MVC

We will describe our design based on MVC mode, the model, the view, and the controller. Each of these components are built to handle specific development aspects of our design. [1]

1. Model/Controller

The Model component corresponds to all the data-related logic. In our design, we choose MySQL, the language that we access the database is PHP. For PHP, it based on PDO. Controllers act as an interface between Model and View, manipulate data using the Model component and interact with the Views to render the final output. In our design, we will create a folder named “command” to perform the command from the user by View components.

Based on our design, because we want to make the architecture not too difficult and also only using MySQL database, we combine Model and Controller aspects into one. In “command” folder, it waits for View’s request by Html Get/Post methods and using PHP to access database.

Besides, for Controllers, there are two aspects: updates and notifies. In our design, we also separate them as far as possible. For updates, we have a file named “command.php”; for notifies, a file named “getReference.php” will engage. But some files, we merge two aspects into one file, that is because we focus on its meaning. For example, a function provides a service to present all names of library which are related to user, and then user can choose the needed library for further operation. (which is included in “searchLibrary.php”)

1. View

The View component is used for all the UI logic of the application [1]. In our design, we have several files to do this part, for instance “Openlibrary.php”, shows the content for user. It involves Html and PHP languages, and also for the stylish, CSS language, and for the front-end language, Javascript.

Security for the database setting

According the security issue, we separate our code base into two parts, one is “Connections”, it includes the database setting; the other is our website code. By this separation, anyone can not access directly by url.

Ajax (Data Loading)

AJAX allows web pages to be updated asynchronously behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page. By this method, browser do not need to query all data when its first loading. It can save bandwidth. In our project, we use this method when we want to query the information to help user to use our website.

Share Library mechanism (database table)

According our design, we have four tables in our database, “userTable”, “referenceTable”, “libraryTable” and “shareLibraryTable”. For “userTable”, it is used to record user profile. Where “userID” is a primary key and it is a foreign key of “libraryTable” and “referenceTable”. By the information in “libraryTable” and “referenceTable”, we can know whose library or reference is. And for the “libraryTable”, it is used to record what library is and created by whom. Where “referenceID” is a primary key and it is a foreign key of “referenceTable” and “shareLibraryTable”. Then we can know which library that a reference belongs to and what user can access this library. And about “referenceTable”, it is used to record the reference information. Any reference is unfiled while it is creating first time. Then we will set the “defaultLibrary” is 1. And for achieving the “Trash” function, we use the “isDelete” to decide it is active (in the any library) or dead (in the Trash). Finally, “shareLibraryTable” is used to state who can access the library. We use “shareUser” to record the multiple “userID”. Which conjoin two different “userID” with a comma “,”. By the method, we can save our database size.

Modal

Modal is a lightbox in Bootstrap. Through it, we can add dialogs to our site and remove scroll from the <body>, using modal content scrolls instead. It is similar to a pop up window. While using “Modal”, we do not need to navigate to the other page to access the database.

JSON

We use JSON to build up a bridge between database and front-end for share the data from database. By JSON, we can access the amount of data once.

Validation

We implement a validation for our password to make sure the password security. By check ing the expression and length, it could make the password strong. Also, we use SHA1 to build up a basic protection on the connection between front-end and back-end to avoid the hacker attack.

[1] <https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm>