Assignment 2
Analysis and Design Document

Student: Radu Petrisel
Group: 30432

Table of Contents

1. Requirements Analysis

1.1 Assignment Specification

The application is used for the management of students in the Computer Science Department of Technical University of Cluj Napoca. The application has two users (student and teacher/administrator) which must provide a username and a password in order to use the application.

1.2 Functional Requirements

- create/read/update/delete students' information (for teacher/administrator)
- generate reports (teacher)
- add/update/view info (all users)
- create/update /view profile (all users)
- process class enrolment (all users)

1.3 Non-functional Requirements

- reliability
- store data in a SQL Database
- store student reports in a NoSQL Database

2. Use-Case Model

Use case: create student
Level: user-goal level
Primary actor: teacher
Main success scenario: insert student info → system validate info → student created
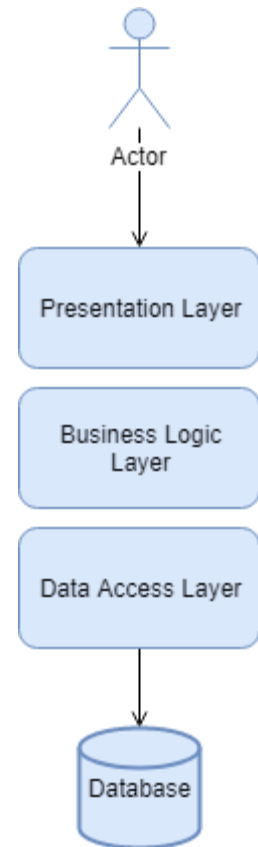**Extensions:** failure → invalid data or student already exists

