

**Whole book:**

# **Second-hand Book Transaction Platform**

Song Tang

University of Pittsburgh  
INFSCI 2710: Database Management Fall 2016(Monday)

# INDEX

<b>WHOLE BOOK:</b> .....	<b>1</b>
<b>SECOND-HAND BOOK TRANSACTION PLATFORM .....</b>	<b>1</b>
<b>1. INTRODUCTION.....</b>	<b>3</b>
1.1 PURPOSE .....	3
1.2 PRODUCT FUNCTIONS.....	3
<b>2.SYSTEM DESIGN .....</b>	<b>5</b>
2.1 DEVELOPING ENVIRONMENT .....	5
2.2 DATABASE DESIGN .....	6
2.3 WEB DESIGN.....	10
<b>3.MODELING REQUIREMENTS .....</b>	<b>17</b>
3.1 USE CASE MODEL .....	17
3.2 FUNCTION DIAGRAM .....	18
<b>4. PROTOTYPE.....</b>	<b>19</b>
4.1 CUSTOMER .....	19
4.2 MANAGER.....	23
<b>5. TEST .....</b>	<b>25</b>
5.1 TEST DESIGN .....	26
5.2 TEST DRIVEN DEVELOPMENT .....	26
<b>6. LIMITATIONS .....</b>	<b>26</b>

# **1. Introduction**

This report is designed to document and describe the specification and detailed introduction of the web system. The web system being produced is called Whole Book: a second-hand transaction platform for System for e-commerce. Its aim is for customers to sell new or second-hand books via the Internet. This system is designed to provide automation support for the process of placing and purchasing books on the Internet and run on local server with each user having a user interface through a web browser to interact with it.

## **1.1 Purpose**

This E-Commerce Book System allows any user to create an account to become a customer. The customer, through the process of account creation, will have the right to sell their used books and purchase any books posted by others. In details, first, the system allows customers to register, select, and add books to a shopping cart; then check out books in shopping cart by the balance in their account. Second, the system allows customers to be sellers to post information selling their own books (no matter new or used). Customers can change their personal information as well as posted information, and view their order histories in real time.

Whole Book (WB) also allows a manager to view and manage all the information posted by customers. Manager has the access with full create, retrieve, update and delete functionality with regards to all books in the system. Also, the system is dynamical-intelligent to adjust the stock of inventory the system maintains so that customers and manager can know the current record of inventory.

## **1.2 Product Functions**

WB is an online bookstore website which supports a number of functions for both the consumer and store's management. WB will provide a number of functions; each is listed below.

- Maintain data associated with the inventory (a collection of books)
  - A book has a title, type, author, price, contributor (stock), state (new or old), and description image.
  - Each book is listed by the category of type and state.
  - The inventory also keeps track of the stock/quantity of each book. Once a certain book is out of stock, it will disappear from the category list.
- Maintain records for each customer
  - Everyone can register as a customer.

- A customer has a username (unique across all users)(it is stored by id in database), password (no restrictions), email address (no restrictions), postal address (unverified), and payment information (card number, expire date and back code).
- A customer can buy books from WB and also can sell books on WB (post goods).
- Every new account has a certain balance for purchase at first. The balance will reduce when customers make order.
- Allow customers to log in and out of the system and they can change personal information anytime as they like.
- Customer can view the purchase history as well as selling history.
- Show a listing of available books
  - Books are to be displayed in order by posted time.
  - Books are posted as classified when posted.
  - Each book will list the title and price on index and can be viewed details when clicking.
  - When a single book is purchased or the books are out of stock, it will not be appeared in catalog.
- Maintain admission of manager
  - Manager can only be added in server's back-end side system.
  - Allow manager to view all transaction logs/orders.
  - Allow manager to view all books posted.
  - Allow manager to change personal information.
- Shopping cart
  - Customer after login is able to add one or more books to the shopping cart.
  - Customer can view shopping cart after the book is added in.
  - Customer can only check out after the item is added in shopping cart.
- Checkout
  - Checkout is only available to logged-in customers.
  - Customer can choose which order to be checked out.
  - The order may have three statuses: waiting to be checked out, check out success and check out fail.
  - Log/record the transaction histories in customer's page and manager's page.
  - Allow manager to modify the order.
  - When the quantity a book falls below a threshold, the manager is notified that the book.

## **2.System Design**

### **2.1 Developing Environment**

WB is a complete e-commerce web system. The language usage is composed by front-end web pages by html/css/JavaScript, back-end server side by php, and database connection by SQL.

WB is developed under the environment of MAMP, which is an archetypal model of web service stacks. MAMP represents its original four open-source components: the Macintosh OS operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language. As a solution stack, MAMP is suitable for building dynamic web sites and web applications on Mac.

#### **2.1.1 Software**

- Web-based application: Atom/ Sublime
- Database information storage system: MySQL

#### **2.1.2 Framework**

In this project, we choose codeIgniter as a framework to develop WB. CodeIgniter is an Application Development Framework , a toolkit for building web sites using PHP.

Why we choose CodeIgniter is because it is light weight framework that the core system requires only a few very small libraries. Additional libraries are loaded dynamically upon request, based on your needs for a given process, so the base system is very lean and quite fast. This advent is largely speed up the development efficiency and the program is easy to be read and maintained.

