

Course Management System

International Institute of Information Technology, Naya Raipur



Ayush Mehar

191000011

BTech CSE 3rd Sem

Ishan Kumar Kaler

191000018

BTech CSE 3rd Sem

Abhik Jain

191000001

BTech CSE 3rd Sem

Project Report

Contents

Introduction	3
1 Technologies Used	4
2 DataBase Design	5

Introduction

We made this project as a part of our **Object Oriented Programming - 2** course. This is is **Course Management System**, written in pure Java. This program provides extensive capabilities to teacher to provide courses to students and gor students to join the courses, along with an admin role to manage all things.

All the information in stored in a database using **SQLite3**, a simple SQL implementation which allows all the data to be stored in form of a password-protected binary file which can be shared without much security concerns. For connecting to the database, we used **JDBC SQLite3 Drivers**, along with [java.sql](#) package.

Users are divided into 3 categories: Student, Teacher & Administrator, and their is proper division of powers and capabilities for each role. Students can only query courses, join courses and see their grades. Teachers can query about all things in database, but cannot modify data, except for data concerning his/her particular course. Admins are given ability to modify all the data in the database.

The entire program is written in pure Java, using **OpenJDK 14**, and is available under MIT License (though some softwares and technologies used in this project might be under a different license, and we don't claim to own them). We only used Java instead of other languages often used for this type of software, like Python or JavaScript because our aim from this project was to get a better understanding of Object Oriented Programming and how it is used to do efficient software development.

1 Technologies Used

Java

Java is a popular programming language, which is used to write a wide range of programs because of its general-purpose nature. It is the *de-facto* language for people who wish to code in abstract and Object-Oriented manner, as Java has a well structured and well-defined ways to do Inheritance, Polymorphism and other things done in Object-Oriented Programming.

We used Java in our project for various reasons:

- **General Purpose Language:** Java is a general purpose language, and thus has extensive capabilities allowing us to easily use it exclusively in our project.
- **Self Contained:** Java was designed to inherently be capable of doing almost everything required in software development, and thus is not dependent on any other language, except for SQL which again could have been done in pure Java but not as easily.
- **Extensive Support:** Java is a popular language, and thus has a wide variety of packages to help us write our software. It also has well integrated coding tools which make coding a lot easier.
- **Excellent Documentation:** Various Java compile systems provide a built-in way to generate documentation, and thus is often used by people writing libraries to provide detailed documentation of all the functionalities of the library

SQLite3

SQLite3 is a Query Language implementation which is often used to with Python. We used it to allow easy sharing of database between the developers, as it can be stored in a single password protected binary file.

In real world applications we would use something like MySQL or PostgreSQL which are designed to be accessed over the internet and have more options to limit permissions and access to the database, along with ability to add users. But because sharing and working together on such a database is difficult when working together remotely, we decided to not use it.

JDBC

JDBC stands for **Java DataBase Connectivity**. In order to connect to database, we need to write a separate module just to handle requests and queries to database and get information and response from it. Writing such a module from scratch is a daunting task, and so almost all SQL implementations provide a JDBC Library which can be used directly in our program, so that the developers need not worry about in-depth details of how the database takes in data and just focus on high level logical operations.

For our project, we used [sqlite-jdbc-3.8.9.1](#) to connect to the the database containing all the information.

2 DataBase Design

Our database is rather simple. We have 5 main tables: **student**, **teacher**, **admin**, **courses** & **department**. Apart from that, each course has it's own table, with name same as [courses.id](#) and contains ID of all the students that are enrolled in that course.

Column Name	Data Type
id	int (primary key)
password	string
name	string
join_year	int
email	string
phone	long
grade	float

Table 1: student

Column Name	Data Type
id	int (primary key)
password	string
name	string
email	string
phone	long
salary	float
department	int (references department.id)

Table 2: teacher

Column Name	Data Type
id	int (primary key)
password	string
name	string
email	string
phone	long

Table 3: admin

Column Name	Data Type
id	int (primary key)
name	string

Table 4: department

Column Name	Data Type
id	string (primary key)
name	string
teacher_id	string (references teacher.id)
prereq	string
dept_id	int (references department.id)
status	bool

Table 5: courses