RAMAIAH UNIVERSITY
OF APPLIED SCIENCES

Scenario

Discuss with the Course leader and identify the project scenario. In this regard students are required to develop DIK model.

PART A — 10 Marks

A.1 Write an abstract statement

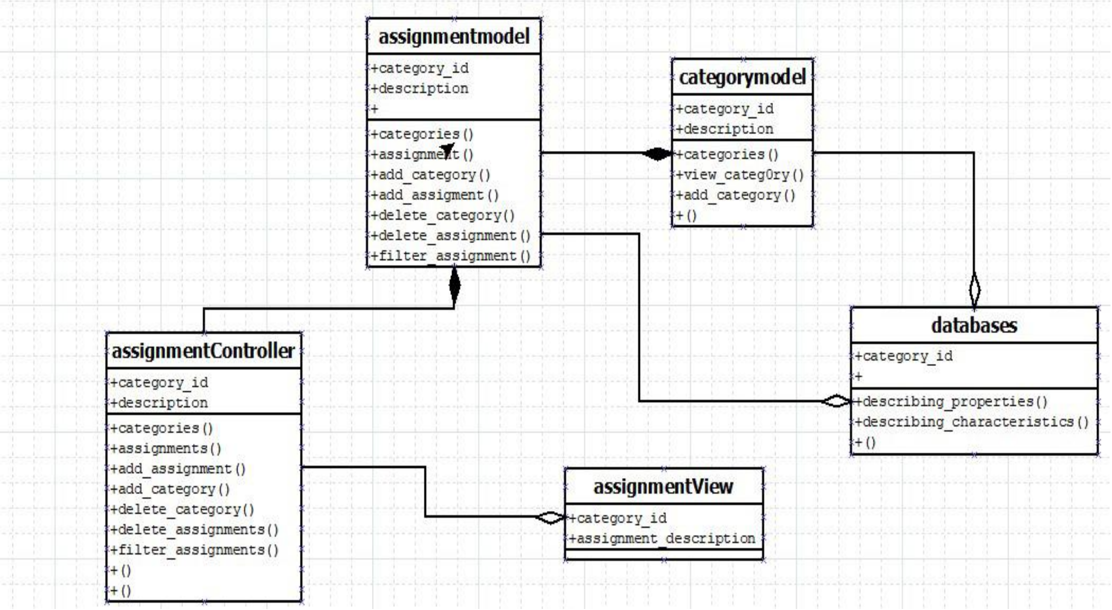
The user opens applications ⇒ category of the assignment ⇒ add assignment for that category ⇒ view the assignment in the selected category ⇒ closes the application

A.2 Do a detail walk through identifying all the transactions and process that happen in each of the transaction?

1. The user opens the application, and in the opening page, he can see all the assignments listed for the selected category and an option to add assignment, manage category
2. The user clicks on the manage category, where he can perform the functions like, view add and deleting the categories
3. The user adds one or more categories
4. The user views the added categories
5. The user if wanted delete the category, where on deleting the category no, assignments are added to that category
6. The user redirect to the home page to manage the assignments
7. The user selects the category for which the assignments should be added
8. The user enters the description for the assignment
9. The user clicks on the add button
10. Then the user selects the category in which the assignments need to be displayed
11. The user views the assignments in the category selected
12. The user can delete the assignment from the list of assignments, by clicking on the cross button.
13. The user closes the application

PART B — 15 Marks

B1.1 Draw a Class Diagram depicting the system.



In this class diagram, the list of classes are assignment model, category model, database, assignment controller, assignment view.

1. The assignment model has a composition relationship with the category model class. This means that the assignment model class owns an instance of category model class and is responsible for creating, using, and destroying it.
2. Assignment controller has a composition relationship with assignment model and category model class. this means that the assignment controller class owns instances of the assignment model and category model class and is responsible for creating, using and destroying them.
3. Assignment controller has an aggregation relationship with the assignment view class. This means that the assignment controller class uses an instance of the assignment view class to display the data and interacts with it to receive user input.
4. Assignment model has an aggregation relation with the database class. This means that the assignment model class uses an instance of the database class to communicate with the database.
5. Category model has an aggregation relation with the database class which means that the category model class uses an instance of the database class to communicate with the database.
6. Assignment view has no relation with other classes in the class diagram, but it interacts with the assignment controller class to receive and display data, and to send user input back to to the controller.

