Lebanese American University

Text

Description automatically generated

Group 6

Database Systems - 10554

Section 31 – MWF

Instructor: Joe Tekli

**Library Management System**

**Apollo Script**

Group:

**Name:** Mahmoud Kebbi

**ID#:** 202100351

Tuesday, 12 December 2023

Table of Contents

[Table of Figures 2](#_Toc153299903)

[Table of Tables 3](#_Toc153299904)

[Introduction 4](#_Toc153299905)

[I. Background: 5](#_Toc153299906)

[Why Apollo Script? 5](#_Toc153299907)

[Problem: 5](#_Toc153299908)

[Information Needs: 5](#_Toc153299909)

[II. Data Models: 6](#_Toc153299910)

[A. Relational Data Model: 6](#_Toc153299911)

[B. Logical Data Model: 7](#_Toc153299912)

[III. Frame Works, Languages, Tools, and Methods: 8](#_Toc153299913)

[A. Languages: 8](#_Toc153299914)

[1- Java 8](#_Toc153299915)

[2- SQL 8](#_Toc153299916)

[B. Frame Works: 8](#_Toc153299917)

[1- Spring Boot: 8](#_Toc153299918)

[2- JavaFX: 9](#_Toc153299919)

[C. Tools: 10](#_Toc153299920)

[1- IDE’s: 10](#_Toc153299921)

[2- GitHub and Git: 10](#_Toc153299922)

[3- MySQL: 10](#_Toc153299923)

Table of Figures

[Figure 1 Library Relational Model 6](#_Toc153299924)

[Figure 2 Logical Data Model 7](#_Toc153299925)

[Figure 3 Spring Inatializr configuration. 8](#_Toc153299926)

[Figure 4 JavaFX SDK file. 9](#_Toc153299927)

[Figure 5 Project dependencies. 9](#_Toc153299928)

[Figure 6 MySQL-Connector dependency. 10](#_Toc153299929)

Table of Tables

**No table of figures entries found.**

Introduction

The goal of this project is to develop an application that allows users to easily manage a database and manipulate the data inside the database. Apollo Script is a management system application which is relatively small size. It is designed to be used by a local library to manage entities including but not limited to users, staff, books, computers, study rooms etc. The data about all the entities and relationships is stored in a SQL relational database. Client users will be able to view books, details and statistics of their profile, and reserve study rooms. On the other hand, staff and admin users will have broader and more advanced privileges. The application will be using the JavaFx and Spring Boot framework, and the SQL database will be designed and managed on a MySQL Server.

# Background:

## Why Apollo Script?

Given the objective of this project, a library management Java application was chosen to be developed for the following reasons:

1. Library management applications allow access and management of valuable information and have many real-world applications.
2. Existing databases for bookshops are plentiful and easy to acquire.
3. A library management system demands management and manipulation of data has significant variety, size, and complexity.
4. Java is a programming language all team members are accustomed to.
5. Opportunity to further enhance our skills in Java, SQL, and XML.

## Problem:

A program was needed to efficiently handle well-structured and normalized data and provide an easy and interactive graphical user interface for both client and admin users of the library management system application.

## Information Needs:

Research was conducted on the following topics to implement certain functions and features in the application:

* Searching and retrieving image files using Java File Chooser to add to the database.
* Inserting small sized image files (less than 500KB) in the database.
* Using Spring Boot Framework to use multiple Java APIs.
* Implementing email scheduler.
* Enabling remote access to the MySQL Server.
* Implementing access control features of MySQL server.
* Implementing an interactive and clean graphical user interface for client users of the library application.

