you can also say regarding design patterns that Spring MVC makes use of the Observer Pattern to implement request handling

also see in the view layer we use lots of APIs. NeoVis for graphs, googleCharts for charts, Bootstrap and Google fonts for css and Thymeleaf for templating

Design Patterns

MVC
Our project was initialised using the spring boot mvc pattern.
Thymeleaf html templates are our views. They receive data from the controller layer, and send requests to it for different pages.

Our controllers are marked by the annotation @Controller
Each page has one controller to separate responsibility. Each controller has a request fulfilling method marked by the annotation @GetMapping or @RequestMapping, which tells the server which URL endpoint this controller is responsible for. Some controllers need to accept URL parameters and, @requestMapping makes this possible. Controllers have also been given helper functions where necessary to interpret and format data from the model layer.

The model layer consists of a small number of java classes, a Neo4j Database, a common point from which template queries can be accessed, and a means by which sentiment analysis can be performed

Prototype:
The class QueryNexus serves as a common point that all other classes can retrieve one of the available cypher queries from. If an object provides a simple string key, they are returned a much longer parameterized (or 0-parameter) cypher query as a string which can be used by the Neo4j Query class

Singleton
The classes database and QueryNexus only require one instance to be running at a time, and running multiple instances is wasteful and possibly damaging to the functioning of the application. For this reason, they have been restricted to singletons

Pipeline
To perform sentiment analysis, we have used the Sentiment capabilities of Stanford CoreNLP. CoreNLP uses the Pipeline pattern to perform a step by step series of Natural Language Processing tasks. It takes a Properties object, which we used to add annotators to the pipeline. This customises the pipeline up for sentiment analysis.

Builder
The sentimentEngine class needs to perform some iterative string concatenation. This is achieved through the use of java StringBuilder, with the method append();

APIs

Neo4j Driver
We use this API to make calls to the Neo4j database. It is used in the model class Database.

Stanford CoreNLP
This is used in the class SentimentEngine to perform sentiment analysis

Spring Boot
This is our server. Spring handles all the complexities of running the server with multiple threads.