3. System Architectural Design

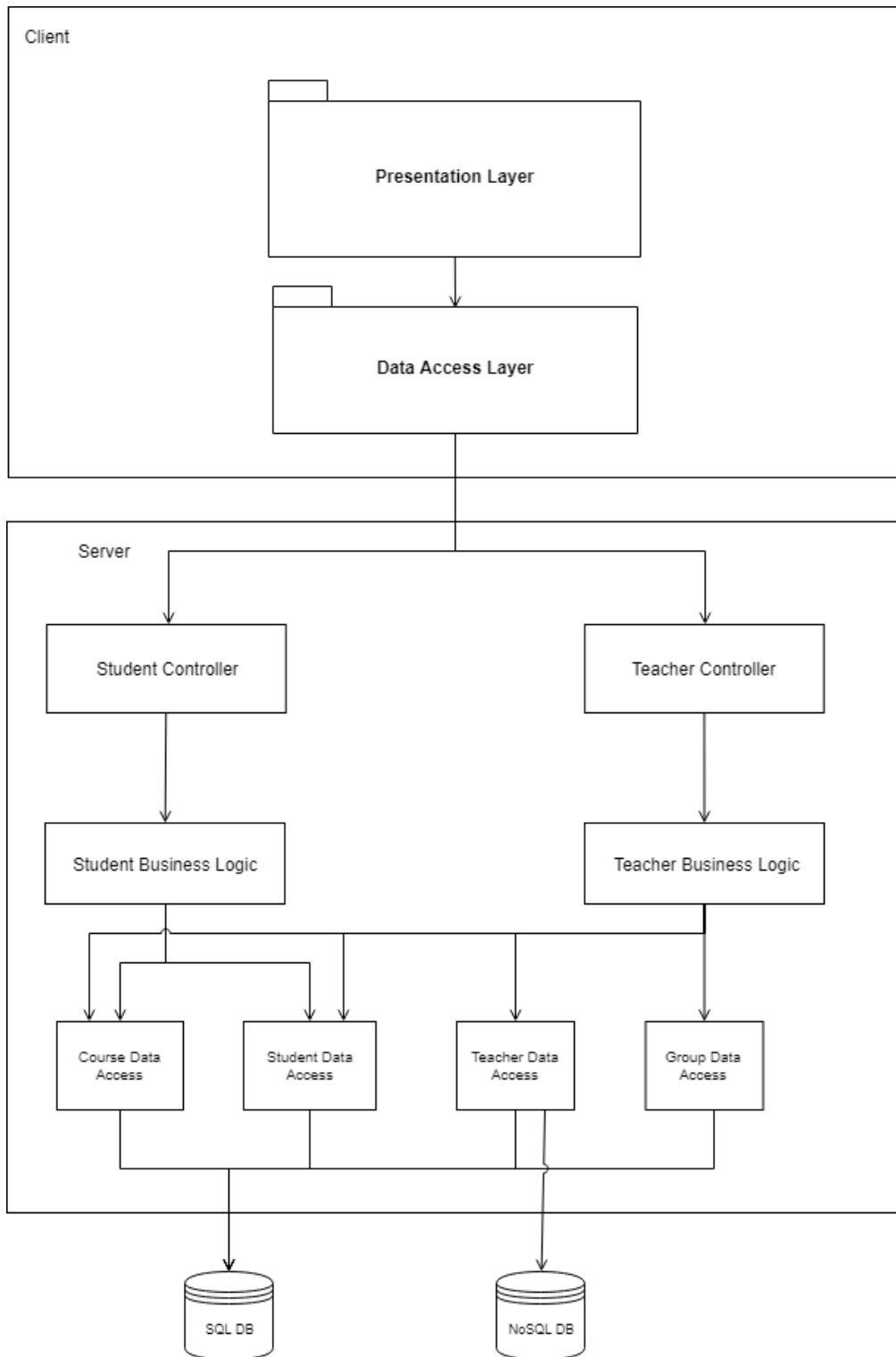## 3.1 Architectural Pattern Description

       This system a Client-Server architecture and MVC (MC – server, VC – client).

       Both the client and the server have a layered architecture. The layered architecture is the most common architectural pattern nowadays. The pattern consists of logically diving the application in layers, each with its own part to play. The server has 3 layers: data access (DAL), business logic (BLL) and presentation layer controllers (PL). The client has two layers – services and presentation layer (views & small controllers – for making requests to the server).

Actor

Presentation Layer

Business Logic Layer

Data Access Layer

Database

3.2 Diagrams

Package Diagram

Deployment Diagram

System 1		System 2		System 4
SQL DB	——	Server	——	Client (presentation)