## 2.2 Database Design

### 2.2.1 ER/Relational schema

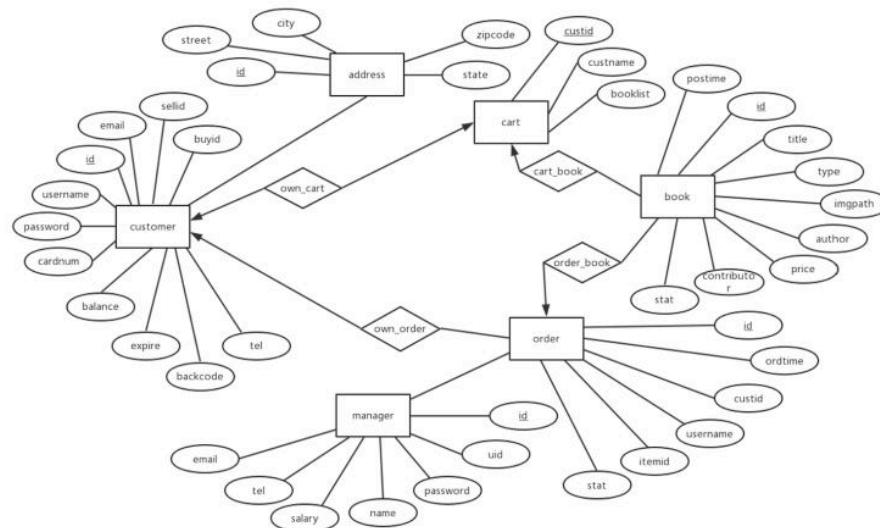


Figure 1

### 2.2.2 DDL

```
/*
Navicat MySQL Data Transfer
```

```
Source Server      : localhost
Source Server Version : 50617
Source Host        : localhost:3306
Source Database    : eshop
```

```
Target Server Type : MYSQL
Target Server Version : 50617
File Encoding     : 65001
```

```
Date: 2016-12-01 12:40:50
```

```
*/
```

```
SET FOREIGN_KEY_CHECKS=0;
```

```
-- Table structure for `eshop_address`
```

```
DROP TABLE IF EXISTS `eshop_address`;
CREATE TABLE `eshop_address` (
  `id` int(11) NOT NULL COMMENT 'Adress_id',
```

```
    `street` varchar(128) DEFAULT NULL COMMENT 'street',
    `city` varchar(128) DEFAULT NULL COMMENT 'city',
    `state` varchar(128) DEFAULT NULL COMMENT 'state',
    `zipcode` varchar(32) DEFAULT NULL COMMENT 'zip',
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-----  
-- Records of eshop_address  
-----
```

```
-----  
-- Table structure for `eshop_book`  
-----
```

```
DROP TABLE IF EXISTS `eshop_book`;
CREATE TABLE `eshop_book` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `title` varchar(128) NOT NULL,
    `type` int(4) DEFAULT NULL COMMENT 'type',
    `imgpath` varchar(1024) DEFAULT NULL COMMENT 'picture base64',
    `author` varchar(32) DEFAULT NULL,
    `price` varchar(10) DEFAULT NULL,
    `isnew` int(1) DEFAULT NULL COMMENT '1-new 0-old',
    `contributor` int(11) DEFAULT '0' COMMENT '0-admin other-customer ID',
    `postime` varchar(64) DEFAULT NULL,
    `stat` varchar(1) DEFAULT '1' COMMENT '1-in-stock 0-out-of-stock',
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;
```

```
-----  
-- Records of eshop_book  
-----
```

```
INSERT INTO `eshop_book` VALUES ('4', 'asz', '1',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'xiaohui',
'2010', '1', '3', null, '1');
INSERT INTO `eshop_book` VALUES ('5', 'sda', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', '123',
'3320', '1', '5', null, '1');
INSERT INTO `eshop_book` VALUES ('6', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'4700', '0', '1', null, '1');
INSERT INTO `eshop_book` VALUES ('7', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'4700', '0', '1', null, '1');
```

```

INSERT INTO `eshop_book` VALUES ('8', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'4500','0','1','','1');
INSERT INTO `eshop_book` VALUES ('9', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'4700','1','1','','1');
INSERT INTO `eshop_book` VALUES ('10', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'1300','0','1','','1');
INSERT INTO `eshop_book` VALUES ('11', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'4700','1','1','','1');
INSERT INTO `eshop_book` VALUES ('12', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'9900','0','1','','1');
INSERT INTO `eshop_book` VALUES ('13', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'5600','1','2','','1');
INSERT INTO `eshop_book` VALUES ('14', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'4400','1','2','','1');
INSERT INTO `eshop_book` VALUES ('15', 'ddfs', '6',
'http://localhost/uploads/f38294ef1bb92da50a2d3e1868c8e889.jpg', 'fsdf',
'7100','0','2','','1');
INSERT INTO `eshop_book` VALUES ('16', 'test', '1',
'http://localhost/uploads/6c563a1834d5fead71b0a9559d75b4e2.jpg',
'testauthor', '332', '1', '2', null, '1');

-----  

-- Table structure for `eshop_cart`  

-----  


```

```

DROP TABLE IF EXISTS `eshop_cart`;
CREATE TABLE `eshop_cart` (
  `custid` int(11) NOT NULL COMMENT 'Customer_id',
  `custname` varchar(64) DEFAULT NULL,
  `booklist` varchar(1024) DEFAULT NULL COMMENT 'Book_list',
  PRIMARY KEY (`custid`),
  FOREIGN KEY custid REFERENCES eshop_customer ( id ) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----  

-- Records of eshop_cart  

-----  


