INTERNET TECHNOLOGY AND WEB SERVICES

Midterm 2 Project

Dynamic Web Application

LibraryOnline

Student: Nina Krstovic SDE 2018230089

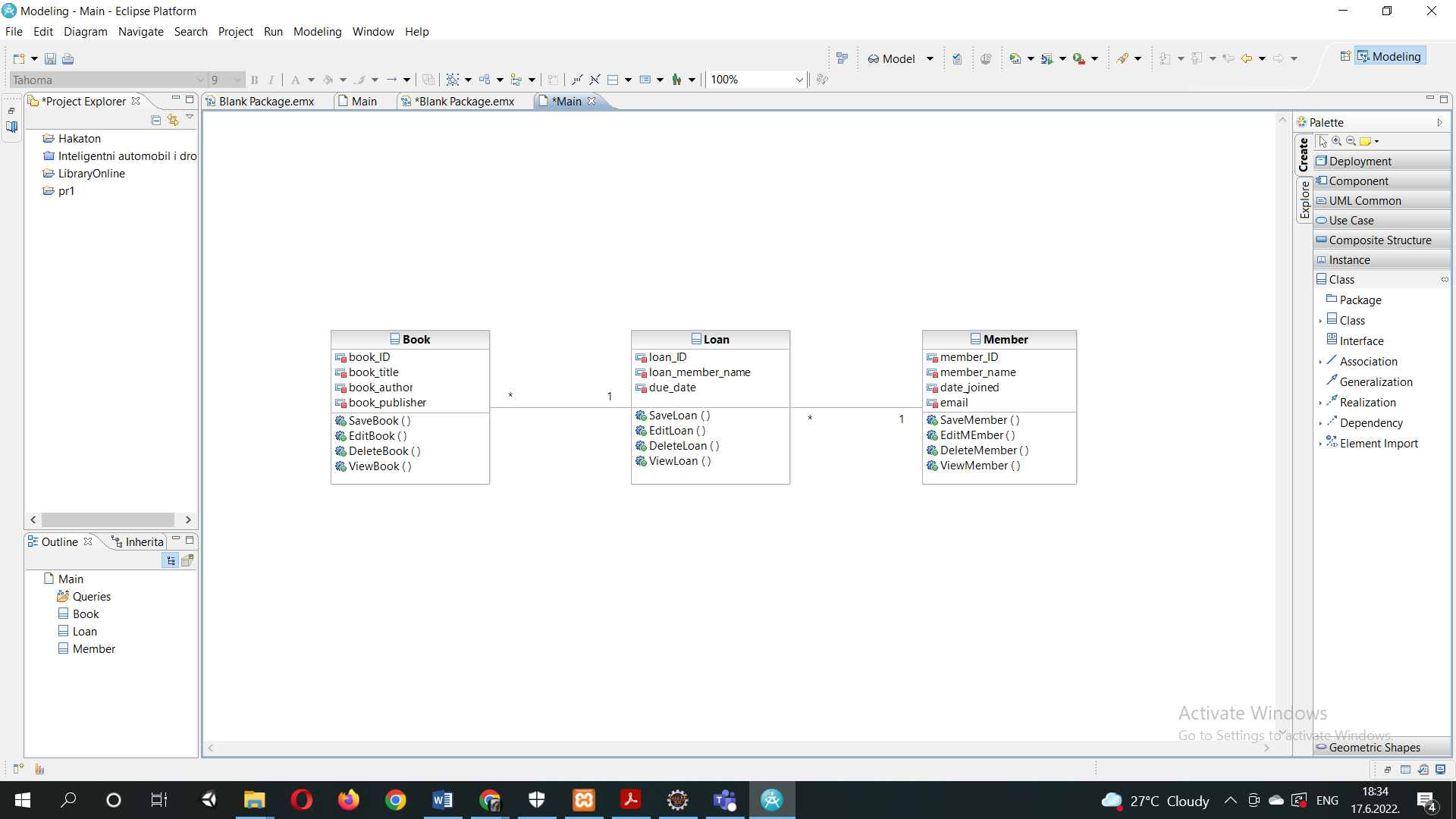
Introduction

For this project I decided to create a dynamic web application called LibraryOnline. It is a type of registry application connected to the MySQL database and serves as a way to record all the books a library might have in their inventory, as well as its registered members and active loans.

