

**Internet Programming Project**

**Tini Blog**

Framework: CodeIgnitor

**Group Index**

Member Wu Zeling

Member He Lijun

Member Zhou Zikai

**Content**

[**1.** **Framework Specification** 3](#_Toc470551732)

[**1.1.** **Description** 3](#_Toc470551733)

[**1.2.** **Framework Characteristics** 3](#_Toc470551734)

[**1.3.** **Framework Restrictions** 3](#_Toc470551735)

[**1.4.** **Framework Working Process** 4](#_Toc470551736)

[**2.** **Website Specification** 5](#_Toc470551737)

[**2.1.** **Technical Structure** 5](#_Toc470551738)

[**2.2.** **Website Functions** 5](#_Toc470551739)

[**3.** **Demo of Framework Features** 6](#_Toc470551740)

[**3.1.** **MVC Based System** 6](#_Toc470551741)

[**3.2.** **Form and Data Validation** 8](#_Toc470551742)

[**3.3.** **Multiple Database Support** 10](#_Toc470551743)

[**3.4.** **File Uploading Class** 12](#_Toc470551744)

[**3.5.** **Error Logging** 13](#_Toc470551745)

[**3.6.** **Unit Test** 14](#_Toc470551746)

[**4.** **Evaluation** 15](#_Toc470551747)

[**4.1.** **Group Work** 15](#_Toc470551748)

[**4.2.** **Task Responsibilities** 15](#_Toc470551749)

# **Framework Specification**

This section will introduce basic information about the CodeIgnitor framework. At first, it talks about the goal of CodeIgnitor framework. And then it will introduce the characteristics of this framework. It is these outstanding features that we choose the CodeIgnitor. Furthermore, it will talk about the disadvantages of the framework that show its restrictions.

# **Description**

CodeIgniter lets you creatively focus on your project by minimizing the amount of code. [1] The feature of extremely light weight could reduce your workload and enable the web application have a better performance. Moreover, zero configuration and no need to use command line enable tyro work more easily and rapidly. More importantly, the MVC development pattern has a clearer logic and enable the web application work more efficiently.

# **Framework Characteristics**

MVC Based Systems: The most outstanding feature of CodeIgnitor is Model-View-Controller design pattern. The controller is the heart of the whole application. Since it determines how HTTP Request will be handled, we put all of the logic of our website except operations of database in the Controller. The model is designed to deal with information in the database. The view is just the appearance of the website, it must be loaded by the controller.

Form and Data Validation: It helps you to write the code in a single line by using effective validation framework system. It generates codes without any errors and ensures various control structures to be placed within HTML form.

Query Builder Database Support: It can simply perform a database connection by using one or two lines of code. More importantly, it allows you to develop database independent applications since the query syntax is generated by each database developer.

Error Logging: It helps web developers to find out the errors in programming codes and fix-up the issues instantly compared to the case of non-framework development.

Unit Test: Unit testing is an approach to software development in which tests are written for each function in your application. CodeIgniter’s Unit Test class is quite simple to evaluate your code to determine if it is producing the correct data type and result.

# **Framework Restrictions**

It is PHP based only and not very object-oriented in some parts

Its MVC structure could result in complex dependencies and tight coupling in the application if developers use heavy controllers’ strategy or heavy models’ strategy.

Company-driven instead of community-driven.

# **Framework Working Process**

This framework start from checking the routes.php under the application/config. The framework then will be redirected to the application/controllers file and run the php file whose name is the value of $route['default\_controller'] back in routes.php. Then in this controller index.php file, it will run index function, load a view and wait for further commands.

# **Website Specification**

# **Technical Structure**

The structure of our website follows the MVC design pattern. Therefore, its structure is divided into four separated main parts: View, Controller, Model and Database.

The view part is just the appearance of our website. In order to manage and update our website more effectively, we put the CSS folder outside of the view part. Our core logic is in the controller. It realized the form validation, initializing Unit Test Class and calling view part. The model part is to extract information from database, add new information to the database and delete information in the database. The database part is to store user editorial information. And the pictures uploaded by users and other pictures we used in the website are stored in the image folder.

# **Website Functions**

The main function of our website is to record life and coding experiences for Software Developers.

Introduction: There are basic information of CodeIgnitor and the user guide of our website.

Content: This is the history of the users’ recording. Here users can read their past life experiences and this part could act as a diary notebook.

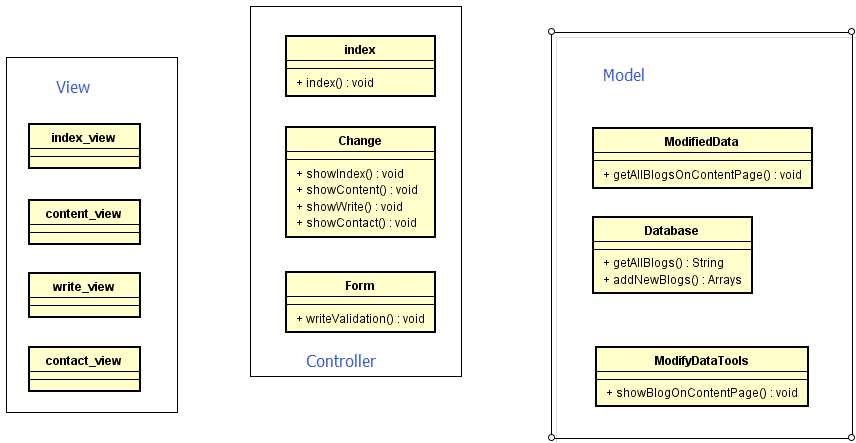
Write: This is the editorial part. Here uses can record their emotions, study experiences and even some useful programming skills. And users can upload a picture as well.

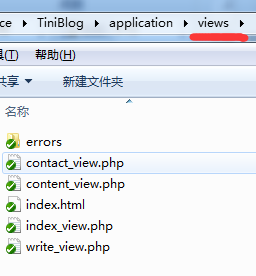
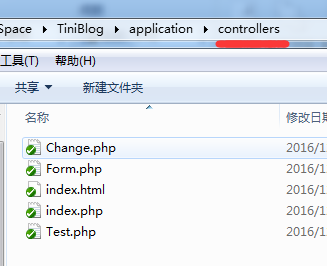
Contact: This is the part of our group members’ detailed contact information.

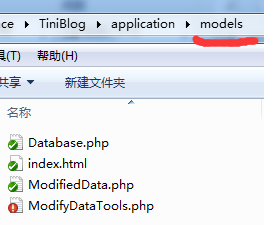
# **Demo of Framework Features**

# **MVC Based System**

There are many files under the application category. But among them, the most important files are controllers, models and views which contains certain relating files.

And these classes are put in:



In the controller and the model, every php file represents one class. In view, every php file contains most of the html code and a few php codes to call controller's function.

To ensure that a web developer is programming properly, CodeIgniter doesn't allow him/her call another controller's function from one controller's function or call controller’s functions from model's functions. The functions of controller can only be called from view.

# **Form and Data Validation**

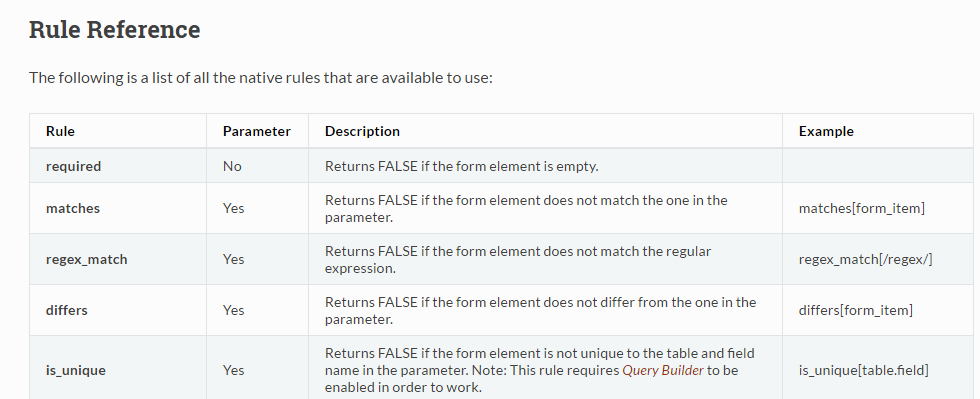
We call the function of form validation in write\_view.php

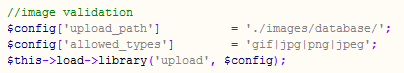


The underlined code will form <form url="<?php base\_url/form/writeValidation?>"> and it will form a special <form> because we need to upload an image here. Then after we click submit, the writeValidation funtion in class form will be called. We can simply set the validation rules by



CodeIgniter has already provided many useful rules that we can use easily. i.e.

 Besides it also provides useful support for image validation. We can define the image type we need just by a few lines of codes.



# **Multiple Database Support**

Traditional PhP web application can only support MySQL. But CodeIgniter framework can support PDO, PostgreSQL, Oracle, ODBC, sqlite with very easy setup. We can simply change some setting and the framework will connect to your database automatically. In this project, we have used sqlite and mysql as our database.





The first image is the setting of MySQL, and the second image is the setting of sqlite. This is the only file that need modification and the rest of the code can be kept as unchanged.

The additional feature of database is Query Database Builder Support. It is independent of database, thus making developer edit information of database easily among different databases.



As the image shows, a developer has no need to know exactly what the query syntax in each database is, but only need to know corresponding database operational functions. Apart from benefiting developer, this feature also improves the reusability and transferability of our code.

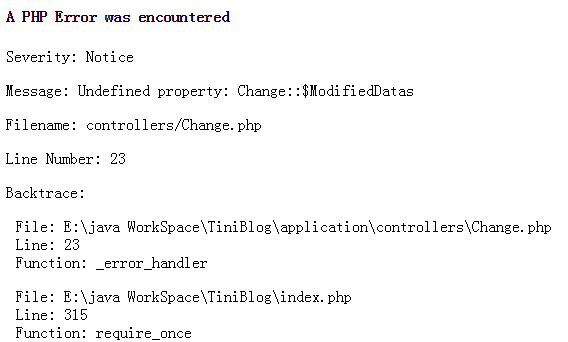
# **File Uploading Class**

In this project, we need to upload images in write\_view which needs to implement File Uploading Class. Besides uploading function, this class also provides *Image Resize Function*.



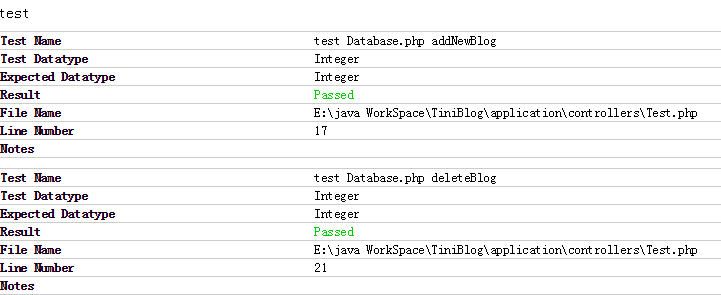
# **Error Logging**

CodeIgniter provides very specific details about syntax errors that helps us to find the errors rapidly and easily.



# **Unit Test**

CodeIgniter provides very simple Unit Test funtion. It doesn't support very complicated test. Because the logic of our project is very function, I only test the Database funtion.



# **Evaluation**

# **Group Work**

Choose the framework.

Analyse the structure, theme and functions of our website.

Setup our website.

Use the framework to implements functions of the webpage.

# **Task Responsibilities**

Wu Zeling(Group Leader): Setup the website, Setup the database and connect to the database, Input validation, Unit test and report writing.

He Lijun: Disign the Website, implement upload function and validation, put upload image into the database.

Zhou Zikai: Disign the Website and report writing.

Reference:

[1]<https://www.codeigniter.com/user_guide/overview/at_a_glance.html>, Dec 25 2016.