

# JPA Entity Classes And its Class Diagram

## User

Stores the data of users.

```
@Entity
@Table(name = "users")
public class User {
    @Id
    @Column
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long users_id;
    @Column
    @NotBlank(message = "Name is mandatory")
    private String name;
    @Column
    @NotBlank(message = "Email is mandatory")
    private String email;
    @Column(name = "imageUrl")
    private String imageUrl;
    @Column(name = "accountType")
    private String accountType ;
    @Column
    private String phone;
    @Column
    private String password;
    @Column
    private String user_role;
```

## Course

Stores the data of Course.

```

@Entity
public class Course {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long courses_id;
    private String course_name;
    @Column(name = "course_key")
    private String courseKey;
    private String course_email;
    private String course_cover_image_url;
    private String course_description;
    private int course_capacity;
    @ManyToOne()
    @JoinColumn(name = "instructor_id")
    private User instructor;

    @ManyToMany
    @JoinTable(
        name = "student_course_enrolment",
        joinColumns = @JoinColumn(name = "course_id"),
        inverseJoinColumns = @JoinColumn(name = "student_id"))
    Set<User> enroledStudents;

```

## Assignment-Module

This Consist of Student Assignment , Assignment Submission and Submission files.

```

@Entity
public class AssignmentSubmissons {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    @ManyToOne()
    @JoinColumn(name = "assignment_id")
    private Assignment assignment;
    @ManyToOne()
    @JoinColumn(name = "student_id")
    private User student;
    Date dateOfSubmission;
    Date dateOfEvaluation;
    @Column(length=10485760)
    String details;
    long obtainedPoints;

    @OneToMany(
        cascade = CascadeType.ALL,
        orphanRemoval = true
    )
    private List<AssignmentSubmissonFiles> files ;
    public List<AssignmentSubmissonFiles> getFiles() { return files; }
}

```

```

@Entity
public class Assignment{
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    @ManyToOne()
    @JoinColumn(name = "course_id")
    private Course course;
    Date dateOpened;
    Date submissionDeadline;
    @Column(length=10485760)
    private String assignmentDescription;
    private String tittle;
    private long maxPoints;
    private String assignmentFileUrl;
}