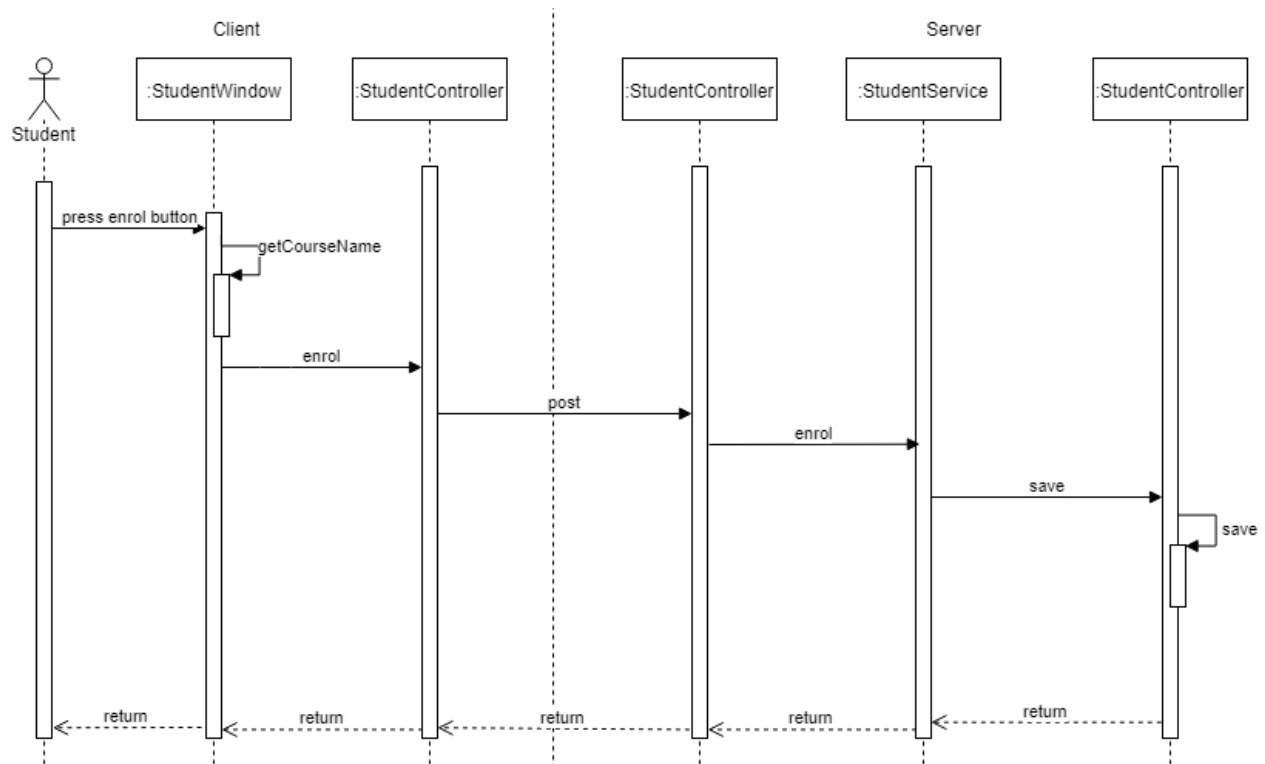| System 3 |
| --- |
| MongoDB |
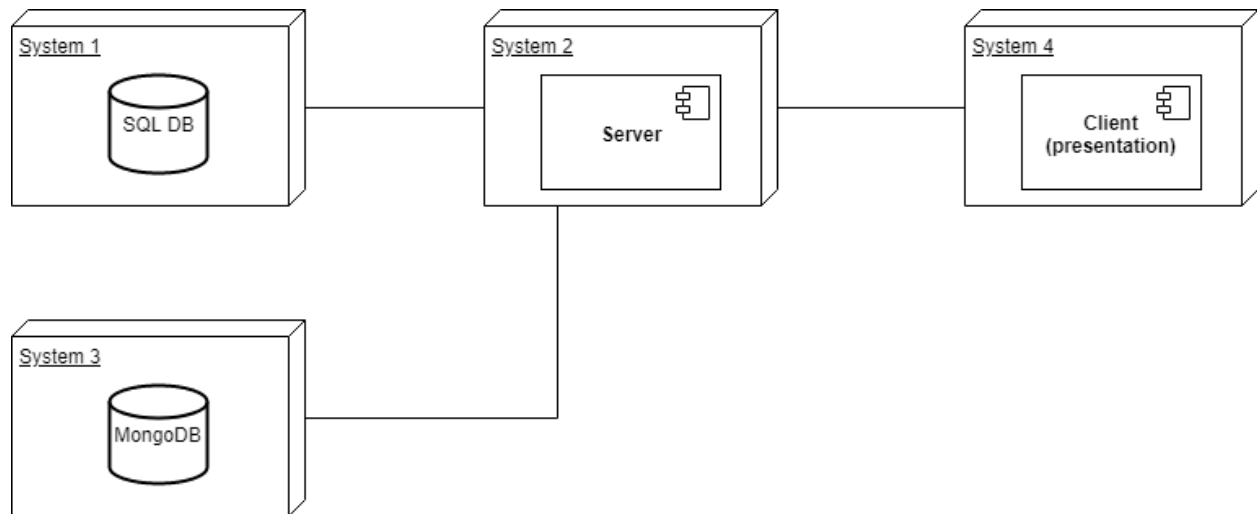
4. UML Sequence Diagram

