Hong Kong Institute of Vocational Education (Tsing Yi)

Department of Information Technology

HD in SE (IT114105)

Assignment of Enterprise Systems Development (ITP4511)

Group project

|  |  |
| --- | --- |
| Cheng Cheuk Nam | 170695922 |
| Yip Yiu Cheung | 170128976 |

### Assumption and the user and system requirements

It is assumed that the user account has been created already. User also knows his Id, password, role and some basic information are already saved in system.

Moreover, the Takeaway King system already have the restaurant record and the owner no need to add his own restaurant information.

### Site map

### System structure on how MVC Model is applied

**Controller**

In our system, there are different servlet, such as LoginController.java, handleRestaurant.java, handleRestaurantEdit.java, as controller to handle the jsp page request, jsp form or response for the navigation.

In login controller, it will receive the post request which include the user id and password parameter and then do the process depend on the request action, such as login or logout. If the user has logged in already, nothing will be do in controller. After the login or logout process is finished, it will redirect to the main page of takeaway king and save the user information in session and UserInfo bean which is a model.

In handleRestaurantEdit.java controller, it can send the command to external entity which is mySQL database. The controller also gets the database result to update (e.g., create, read, editing or delete the restaurant information) and save the status in the model.

The handleRestaurant.java is also as a controller to forward the page from different request. For example, the controller will return a model which is a restaurant bean if the “view restaurant” action and other related parameters are received from request.

handleMenu.java, handleMenuEdit.java, uploadNewMenu.java and so on in “ict. servlet” package are also the some design.

**View**

In our system design structure, there are different jsp page for handle different functions and model presentation. ViewOwnRestaurant.jsp, editMenu.jsp, viewRestaurant Details.jsp as different views that request information from the different model which is restaurant bean, menu and so on that the view needs for generating an output representation to the user.

**Model**

There are many models in our system. “Comment.java”, “Menu.java”, “Restaurant.java” and “UserInfo.java” etc. are the bean. They become an entity or a record of database. When the controller receives the request from view, the controller may run the process and receive the status and store it in its corresponding model such as the user information status will store in user info bean. The model is associated with views and controllers when there has been a change in model’s state. This notification allows the views to output the updated information, and the controllers to change the available set of commands. The model is a "passive" object and its status would be update by the information inputted from view and changed by controller.

### Database structure

### Characteristics and design of application

**Login security**

The Takeaway King system is run with local database. In the main page, user who is not login can view the restaurant and menu as guest. When the user login the website, there are JavaScript and Controller to check the user login information. The user is no way to input illegal statement and wrong information to login. The user id and password are restricted to input special symbols or space by JavaScript in order to prevent the sql injection. Also, the Login Controller will check again and remove all blank symbol for double security.

**Edit information Security**

Only the restaurant owner would show the “maintain restaurant button” to maintain his/her own restaurant. Although the people input the site url directly, there are still different checking procedure to prevent other unrelated users to enter this page.

Moreover, all the edit form is processed by post method, such as edit restaurant or edit the menu. Before updating the information, the controller would check the current user id who is or not the owner of the restaurant or menu. Other restaurant owner or users cannot edit the restaurant or menu information which its owner not the current logged in user.

**Edit User Favorite**

Before user add or remove favorite, the JSP will call servlet to check that user which is login. If user is login, user can press the button to add restaurant or menu to “My Favourite”. Also user can remove restaurant or menu in “My Favourite” page.

In edit favorite, the system will check the favorite in the user’s myFavourite. If that item which is restaurant or menu in user’s myFavourite. The button will show black star and user cannot add that to myFavourite again. Otherwise, the button will show white star and user is allowed to add that to myFavourite.

### Conclusions

In conclusion, we have described the assumption of our system, Takeaway King site map, MVC model description, database structure and the techniques of application.

All the limitation is shown in “assumption of our system” part. You can see all the Takeaway King Web pages and related function in site map. The our MVC Model structure, the View is the Java Server Page (JSP). The code that generates the View can be part of a servlet, especially when the client and server interact. Then, the Controller is servlet that process the request and be written with the frameworks and use the model which is an entity bean to save the status.

That is all the design structure of our website application and the following is the Skill checklist. All the technologies we have used are shown below and described that how its applied.

### Skill checklist

|  |  |
| --- | --- |
| Skills | Apply in |
| Servlets | Controller of login, upload photo, form editing |
| JavaBean | Save all the status from database result |
| JSP Action | Pass the form parameter |
| Taglib | Sorting the popular restaurant |
| JDBC | Connect the database with java |
| Session | Save the user who logged in |
| MVC Model | Jsp for view, controller for process the request, model for save the status. |
| JavaScript | Basic form checking |