**PROJECT REPORT**

On

**STUDENT MANAGEMENT SYSTEM USING REST API**

By

(**Batch: 2022-7469**)

Vaishnavi Jadhav

Neha Sawant

Pooja Chavan

Rupali Khairnar

Sima Jadhav

Under the Supervision of

**Pooja Mehta**



**India's leading Workforce Development Platform**

**Table of Content**

|  |  |  |
| --- | --- | --- |
| SR NO | TOPIC | PAGE NO |
| 1 | Introduction |  |
| 2 | Objective |  |
| 3  3.1  3.2  3.3 | Technologies used  Spring Tool Suite  MySql  Postman |  |
| 4  4.1  4.2  4.3  4.4 | Modules  Entity  Repository  Service  Controller |  |
| 5 | Annotations |  |
| 6  6.1  6.2  6.3 | Diagrams  Class Diagram  UseCase Diagram  ER Diagram |  |
| 7 | Screenshots |  |
| 8 | Conclusion |  |
| 9 | References |  |

1. **Introduction**

The Student Management System can handle all the details about a Student.

The details include Course details, Student personal details ,etc.

The Student Management System is an automated version of manual Student Management System.

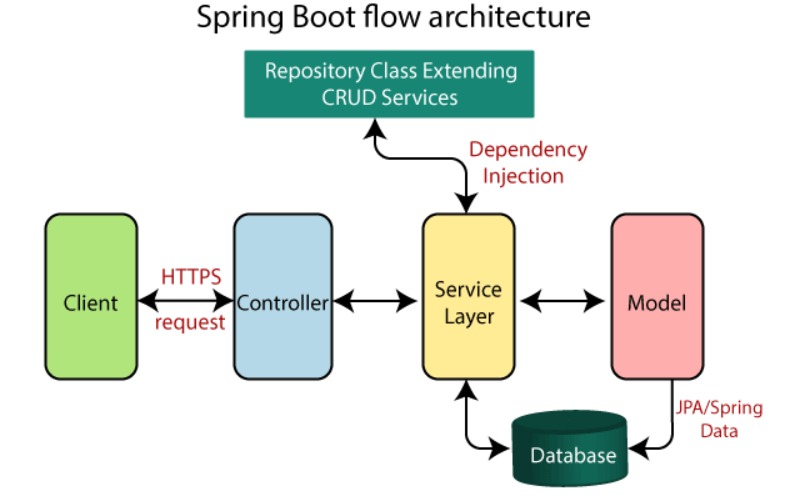
1. **Objective**

It is designed to make the existing manual system automatic with the help of computerised equipment and full-edged computer software, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with.

1. **Technology Used**

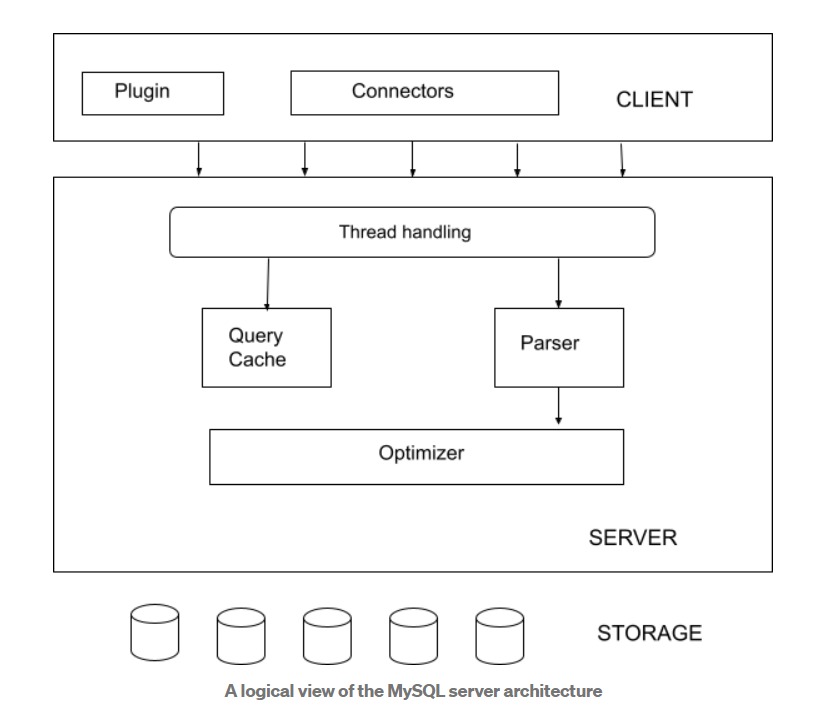
* Spring Tool Suite

Spring Boot is a java framework used for develop standalone application. Need strong knowledge in OOPS & Java concept then only working with spring framework.Mostly all applications are developed by spring boot. Because it was very secure no one hack the information. Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications.



