**Comp 3550: Internet Technologies II**

**Lecturer: Mr Kyle De Freitas**

**The University of the West Indies**

**Student Tracking Application**

**Group Project**

**Stefan Hosein |Christopher Flus |Allan Gittens |Nicholas Mendez**

2013 – Semester 1

**Table of Contents**

Introduction

Description

Database

**Introduction**

This project was developed in response to an identified problem facing the Computer Labs of the Department of Computing and Information Technology, The University of the West Indies. The current system utilizes a manual sign in process for students using the lab. There are no effective records of recording which computer is used or how long it has been in use. The project therefore required the development of a system that, working along with a simultaneously developed means of recording usage data to a central server, enables the analysis of this usage data for the individual computers as well as the lab as a whole. This will also provide for the identifying of any possible trends in lab usage as a whole or by individual students. The system is to be used by varied groups of persons including students, lab supervisors and managers.

Inherent in the development of the system was the need to build adequate application management capabilities for the system’s end users with their different privileges. The web-based application developed utilizes various internet technologies and techniques learned throughout the course as well as the addition of other minor out-of-course components. Using a database as its backend, the application uses a browser based front end.

The application allows for four user types each accorded specific privileges and who are provided with secure access to the application with individual accounts. Each category of users is afforded different account privileges which limit capability and access to information. The application monitors the usage of each computer in the lab as well as the usage of the individual users. Reports may be generated on the usage of these computers and filtered as desired to provide the necessary information. In order to provide easier analysis of data and a more user friendly interface, the application heavily utilises graphs for the presentation of data.

**Description**

The emphasis of the web application built is for it to be Modular and Extensible. Taking these two properties into consideration it was decided that an MVC framework was the most logical choice. Upon deciding to use the MVC strategy, the best framework for our use was Codeigniter due to its minimal footprint and extensive documentation which made understanding the framework trivial.

Moreover, our model contained 5 files, each for a specific user and one for login and logout. Each model was built with additional features specific to each user type. This prevented the grotesque method of repeating code for each user and simplified adding a new user type.

Additional, the view contained many files with not only views specific to the user type, but header, footer and login views. This increased Modularity since each view did not depend on one another which is one of the goals of the application and enhanced Extensibility since the developer can modify a view without worrying about the underlying features.

Lastly, there are controllers for each user type and one for logging in and out. Each controller related to a specific user thus it enhanced Modularity and Extensibility because changes to one controller would not affect the other controllers and adding on functionality to a user is easy by simply calling the functionality from a model without knowing how the models work.