# Data Models:

## Relational Data Model:

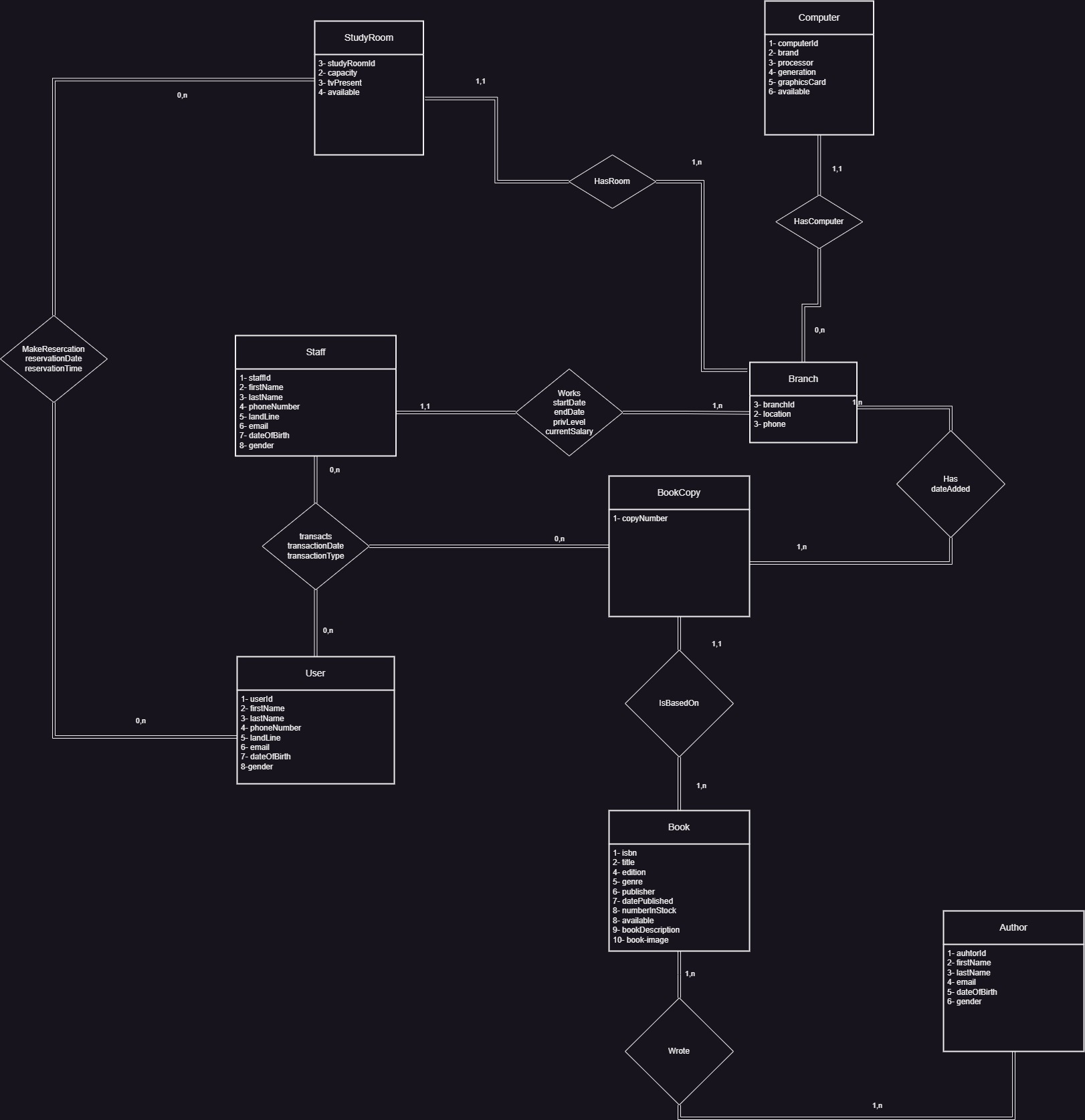


Figure Library Relational Model

## Logical Data Model:

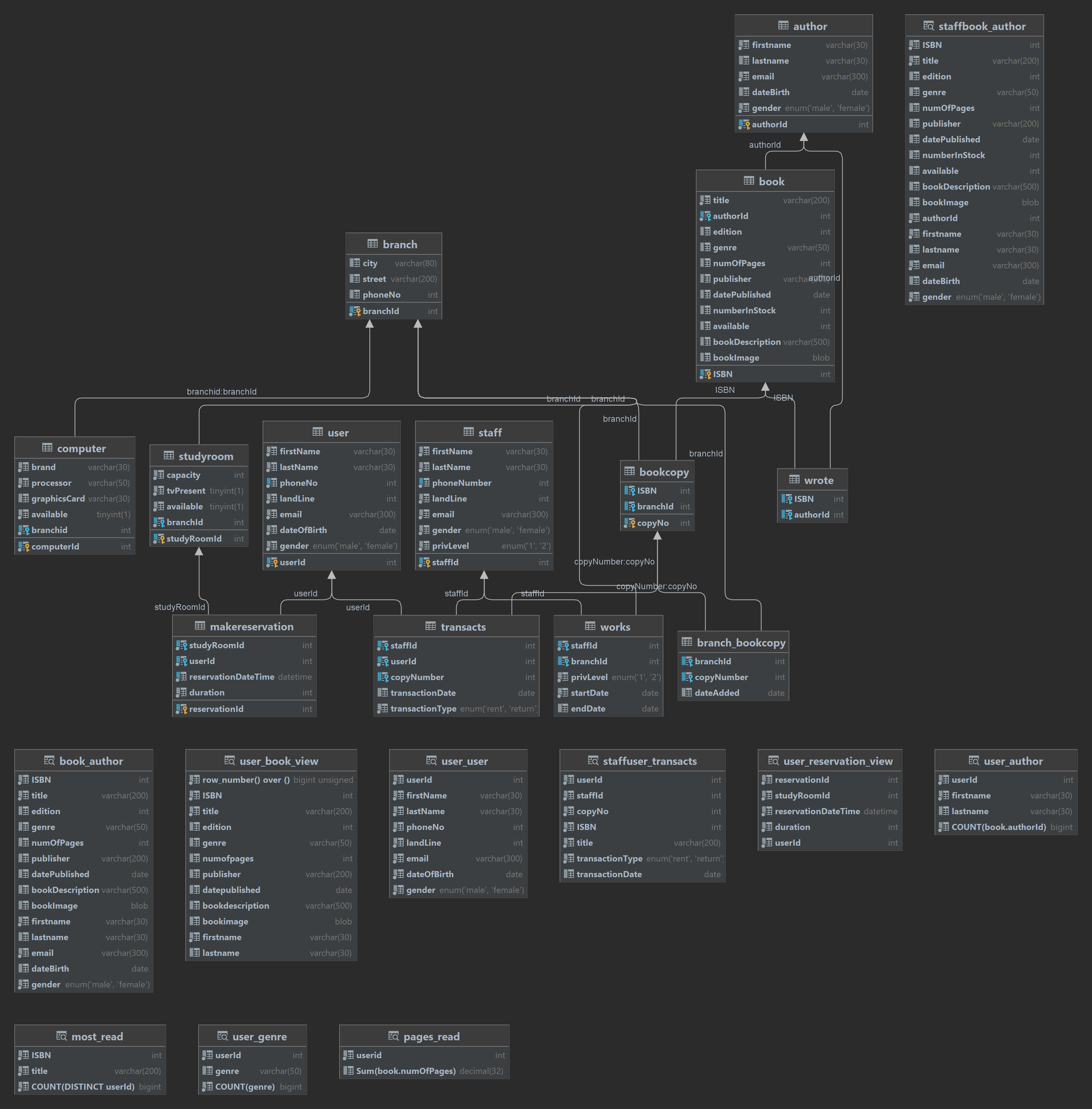


Figure Logical Data Model

# Frame Works, Languages, Tools, and Methods:

## Languages:

### Java

Java was used to develop desktop applications.

### SQL

SQL was used to define and manipulate the project’s relational Database

## Frame Works:

### Spring Boot:

Spring Framework is an open-source Java framework that facilitates building and launching Java applications and services. Spring Boot comes with a default embedded web server, namely Apache Tomcat, which enables developers to develop production-grade applications. It also provides starter dependencies to simplify the build configuration. Additionally, it offers production-ready features (dependency injection, metrics, health checks). To use the Spring Boot framework the Sprin Initializr was used through the following link [Spring Initializr](https://start.spring.io/) and the Spring project was built according to this configuration:

A screenshot of a computer

Description automatically generated

Figure Spring Inatializr configuration.

The Java Mail Sender API was added to the dependencies of the project. This framework was used in this project specifically to enable sending and scheduling emails in the java application, in addition to its dependency injection capabilities.

### JavaFX:

JavaFX is a Java framework that allows developers to create rich client application through APIs for building GUI application. These applications can run on any device that offers Java support. The JavaFX framework is also highly compatible with the Spring Boot framework. To use JavaFX in the Spring project the JavaFX SDK folder was added to the project folder, then dependencies on the JavaFX libraries were added to the project object model xml file. Next, the JavaFX libraries were added to the project modules.