5. Class Design

5.1 Design Patterns Description
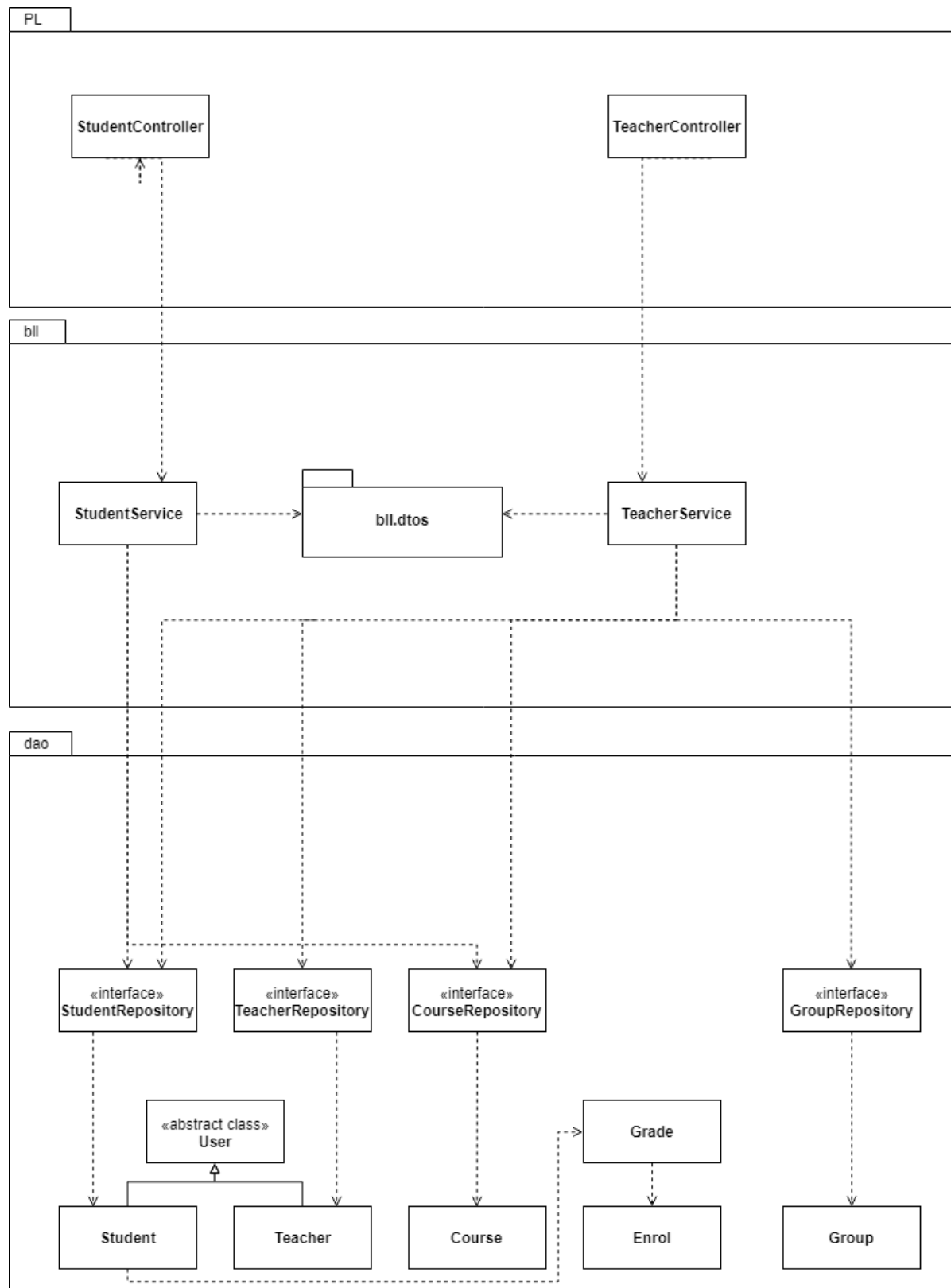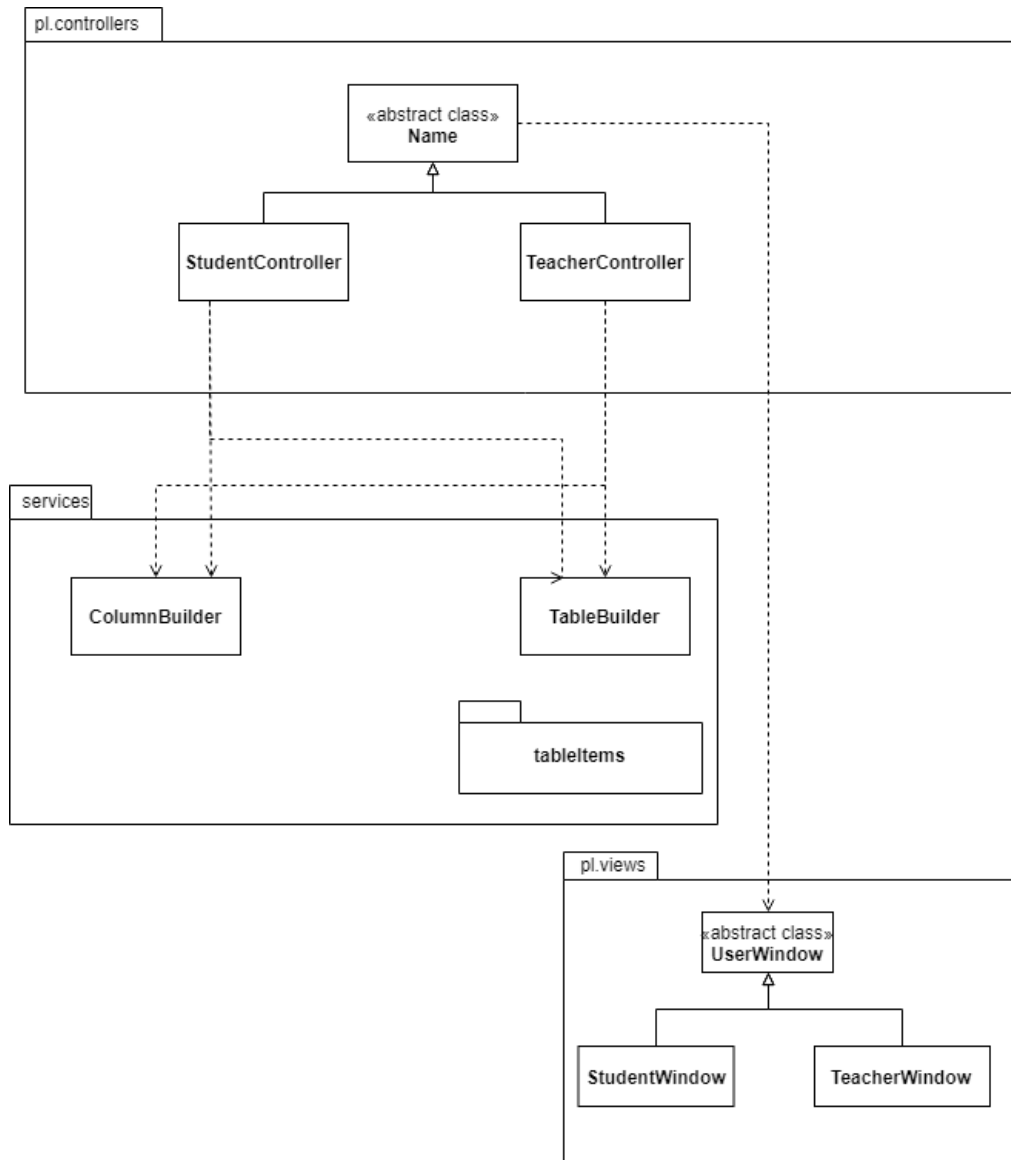
- Singleton – database connection has only one instance
- Design by contract – database access methods validate