B1.2 Discuss the architectural style and design pattern used to implement the design. Justify

The Model-View-Controller (MVC) is an architectural design pattern that separates an application into three interconnected components: the Model, the View, and the Controller. This pattern provides a clear separation of concerns and helps developers to manage complex applications in a more organized manner.

The Model represents the application's data and defines the logic for manipulating that data. It responds to requests from the View and sends updates to the Controller when the data changes. In PHP, the Model is typically implemented as a set of classes that interact with the database or other data sources.

The View is responsible for rendering the data from the Model to the user interface. It receives requests from the Controller and updates the user interface accordingly. In PHP, the View is typically implemented as a set of templates or pages that display the data to the user.

The Controller handles user input and communicates with the Model and View to perform the necessary actions. It receives requests from the user interface and updates the Model or View accordingly. In PHP, the Controller is typically implemented as a set of scripts that handle user input and interact with the Model and View.

MVC is a popular design pattern in PHP because it allows for a clear separation of concerns and promotes modular code development. It is also widely used in web development because it enables developers to create dynamic web applications that are scalable, maintainable, and easy to extend.

Some of the benefits of using the MVC pattern in PHP include:

1. Improved code organization: The separation of concerns makes it easier to maintain and extend the codebase.
2. Code reusability: The modular nature of the pattern allows for code reuse across different parts of the application.
3. Scalability: MVC-based applications are easier to scale because the components are decoupled and can be modified independently.
4. Testability: The separation of concerns makes it easier to write unit tests for each component of the application.
5. Enhanced user experience: MVC promotes the use of templates, which enables developers to create responsive and user-friendly web applications.

In conclusion, the MVC pattern is a powerful design pattern that can help developers to create scalable and maintainable PHP applications. By separating the application into three components - Model, View, and Controller - developers can improve code organization, promote code reusability, and enhance the user experience.

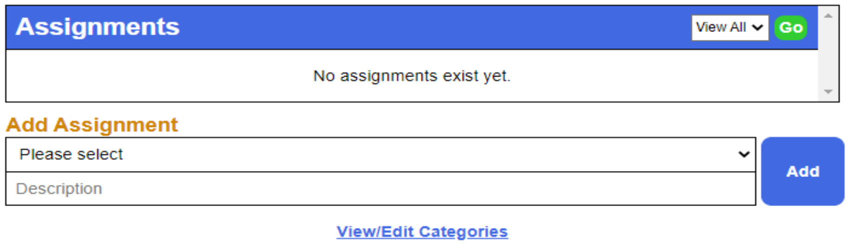
Justification

The Assignment Tracking application follows the MVC (Model-View-Controller) architectural style and design pattern because it separates the application's concerns into distinct components, promoting modularity, reusability, and maintainability.

In the MVC pattern, the Model component manages data and logic, the View component manages the user interface, and the Controller component manages the flow of information between the Model and View. This separation allows each component to be developed and tested independently of the others, which simplifies development and promotes code reuse.

In the context of the Assignment Tracking application, the Model component could represent the data storage and management, such as the database tables for assignments and categories. The View component could represent the user interface, such as the HTML templates and CSS styles. The Controller component could represent the logic that retrieves data from the Model and passes it to the View for display.

Results of the design:

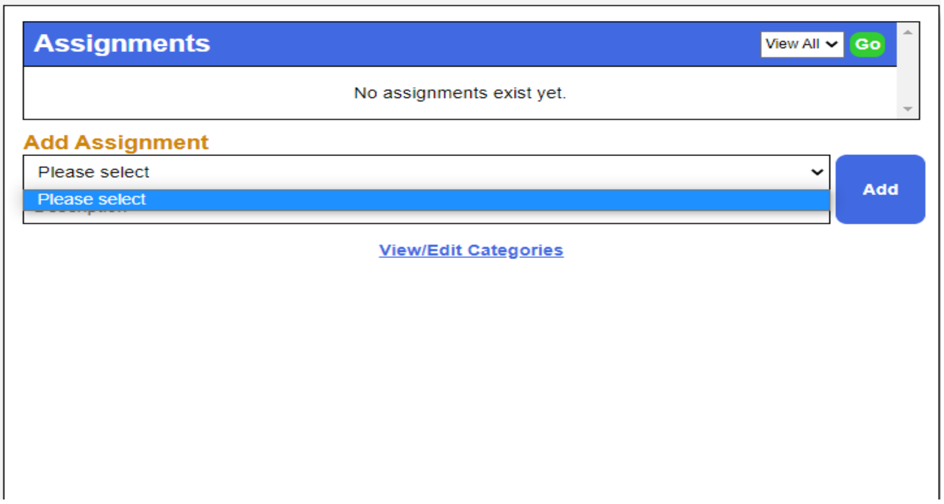


No assignments exist yet.

