

Reported by Wanen Gao

系统详细设计说明书

Completion Time: 2021.11.12

By Wanen Gao

1、系统功能概述

本平台参照 12306 选票系统设计，实现功能如下：

1. 用户注册功能
2. 用户登录功能
3. 用户修改信息功能
4. 用户注销账户功能
5. 用户查询列车余票功能
6. 用户选票功能
7. 用户退票功能

该工程部署在 web 服务器上，通过 Apache 实现。用户通过访问浏览器来实现余票查询功能。

2、系统模块结构

本项目的前端 html 页面主要采用 bootstrap 的组件进行设计。

2.1 前端功能

前端功能：

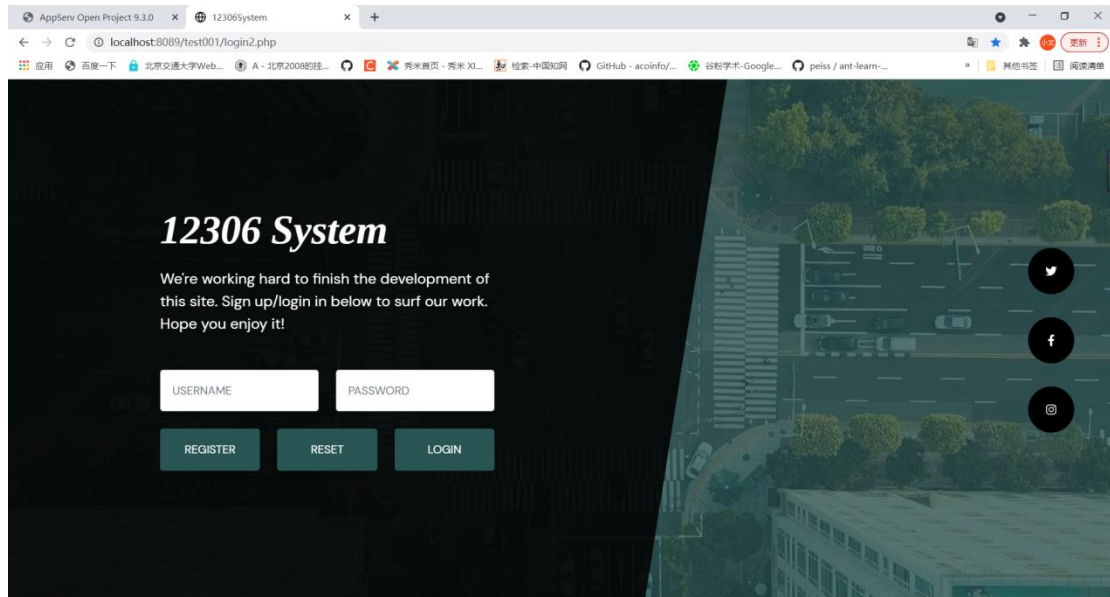
前端界面包括：登录界面、注册界面、系统首页界面、余票查询界面、购买通道界面、列车查询界面、我的订单界面、修改订单界面界面、我的信息界面。

