

Explanation:

MVVM (Model View ViewModel) is an architectural pattern based on MVC and MVP, which attempts to more clearly separate the development of user interfaces (UI) from that of the business logic and behavior in an application. To this end, many implementations of this pattern make use of declarative data bindings to allow a separation of work on Views from other layers.

This facilitates UI and development work occurring almost simultaneously within the same code base. UI developers write bindings to the ViewModel within their document markup (HTML), where the Model and ViewModel are maintained by developers working on the logic for the application.

Models hold information, but typically don't handle behavior.

A passive View only outputs a display and does not accept any user input.

The ViewModel can be considered a specialized Controller that acts as a data converter. It changes Model information into View information, passing commands from the View to the Model.