Please select

Description

Figure 1.1. page showing assignment

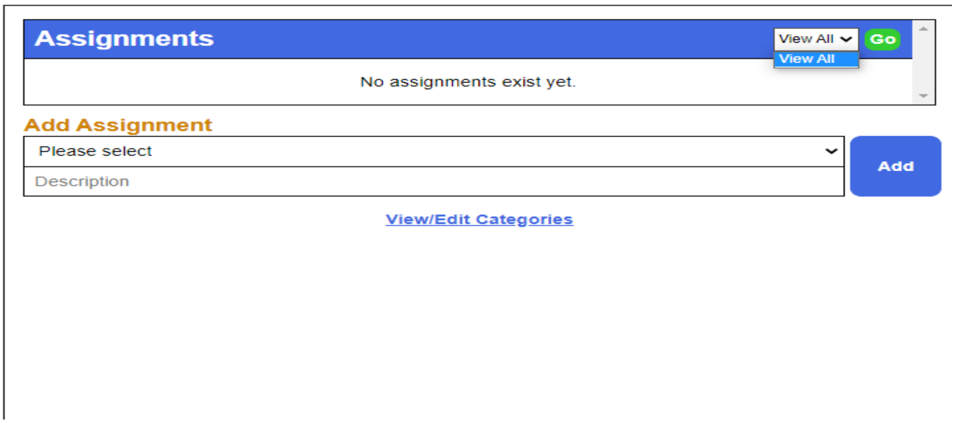


No assignments exist yet.

Please select

Please select

Figure 1.2. you can select the assignment which need to be added.

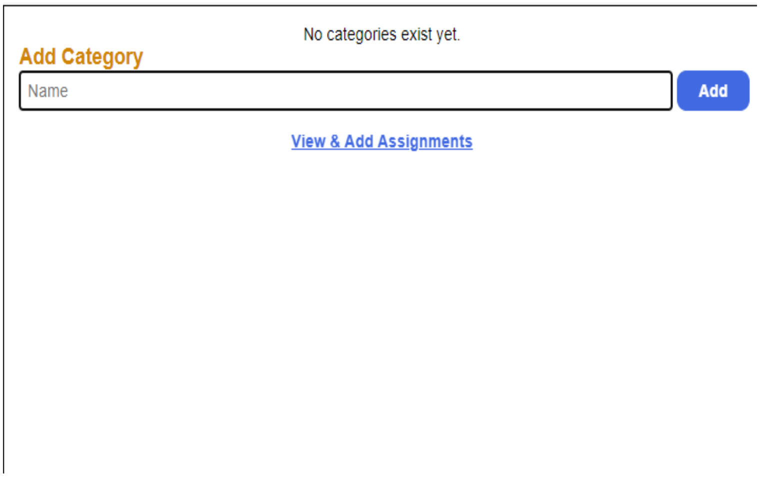


No assignments exist yet.

Please select

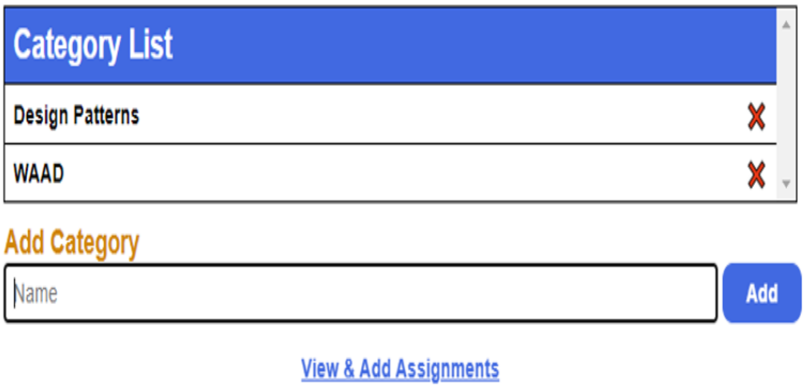
Description

Figure 1.3. page showing the categories of assignment. In which all course are included.



No categories exist yet.

Figure 1.4 you can add categories.

