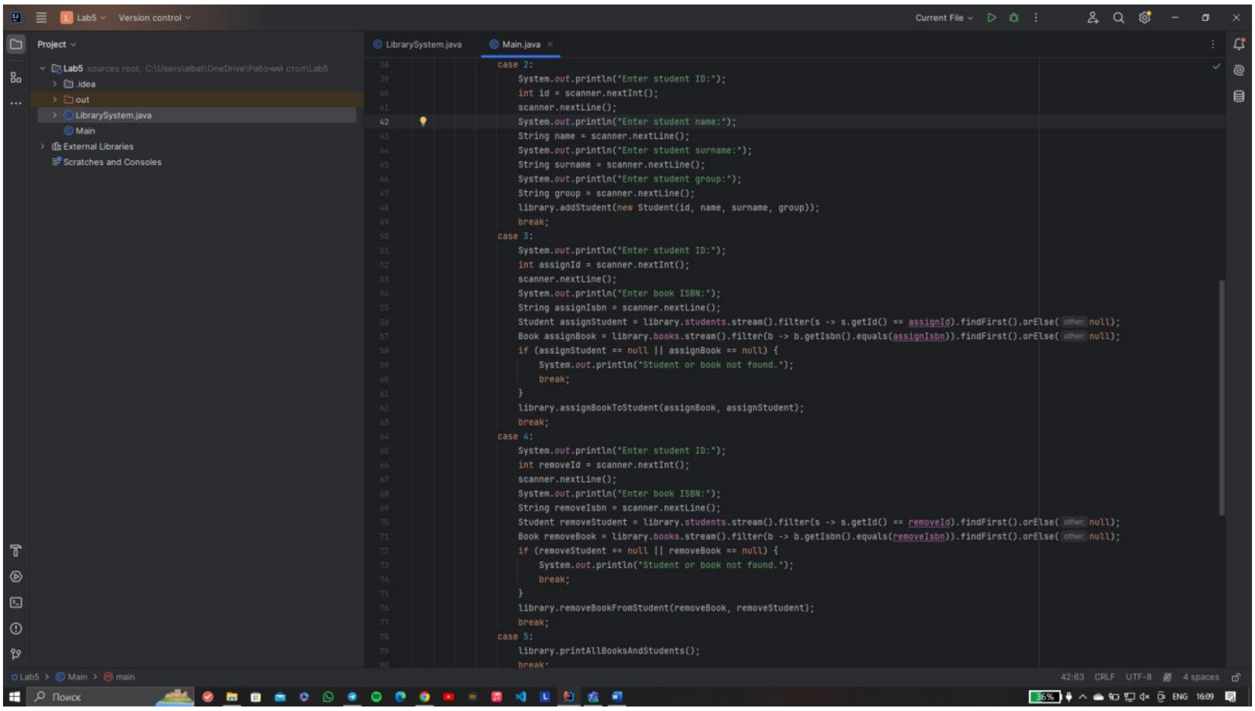
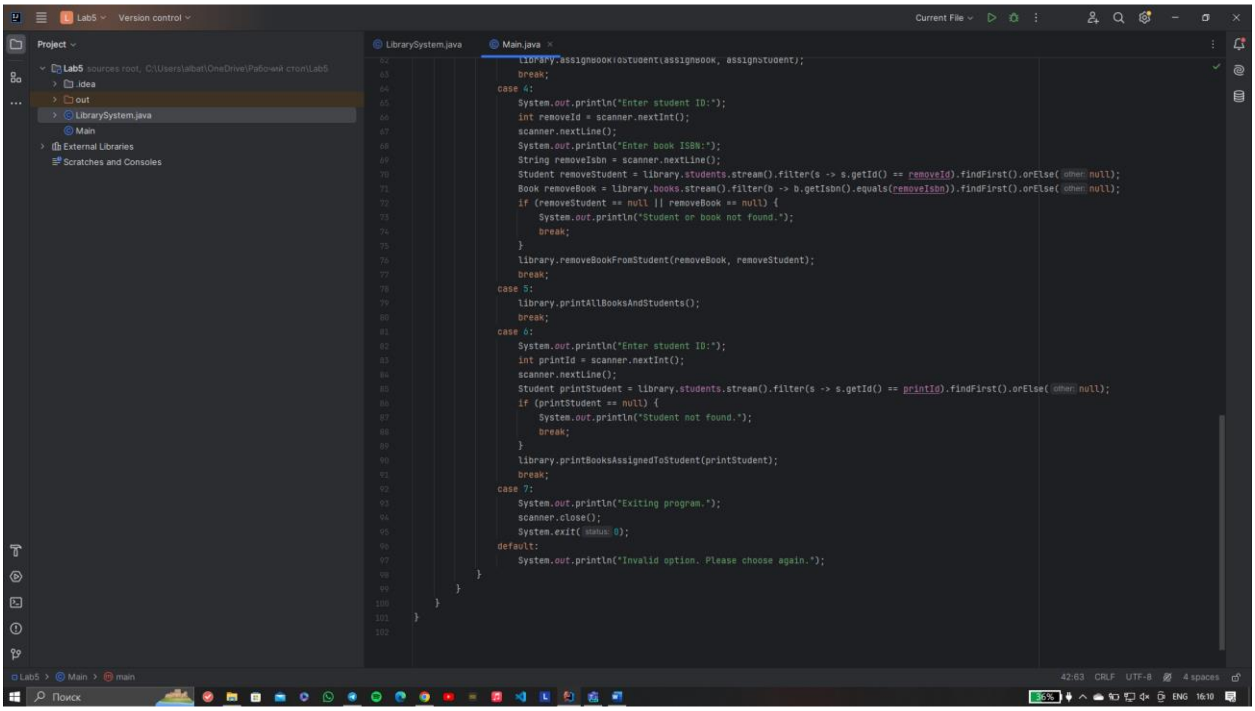


```java
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Library library = new Library();
        Scanner scanner = new Scanner(System.in);

        System.out.println("Welcome to the Library System!");

        while (true) {
            System.out.println("\nChoose an option:");
            System.out.println("1. Add Book");
            System.out.println("2. Add Student");
            System.out.println("3. Assign Book to Student");
            System.out.println("4. Remove Book from Student");
            System.out.println("5. Print All Books and Students");
            System.out.println("6. Print Books Assigned to Student");
            System.out.println("7. Exit");

            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.println("Enter book title:");
                    String title = scanner.nextLine();
                    System.out.println("Enter author name:");
                    String author = scanner.nextLine();
                    System.out.println("Enter ISBN:");
                    String isbn = scanner.nextLine();
                    System.out.println("Enter publication year:");
                    int year = scanner.nextInt();
                    System.out.println("Enter quantity:");
                    int quantity = scanner.nextInt();
                    scanner.nextLine();
                    library.addBook(new Book(title, author, isbn, year, quantity));
                    break;
                case 2:
                    System.out.println("Enter student ID:");
                    int id = scanner.nextInt();
                    scanner.nextLine();
                    System.out.println("Enter student name:");
                    String name = scanner.nextLine();
```



```java
                case 2:
                    System.out.println("Enter student ID:");
                    int id = scanner.nextInt();
                    scanner.nextLine();
                    System.out.println("Enter student name:");
                    String name = scanner.nextLine();
                    System.out.println("Enter student surname:");
                    String surname = scanner.nextLine();
                    System.out.println("Enter student group:");
                    String group = scanner.nextLine();
                    library.addStudent(new Student(id, name, surname, group));
                    break;
                case 3:
                    System.out.println("Enter student ID:");
                    int assignId = scanner.nextInt();
                    scanner.nextLine();
                    System.out.println("Enter book ISBN:");
                    String assignIsbn = scanner.nextLine();
                    Student assignStudent = library.students.stream().filter(s -> s.getId() == assignId).findFirst().orElse( null);
                    Book assignBook = library.books.stream().filter(b -> b.getIsbn().equals(assignIsbn)).findFirst().orElse( null);
                    if (assignStudent == null || assignBook == null) {
                        System.out.println("Student or book not found.");
