## Conclusions and Future Work

As a conclusion, we analyzed the main architecture design of this work-flow centric scientific social network project. In general, the design of the system follows Model-View-Controller (MVC) architectural pattern by using Play framework, with support of MySQL database at the back-end, and Bootstrap and jQuery at the front-end.

Our team successfully applied 8 different useful design patterns to improve the existing architecture design of the legacy CMDA project. The 8 design patterns are Factory, Flyweight, Façade, Strategy, Builder, Visitor, Template and Command. The benefits of applying these design patterns include supporting a stronger reusability, extensibility, and maintainability for the system, and making the system architecture well-structured. There are also some constraints of using design patterns. For example, design patterns may bring increased effort to apply, and they also define more classes and interfaces to maintain, thus may leading to a poor performance due to indirection, and gaining the complexity of code.
In short, a perfect design patterns for all design problems do not exist. Every design pattern will has its own fitable scenario, benefits and constraints. We need to carefully analyze each design pattern's cons&pros, and then make a trade-off of whether should use it or not given different specific scenarios.

There are also some other good suggestions for improving this CMDA project. Micro-services architecture style and SOA architecture style are also good practices to improve the architecture design of the workflow-centric scientific social network project.

Both of these two styles are the most popular technologies for software development. We could decompose the whole project into multiple sub-systems, then implement micro-services for each of these sub-systems. By using micro-services, we could achieve the good standard of low coupling and high cohesion. What's more, micro-services enable a stronger composition freedom for development, and can make the whole project easier to reuse, extend, scale, compose and manage. And SOA style is mainly focused on enabling business-level programming by using the business processing technologies like BPEL. In SOA, the high-level business service could easily make use of the reusable utility and IT services to achieve a business goal. So if we can adopt the SOA style for the CMDA project, it will greatly promote the sharing and communication process of the system with the public.