**Justification**

**Justification of the Models:**

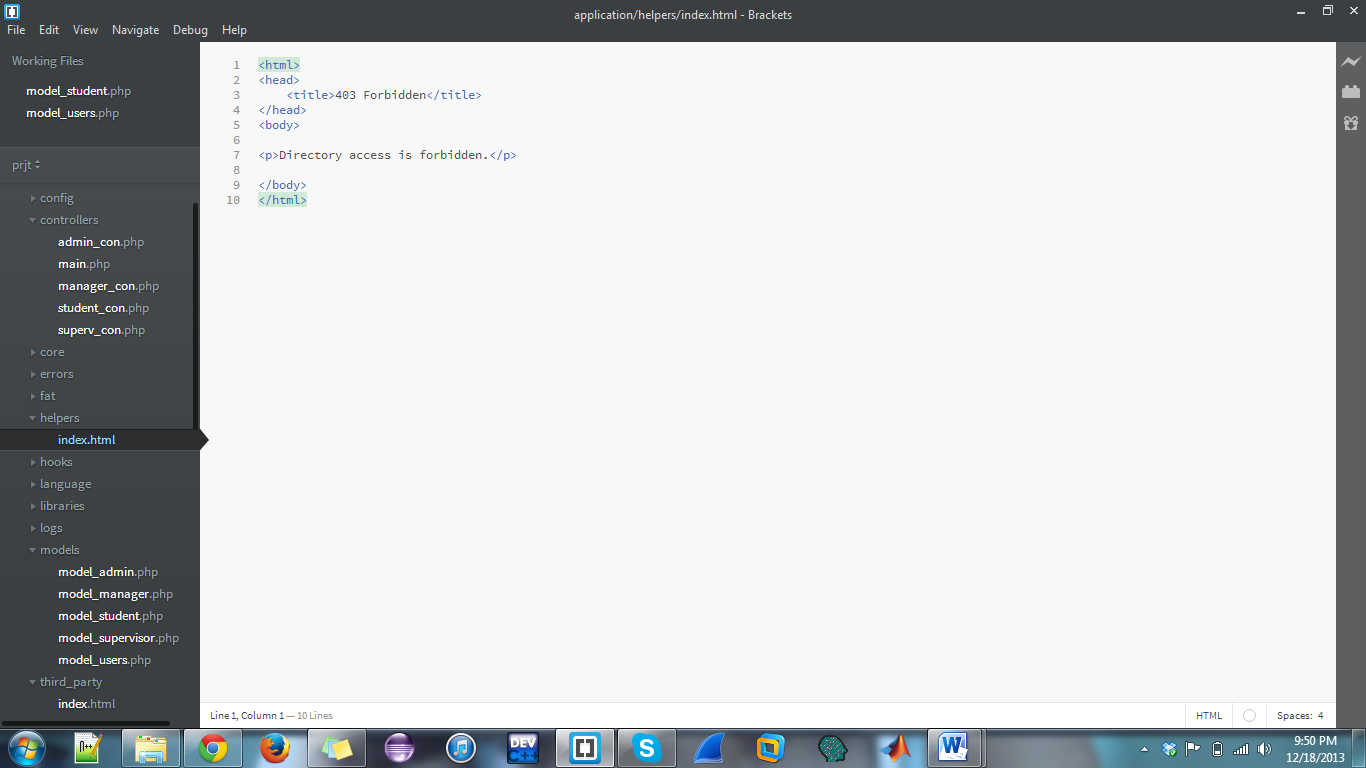


Figure 1. The figure shows the different Models in the application

In our MVC structured application, we had a Model for each type of user and as well as a general model for all users. These models had names respecting their operation such as “model\_admin” and “model\_supervisor”; which would be the models corresponding to their type of user.

In the “model\_student” model, there are functions to accomplish checking their personal information as well as seeing their usage graphs and tables. Admins, supervisors and managers can also view this information for them and will access this via the same code as from the “model\_student” model instead of repeating the same code in its own respective model. This removes code repetition and increases modularity.