                        break;
                    }
                    library.assignBookToStudent(assignBook, assignStudent);
                    break;
                case 4:
                    System.out.println("Enter student ID:");
                    int removeId = scanner.nextInt();
                    scanner.nextLine();
                    System.out.println("Enter book ISBN:");
                    String removeIsbn = scanner.nextLine();
                    Student removeStudent = library.students.stream().filter(s -> s.getId() == removeId).findFirst().orElse( null);
                    Book removeBook = library.books.stream().filter(b -> b.getIsbn().equals(removeIsbn)).findFirst().orElse( null);
                    if (removeStudent == null || removeBook == null) {
                        System.out.println("Student or book not found.");
                        break;
                    }
                    library.removeBookFromStudent(removeBook, removeStudent);
                    break;
                case 5:
                    library.printAllBooksAndStudents();
                    break;
```



```
                    library.assignBookToStudent(assignBook, assignStudent);
                    break;
                case 4:
                    System.out.println("Enter student ID:");
                    int removeId = scanner.nextInt();
                    scanner.nextLine();
                    System.out.println("Enter book ISBN:");
                    String removeIsbn = scanner.nextLine();
                    Student removeStudent = library.students.stream().filter(s -> s.getId() == removeId).findFirst().orElse( other: null);
                    Book removeBook = library.books.stream().filter(b -> b.getIsbn().equals(removeIsbn)).findFirst().orElse( other: null);
                    if (removeStudent == null || removeBook == null) {
                        System.out.println("Student or book not found.");
                        break;
                    }
                    library.removeBookFromStudent(removeBook, removeStudent);
                    break;
                case 5:
                    library.printAllBooksAndStudents();
                    break;
                case 6:
                    System.out.println("Enter student ID:");
                    int printId = scanner.nextInt();
                    scanner.nextLine();
                    Student printStudent = library.students.stream().filter(s -> s.getId() == printId).findFirst().orElse( other: null);
                    if (printStudent == null) {
                        System.out.println("Student not found.");
                        break;
                    }
                    library.printBooksAssignedToStudent(printStudent);
                    break;
                case 7:
                    System.out.println("Exiting program.");
                    scanner.close();
                    System.exit( status: 0);
                default:
                    System.out.println("Invalid option. Please choose again.");
            }
        }
    }
}
```