data before writing it to the database
- Builder pattern – students, tables and table columns have builders associated
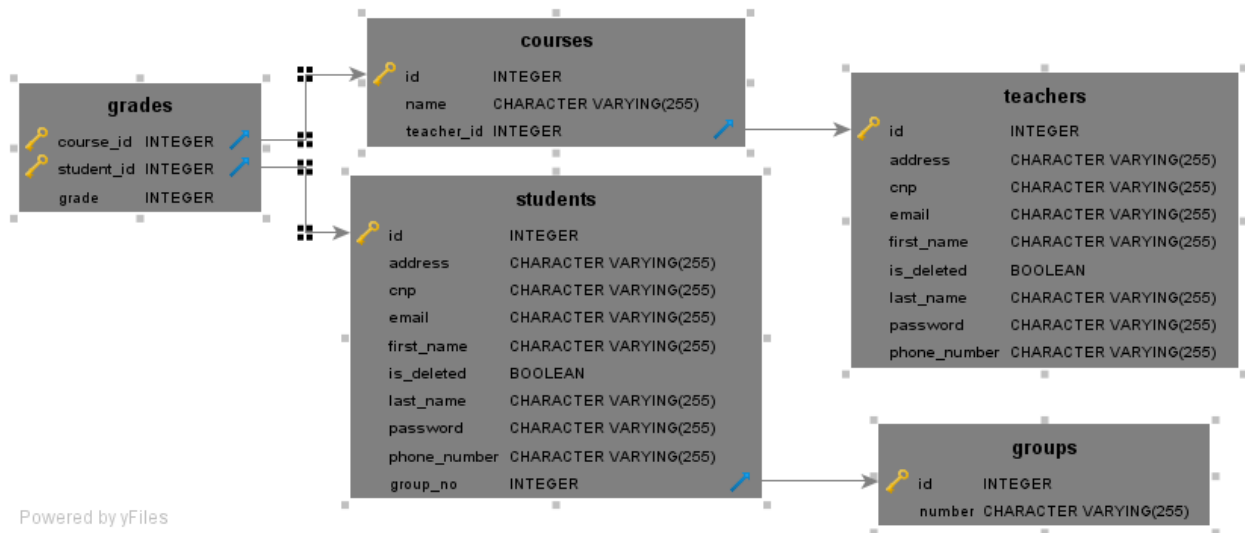- Dependency Injection – repositories and services use Spring injection mechanism