```

```
-- Table structure for `eshop_customer`  
-----  
DROP TABLE IF EXISTS `eshop_customer`;  
CREATE TABLE `eshop_customer` (  
    `id` int(11) NOT NULL AUTO_INCREMENT,  
    `username` varchar(64) NOT NULL,  
    `password` varchar(512) NOT NULL,  
    `cardnum` varchar(36) DEFAULT NULL,  
    `balance` varchar(10) DEFAULT '10000',  
    `expire` varchar(12) DEFAULT NULL COMMENT 'Expire_date',  
    `backcode` varchar(3) DEFAULT NULL COMMENT 'Secret_code',  
    `tel` varchar(20) DEFAULT NULL,  
    `address` varchar(64) DEFAULT NULL COMMENT 'Adress',  
    `email` varchar(64) DEFAULT NULL,  
    `sellid` varchar(4096) DEFAULT NULL,  
    `buyid` varchar(4096) DEFAULT NULL,  
    PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;
```

```
-- Records of eshop_customer  
-----
```

```
INSERT INTO `eshop_customer` VALUES ('1', 'liuhui',  
'2219d01fe0f75d05bcae94c99ec3f58b', '6213002201123', '10000', '20201102',  
'202', '18576676197', 'shenzhen', '331234958@qq.com', null, null);  
INSERT INTO `eshop_customer` VALUES ('2', 'huipang',  
'2219d01fe0f75d05bcae94c99ec3f58b', '6213002201123', '10000', '20201102',  
'202', '18576676197', 'shenzhen', '331234958@qq.com', null, null);  
INSERT INTO `eshop_customer` VALUES ('6', 'luv',  
'2219d01fe0f75d05bcae94c99ec3f58b', '6213002201123', '199877000',  
'20201102', '123', '18576676197', 'shenzhen', '331234958@qq.com', null, null);
```

```
-- Table structure for `eshop_manager`  
-----
```

```
DROP TABLE IF EXISTS `eshop_manager`;  
CREATE TABLE `eshop_manager` (  
    `id` int(11) NOT NULL,  
    `uid` varchar(32) NOT NULL,  
    `password` varchar(128) NOT NULL,  
    `name` varchar(128) DEFAULT NULL,  
    `salary` varchar(12) DEFAULT NULL COMMENT 'Salary',
```

```

`tel` varchar(20) DEFAULT NULL,
`email` varchar(64) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of eshop_manager
-----
INSERT INTO `eshop_manager` VALUES ('0', '1000',
'21232f297a57a5a743894a0e4a801fc3', 'admin', '30K', '18576676197',
'331234958@qq.com');

-----
-- Table structure for `eshop_order`
-----
DROP TABLE IF EXISTS `eshop_order`;
CREATE TABLE `eshop_order` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`ordtime` varchar(32) DEFAULT NULL COMMENT 'Order_time',
`custid` int(11) DEFAULT NULL COMMENT 'Customer_ID',
`username` varchar(64) DEFAULT NULL COMMENT 'Customer_name',
`itemid` varchar(512) DEFAULT NULL COMMENT 'Book_ID',
`stat` varchar(1) DEFAULT '0' COMMENT '0-wait_pay 1-pay_success
9-pay_fail',
PRIMARY KEY (`id`),
FOREIGN KEY custid REFERENCES eshop_customer ( id ) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of eshop_order
-----

```

## 2.3 Web Design

### 2.3.1 Web Structure

WB uses the Model-View-Controller approach, which allows great separation between logic and presentation. In practice, it permits web pages to contain minimal scripting since the presentation is separate from the PHP scripting.

- The Model represents data structures. Typically the model classes will contain functions that help to retrieve, insert, and update information in database.
- The View is the information that is being presented to a user. The View

will normally be web pages.

- The Controller serves as an intermediary between the Model, the View, and any other resources needed to process the HTTP request and generate a web page.

Thus, the main source code can be divided into three parts, Model part, Controller part and View part. The Model part in CodeIgniter is not necessary since it can be invoked by the existing scripts. The Controller part is a class in php containing a set of functions. The functions are in different subclasses inherited from CI\_Controller to transit user interaction from web pages to server side and database. The implementation of View fit in with the prototype we designed to be shown as web pages for user interfaces. Additionally, the configuration part under the folder config is a complement for Controller to map URI and connect database.

### 2.3.2 Application Flow Chart

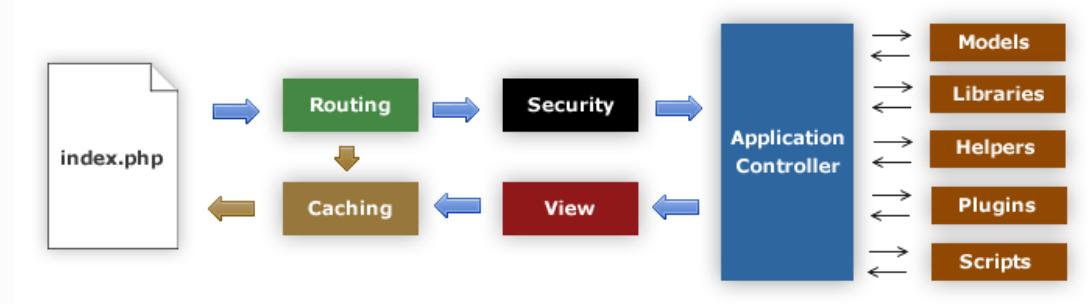


Figure 2

- The index.php serves as the front controller of homepage, initializing the base resources needed to run WB system.
- The Router examines the HTTP request to determine what should be done with it.
- The cache file sent directly to the browser, bypassing the normal system execution.
- Security. Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.
- The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.
- The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