1) The `Main` class in this code snippet is the entry point for the library management system. It uses a `Scanner` object to read user input from the console and interacts with the `Library` class to perform various operations.

1. Initialization:

- Creates an instance of the `Library` class to manage the library operations.

- Creates a `Scanner` object to read user input.

2. Menu Display and User Input:

- Displays a menu of options for the user to choose from.

- Reads the user's choice using `scanner.nextInt()` and consumes the newline character with `scanner.nextLine()`.

3. *Switch Statement:

- Handles different user choices by executing the corresponding case.

- For each case:

  - Prompts the user for necessary information (e.g., book title, author, ISBN, student ID, etc.).

  - Reads the input using `scanner.nextLine()` or `scanner.nextInt()` based on the data type.

  - Calls the corresponding method of the `Library` class to perform the operation (e.g., `addBook`, `addStudent` and so on)

4. Loop:

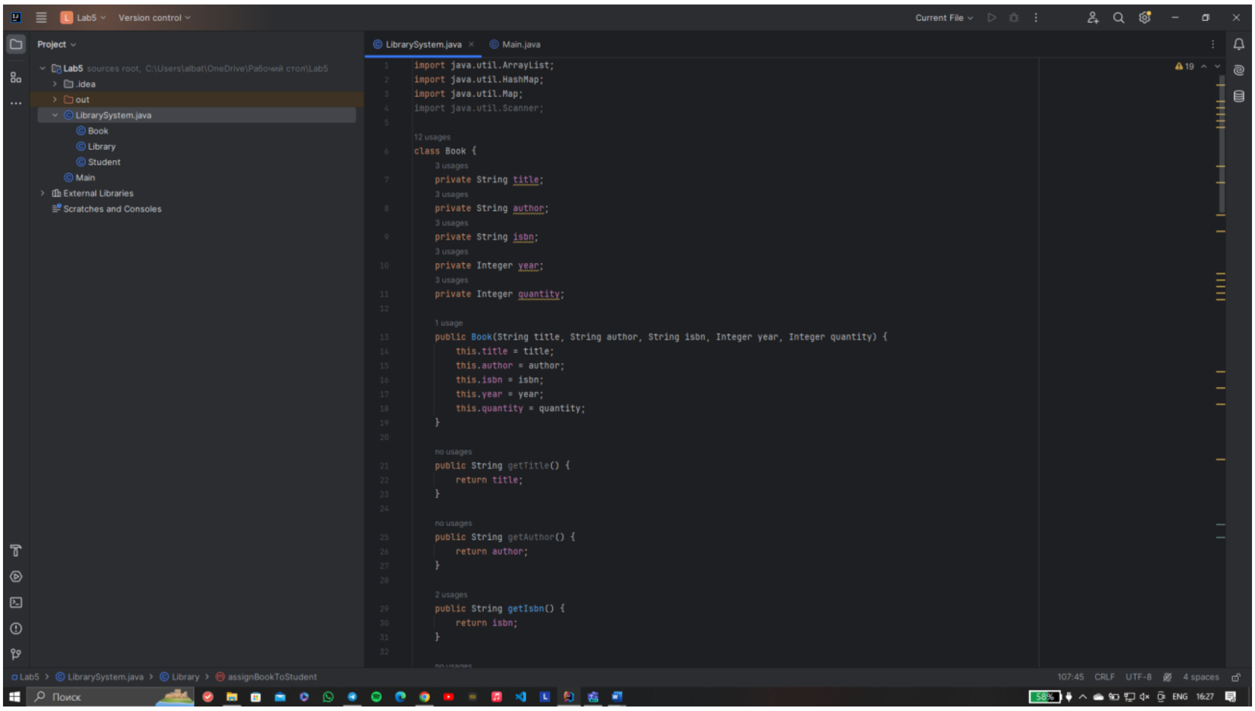
- Uses a `while (true)` loop to keep the program running until the user chooses to exit (option 7).

- Ensures that the user can perform multiple operations without restarting the program.

