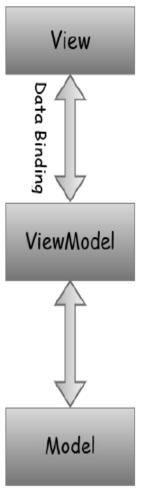
## **Design patterns**

The design pattern that selected to our project is **MVVN** in the **presentation layer**.

The **Model**, **View**, **ViewModel** (**MVVM** pattern) is all about organize and structure our code for maintainable, testable, and extensible applications.



**View** – It simply holds the formatted data and essentially delegates everything to the Model.

In our project we put all the relevant windows in the view layer (what the user sees) – log in window, board's view, specific board view, and windows for adding task, board, column, and to join a new board. In addition, a window of the in-progress tasks of the user. – those windows are the "PL".

In addition, we used **data binding** to connect between the presented view of the user to the entities at the viewModel. We used **Obseverable Collection** to maintain connectivity with the user actions on the site to the ServiceLayer and vice versa.

**ViewModel** – It acts as the link/connection between the Model and View and makes stuff look pretty.

In our project we put all the code "BI" of the view. It means that all windows we have in the view we have a matching class in the viewmodel that making actions on the objects at the Model.

**Model** – It simply holds the data and has nothing to do with any of the business logic.

In our project we have all the entity from the business layer and make the connection between the objects of service layer and PL layer