Architecture and technology

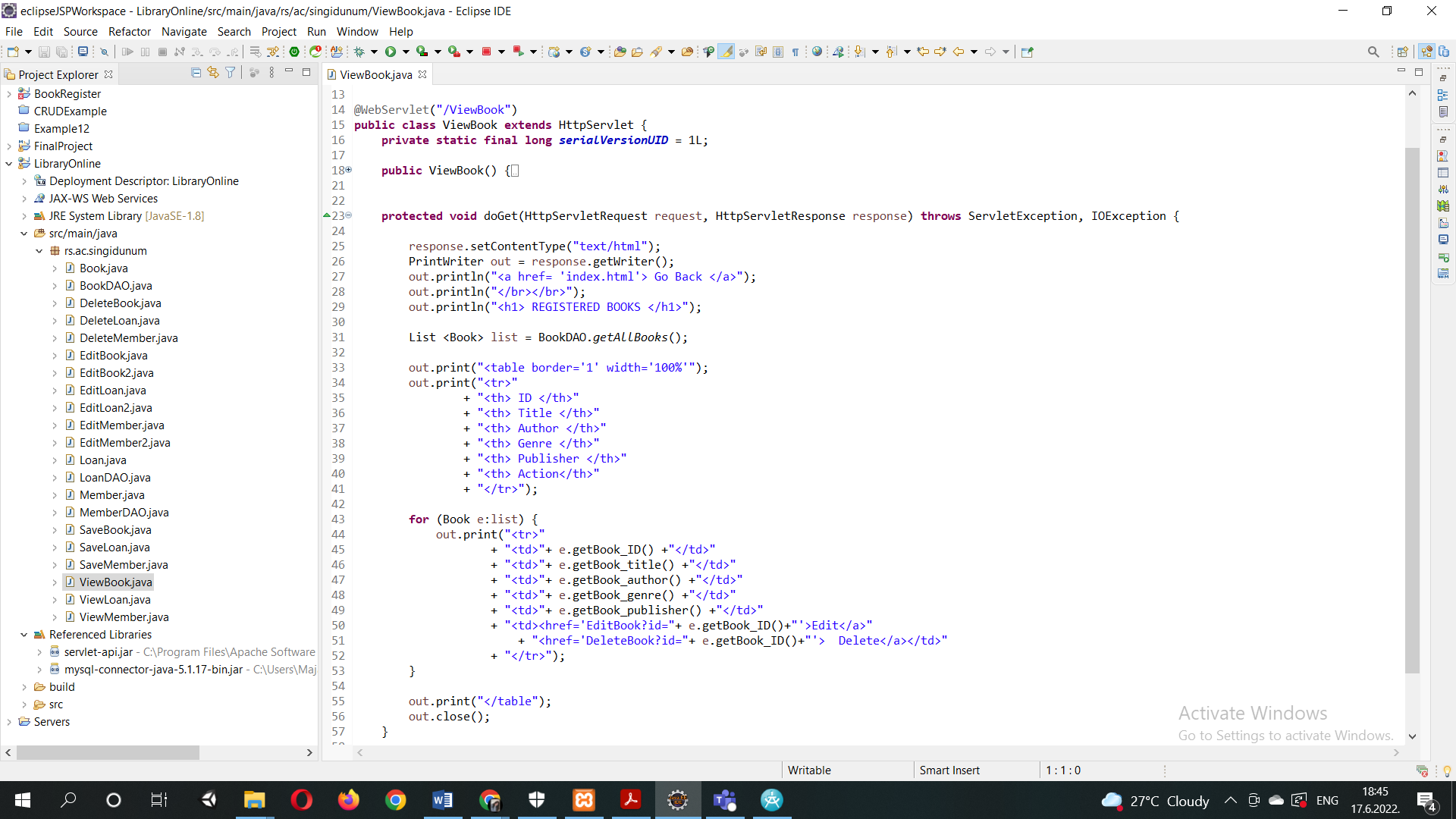
LibraryOnline is written using java programming language. It utilizes java ordinary java classes (i.e. Book.java) as well as DAOs or Data Access Objects and Servlets.  
 In our database, there are three tables – book, member and loan, each table has their own java class and DAO.  
 Java classes are used mainly for getters and setter, whereas DAOs are used for functionalities regarding data stored or to be stored in the database.   
 Servlet classes used allow us to manipulate provided data, to view it, edit it, save it or delete it. With that being said, each ordinary java class has their own respective servlet classes for actions just mentioned. These are: SaveBook, SaveMember, Save Loan, EditBook, EditBook2, EditMember, EditMember2, EditLoan, EditLoan2, ViewBook, ViewMember, ViewLoan, DeleteBook, DeleteMember and DeleteLoan.  
 All of the servlets above mentioned are accessed through html pages. In this project there   
are four such files: index.html which serves are a “landing” page, Book.html which provides a form to be filled for new book submissions, Member.html for new member registration and Loan.html for new Loan.html for new loans. These forms once saved are connected to the saving servlets which will the the provided data and upload it to the MySQL database.   
 As for the view servlets they are also accessed through the index.html page and lead to a table that displays all the data retrieved for that specific table from the database.   
 The edit and delete servlets are shown only once there is data in the table, and they can be accessed through their respective links.

