

Mini Project Document

Class Diagram

Team Id - A 10

Ankitha. N - PES1UG20CS059

Aishwarya.N- PES1UG20CS024

Class Name:

1. Register class

Sl . no	Attribute name Data type Access Prototype	Default Value
1	Name string Public (+)	null
2	password string Private (-)	null

2. Admin class

Sl no	Attribute name	Data type Access Prototype	Default Value
1	Admin name	String Public (+)	null
2	password string	Private (-)	null

Methods:

- ★ login(): ; Void ; – This is used to log into the account.
- ★ logout(): ; Void; – This is used to log out of the account.
- ★ Settings(): ; void; – this is used to manage the settings of the account.

3.Student class

Sl no	Attribute name	Data type Access	Default Value Prototype
1	Student name	String Public (+) null	null
2	password string	Private (-)	null
4	Take quiz	Unsigned int Public (+) array	null

Methods:

- ★ takeQuiz(): ; void – This allows to take the quiz.
- ★ reviewQuiz(): ; void – This allows to review the quiz.

4.Question Bank:

Sl no	Attribute name	Data type Access	Default Value Prototype
1	Questions	Unsigned int Private (-)	0

5.Quiz class

Sl no	Attribute name	Data type Access Prototype	Default Value
1	Quiz name	String Private (-) null	
4	Quiz type	Boolean Private (-) False	

6.Question class

Sl no	Attribute name	Data type Access Prototype	Default Value
1	Question	String Public (+) null	
3	Options	String array Public (+) null	
4	Answer	Unsigned int Private (-) 0	

7.Review class

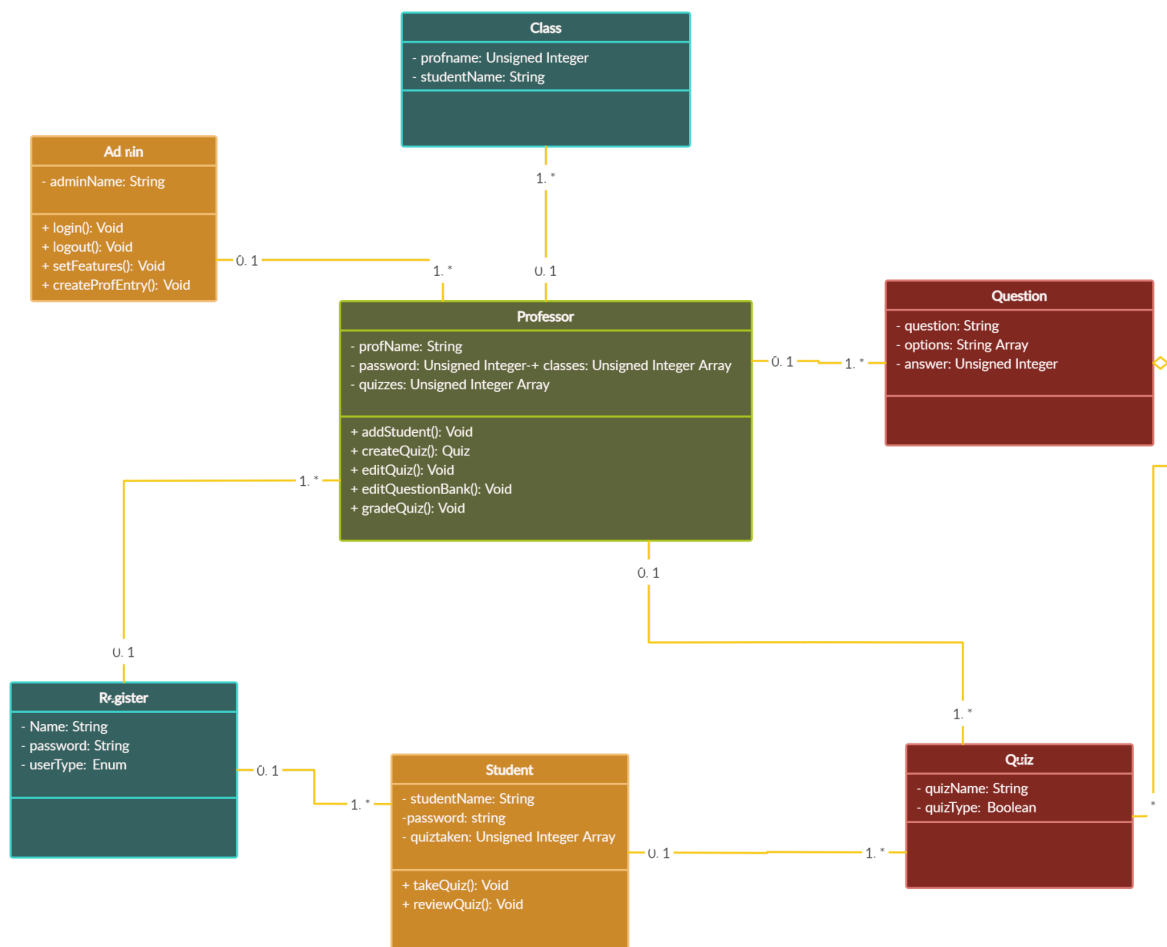
Sl no	Attribute name	Data type Access Prototype	Default Value
1	Score	Unsigned int Public (+) 0	
2	Answer	Unsigned int Private (-) 0	