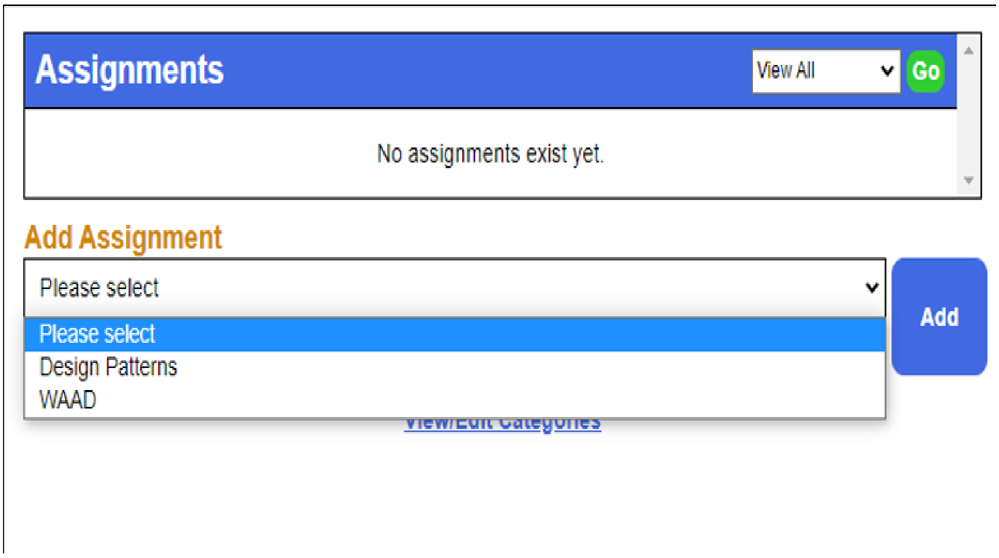


Category List

Design Patterns

WAAD

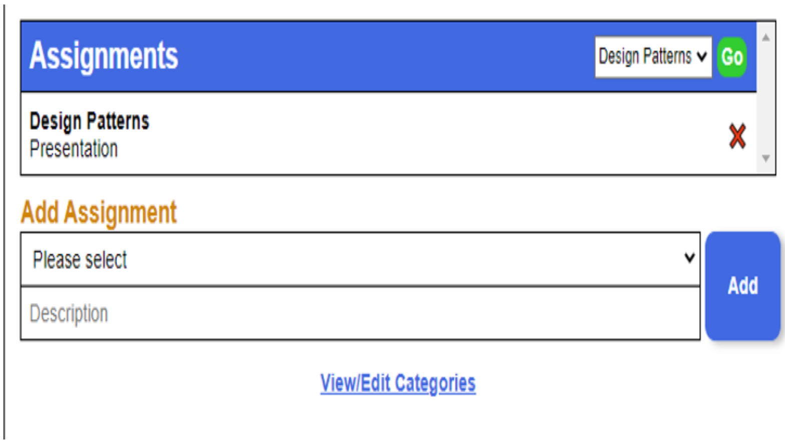
Figure 1.5. from the given categories list you can add categories.



Assignments

No assignments exist yet.

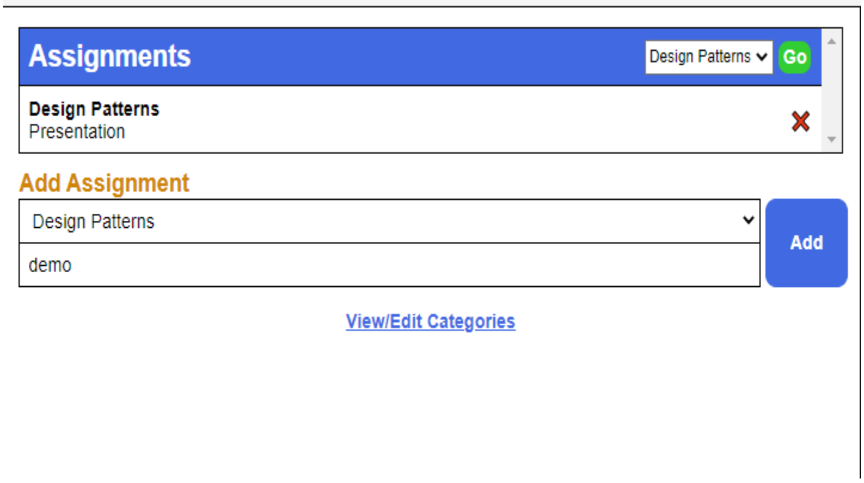
Figure 1.6. from the categories we can add assignment that need to be added



