

System Architectural Design

Three-tier architecture is a client–server software architecture pattern in which the user interface (presentation), functional process logic (business rules), computer data storage and data access are developed and maintained as independent modules. *(source: Wikipedia)*

Our system has 3 essential modules; each module has particular responsibility to operate its function:

* **Presentation tier:**
  + This tier displays information to end-user that they can manage with their data or interact with another user.
  + Web application: based on AngularJS project structure, combine using of HTML, Javascript and CSS to one project. Controller components (Javascript files) interact with service layer of Logic tier to request and receive data from database. Model components use Angular local storage technique to storing client data. View components are a presentation of data in HTML format, triggered by controller’s decision.
* **Logic tier:**
  + Service layer: Use Representational state transfer (REST) web service to provide API for accessing data. RESTful web service is supported by Spring framework by using Spring annotation mapping and configuration.
  + Business logic layer: Use Spring framework to operate system business logic function by Spring technique, includes: transaction management and session management.
  + Persistence layer: Use hibernate framework to manage database accessing operation includes: data source management, connection pool support, query language support…
* **Data tier:**
  + Our system use MySQL to manage database. The persistence layer of Logic tier connects to Data tier by using MySQL connector driver.



### 🡪 Discussion of chosen AngularJS in presentation tier

AngularJS has many strengths that can meet our system requirement and goal that we have decided to choose this framework. AngularJS strength is about:

#### 1. MVC pattern familiar

AngularJS splits web application into MVC components, which is familiar with our team member to understand and manage code.

#### 2. Client side advantages

Different with JSP, AngularJS can be developed separately from web services that we can archive:

* More interactivity by immediately responding to users’ actions.
* Execute quickly because they do not require waiting response from server for loading pages.
* Easy to reuse resource and manage HTML template separately from server.

#### 3. Modularity

In AngularJS, we can create application combining separated modules and manage each module separately.

For example: Staff or Admin will have different views from User so that we cannot apply one UI for all roles. AngularJS allow us to separates from project to modules so that we can handle UI well although both modules use the same resources.

#### 4. Dependency Injection

Different from some other client side framework, AngularJS allow us to design MVC pattern that can inject wrapper, setter, getter or class declaration across the application. AngularJS do it themselves so that we can save the most of development cost.