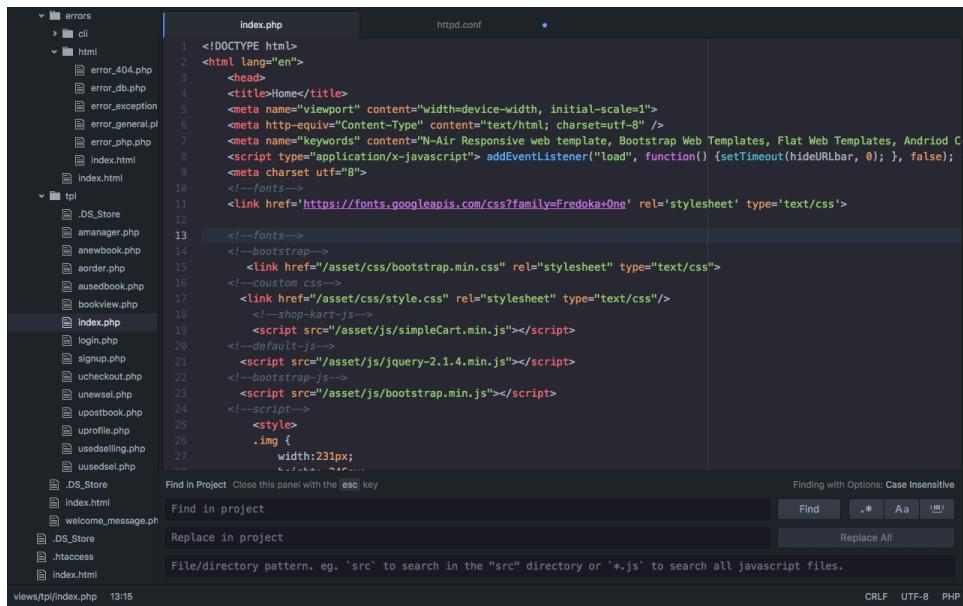
### 2.3.1 Front-end Design

The front-end design is mainly based on HTML, CSS and Java script. HTML is the standard mark up language for creating web pages. Web browsers receive HTML documents from a webserver and render them into multimedia web pages.

HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. CSS is a style sheet language used for describing the presentation of a document written in a markup language. It enables the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It is used to add interactive behavior to the HTML page passing user interaction of web pages.

In WB, HTML pages are all under the folder of View, the CSS and JavaScript are under the folder of asset and embedded into HTML.

- Example of html page



```

<!DOCTYPE html>
<html lang="en">
<head>
<title>Home</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="Keywords" content="N-Air Responsive web template, Bootstrap Web Templates, Flat Web Templates, Andriod C" />
<script type="application/x-javascript"> addEventListener("load", function() {setTimeout(hideURLbar, 0); }, false);
<meta charset utf="8">
</head>
<body>
<!--fonts-->
<!--bootstrap-->
<link href="/asset/css/bootstrap.min.css" rel="stylesheet" type="text/css">
<!--custom css-->
<link href="/asset/css/style.css" rel="stylesheet" type="text/css"/>
<!--shop-kart.js-->
<script src="/asset/js/simpleCart.min.js"></script>
<!--default.js-->
<script src="/asset/js/jquery-2.1.4.min.js"></script>
<!--bootstrap.js-->
<script src="/asset/js/bootstrap.min.js"></script>
<!--script-->
<style>
    .img {
        width:231px;
    }
</style>
</body>

```

Figure 3

```

index.php
32 <body>
33     <div class="header">
34         <div class="container">
35             <div class="header-top">
36                 <div class="logo">
37                     <a href="index.html">Whole Books</a>
38                 </div>
39                 <div style="float:right;margin-top:10px;margin-right: 10px;">
40                     <?php
41
42                         if(empty($username)&&empty($name)) { ?>
43                             <a class="btn btn-default log-bar" href="/user/register.html">Sign up</a>
44                             <a class="btn btn-default log-bar" href="/user/login.html" role="button">Login</a>
45                         <?php } else { ?>
46                             <a class="btn btn-default log-bar" href="/uadmin/index.html">
47                                 <?php
48                                     if(!empty($username)) {
49                                         echo $username;
50                                     } else {
51                                         echo $name;
52                                     }
53                                 ?>
54                             </a><a class="btn btn-default log-bar" href="/user/logout.html" >logout</a>
55                         <?php } ?>
56
57                         <div class="cart_box_1">
58                             <a href="/uadmin/checkout.html">

```

Find in Project Close this panel with the **esc** key      Finding with Options: Case Insensitive

Find in project      Find      Replace All

Replace in project

File/directory pattern. eg. `'src'` to search in the "src" directory or `'*.js'` to search all javascript files.

CRLF    UTF-8    PHP

Figure 4

- Example of css page

```

index.php
1 h4, h5, h6,
2 h1, h2, h3 {margin: 0;}
3 ul, ol, li {margin: 0; padding: 0;}
4 p {margin: 0; padding: 0;}
5 body{
6     margin:0;
7     padding:0;
8 }
/*--fonts--*/
10 @font-face{
11     font-family: 'Didact Gothic';
12     src:url('../fonts/Didact_Gothic/DidactGothic.ttf) format('truetype');
13 }
14 @font-face{
15     font-family: 'Pathway Gothic One';
16     src:url('../fonts/Pathway_Gothic_One/PathwayGothicOne-Regular.ttf) format('truetype');
17 }
18 .header-top{
19     position: relative;
20     padding: 0;
21     text-align: center;
22 }
23 .login-bars{
24     position: absolute;
25     top: 0%;
26     right: 0%;
27 }

```

Find in Project Close this panel with the **esc** key      Finding with Options: Case Insensitive

Find in project      Find      Replace All

Replace in project

File/directory pattern. eg. `'src'` to search in the "src" directory or `'*.js'` to search all javascript files.