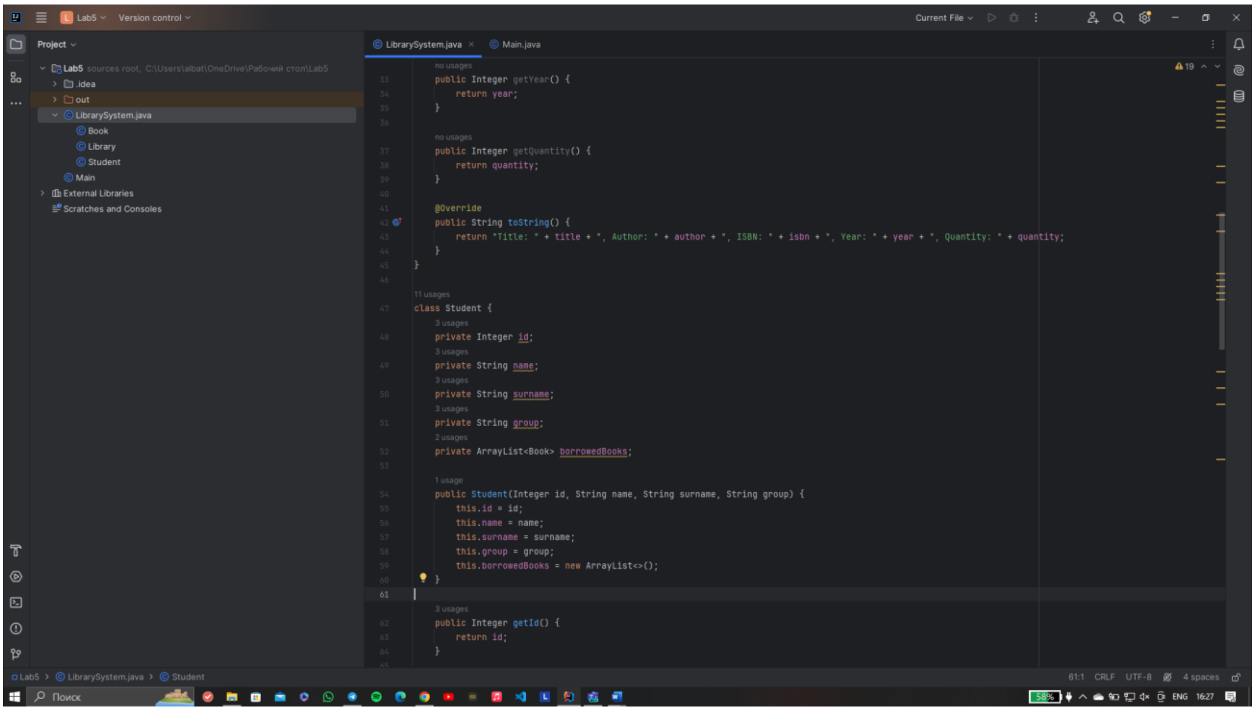
5. Resource Management:

- Closes the `Scanner` object (`scanner.close()`) and exits the program (`System.exit(0)`) when the user chooses to exit.

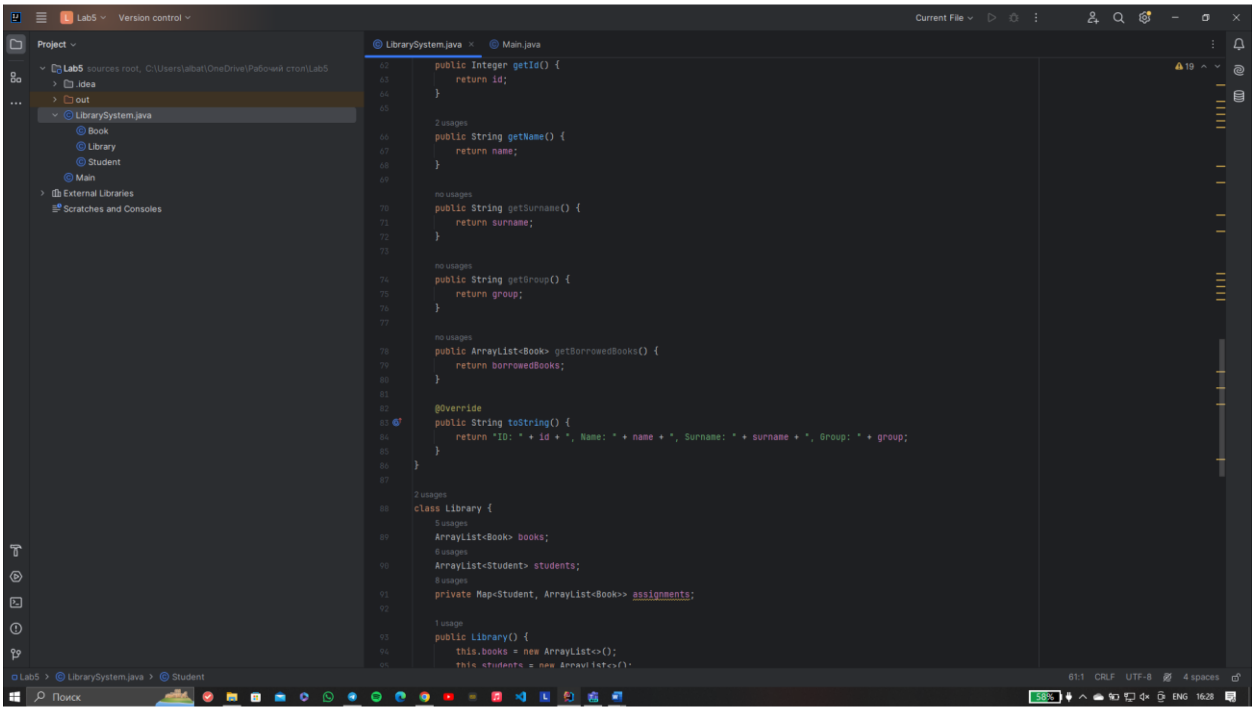
`Main` class provides a user-friendly interface for managing a library, allowing users to add books and students, assign books to students, remove books from students, and view information about books and students.