5.2.1 UML Class Diagram – server

5.2.2 UML Class Diagram – client

6. Data Model

grades

🔑	course_id	INTEGER
🔑	student_id	INTEGER
	grade	INTEGER

courses

🔑	id	INTEGER
	name	CHARACTER VARYING(255)
	teacher_id	INTEGER

students

🔑	id	INTEGER
	address	CHARACTER VARYING(255)
	cnp	CHARACTER VARYING(255)
	email	CHARACTER VARYING(255)
	first_name	CHARACTER VARYING(255)
	is_deleted	BOOLEAN
	last_name	CHARACTER VARYING(255)
	password	CHARACTER VARYING(255)
	phone_number	CHARACTER VARYING(255)
	group_no	INTEGER

teachers

🔑	id	INTEGER
	address	CHARACTER VARYING(255)
	cnp	CHARACTER VARYING(255)
	email	CHARACTER VARYING(255)
	first_name	CHARACTER VARYING(255)
	is_deleted	BOOLEAN
	last_name	CHARACTER VARYING(255)
	password	CHARACTER VARYING(255)
	phone_number	CHARACTER VARYING(255)

groups

🔑	id	INTEGER
	number	CHARACTER VARYING(255)

Powered by yFiles

8. Bibliography

MySQL jdbc guide
JavaFX Docs
Stackoverflow