/Applications/MAMP/htdocs/asset/css/style.css 56:16      CRLF    UTF-8    CSS

Figure 5

- Example of JavaScript page

```

1 | jQuery.extend({
2 |   handleError: function( s, xhr, status, e ) {
3 |     // If a local callback was specified, fire it
4 |     if ( s.error ) {
5 |       s.error.call( s.context || s, xhr, status, e );
6 |     }
7 |
8 |     // Fire the global callback
9 |     if ( s.global ) {
10 |       (s.context ? jQuery(s.context) : jQuery.event).trigger( "ajaxError", [xhr, s, e] );
11 |     }
12 |   },
13 |
14 |   createUploadIframe: function(id, uri)
15 |   {
16 |     //create frame
17 |     var frameId = 'jUploadFrame' + id;
18 |
19 |     if(window.ActiveXObject) {
20 |       var io = document.createElement('<iframe id="' + frameId + '" name="' + frameId + '" />');
21 |       if(typeof uri=='boolean'){
22 |         io.src = 'javascript:false';
23 |       }
24 |       else if(typeof uri=='string'){
25 |         io.src = uri;
26 |       }
27 |     }
28 |   }
29 |
30 | });

```

Figure 6

### 2.3.2 Back-end Design

The back-end design is used in programming language: php. PHP is a server-side scripting language designed primarily for web development. In WB, PHP code may be embedded into HTML as well as separately in Controllers and Configuration.

On the one hand, the back-end system should receive the user action from interface. All functions are written in Controller, and every function is executed user instruction as a target. On the other hand, the php page of controller executes SQL to pass value in websites. For example, these codes are part of a function to view all books listing in homepage. The php function transits all book information from database: *eshop\_book* and show website: *bookview* page. The URL is redirected as *localhost*.

```

public function viewbook($viewid)
{
    if(empty($viewid)) {
        $this->load->helper('url');
        redirect('http://localhost/');
        exit;
    }
    $sql = "select * from eshop_book where id='".$viewid."'";
    //echo $sql;
    $query = $this->db->query($sql);
    $retdata = $query->result_array();
    $data['bookinfo'] = $retdata;
    $this->load->view('tpl/bookview',$data);

}

```

Figure 7

### 2.3.3 Connection

For the connection of each website, there is a one-to-one relationship between a URL and its corresponding controller class/method in route.php. The segments in a URL normally follow this pattern: example.com/class/method/id/

```
62 //用户操作
63 $route['user/register.html'] = 'User/dosignup';
64 $route['user/login.html'] = 'User/dologin';
65 $route['user/checklogin'] = 'User/checklogin';
66 $route['user/logout.html'] = 'User/logout';
67
68 $route['category/(:num)/(:num)'] = 'Home/cshow/$1/$2';
69
70
71
72 //管理员管理
73 $route['admin/index.html'] = 'Admin/index';
74 $route['admin/order.html'] = 'Admin/orderlist';
75 $route['admin/newbook.html'] = 'Admin/newbooklist';
76 $route['admin/usedbook.html'] = 'Admin/usedbooklist';
77
78 //用户管理
79 $route['uadmin/index.html'] = 'Profile/index';
80 $route['uadmin/usedsel.html'] = 'Profile/usedsel';
81 $route['uadmin/newsel.html'] = 'Profile/newsel';
82 $route['uadmin/order.html'] = 'Profile/order';
83 $route['uadmin/postusedbook.html'] = 'Profile/postbook';
84 //结算
85 $route['uadmin/checkout.html'] = 'User/checkout';
86
87 $route['api/upfile'] = 'Api/upfile';
88 $route['api/adduser'] = 'Api/adduser';
```

Figure 8

For the connection of websites and database, the framework CodeIgniter has provided an interface to connect database. The configuration is under database.php. This file contains the codes needed to access database.

Thus, in MySQL of MAMP, the database name of WB is *eshop*. And the database permission is opened to *sqltester*, with password *123*.

```

1 | // ...
2 | | the query builder class.
3 | */
4 | $active_group = 'default';
5 | $query_builder = TRUE;
6 |
7 | $db['default'] = array(
8 |     'dsn' => '',
9 |     'hostname' => 'localhost',
10 |     'username' => 'sqltester',
11 |     'password' => '123',
12 |     'database' => 'eshop',
13 |     'dbdriver' => 'mysqli',
14 |     'dbprefix' => 'eshop',
15 |     'pconnect' => TRUE,
16 |     'db_debug' => TRUE,
17 |     'cache_on' => FALSE,
18 |     'cachedir' => '',
19 |     'char_set' => 'utf8',
20 |     'dbcollat' => 'utf8_general_ci',
21 |     'swap_pre' => '',
22 |     'encrypt' => FALSE,
23 |     'compress' => FALSE,
24 |     'stricton' => FALSE,
25 |     'failover' => array(),
26 |     'save_queries' => TRUE
27 | );
28 |
29 | Find in Project Close this panel with the esc key
30 | Finding with Options: Case Insensitive
31 | Find in project Replace in project
32 | File/directory pattern. e.g. `src` to search in the "src" directory or `*.js` to search all javascript files.
33 | Find Replace All
34 | LF UTF-8 PHP
35 | config/database.php 8:70 (1, 66)

```

Figure 9

Also, this frame work of WB doesn't need to connect database again and again when needed. It just uses `$this->load->database()` in function of controller to set connection since all establishment configuration have saved in database.php and will be invoked once needed.

When selecting data or passing value in website from database, the function use `$query = $this->db->query` to get specific data from MySQL. For example, get the information of customer's shopping cart from database to show in website:

```

public function addtocart()
{
    $this->load->library('session');
    $this->load->helper('url');
    //var_dump($_POST);
    $id = $_POST['id'];
    $uname = $_POST['username'];
    $bookid = $_POST['bookid'];
    $query = $this->db->query("select * from eshop_cart where custid='$id'");
    $retdata = $query->result_array();

    $query1 = $this->db->query("select * from eshop_book where id='$bookid'");
    $retdata1 = $query1->result_array();

```

Figure 10

Additionally, it uses `$query=$this->db->insert()`, `$query=$this->db->update` to insert and update data.