```

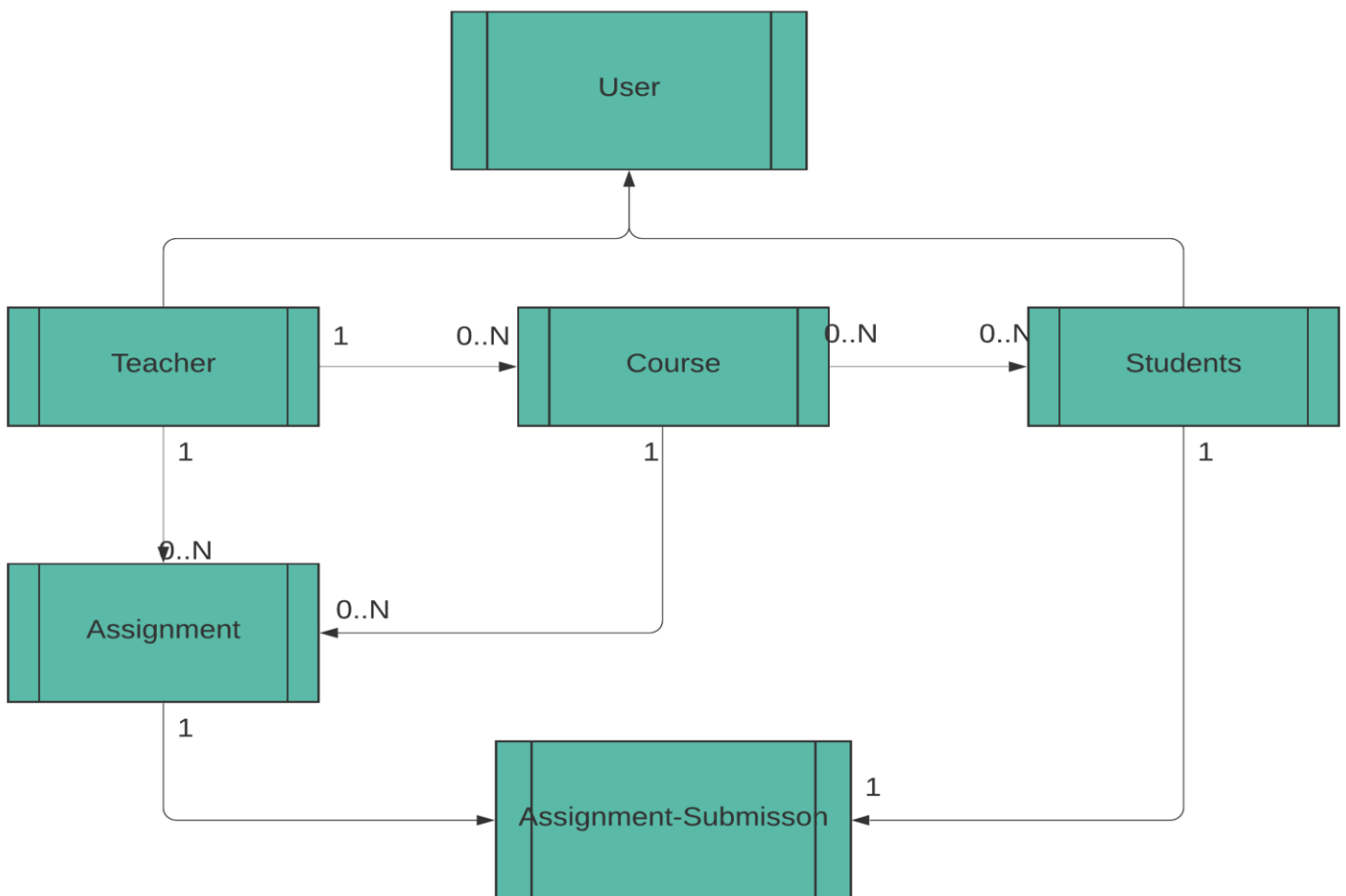
```

@Entity
public class AssignmentSubmissionFiles {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;

    @ManyToOne(cascade=CascadeType.ALL)
    @JoinColumn(name = "assignment_submission_id")
    private AssignmentSubmissons assignmentSubmission;
    private String fileUrl;

    public long getId() { return id; }
}