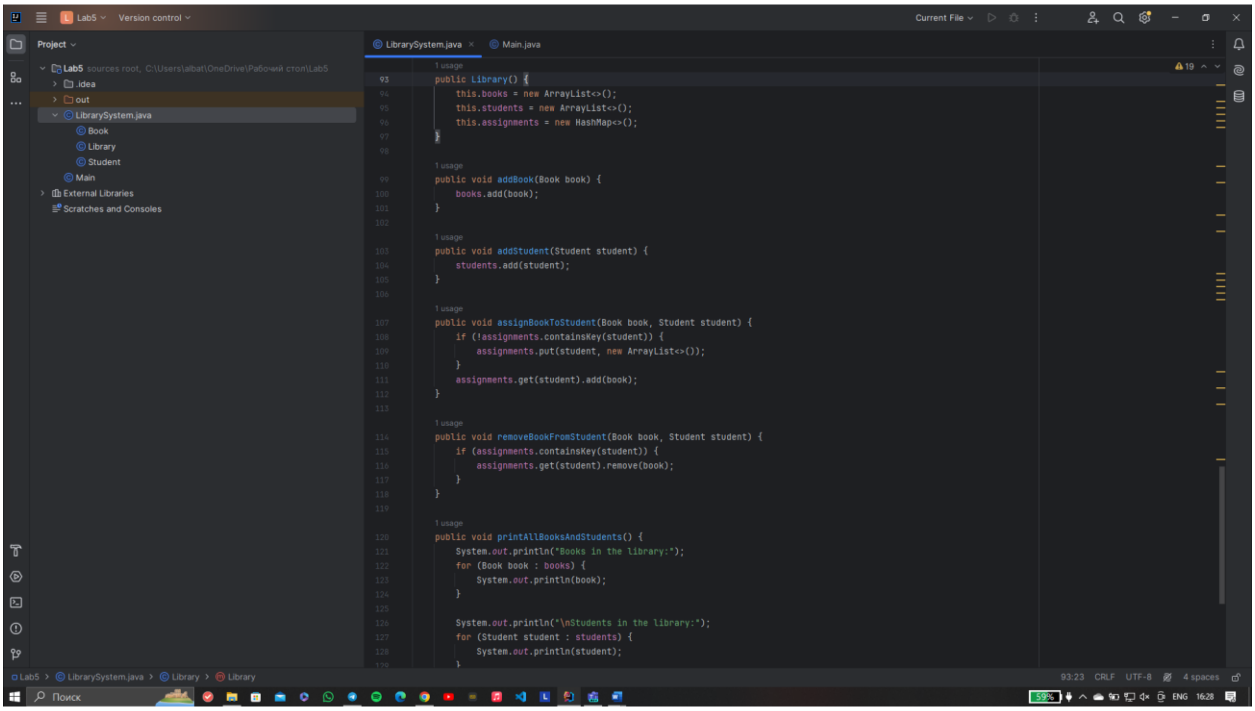
```java
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

class Book {
    private String title;
    private String author;
    private String isbn;
    private Integer year;
    private Integer quantity;

    public Book(String title, String author, String isbn, Integer year, Integer quantity) {
        this.title = title;
        this.author = author;
        this.isbn = isbn;
        this.year = year;
        this.quantity = quantity;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public String getIsbn() {
        return isbn;
    }

```



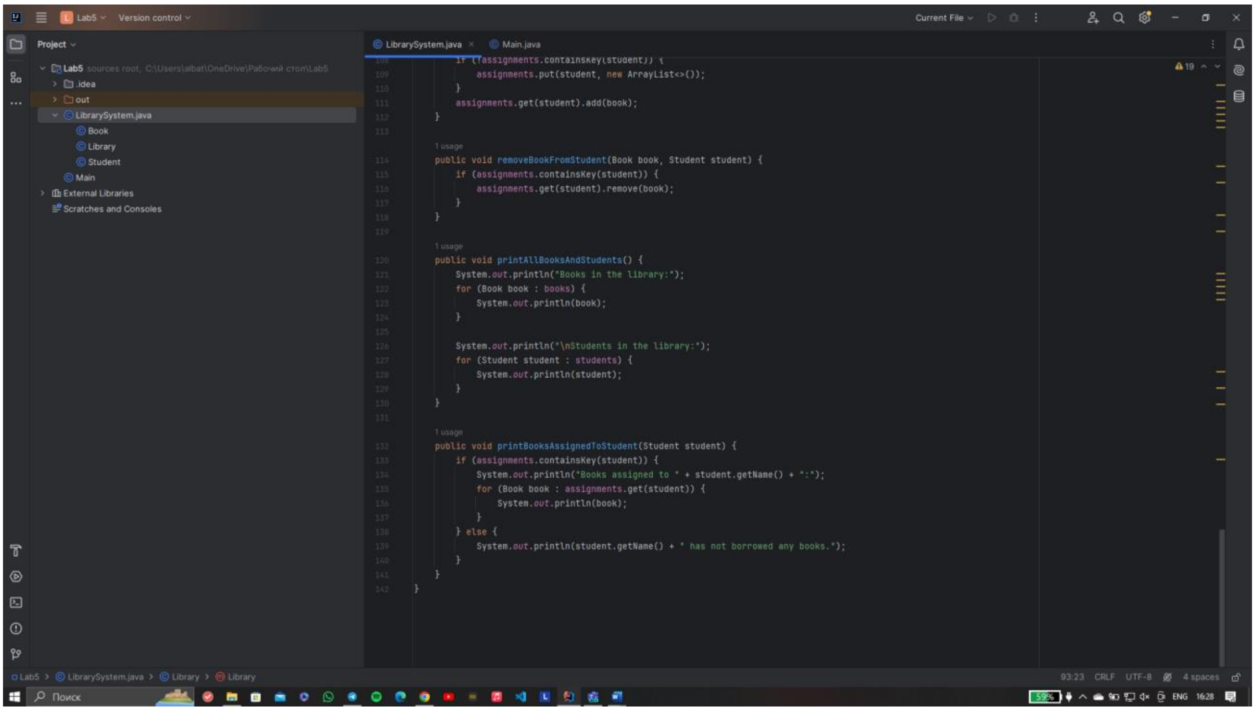
```java
    public Integer getYear() {
        return year;
    }

    public Integer getQuantity() {
        return quantity;
    }

    @Override
    public String toString() {
        return "Title: " + title + ", Author: " + author + ", ISBN: " + isbn + ", Year: " + year + ", Quantity: " + quantity;
    }
}

class Student {
    private Integer id;
    private String name;
    private String surname;
    private String group;
    private ArrayList<Book> borrowedBooks;

    public Student(Integer id, String name, String surname, String group) {
        this.id = id;
        this.name = name;
        this.surname = surname;
        this.group = group;
        this.borrowedBooks = new ArrayList<>();
    }

    public Integer getId() {
        return id;
    }