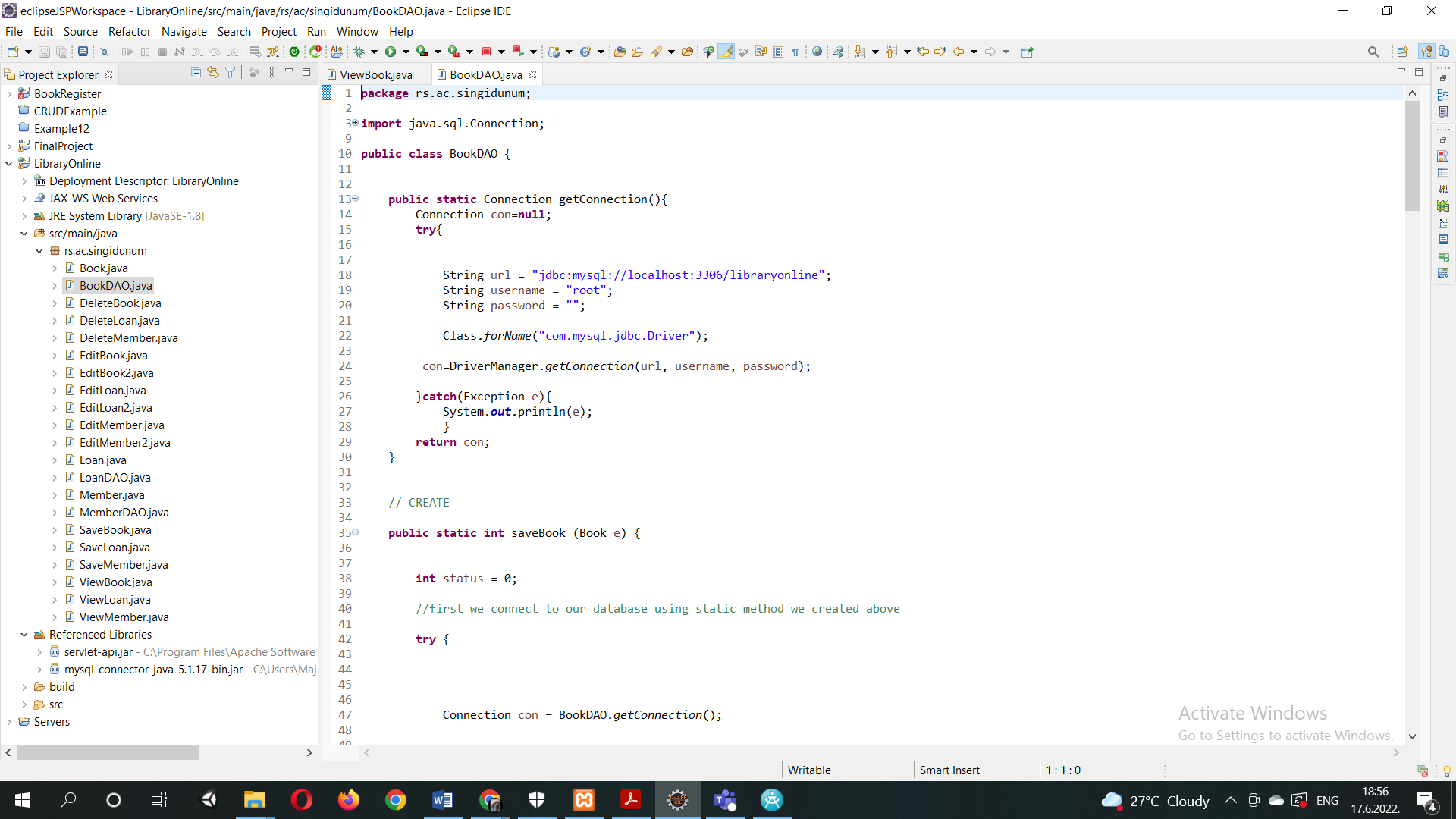
Database Model

In this project, as mentioned, I used a relational MySQL database to store the data. The database has three tables connected with “one to many” relationships that function as:  
- A member can take out multiple loans, but one loan can only be made to one member (1-\*,1-1)  
- A loan can have multiple books, but a book can only be given out on one loan (1-\*,1-1)   
  
 

Implementation

As previously stated, the whole project is based around servlet technology and how they can be used to manipulate data in the database.   
 Here is a code snipped showing a creation of ViewBook servlet:

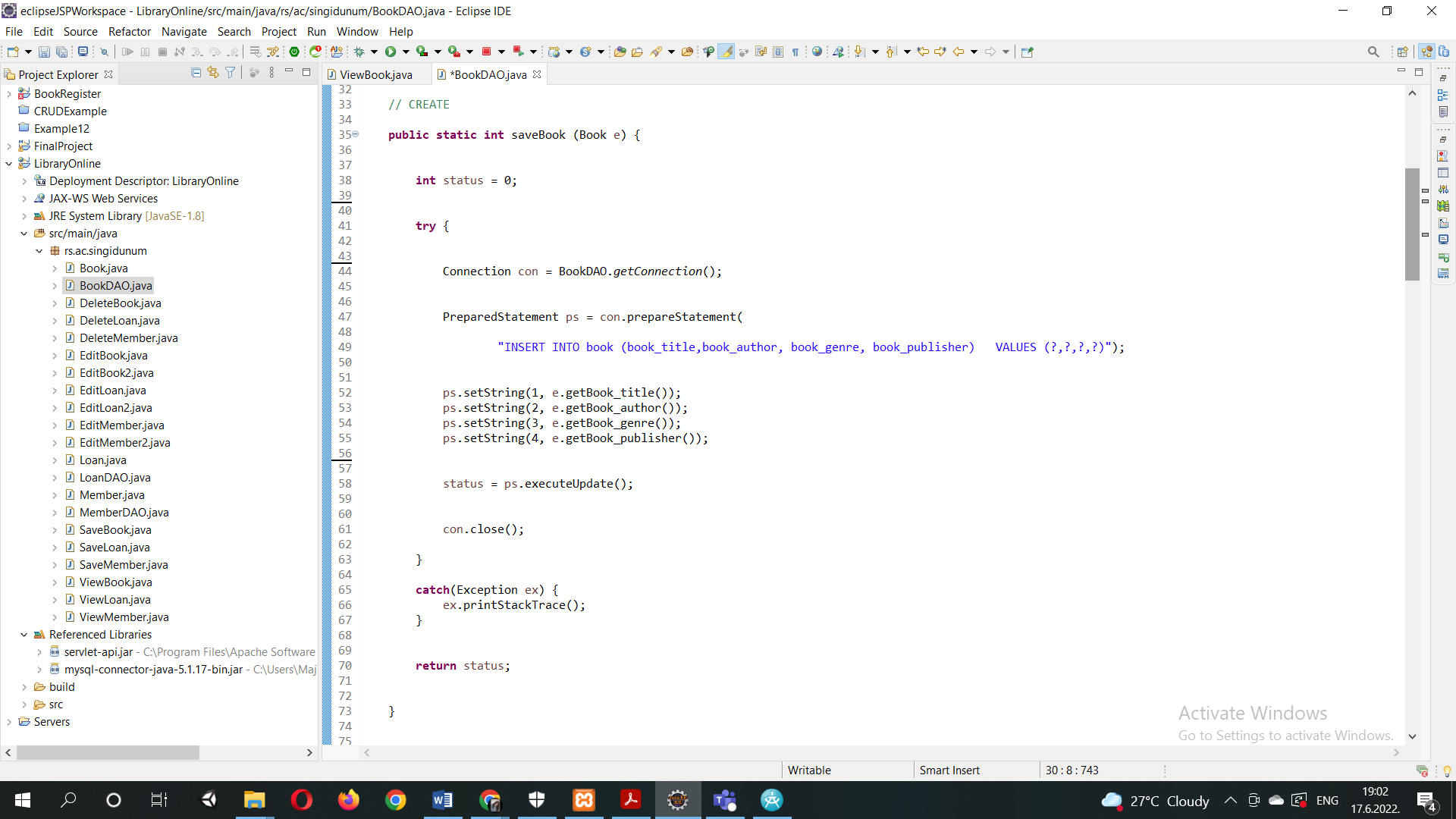


Servlets can either run as a doGet or as a doPost method. DoGet method is used to request information, whereas a doPost is used to provide information. In this particular servlet a table is created and data pulled from the database through the invocation of DAO’s functions and with commands such as e.getBook\_author is inserted into described table fields.