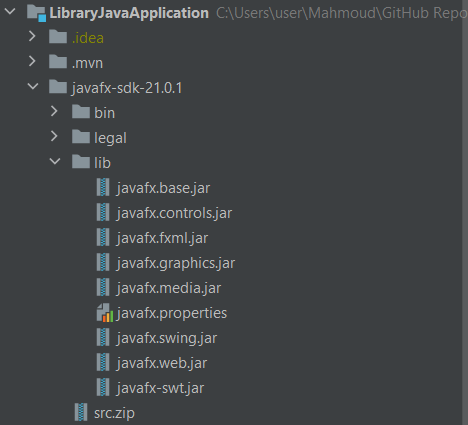


Figure JavaFX SDK file.

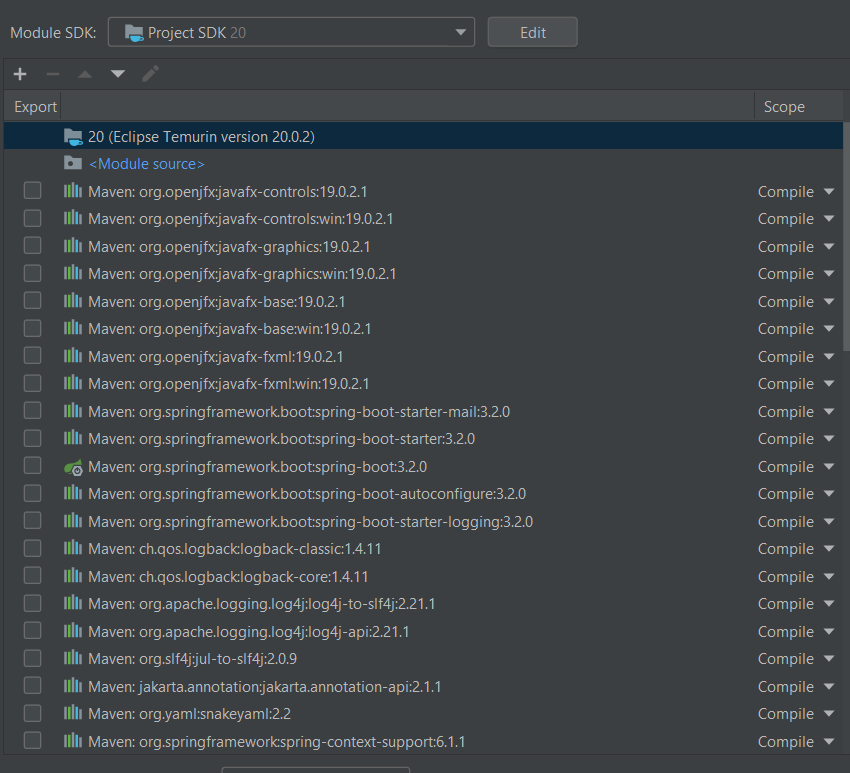


Figure Project dependencies.

## Tools:

### IDE’s:

JetBrains development tools were used by all team members throughout the project.

### GitHub and Git:

The collaboration on the project was done via GitHub through an online repositor, and Git was used for version control. Using GitHub and Git facilitated the development process by allowing all team members to track changes in the project, and allowing each member to work on separate branches to implement features one by one, The collaboration was done the following repository [AbedAlRahmanMneimneh/Library-Management-System (github.com)](https://github.com/AbedAlRahmanMneimneh/Library-Management-System).

### MySQL:

MySQL was chosen as the project’s database management system for the following reason:

1. Used throughout the COE 418 course.
2. Open source.
3. Fast and reliable.
4. Intuitive and easy to use.

To connect the MySQL Server to the Java project the J Connector JAR was added to the project module.

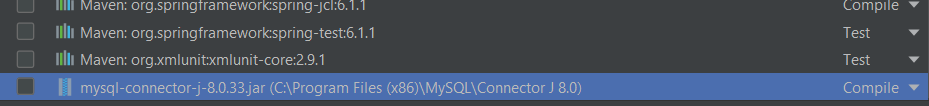


Figure MySQL-Connector dependency.