In this project, we have used several OOP patterns that would help us keep development clean and maintain the system easier.

The first one is Builder. As we create our Response, we gradually build the instance of the class. It is much easier to do that than just using messy constructors with dozens of arguments.

Another pattern we've made use of is Proxy. If one wants to get some information about a particular company, one can form a request or Query, which then will create a Response to the user. Hence, speaking with the user is organized via the interface.
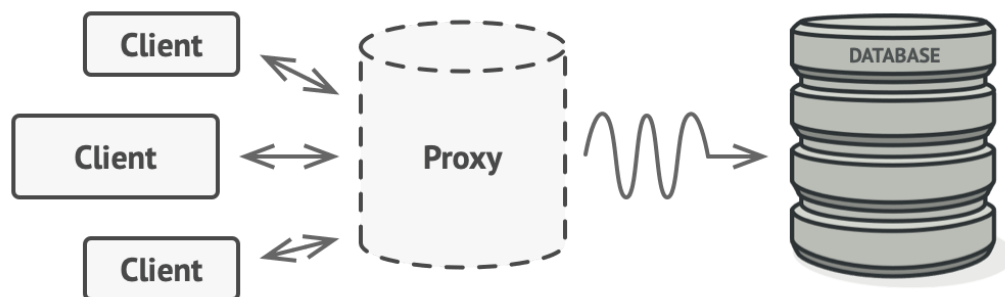
An Error code is an example of a simple FSM (Finite State Machine) with OK and Error States. Our Application might then vary its contents based on the state of the request process.

The chosen architecture lies under a famous MVC (Model-View-Controller) paradigm, which allows for easy application expansion and layer decomposition..