**Justification of the Views:**

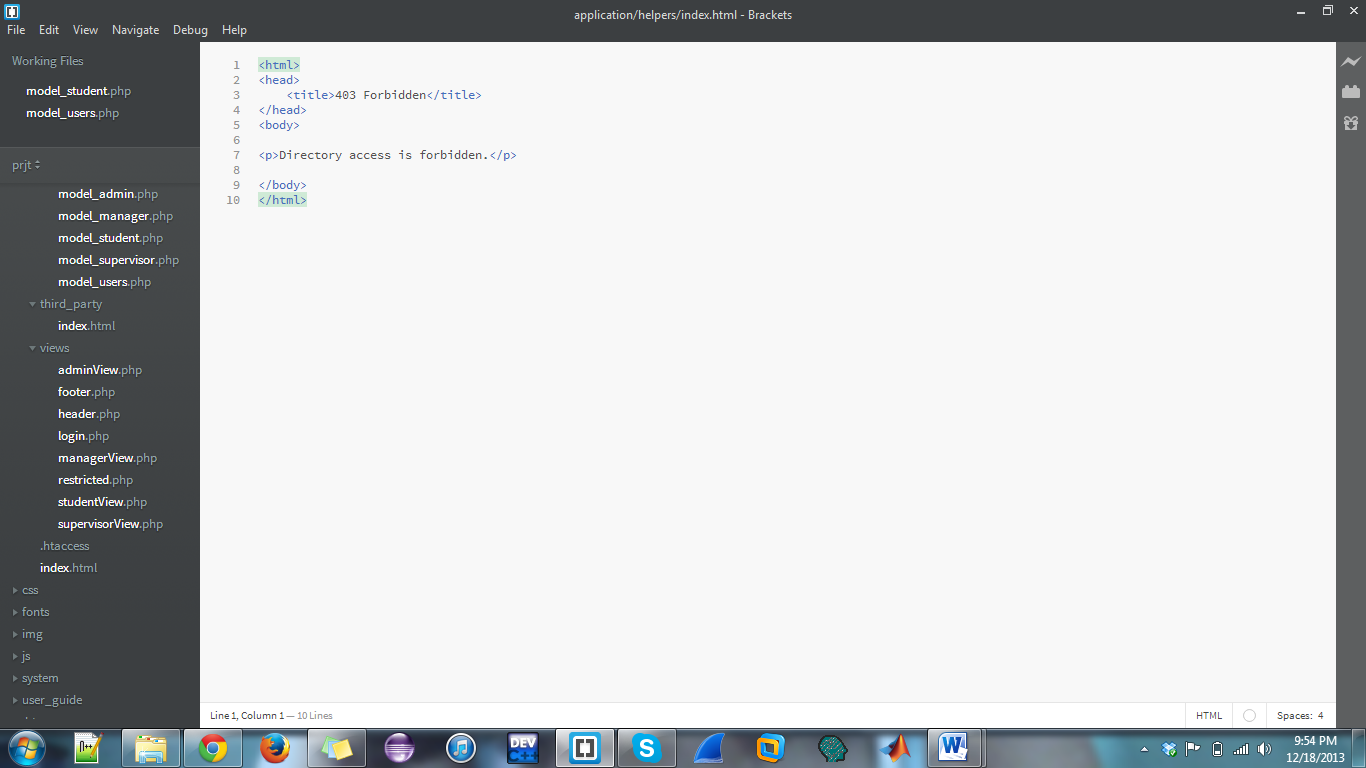


Figure 2. The figure shows the different Views in the application

**Justification of the Controllers:**

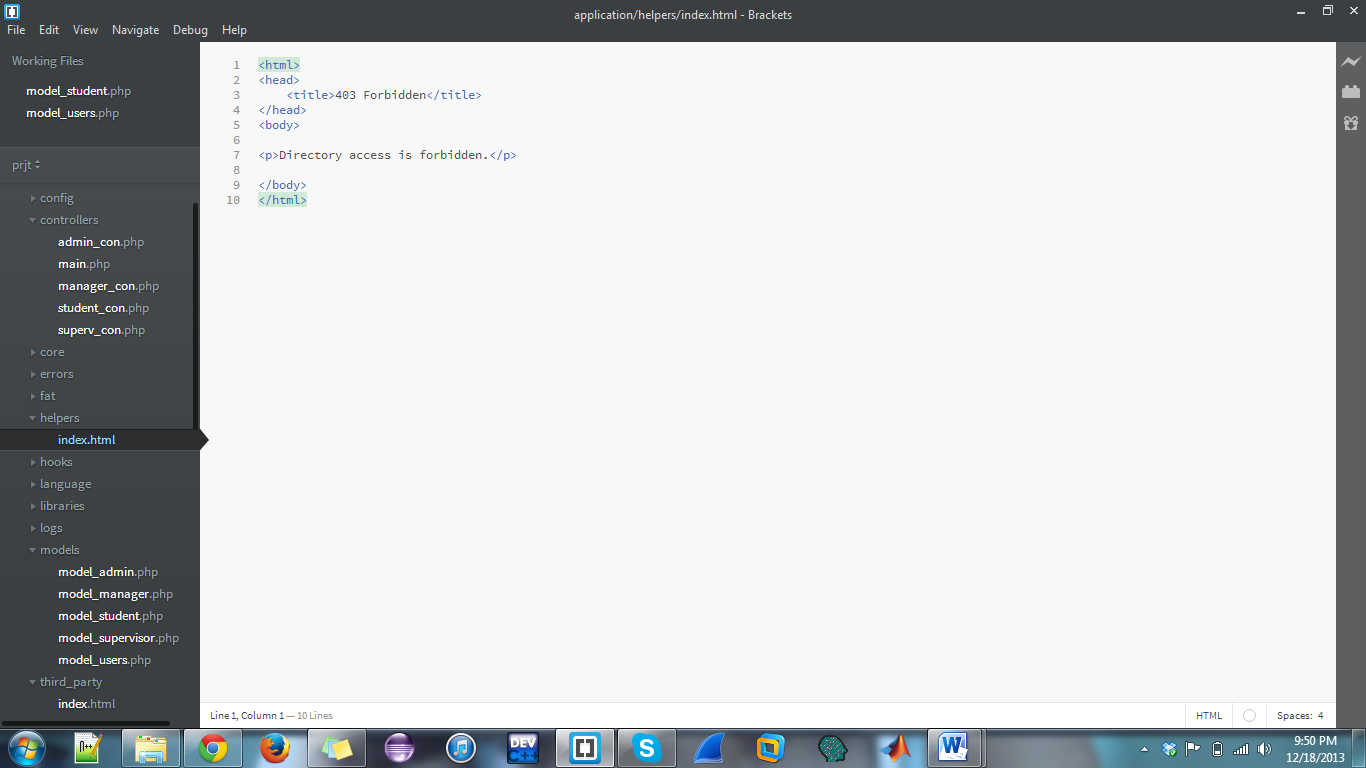


Figure 3. The figure shows the different Controllers in the application

There are 5 controller files, each representing a user type and one (main.php) for the logging in and out of users. Since a controller can send commands to the model to update the model's state and also send commands to its associated view to change the view's presentation of the model, we ensured that the controllers adhered to this format.

When a user logs into the application, it get the user type and displays a view corresponding to the user. An example is that if a Manager singed in then he would be presented with the Manger View and is not able to modify the URL to access another view as he/she would be presented with the restricted view. All this is done in the controller to ensure Modularity, Extensibility and prevent repeated code.

In addition, the controller is responsible for accessing the model to get functionality for that user. If the Manager decides to view all the information in the log table, then he/she would click the button for this and the information would be sent to the controller which would then call the model. The model would perform the task of getting all the log information and send it to the controller which would replay this information and it would be presented in the view.

**Justification of other components:**

Other components that we used in this project were Highcharts, Bootstrap and Codeigniter. Highcharts were used because it is a very easy-to-use and intuitive way of incorporating interactive and dynamic charts in a web application.

We used Bootstrap for designing our pages. It has a great set of readymade libraries which we incorporated into our design. Also bootstrap is responsive; this means that if our web application is viewed on some other device, such as an Ipad or Iphone, bootstrap will adapt to the change.

CodeIgniter is an MVC framework that we used for this project. It has the smallest footprint for any MVC structured template. It also had extremely user-friendly features and easy-to-follow tutorials in order to accomplish our needs. It made creating our own MVC based application a breeze.