2.1.1 登录界面

文件名称：test001/login2.php

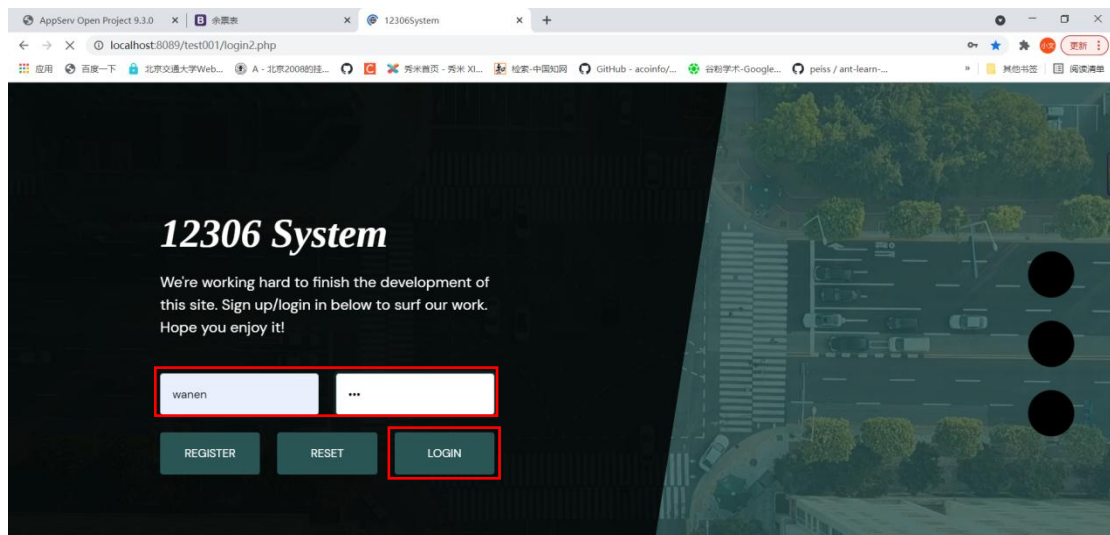
整体界面展示：

Reported by Wanen Gao



该界面设有注册、重置、登录功能。

输入正确的用户名和密码，然后点击“登录”，方能进入系统。

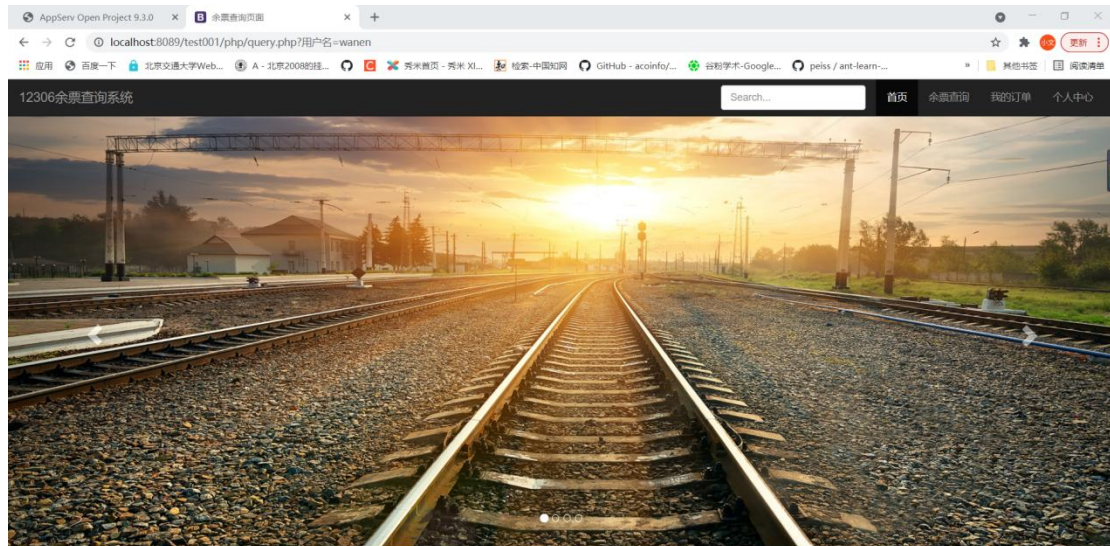


2.1.2 系统首页

文件名称：test001/php/query.php

整体界面展示：

Reported by Wanen Gao



该界面主要用于放置一些系统公告、通知等需要用户一眼注意到的信息。

2.1.3 导航功能



在每个界面的上栏区域都有四个界面的入口，鼠标点击即可打开相应界面，其中包括：

- 1、首页 -> 通知界面
- 2、余票查询 -> 余票查询界面
- 3、我的订单 -> 查询自己已经预定的订单
- 4、个人中心 -> 查询和修改个人信息，注销功能

2.1.4 查询界面

文件名称：test001/php/query_raw.php

界面展示：

Reported by Wanen Gao

AppServ Open Project 9.3.0

余票表

localhost:8089/test001/php/query_raw.php?用户名=wanen

更新

应用 百度一下 北京交通大学Web... A - 北京2008年的经... 秀半真页 - 秀半 XL... 检索-中国知网 GitHub - accinfo/... 谷粉学术-Google... peiss / ant-learn...

12306余票查询系统

Search...

