System Architecture and System Design

**Architectural styles**

The most benefiting style for are system would be a multitiered architecture. Specifically, a three-tiered architecture. In this style we can separate all the components into three main systems, presentation logic and application. This is the classical design for web-based apps and websites, so it is the most fitting for our programming solution. In this style the presentation front-end is responsible for showing user the data that the logical application layer calculated or retrieved from the data tier.

This benefits us the must because it allows us to format the data as needed, as we are pulling from multiple sites. It is beneficial to have the database separate in these layers we only need to make changes in one subsystem to accommodate the change, as opposed to more centralized styles which would be more work to adapt and accommodate multiple sites.

**Identifying Subsystems**

**Presentation/Interface:**

The first subsystem that is present within is the presentation. This contains both are user interface and are back-end controller, this subsystem is responsible for requesting the builder subsystem to retrieve/store data and present the users with the outputted data. This subsystem doesn’t care about what or how retrieved data was handled, calculated or stored.

**Application/Worker:**

Are worker subsystem being what creates data to display to user. This includes are finders, and builder classes. This layer is both responsible for bridging the users with the database. It both formats for display and storage. There should be no direct interaction with the database here. It only manipulates data that has been acquired already. If a change to the database should occur this subsystem should mostly be unaffected.

**Data/Database:**

The third and final subsystem is the database. This also includes inserters and the result handler classes. In this stage the defining feature is if we were to change the database system would the classes need to be changed. Only classes that have direct interaction with the database here.