### 3. Modeling Requirements

#### 3.1 Use Case Model

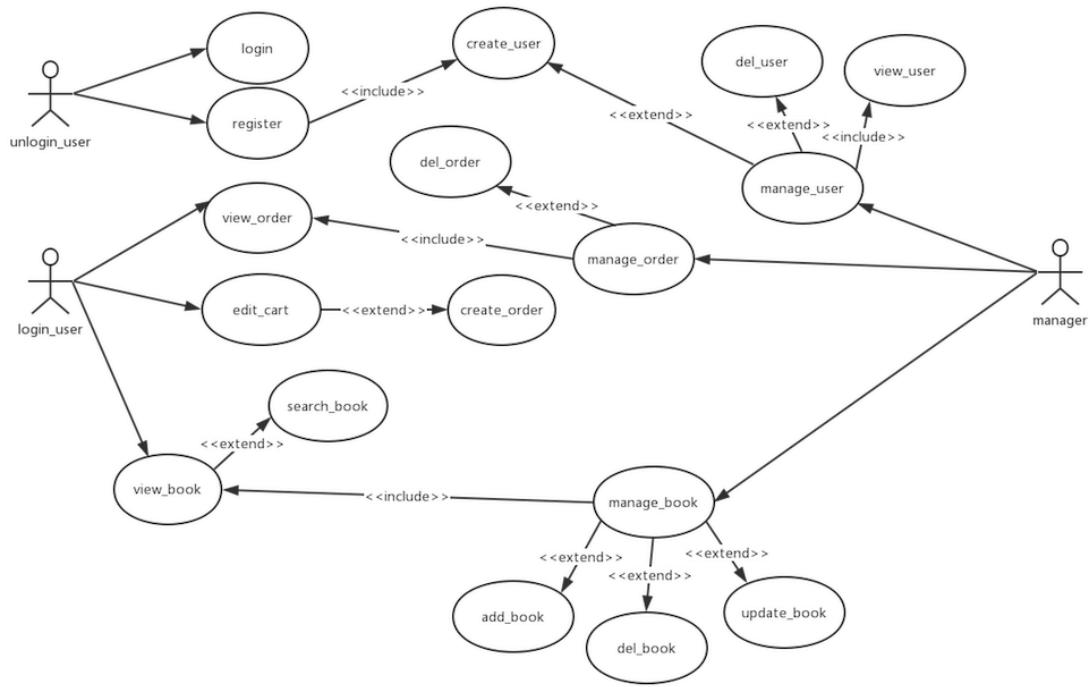


Figure 11

### 3.2 Function Diagram

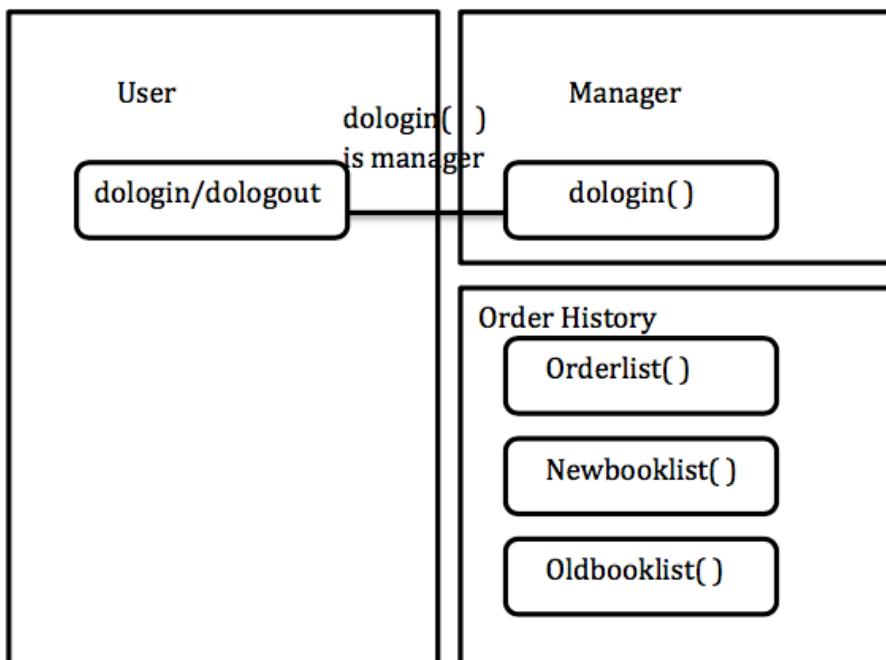


Figure 12

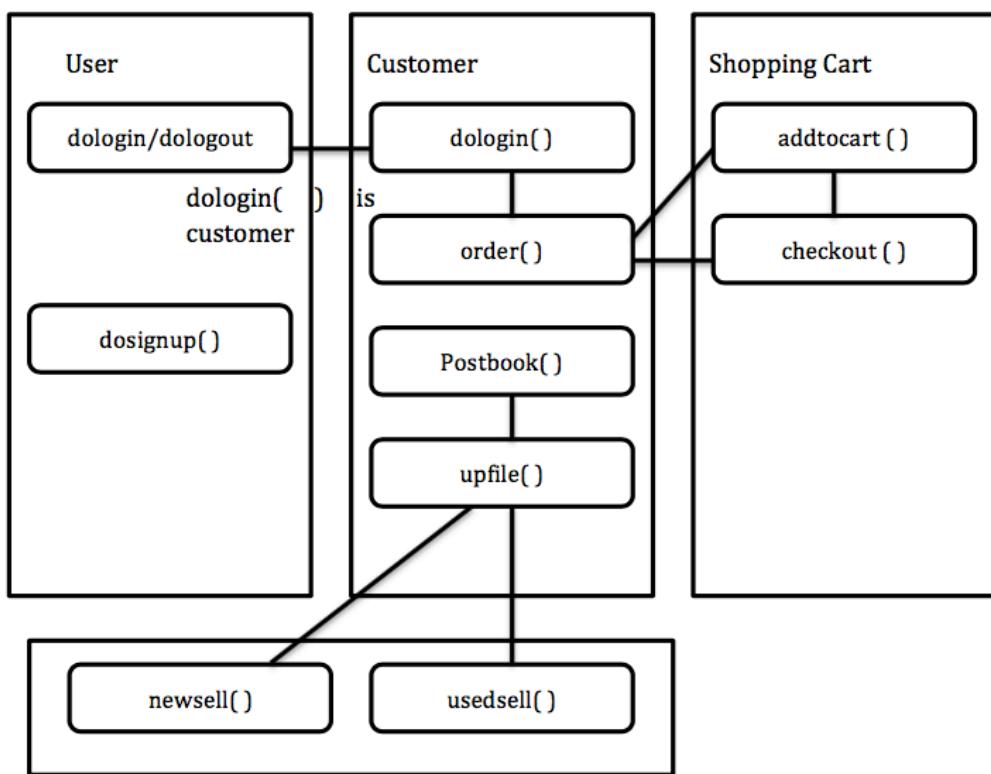


Figure 13

## 4. Prototype

### 4.1 Customer

#### 4.1.1 Homepage



#### 4.1.2 Login

A screenshot showing two side-by-side forms. On the left is the "Login" form, which includes fields for "User Name" (with placeholder "Input your name") and "Password" (with placeholder "Input password"). It also features a "Login" button and a "Forgot Password ?" link. On the right is the "New Registration" form, which includes a "Create An Account" button. Both forms are set against a background with a "HOME / LOGIN" link at the top and a "Sign Up" and "Login" button at the very top of the page.

### 4.1.3 Register

HOME / LOGIN / REGISTER

#### Register Now

Welcome, please enter the following details to continue.  
If you have previously registered with us, [click here](#)

User Name:

Password:

Cardnum:

Expire:

Barcode:

Tel Number:

Post Address:

User Email:

Register Now

By clicking this button, you are agree to my [Policy Terms and Conditions](#).

### 4.1.4 Personal Information

The screenshot shows a web application interface for managing user profiles. On the left, there is a vertical sidebar with navigation links: 'Home' (selected), 'User Profile', 'Selled Used Books' (with 'Used Books Info.'), 'Selling New Books' (with 'New Books Info.'), 'Post goods' (with 'Upload books'), and 'Check Out' (with 'Pay page'). The main content area displays four input fields for personal information: 'User name' (tang), 'Password' (redacted), 'Email' (sol13@pitt.edu), and 'Tel' (82899874). At the top right, there are 'Home' and 'Logout' buttons, and at the bottom right, there is a 'Copyright ©' notice.