```

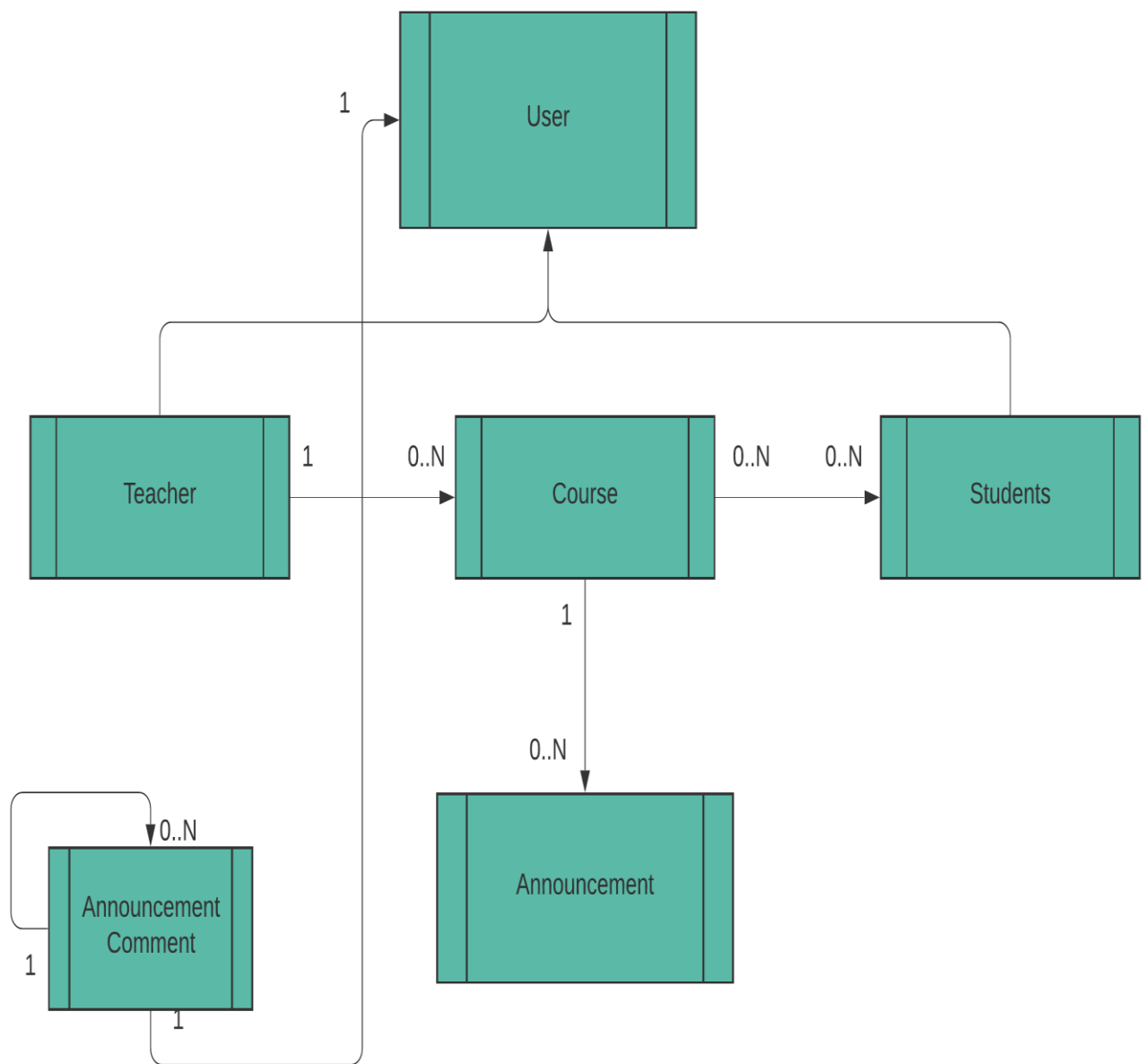


## Announcement-Module

This Consist of Announcement, Announcement Comment and Submission files.

```
@Entity
public class Announcements {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private
    long id;
    private Date dateMadeOn;
    @ManyToOne()
    @JoinColumn(name = "course_id")
    private Course course;
    private String title;
    @Column(length=10485760)
    private String details;
```

```
@Entity
public class AnnouncementComments {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    @ManyToOne()
    @JoinColumn(name = "announcement_id")
    private Announcements announcement;
    @ManyToOne()
    @JoinColumn(name = "commenter_id")
    private User commenter;
    @Column(length=10485760)
    private String comment;
```



## Doubt-Question-Answers-Module

This Consist of StudentQuestion Answer, Answer-comment

```

@Entity
public class StudentQuestion {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    private Date dateAskedOn;
    @ManyToOne()
    @JoinColumn(name = "course_id")
    private Course course;
    @ManyToOne()
    @JoinColumn(name = "student_id")
    private User student;
    private String question;

    private String image_url;
}