* MySQL

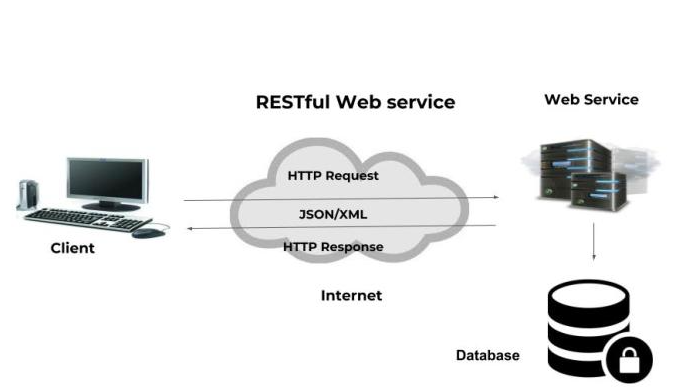
MySQL creates a database for storing and manipulating data, defining the relationship of each table. Clients can make requests by typing specific SQL statements on MySQL. The server application will respond with the requested information and it will appear on the clients' side. MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. MySQL is ideal for both small and large applications.



* Postman

Postman is an API client that makes it easy for developers to create,share, test and document APIs. With this open-source solution, users can create and save simple and complex HTTP/s requests, as well as read their responses. When you are signed into your account, you are able to access your files. You can execute Postman API tests anytime, anywhere.

Postman is very convenient when it comes to executing APIs. Once you’ve entered and saved them, you can simply use them over and over again, without having to remember the exact endpoint, headers, or API keys.



1. **Module Used**

* Entity

The entities are the persistence objects stores as a record in the database. Persistence Unit: It defines a set of all entity classes. In an application, EntityManager instances manage it. The set of entity classes represents the data contained within a single data store.

* Repository

Spring boot framework provides us repository which is responsible to perform various operations on the object.Repository classes are auto detected by spring framework through classpath scanning.

* Service

Spring boot service component is defined as a class file that includes the @Service annotation and allows developers to add business functionali

-ties. The annotation is used with the classes that provide these business functionalities.

* Controller

In Spring Boot, the controller class is responsible for processing incoming REST API requests, preparing a model, and returning the view to be rendered as a response. The controller classes in Spring are annotated either by the @Controller or the @RestController annotation.

**5.Annotation**

1.@Autowired:

Spring provides annotation-based auto-wiring by providing @Autowired annotation. It is used to autowire spring bean on setter methods, instance variable, and constructor. When we use @Autowired annotation, the spring container auto-wires the bean by matching data-type.

2. @RequestMapping:

It is used to map the web requests. It has many optional elements like consumes, header, method, name, params, path, produces, and value.

3.@RestController:

It is a convenience annotation for creating Restful controllers. It is a specialization of @Component and is autodetected through classpath scanning.

4. @GetMapping:

It maps the HTTP GET requests on the specific handler method. It is used to create a web service endpoint that fetches It is used instead of using: @RequestMapping(method = RequestMethod.GET)

5.@PostMapping:

It maps the HTTP POST requests on the specific handler method. It is used to create a web service endpoint that creates It is used instead of using: @RequestMapping(method = RequestMethod.POST)

6.@PutMapping:

It maps the HTTP PUT requests on the specific handler method. It is used to create a web service endpoint that creates or updates It is used instead of using: @RequestMapping(method = RequestMethod.PUT)

7.@DeleteMapping:

It maps the HTTP DELETE requests on the specific handler method. It is used to create a web service endpoint that deletes a resource. It is used instead of using:@RequestMapping(method =RequestMethod.DELETE)

8.@PatchMapping:

It maps the HTTP PATCH requests on the specific handler method. It is used instead of using: @RequestMapping(method = RequestMethod.PATCH)

9. @NotNull:

It is actually, an explicit contract declaring that: A method should not return null. Variables (fields, local variables, and parameters) cannot hold a null value.

10. @SpringBootApplication:

This annotation is used to mark a configuration class that declares one or more @Bean methods and also triggers auto-configuration and component scanning.

11.@Id:

This annotation is inherited from javax.persistence.Id, indicating the member field below is the primary key of the current entity.

12.@Generated:

The Generated annotation is used to mark source code that has been generated.

14. @ManyToOne:

This mapping means that many instances of this entity are mapped to one instance of another entity – many items in one cart.

15.@ GeneratedValue:

This annotation specifies the generation strategies for the values of primary keys

16.@ Column:

This annotation is used for Adding the column the name in the table of a particular MySQL database.

17. @JoinTable:

This Specifies the mapping of associations. It is applied to the owning side of an association.

18.@Service:

It is also used at class level. It tells the Spring that class contains the business logic.