8.Professor/Quiz handler class

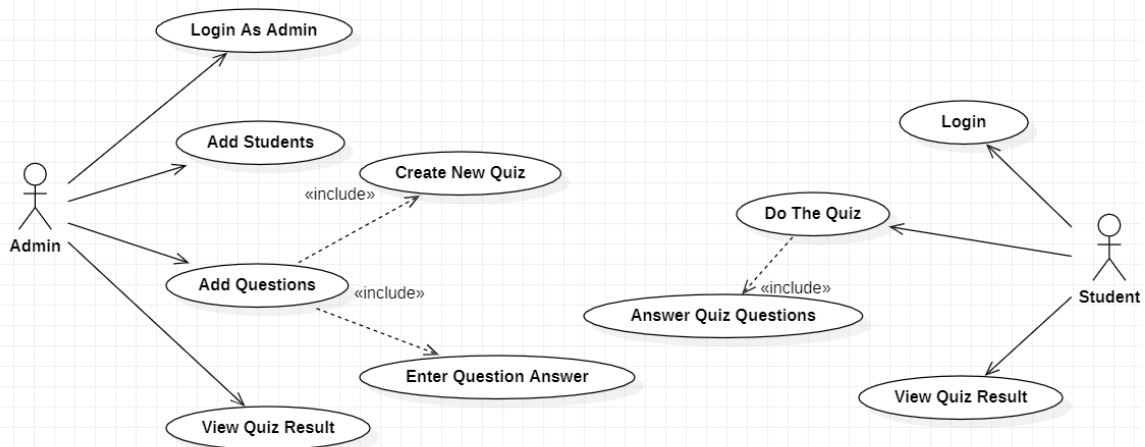
Sl no	Attribute name	Data type Access Prototype	Default Value
1	Professor name	Unsigned int Private (-) 0	
2	Password	Unsigned int Private (-) 0	
3	Add quiz	Unsigned int Private (-) 0	

Methods:

- ★ addStudent(): void – this can add a student into the course.
- ★ createQuiz(): void – this can create a quiz.
- ★ editQuiz(): void – this can allow editing of a created quiz
- ★ gradeQuiz(): void – this can allow to grade a quiz.



USE-CASE DIAGRAM:



DESIGN PATTERNS USED ARE:

1.REPOSITORY:

It provides 2 main benefits: The pattern abstracts the data store and enables you to replace your data store without changing your business code.

2.A Decorator Pattern says that just "**attach a flexible additional responsibilities to an object dynamically**".

In other words, The Decorator Pattern uses composition instead of inheritance to extend the functionality of an object at runtime.

The Decorator Pattern is also known as **Wrapper**.