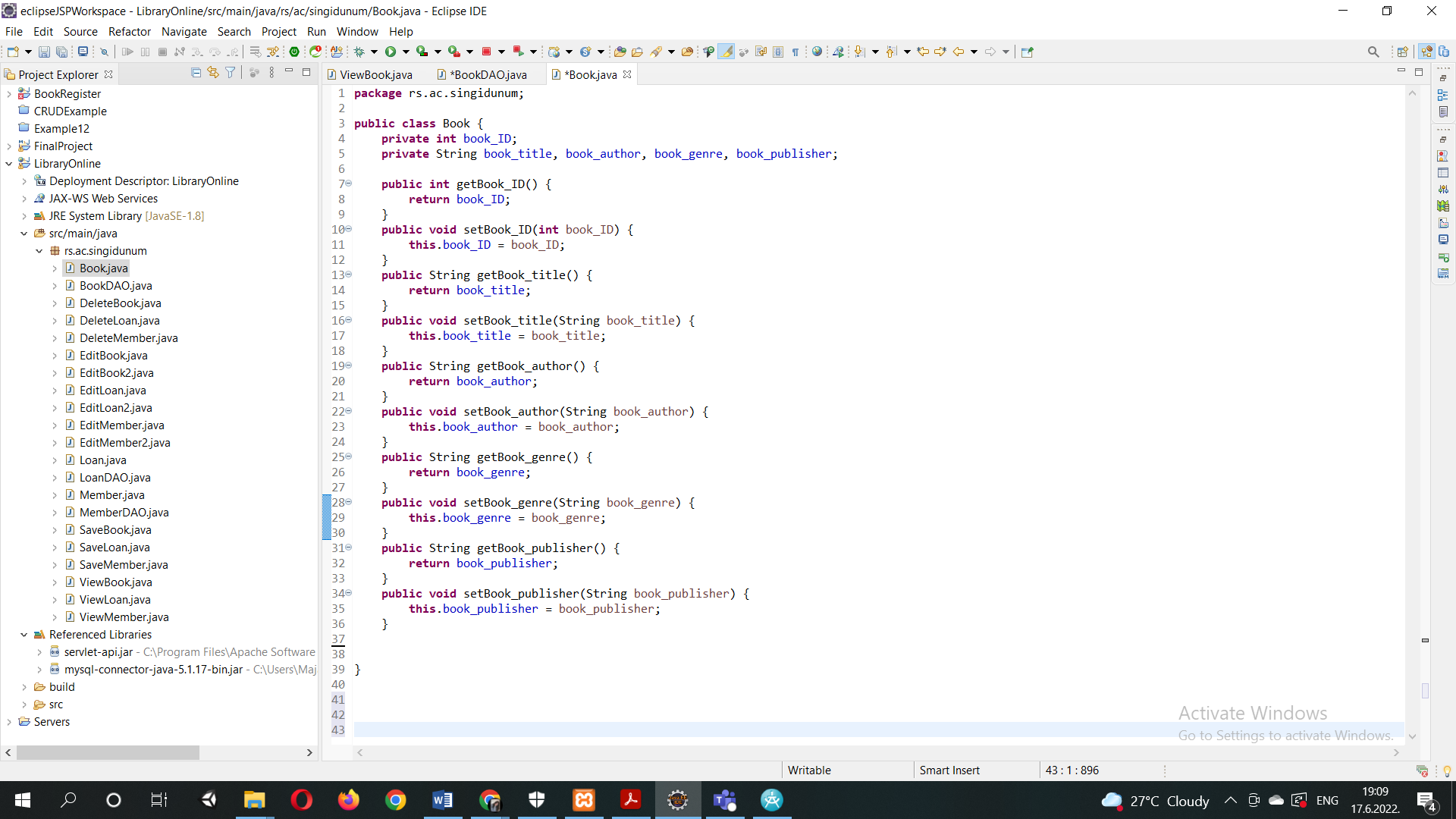
DAOs are used primarily to connect to the database via the JDBC Driver, but besides that they contain within themselves CRUD operations that are invoked inside of servlets.   
Here is how the database connection is established:

As for the CRUD operations, they all utilize database connectivity and prepareStatment() method to execute a desired quary.

For example:

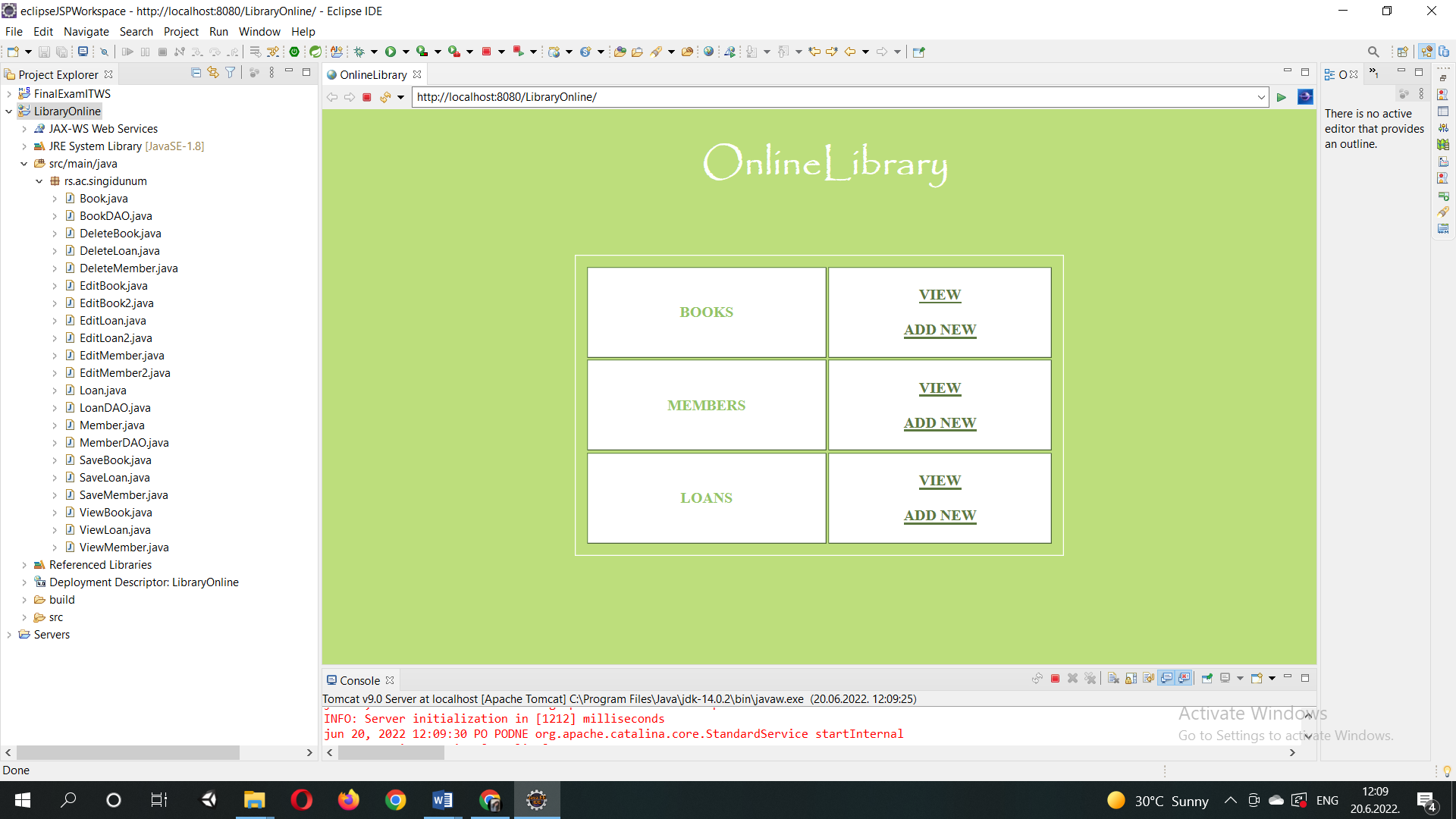


Within these CRUD operations getters and setters from the ordinary java classes are being called. This is what those java classes look like:

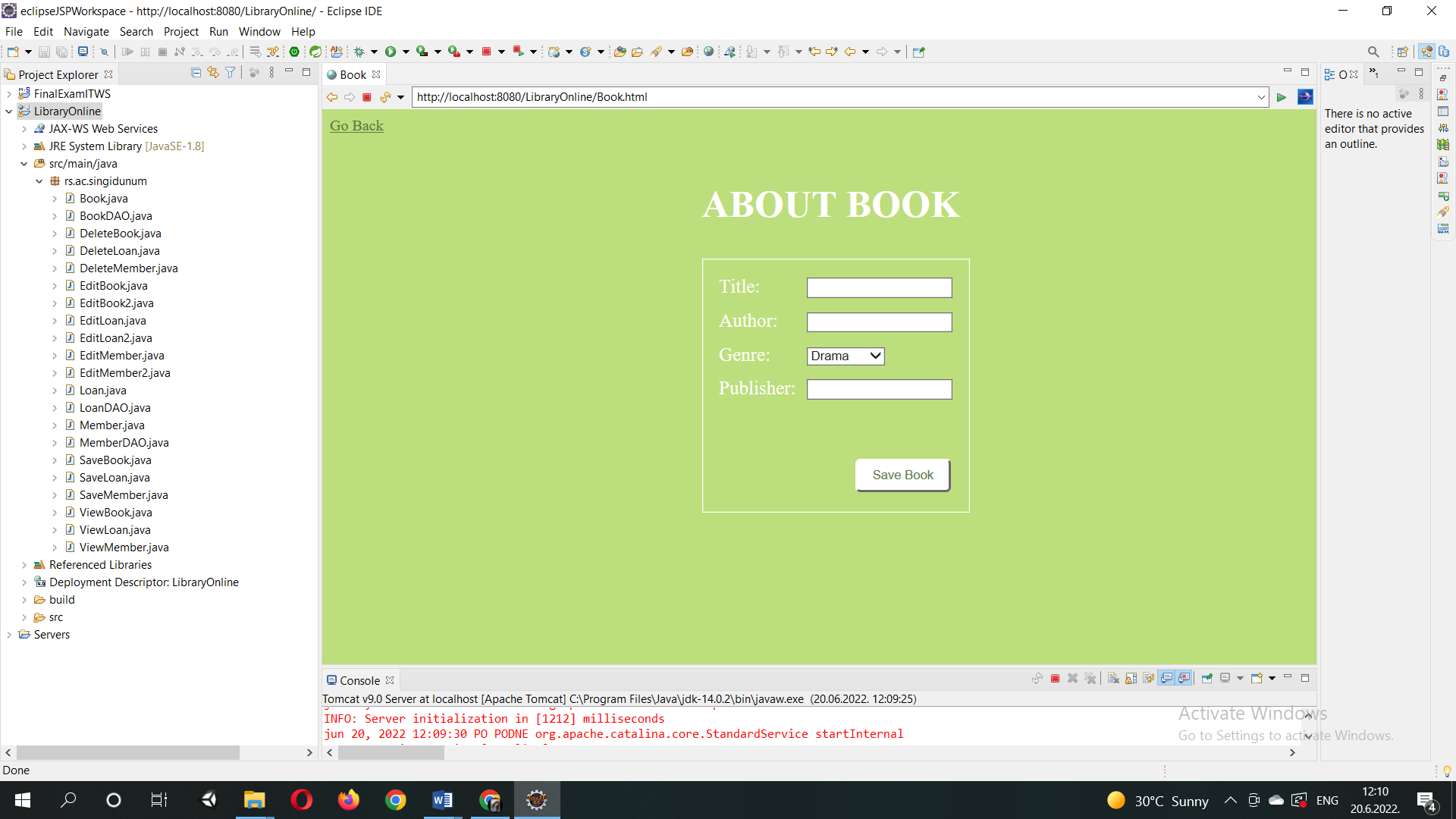


Functionalities

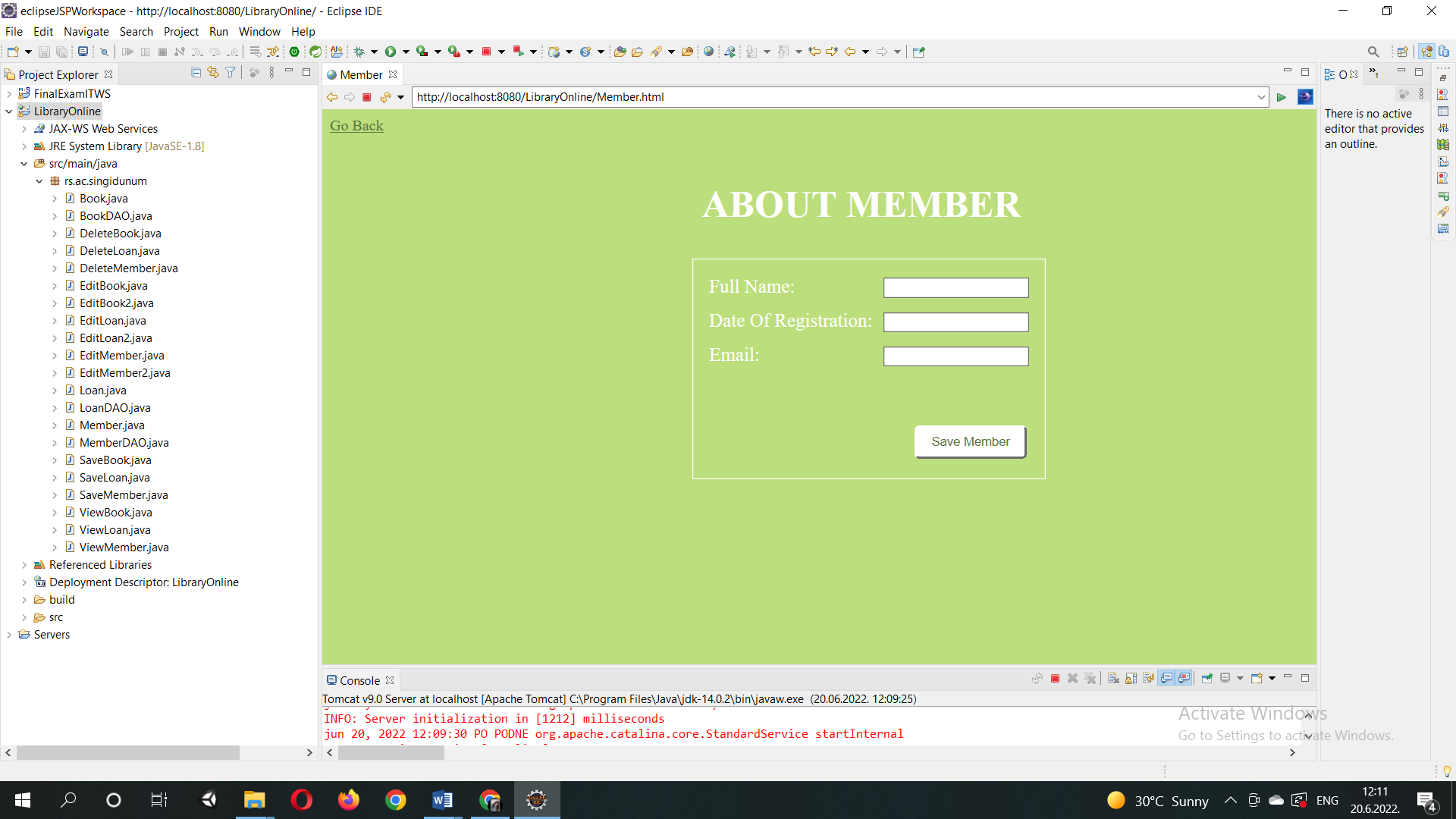
In order to be able to present the code we made, we need to be able to contain it within files that a web browser can “understand”. Therefore, to be able to present and provide all the functionalities of this project, four .html with respective .css files exist. Here are their views:  
  
