

Question 2: Reasons for pattern choices

On this question, we have used composite, decorator, proxy and façade pattern. And architecturally we have used MVC architecture. Since the view of the MVC structure is automatically generated using drag and drop capabilities of the Visual Studio IDE, we have put our attention to the Model and Controller part.

On the controller we used façade and composite pattern.

- Façade – The controller provides an interface for the view to interact with the model easily
- Composite – The controller contains instance of the view proxy object that controls access to the ProductData object.

On the model we used composite, decorator and proxy pattern.

- Composite – The Product object contains a list of Sale object. Then we have a ProductData class which contains the list of Product making another composite object.
- Decorator - The ProductData object has three more objects that will modify it at runtime: FilterProductSale, FilterSignificantProduct and SortProduct.
- Proxy – The CustViewProxy and ExecViewProxy control the users' access to the Customer and Executive view.