```



```
    public Integer getId() {
        return id;
    }

    2 usages
    public String getName() {
        return name;
    }

    no usages
    public String getSurname() {
        return surname;
    }

    no usages
    public String getGroup() {
        return group;
    }

    no usages
    public ArrayList<Book> getBorrowedBooks() {
        return borrowedBooks;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Surname: " + surname + ", Group: " + group;
    }
}

2 usages
class Library {
    5 usages
    ArrayList<Book> books;
    6 usages
    ArrayList<Student> students;
    8 usages
    private Map<Student, ArrayList<Book>> assignments;

    1 usage
    public Library() {
        this.books = new ArrayList<>();
        this.students = new ArrayList<>();
```



```
    1 usage
    public Library() {
        this.books = new ArrayList<>();
        this.students = new ArrayList<>();
        this.assignments = new HashMap<>();
    }

    1 usage
    public void addBook(Book book) {
        books.add(book);
    }

    1 usage
    public void addStudent(Student student) {
        students.add(student);
    }

    1 usage
    public void assignBookToStudent(Book book, Student student) {
        if (!assignments.containsKey(student)) {
            assignments.put(student, new ArrayList<>());
        }
        assignments.get(student).add(book);
    }

    1 usage
    public void removeBookFromStudent(Book book, Student student) {
        if (assignments.containsKey(student)) {
            assignments.get(student).remove(book);
        }
    }

    1 usage
    public void printAllBooksAndStudents() {
        System.out.println("Books in the library:");
        for (Book book : books) {
            System.out.println(book);
        }

        System.out.println("\nStudents in the library:");
        for (Student student : students) {
            System.out.println(student);
        }
```



```java
        if (!assignments.containsKey(student)) {
            assignments.put(student, new ArrayList<>());
        }
        assignments.get(student).add(book);
    }

    1 usage
    public void removeBookFromStudent(Book book, Student student) {
        if (assignments.containsKey(student)) {
            assignments.get(student).remove(book);
        }
    }

    1 usage
    public void printAllBooksAndStudents() {
        System.out.println("Books in the library:");
        for (Book book : books) {
            System.out.println(book);
        }

        System.out.println("\nStudents in the library:");
        for (Student student : students) {
            System.out.println(student);
        }
    }

    1 usage
    public void printBooksAssignedToStudent(Student student) {
        if (assignments.containsKey(student)) {
            System.out.println("Books assigned to " + student.getName() + ":");
            for (Book book : assignments.get(student)) {
                System.out.println(book);
            }
        } else {
            System.out.println(student.getName() + " has not borrowed any books.");
        }
    }
}
```

This part of code defines three classes: `Book`, `Student`, and `Library`, which together form a basic library management system:

1. *Book Class:

   - Represents a book with properties such as title, author, ISBN, publication year, and quantity.

   - Provides a constructor to initialize these properties.

   - Includes getter methods to access the properties.

   - Overrides the `toString` method to return a formatted string representation of the book.

2. Student Clas:

   - Represents a student with properties such as ID, name, surname, group, and a list of borrowed books.

   - Provides a constructor to initialize these properties, including an empty list for borrowed books.

   - Includes getter methods to access the properties.

   - Overrides the `toString` method to return a formatted string representation of the student.

3. Library Class:

   - Manages the books, students, and their assignments.

   - Contains ArrayLists to store books and students, and a HashMap to store assignments (books assigned to students).

