## **Reason for the candidate architecture**

The selected candidate architecture follows the pattern of Model-View-Controller(MVC) architecture pattern. The one of the reasons choosing this architecture is MVC design pattern allows the separation of concerns during development phases. MVC design pattern breaks the software components into three main categories, namely model, view and controller. The model is the part of the application that handles the logic for the application data. It formats the data from database into an object by following strict rules of conversion. Besides that, the model formats the incoming data from client side before it is saved into database. It improves the data correctness and reliability. Moreover, the view is presentation of the data that model object contains, whenever the data in the model object is changed, it will generate the view’s presentation of the model based on changes in the model. Besides that, the controller is the part connecting the model and view and control communication between model and view. It controls the data flow into model object and updates the view when there is any change of the data in model object. Thus, it promotes the reusability of business logic and template across the application.

Furthermore, MVC architecture pattern allows parallel development process among team members. MVC architecture pattern suggests the development of business logic, view and model separately and improve the code reusability. It allows all team members to build business logic and design templates independently without affecting other progress and commitment. MVC architecture allows the developers to focus and concentrate on their own part like UI, model or business logic implementations and modifications without bogged down with other developers. And also, the delay caused by user interface design does not affect the progress of business logic design as the processes are carried out independently. Besides that, the minor change in business logic has very less effect to the user interface design, thus it avoids the revision to user interface design as long as the data are formatted according to models.

Furthermore, the candidate architecture follows the 3-tier layered architecture pattern. It suggests the separation of processes on different nodes or machines. The candidate architecture splits the application into different layers. The layers suggested are user interface, business logic and data storage and data access which are developed and maintained as independent modules or most often on separate platforms. In order to achieve the candidate architecture in the software development, several frameworks are introduced to the software implementation, namely Slim Framework, Propel ORM, Twig Framework and AngularJS Framework. First of all, AngularJS Framework takes care of the presentation layer of the software. It allows the presentation layer to work independently in client side. It provides several useful dependencies in data binding, standard REST call and template rendering. Besides that, Slim Framework builds the environment for business logic layer of the software application. It provides several features including middleware control and router which allow the development of software application based on MVC architecture pattern. It allows the software application to work in a separate node or server to provide computation and basic template rendering features. Moreover, Propel ORM realizes the object-oriented features to PHP application. It builds the data access layer and maintains database according to a designed schema. Besides, it also generates the optimized object-oriented PHP models which speed up the development process of complicated entity-relation design in software application. It handles the modification and changes to the database structure and records the migrations done in organized manner. It helps to maintain the database version and revision.

.