```

```

@Entity
public class StudentQuestionAnswers {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    private Date dateAnsweredOn;
    @ManyToOne()
    @JoinColumn(name = "question_id")
    private StudentQuestion studentQuestion;
    private String answer;
    private String image_url;

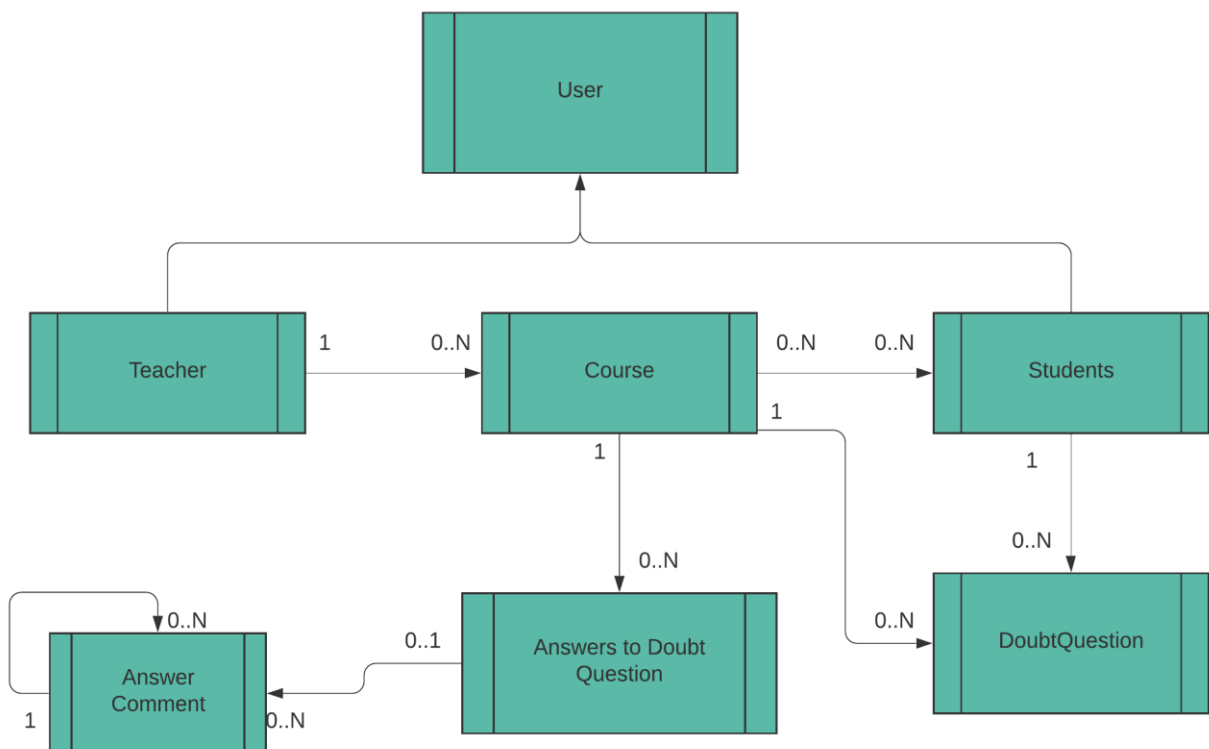
    public long getId() {
        return id;
    }
}

```

```

@Entity
public class AnswerComment {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    @ManyToOne()
    @JoinColumn(name = "answer_id")
    private StudentQuestionAnswers studentQuestionAnswers;
    @ManyToOne()
    @JoinColumn(name = "commenter_id")
    private User commenter;
    @Column(length=10485760)
    private String comment;
    @ManyToOne()
    @JoinColumn(name = "parent_id")
    AnswerComment parentComment;
}

```





# Lecture-Feedback-Module

This consist of lecture , isFeedBackRegistered StudentFeedBack

```
@Entity
public class Lecture {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    private Date lectureDate;
    private Time StartTime;
    private Time EndTime;
    private String LectureTitle;
    @Column(length=10485760)
    private String LectureDescription;
```

```
@Entity
public class StudentFeedBack {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private long id;
    @ManyToOne()
    @JoinColumn(name = "lecture_id")
    private Lecture lecture;
    @ManyToOne()
    @JoinColumn(name = "student_id")
    private User student;
    private short ratings;
    private String comment;
```

@Entity

```
public class isFeedBackRegistered {  
    @Id  
    @GeneratedValue(strategy = GenerationType.SEQUENCE)  
    private long id;  
    @ManyToOne()  
    @JoinColumn(name = "student_id")  
    private User student;  
    @ManyToOne()  
    @JoinColumn(name = "lecture_id")  
    private Lecture lecture;  
}
```