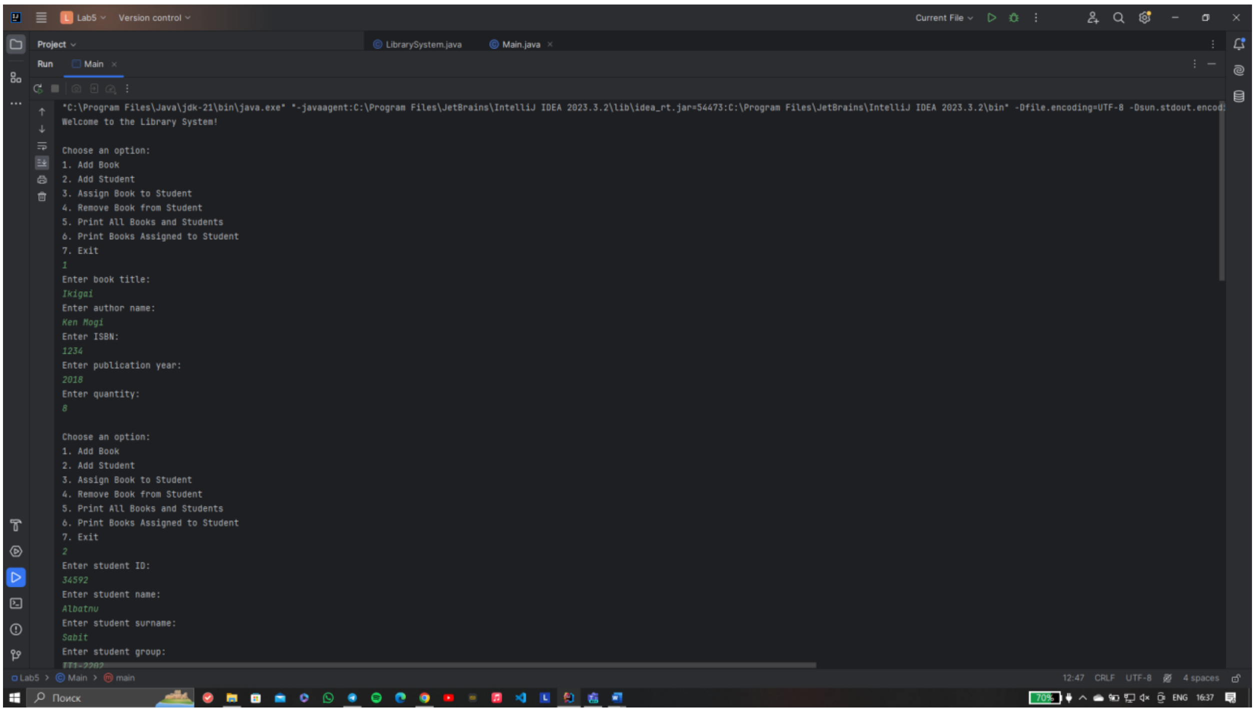
   - Provides methods to add books and students to the library, assign books to students, remove books from students, and print information about books and students.

- The `printAllBooksAndStudents` method iterates over the books and students to print their information.

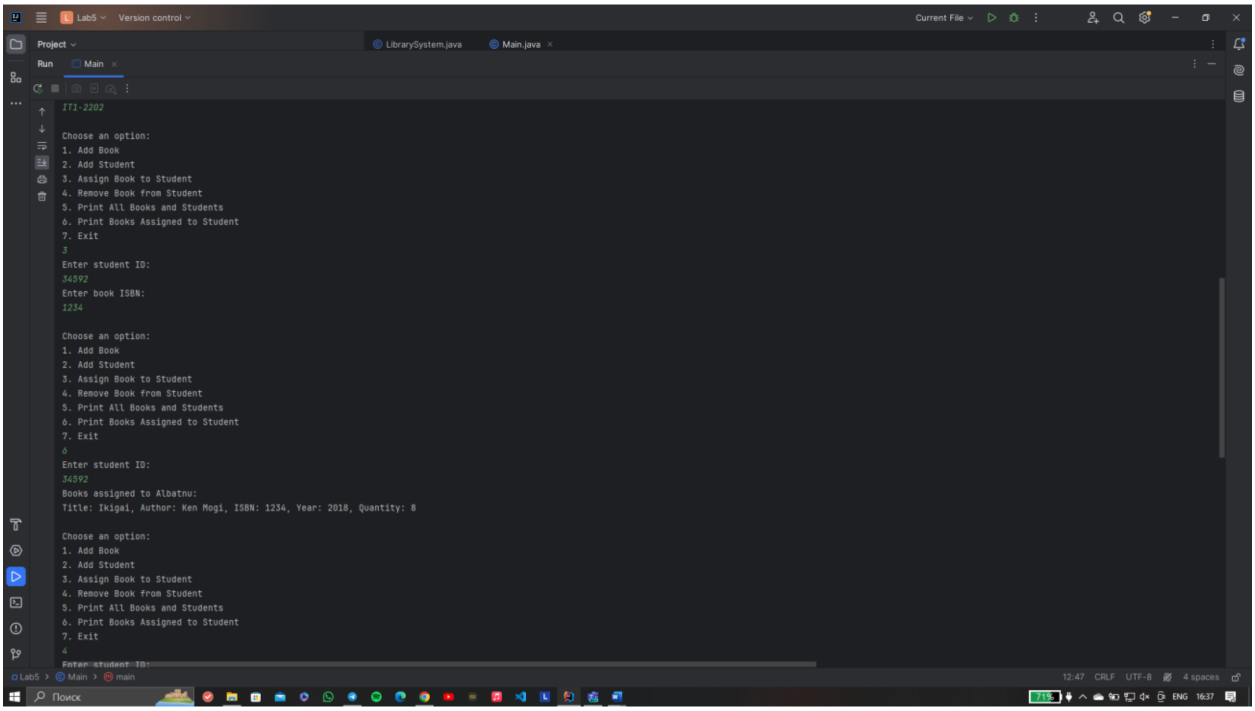
- The `printBooksAssignedToStudent` method prints the books assigned to a specific student, if any.