**Database**

PHP MyAdmin was utilised to create the database as it provided an easy means of connecting to the browser-based application through PHP. The application was designed so that the interaction with the actual database will be minimal for the users of the application and that most operations will be performed through the application and its user interface.

The database used in the application consists of four tables: Users, Usage, Computers and Log

The following relational schema was created:

**Users** (*id, username, password, usertype*)

**Usage** (*id, username, date, time, comp*)

**Log** (*id, user, start, finish*)

Computers ( )

The **Users** table contained the records of all user accounts and was indexed by the id. Although the id field was made the primary field of this table, as the username was of key importance to the application, the data rules enforced a unique constraint on the username field as well. The password field was set to encrypt all values in the field with the SHA1 encryption upon entry. This provides security to the users and prevents the accessing of passwords from those with viewing privileges to the database.

**Usage**

This table provided the main basis of the data for the application and as such is a key component of the database. The table was designed to hold the records of the usage of the lab computers by users. Again the table was indexed by an id field. Each record represents an instance of a user on a lab computer and would provide the username of the specific user, the date of usage (*date*) as well as the total time spent on the computer (*time*) and its given id (*comp*). This data is used to create the graphs and other reports of the application.

The **Log** table records the instances of the application being assessed. Every time a user logs on to the application, an entry is created containing the user’s username and the exact system time of the logon (using timestamp). On logging out the corresponding record is updated with the logout time being recorded with another timestamp entry. This table is also indexed by an id field to provide a unique means of identification for the records. The user field records the username while the start and finish fields of the table record timestamps for the user login and logout respectively.

**Conclusion**

dd