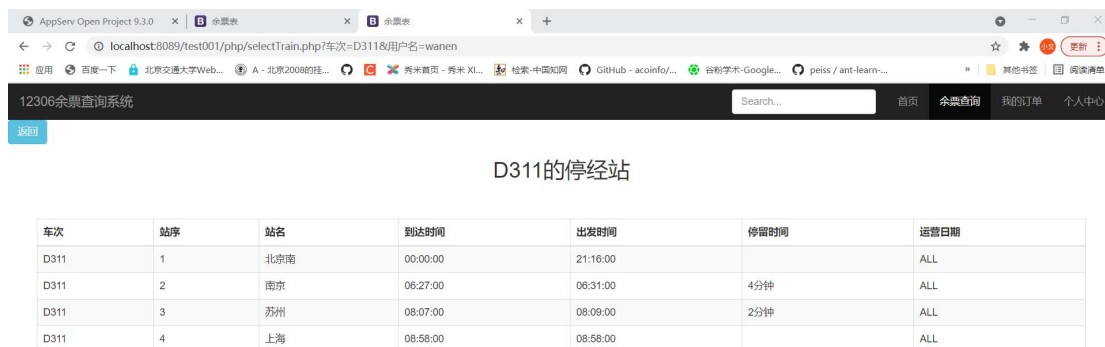
首页 余票查询 我的订单 个人中心

from 北京 to 上海 date: 2021-12-17 查询

添加订单	车次	出发站	到达站	起始时间	到达时间	历时	车型	商务座特等座余票	商务座特等座票价	一等座余票	一等座票价	二等座票价	二等座余票
ADD	D311	北京南	上海	21:16:00	08:58:00	11:42:00	D 3	495		238	264	165	29
ADD	D313	北京南	上海	19:34:00	07:27:00	11:53:00	D 36	720		170	384	240	228
ADD	D321	北京南	上海	21:23:00	09:12:00	11:49:00	D 226	693		170	369.6	231	140
ADD	G1	北京南	上海虹桥	09:00:00	13:48:00	04:48:00	G 138	552		16	294.4	184	141
ADD	G101	北京南	上海虹桥	07:00:00	12:37:00	05:37:00	G 146	2349		56	1252.8	783	89
ADD	G103	北京南	上海虹桥	07:05:00	12:42:00	05:37:00	G 105	2295		29	1224	765	322
ADD	G105	北京南	上海虹桥	07:36:00	13:21:00	05:45:00	G 140	2403		252	1281.6	801	37
ADD	G107	北京南	上海虹桥	08:05:00	13:26:00	05:21:00	G 75	1386		140	739.2	462	387
ADD	G109	北京南	上海虹桥	08:15:00	14:00:00	05:45:00	G 46	2184		222	1164.8	728	304
ADD	G11	北京南	上海虹桥	08:00:00	13:09:00	05:09:00	G 50	1395		175	744	465	80
ADD	G111	北京南	上海虹桥	08:35:00	14:21:00	05:46:00	G 294	1701		201	907.2	567	310
ADD	G113	北京南	上海虹桥	09:05:00	14:29:00	05:24:00	G 151	2016		65	1075.2	672	29
ADD	G115	北京南	上海虹桥	09:32:00	15:14:00	05:42:00	G 213	1224		109	652.8	408	274
ADD	G117	北京南	上海虹桥	09:43:00	15:41:00	05:58:00	G 34	1458		98	777.6	486	393

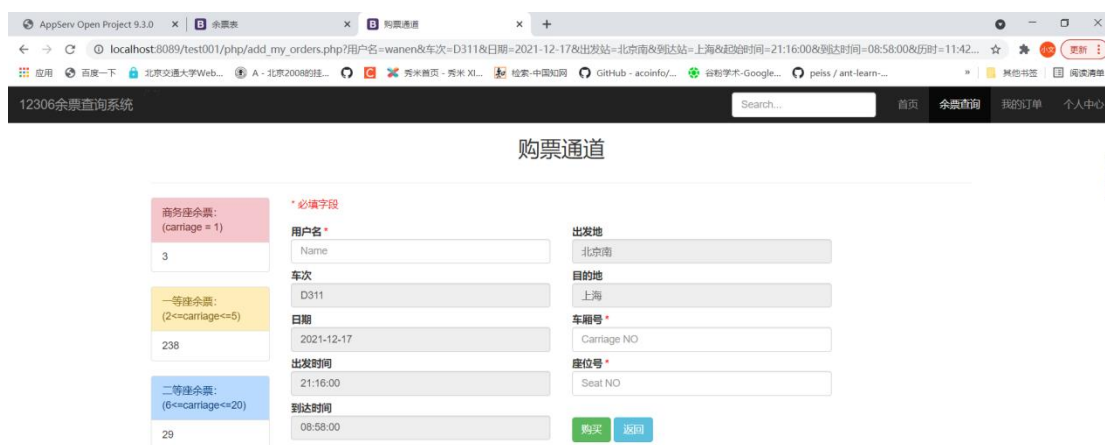
在输入框中输入出发城市、目的城市和日期，再点击查询，即可查询到所有的余票信息。

①当点击每一行中①的“车次”号，即可查询该车次的停经站。如下所示



车次	站序	站名	到达时间	出发时间	停留时间	运营日期
D311	1	北京南	00:00:00	21:16:00		ALL
D311	2	南京	06:27:00	06:31:00	4分钟	ALL
D311	3	苏州	08:07:00	08:09:00	2分钟	ALL
D311	4	上海	08:58:00	08:58:00		ALL