19. @Entity:

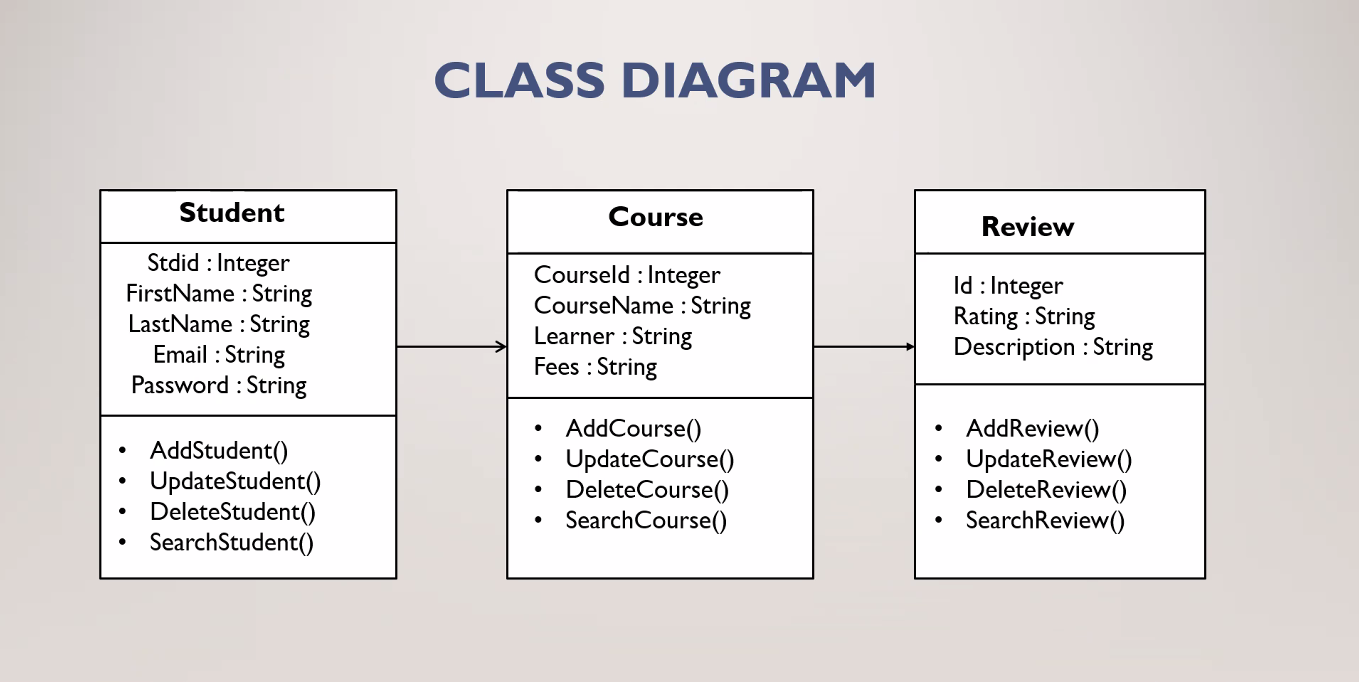
This annotation defines that a class can be mapped to a table. And that is it, it is just a marker, like for example Serializable interface.

20.@JoinColumn:

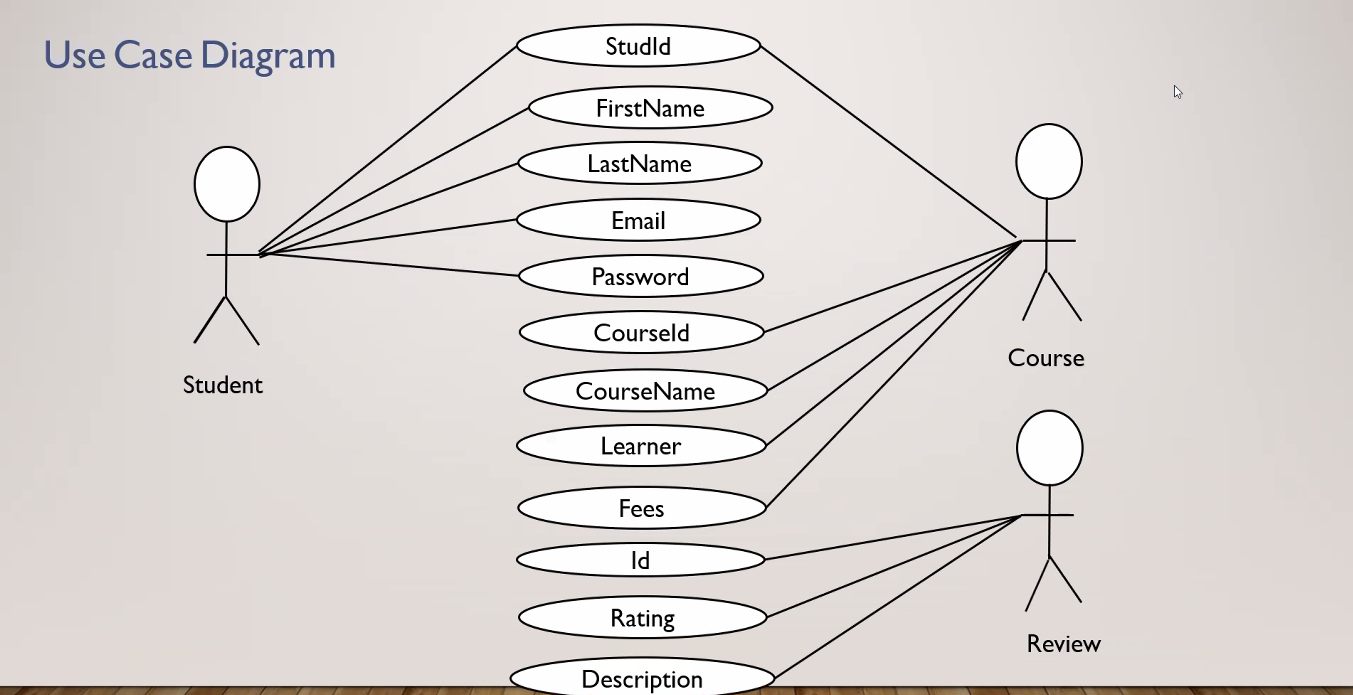
Specifies a column for joining an entity association or element collection.

**6.Diagrams**

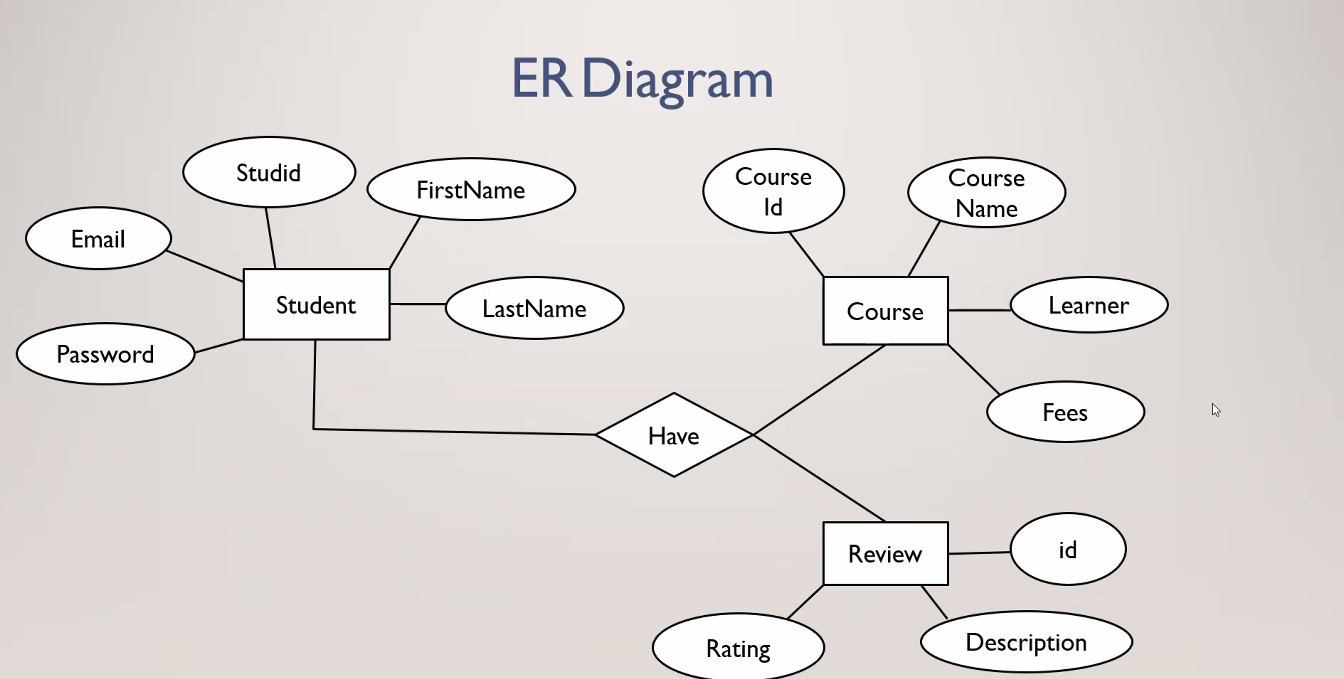
* **Class Diagram**

****

* **Use Case Diagram**

****

* **ER Diagram**

****

**7.Screenshots**