Index.html:



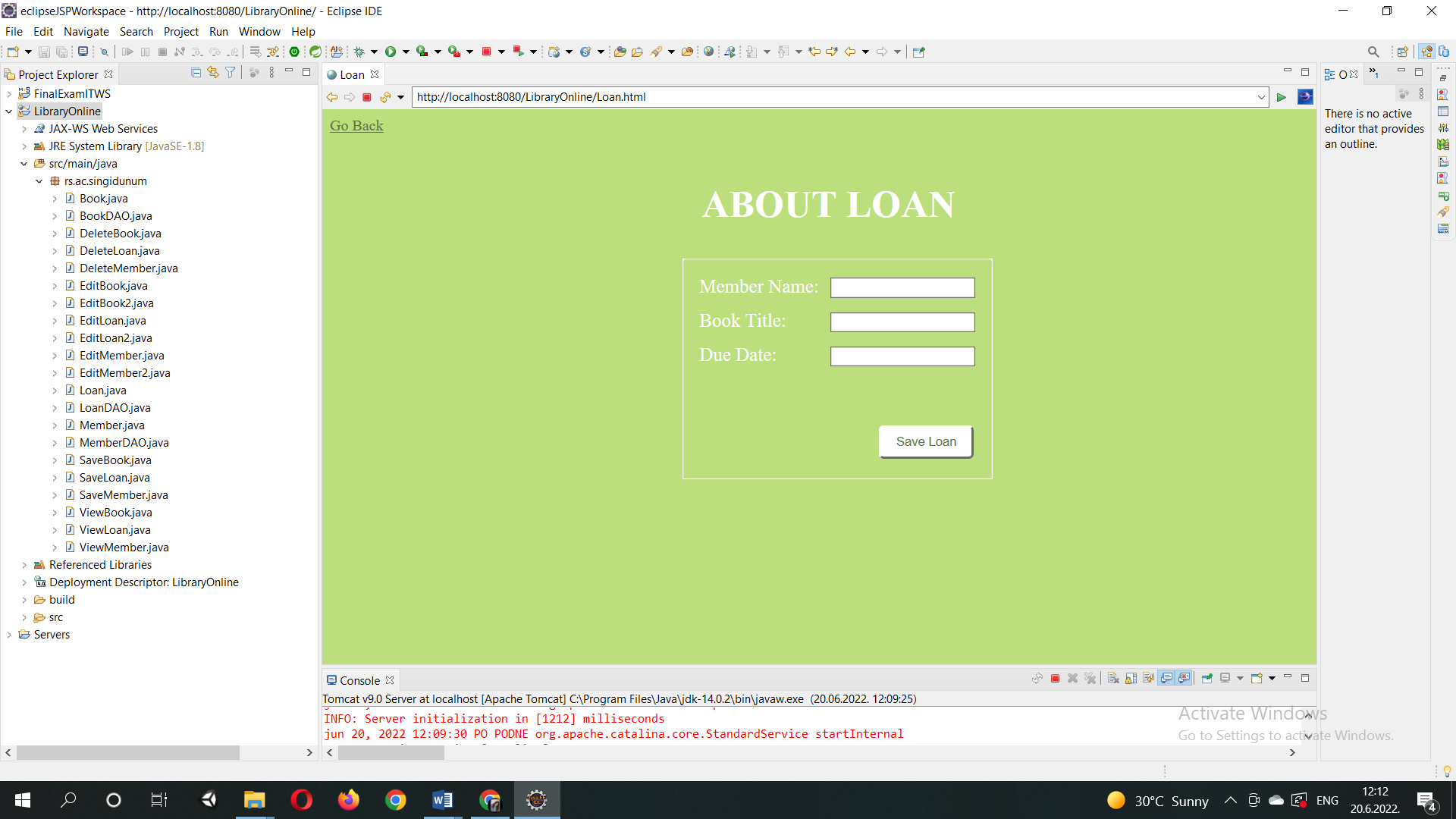
Book.html:



Member.html:



Loan.html:



Conclusion

This project is a fairly simple one, yet as such it still can serve a purpose. However, if it were to be implemented further or to be used in actual real world situations, it would have to have login capabilities as well as many other functionalities regarding the contents themselves. Other tables could be added to the database and implemented in the similar manner as shown here, such as different types of members (i.e. student member, professor member), librarian, administrator etc.