Design Patterns
Presentation

Please select

Description

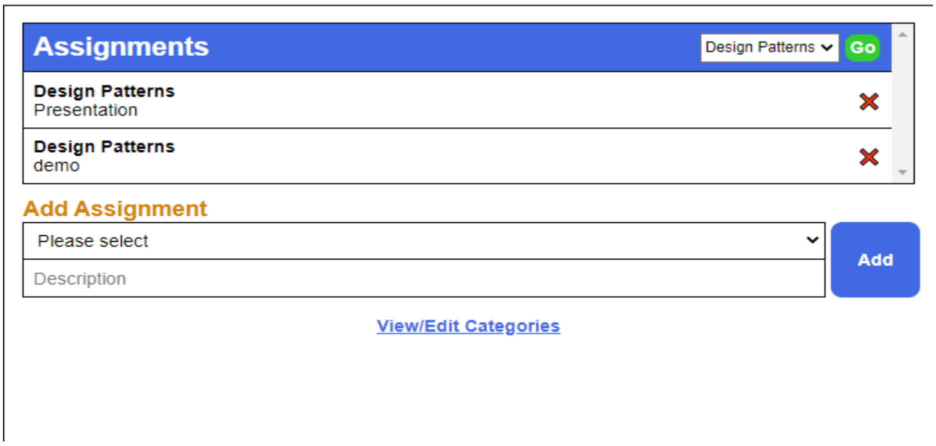


Design Patterns
Presentation

Design Patterns

demo

Figure 1.7. in description we can add demo



Design Patterns
Presentation

Design Patterns
demo

Please select

Description

Figure 1.8 after adding assignments we can see at top.

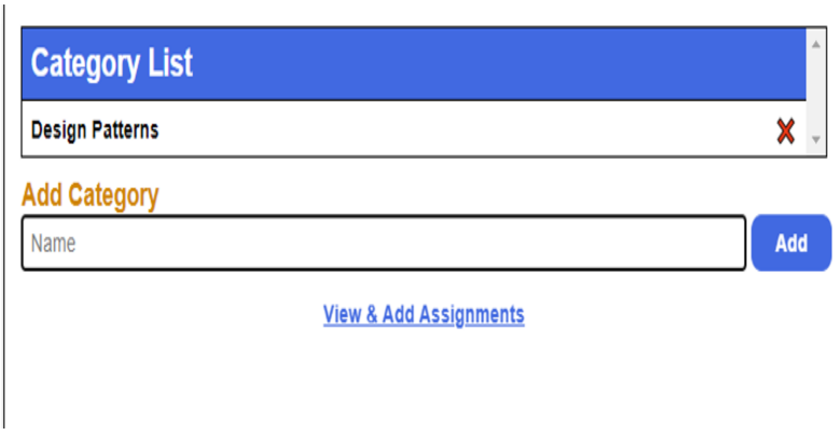


Figure 1.9. we can add new category and proceed.

B1.3 Discuss salient quality parameters that is addressed. Justify your design.

The following are the salient features of the design.

When designing an assignment tracking application, several quality parameters need to be addressed to ensure that the application meets the user's requirements and expectations. These parameters include:

1. Usability: The application should be easy to use and navigate, even for users with minimal technical skills. The user interface should be intuitive, and user input should be minimal.
2. Reliability: The application should function reliably, with a low rate of errors or failures. Any errors should be handled gracefully, without compromising the integrity of the data.
3. Security: The application should be secure, protecting users' personal information and preventing unauthorized access. This includes ensuring that data is stored securely and that access controls are implemented.
4. Performance: The application should be fast and responsive, with minimal loading times and efficient data processing.
5. Maintainability: The application should be easy to maintain and update, with clean and well-organized code. This will help ensure that any bugs or issues can be quickly identified and resolved.

By using the MVC architectural style and design pattern, the assignment tracking application can address these quality parameters effectively. The separation of concerns provided by MVC makes the application easier to maintain and update, while the use of reusable code and components increases reliability and reduces the likelihood of errors. The modular structure also makes the application more scalable, allowing for additional features and functionality to be added as needed. Finally, the focus on user input and feedback in the design process helps ensure that the application is easy to use and navigate, improving overall usability.

Note: Make appropriate assumptions to make the specification complete.

****