②当点击每一行中②的“ADD”键，即可将进入购票通道界面。如下所示



12306余票查询系统		购票通道
商务座余票: (carriage = 1)	3	用户姓名 * Name
一等座余票: (2 <= carriage <= 5)	238	车次 D311
二等座余票: (6 <= carriage <= 20)	29	日期 2021-12-17
		出发时间 21:16:00
		到达时间 08:58:00
		出发地 北京南
		目的地 上海
		车箱号 * Carriage NO
		座位号 * Seat NO
		购买 返回

在购票通道界面中，最左边三个框显示该趟列车的不同座次的余票数量。在中间

Reported by Wanen Gao

部分，用户名、车厢号、座位号是必填选项。用户名可以填自己的用户名，给自己买票；也可以填别的用户的用户名，给他人买票。其中，用户名会检查是否存在该用户名，否则，不能重新买票。同时，也需要填车厢号和座位号，但是当该趟车次的某个作为被重复选择时，系统则会报错。

当输入的用户名不存在时：

输入界面：

The screenshot shows a web browser window with the URL `localhost:8089/test001/php/add_my_orders.php`. The page title is "12306余票查询系统". The main heading is "购票通道". On the left, there are three sections for selecting ticket types: "商务座余票: (carriage = 1)" with value 210, "一等座余票: (2<=carriage<=5)" with value 71, and "二等座余票: (6<=carriage<=20)" with value 53. The main form contains fields for "用户名" (Username) with the value "iii" and a red error message "该用户名不存在", "出发地" (Origin) with "深圳", "目的地" (Destination) with "广州", "车次" (Train No.) with "C7002", "日期" (Date) with "2021-12-17", "出发时间" (Departure Time) with "07:40:00", "到达时间" (Arrival Time) with "09:06:00", "车厢号" (Carriage No.) with "3", and "座位号" (Seat No.) with "7". At the bottom right are "购买" (Buy) and "返回" (Return) buttons.

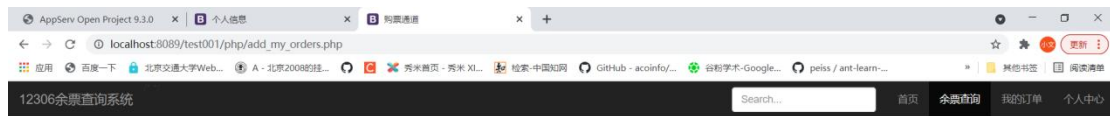
提交后显示：“该用户名不存在”

This screenshot is identical to the one above, showing the same ticket purchase form with the username validation error.

当选中的作为已经被其他用户选中时：

输入界面：

Reported by Wanen Gao

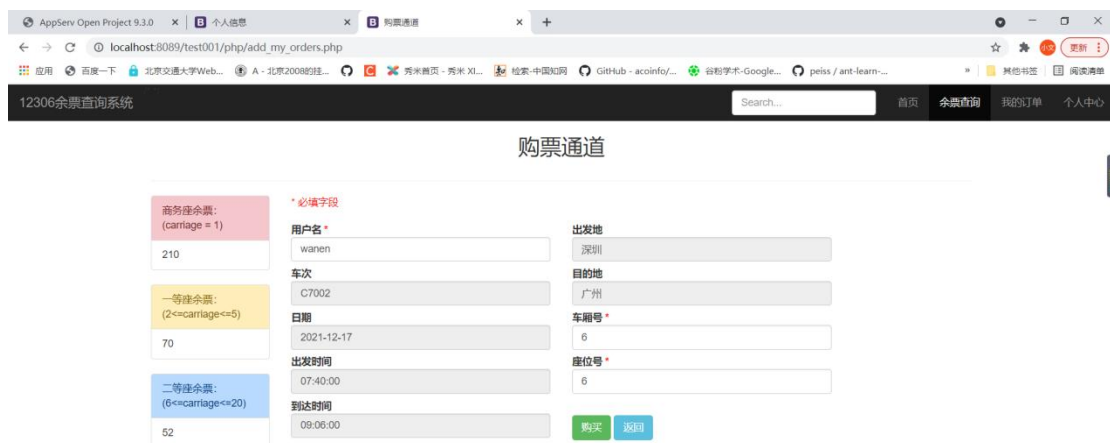


提交后提示：“当前用户不可选，请另选位置”