So, these classes work together to create a simple library management system that allows adding books and students, assigning books to students, removing books from students, and viewing information about books and students.

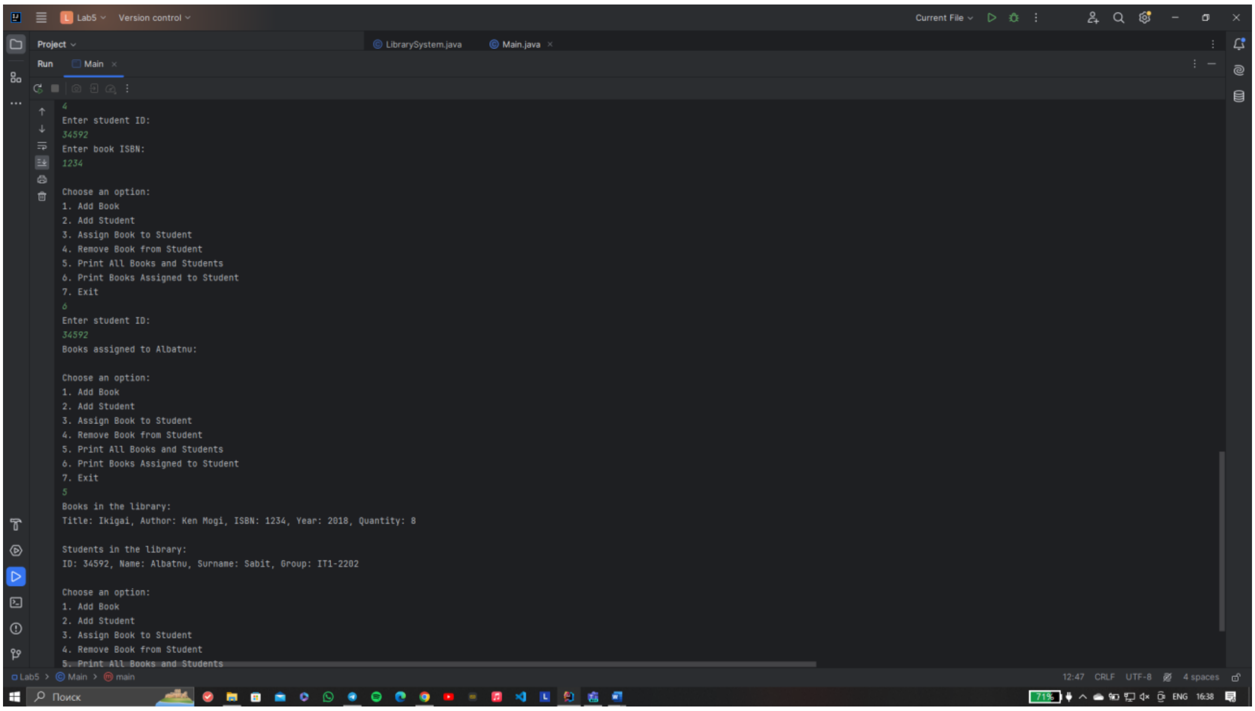
Here is output:



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\lib\idea_rt.jar=54473:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encod
Welcome to the Library System!

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
1
Enter book title:
Ikigai
Enter author name:
Ken Mogi
Enter ISBN:
1234
Enter publication year:
2018
Enter quantity:
8

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
2
Enter student ID:
34592
Enter student name:
Albatnu
Enter student surname:
Sabit
Enter student group:
IT1-2202
```



```
IT1-2202

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
3
Enter student ID:
34592
Enter book ISBN:
1234

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
6
Enter student ID:
34592
Books assigned to Albatnu:
Title: Ikigai, Author: Ken Mogi, ISBN: 1234, Year: 2018, Quantity: 8

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
4
Enter student ID:
```



```
4
Enter student ID:
34592
Enter book ISBN:
1234

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
6
Enter student ID:
34592
Books assigned to Albatnu:

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
6. Print Books Assigned to Student
7. Exit
5
Books in the library:
Title: Ikigai, Author: Ken Mogi, ISBN: 1234, Year: 2018, Quantity: 8

Students in the library:
ID: 34592, Name: Albatnu, Surname: Sabit, Group: IT1-2202

Choose an option:
1. Add Book
2. Add Student
3. Assign Book to Student
4. Remove Book from Student
5. Print All Books and Students
```