### 4.1.5 Selling History

Home | Logout

Home  
User Profile.

Sold Used Books  
Used Books Info.

Purchased books  
Purchased Info.

Post goods/h4>  
upload books

Check Out  
Pay page

Used Books Info

id	title	type	price	author	stock
0	software	science	2200	sun	2
1	ddfs	comics	1300	fsdf	1

Whole Books  
All rights reserved.  
[please view my site](#)

#### 4.1.5 Purchasing History

Home | Logout

Home  
User Profile.

Sold Used Books  
Used Books Info.

**Purchased Books**  
Purchased Info.

Post goods  
Upload books

Check Out  
Pay page

Purchased books Info

id	title	type	price	author	stock
0	base	culture	12300	song tang	1
1	database	art	8800	tangsong	1
2	asd	culture	123	asd	1
3	asd	culture	osd	asd	2
4	4354	culture	4444	345	1
5	ddfs	comics	4700	fsdf	1
6	ddfs	comics	4700	fsdf	1

Whole Book  
All rights reserved.  
[please view my site](#)

#### 4.1.5 Post Selling Information

Home | Logout

Home  
User Profile.

Sold Used Books  
Used Books Info.

Purchased books  
Purchased Info.

**Post goods**  
Upload books

Check Out  
Pay page

title:

type:

author:

price:

contributor:

newold:  new  old

image:  No file chosen

Copyright ©

#### 4.1.5 Select Book Category



#### 4.1.5 Click Icon “Shopping Cart” to Order



DDFS

🛒 \$13

#### 4.1.5 View Details of Item

**WHOLE BOOKS**

[Sign Up](#) | [Login](#)  
 \$ cart: (0)

home    NewBooks▼    UsedBooks▼



#### 4.1.5 Check Out

Home  
User Profile.

Selled Used Books  
Used Books Info.

Purchased books  
Purchased Info.

Post goods  
Upload books

**Check Out**  
Pay page

shopping cart

ID	Title	Type	Price	Author	Contributor
0	database	art	88	tangsong	<input checked="" type="checkbox"/>

check out

Home | Logout

Whold Book  
All rights reserved.

