]Project Developed by

P Joy Philip

Project Brief Introduction

Developed a Simple and Straight Forward University **Data Management Application** using Spring Boot along with **Spring Web**, **Spring Data** frameworks linked to **MySQL Database**.

Demonstrated Relational Database on the aspects of OneToOne, ManyToOne, ManyToMany applied on MySQL tables, which are created and/or linked **using Java Classes**

Implemented MVC(Model, View, Controller) Architechture

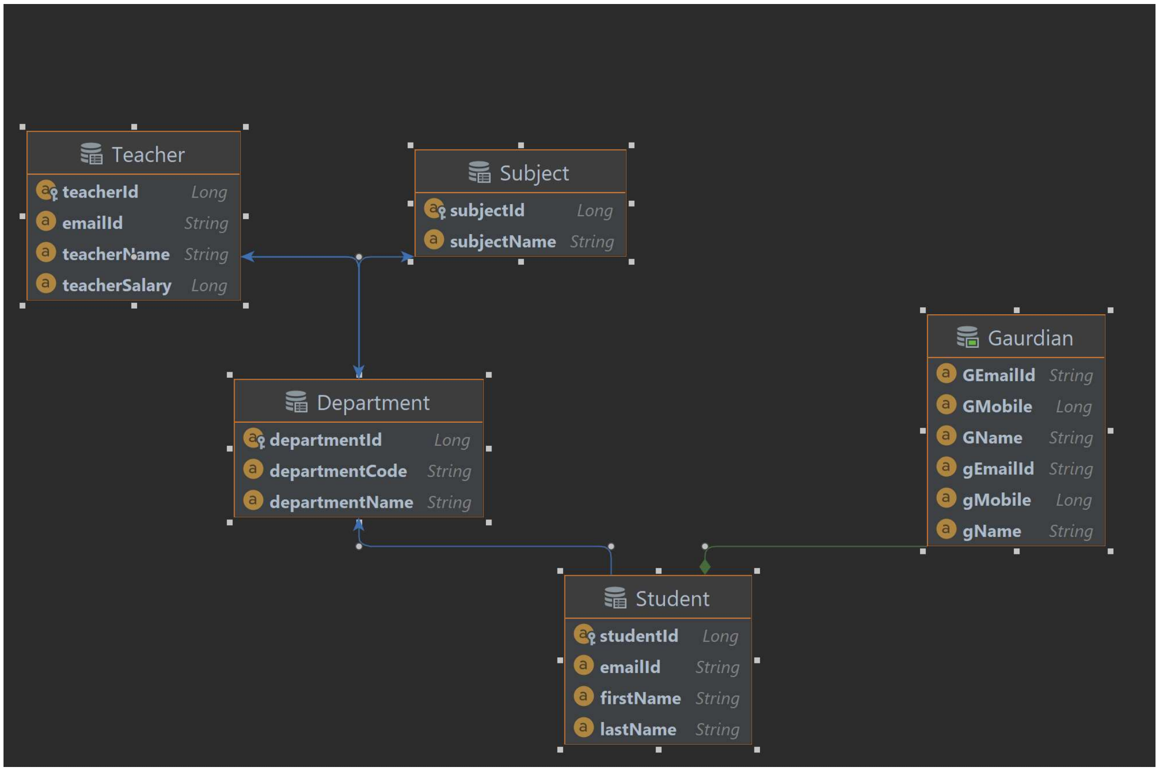
- To ensure isolation between multiple layers
- Easier to debug and maintain the structure of the Application

Handled Various **C.R.U.D.** (Create, Read, Update, Delete) Operations according to the **Relational functions** of Entities in Service Layer

- Usage of Native Queries / JPQL Queries were kept to minimum to maintain simple presentaion of C.R.U.D Operatoins

Brief Explanation of Relationships

Entities (Student, Gaurdian, Department, Teacher, Subject)



Gaurdian

Gaurdian has a OneToOne Relationship with Student without existing as an individual table in the database, In-Depth Relationship between Gaurdian and Student is stated below in the Student Section

Student

Student has a seperate Entity named Gaurdian built into the Same table, Because having two different tables for Each Student and Gaurdian won't make sense to maintain and retrive information (Mapped using **@Embeddable**, **@Embedded**).

Student has a ManyToOne Relationship with Department, which implies there can be multiple Students belonging to a Department (Mapped using **@ManyToOne**),

Student table Contains a Foreign Key pointing to Primary Key of the appropriate department,

There are multiple Departments but a student can only belong to one department at a time

Teacher

Teacher has a ManyToOne Relationship with Department, which implies there can be multiple Teachers belonging to a Department (Mapped using **@ManyToOne**)

Teacher table Contains a Foreign Key pointing to Primary Key of the appropriate department,

There are multiple Teachers belonging to single Department but Single Teacher cannot belong to multiple Departments at a time

Department

Department has OneToOne Relationship with Teacher, Which implies there can be only be one Teacher belonging to a department, this teacher acts as the HOD of a department (Mapped using **@OneToOne**)

Department table contains a Foreign Key pointing to the appropriate teacher

Subject

Subject has a ManyToMany Relationship with Department, which implies there can be one subject belonging to multiple departments and many subjects can belong to single department (Mapped using **@ManyToMany**)

Mapping Table is created which Contains Foriegn Key of Subject and Forieng Key of Department

- Known Issue :- Relationship between Teacher and Department creates a buffer overflow error causing recursion of parent and child class while using Json (Has been Solved through **@JsonManagedReference**, **@JsonBackReference**)

This does not effect the functionality or the performance of the Database, it only effects how data is displayed through Json format, not the way it's retrieved

The retrieval of data has been dealt with by implementing an api to retrive necessary data of teacher and department objects without causing recurrsion

Personal Take on the Project

As new comer to the Spring Framework and it's Applications, i noticed the learning curve impacting the compilation of project, Regarless, Absorbing *Practical Knowledge* on Building Web APIs to manage Database and it's functional flow was pleasurable experience to learn of

P Joy Philip

joyphilip.p2001@gmail.com