

Design Patterns

MVVM Pattern:

We used the MVVM pattern in the Presentation layer, each window's data is bound with a view model, which the window has a reference to.

Every service that the window supplies is delegated to the view model where the necessary methods are called, every view model contains a reference to one or more models, which are the representation of an object from the Business layer and contain the logic that addresses the Service layer.

Observer Pattern:

We used Observer pattern in the Presentation layer as well, every view model and model is a notifiable object, meaning that they will send a notification about a property in their class being modified, we implemented this by notifying the window when a property in the view model has changed, so that it will be updated in the window as well.

We also used Observable collections of models which represents a dynamic data collection that provides notifications when items get added, removed, or when the whole list is refreshed.

In our implementation we used Observable collections to advance tasks if they are removed from the column, and to join or delete a board if it is added or removed.