[please view my site](#)

#### 4.2 Manager

##### 4.2.1 Manager Information

Home | Logout



User name

Password

Salary

Email

Tel

Whole Books  
All rights reserved.  
[please view my site](#)

#### 4.2.1 All Order History

Home | Logout

order list

order info						
Sequence	order ID	title	price	buyer	contributor	order date
1	20161107001	书本1	11.2	abner	admin	2016/11/17 21:58
2	20161107002	书本1	11.2	abner	admin	2016/11/17 21:58
3	20161107003	书本1	11.2	abner	admin	2016/11/17 21:58

Whole Books  
All rights reserved.  
[please view my site](#)

#### 4.2.1 All New Book List

Home | Logout

newbook list

new book info						
Sequence	order ID	title	price	buyer	contributor	order date
1	20161107001	书本1	11.2	abner	admin	2016/11/17 21:58
2	20161107002	书本1	11.2	abner	admin	2016/11/17 21:58
3	20161107003	书本1	11.2	abner	admin	2016/11/17 21:58

Whole Books  
All rights reserved.  
[please view my site](#)

#### 4.2.1 All Old Book List

The screenshot shows a web-based application interface. On the left, there is a vertical sidebar with a light gray background containing several menu items:

- Home
- Manager Profile.
- All Order Info
- Detail
- New Books Info
- Detail.
- Used Books Info**
- Detail.

The "Used Books Info" item is highlighted with a blue background and white text. To its right, the main content area has a title "usedbook list". Below the title is a table header "used book info" with columns: Sequence, order ID, title, price, buyer, contributor, and order date. Underneath the header is a table body containing three rows of data:

Sequence	order ID	title	price	buyer	contributor	order date
1	20161107001	书本1	11.2	abner	admin	2016/11/17 21:58
2	20161107002	书本1	11.2	abner	admin	2016/11/17 21:58
3	20161107003	书本1	11.2	abner	admin	2016/11/17 21:58

At the bottom of the table, there is a navigation bar with four buttons: "?", "1", "?", and "总共1页" (Total 1 page).

In the top right corner of the main content area, there is a small "Logout" button. In the top right corner of the entire window, there is a "Whole Books" logo with the text "All rights reserved." and a link "please view my site".

## 5. Test

Test is the necessary procedure in developing a system. There are different levels of component for component level. In WB, there are 7 components being tested which are Login, Operations (Admin), Operations (Customer), Add Item (for shopping), Check Out, Sell Item and Logout.

## 5.1 Test Design

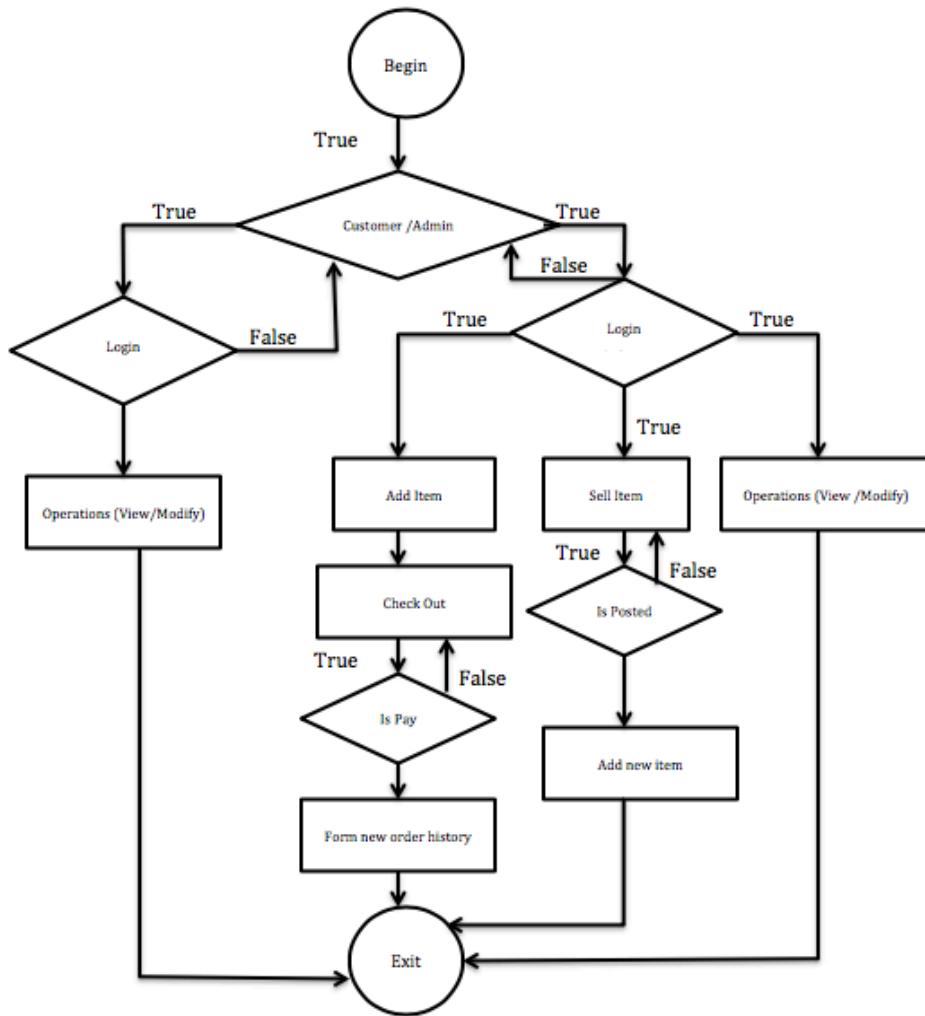


Figure 14

## 5.2 Test Driven Development

By doing the test path listing above, it is finished testing all the basic functions of WB without problems.

Then as for database, WB system is robust enough to store data from user.

## 6. Limitations

WB system has already satisfied the basic e-commerce functions. However, it still

exists some limitations. First, the manager doesn't have the access with full create, update and delete functionality with regards to all new or old books and orders in the system. Second, the customer cannot change the content of selling book's information once they have posted.

Those two limitations will improve in the future development.