# Current Architectural Style Analysis

The design of this system is following model-view-controller (MVC) architectural pattern, with MySQL as the consistent data management and storage. Play is the main framework used in the web system.

**Model**: the model is the domain-specific representation of the information on which

the application operates. It encapsulates the data access layer which is in charge of persistent storage mechanism.

**View**: the view renders the model into a form suitable for interactions., in the case of this system, a user interface. It’s responsible to display every change to users.

**Controller**: the controller responds to all events from outside. It handles all events and manipulate model corresponding to events, and then render views to response.

# Recommendation

**MVC in front-end for Reusability**:

in front-end, it is suggested to use Angular.js or React.js to bring MVC pattern in front-end. It will bring more reusability into system.

**OOD for Reusability**:

utilize object oriented design into system. For example, the backend should be made up by kinds of different classes. Utilize inheritance and composition between classes.

**Publish Subscribe Pattern in backend:**

It is suggested to utilize publish subscribe pattern in back-end as the communication mechanism to decouple the components in backend. All events trigger subscription distributed to subscriber, and be handled by callback of subscribe class.

**Adapter over database:**

It is suggested to set up an adapter layer over database. This will bring extensibility and reusability when the system changes to other platform or database.

**Distributed Processing for Scalability:**

When the whole system is scale up, we need distributed processing over distributed back-end to support increasing clients.