若当前用户以购买过该趟列车时：

输入界面：



系统显示：“该用户 已购买过这趟旅程的车票，一人只能购买同一趟旅程的一张票”。

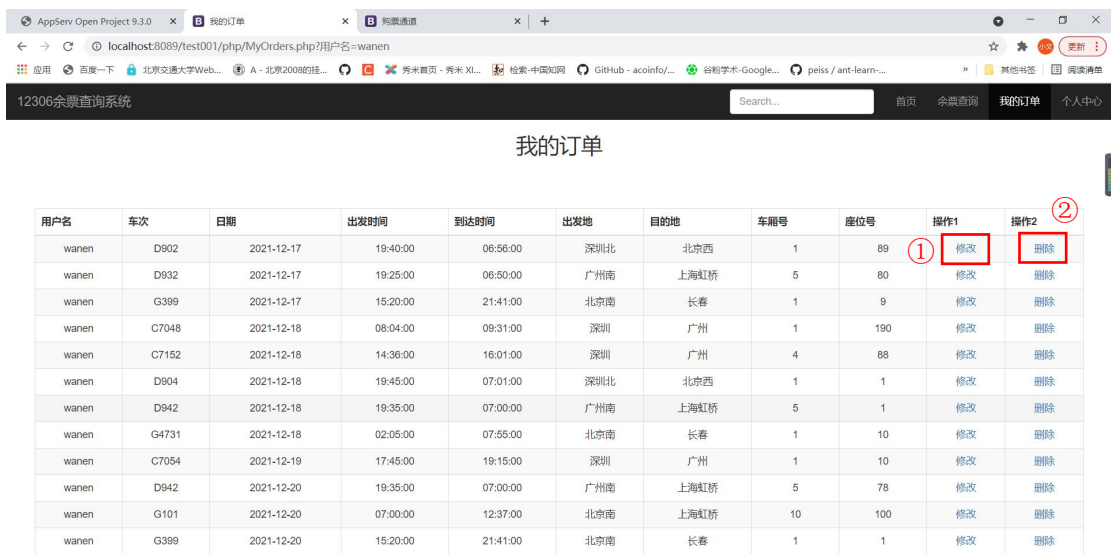
Reported by Wanen Gao



2.1.5 我的订单界面

文件名称：test001/php/MyOrders.php

界面展示：



在该界面中展示了当前用户的所有订单。

①：点击每一行中的“修改”，即可对进入修改界面，对该订单进行修改：

Reported by Wanen Gao

12306余票查询系统

修改订单信息

商务座余票:
(carriage = 1)

7

一等座余票:
(2<=carriage<=5)

2

二等座余票:
(6<=carriage<=20)

20

* 必填字段

用户名 *

wanen

车次

D902

日期

2021-12-17

出发时间

19:40:00

到达时间

06:56:00

出发地

深圳北

目的地

北京西

车厢号

1

座位号(座位号1~100) *

89

提交 返回

该界面最左边展示了当前车厢等级的余票，中间可以修改该订单的车厢号和座位号，其余列车信息用户不能修改。

②：点击每一行中的“删除”，即可将当前订单从订单表中删除掉。

2.1.6 注册界面

文件名称：test001/register2.php

整体界面展示：

REGISTER

用户名

手机号

用户密码

邮箱

确认密码

证件号

真实姓名

旅客类型

LOGIN RESET SUBMIT

在该界面中输入所有的注册信息。若注册信息有误或者该信息已被使用，则系统会出现错误提示。（如下）

Reported by Wanen Gao



若信息没有填完时，系统显示“请填写此字段”，如下：



A screenshot of a registration form titled 'REGISTER'. The form has two columns of input fields. The first column contains '用户名' (Username) with value 'wanen', '用户密码' (User Password) with three asterisks, '确认密码' (Confirm Password) with three asterisks, and '真实姓名' (Real Name) with value '高婉恩'. The second column contains '手机号' (Mobile Number), '邮箱' (Email), '证件号' (ID Number), and '旅客类型' (Passenger Type) with a dropdown menu. A red 'LOGIN' button, an orange 'RESET' button, and a green 'SUBMIT' button are at the bottom. A yellow tooltip with a red exclamation mark and the text '请填写此字段。' (Please fill in this field.) points to the '邮箱' field.

当需要注册的用户名已存在时：

输入界面如下：



A screenshot of a registration form titled 'REGISTER'. The form has two columns of input fields. The first column contains '用户名' (Username) with value 'wanen', '用户密码' (User Password) with three asterisks, '确认密码' (Confirm Password) with three asterisks, and '真实姓名' (Real Name) with value '高婉恩'. The second column contains '手机号' (Mobile Number) with value '13590619688', '邮箱' (Email) with value '1922222@bjlu.edu.cn', '证件号' (ID Number) with value '123232323', and '旅客类型' (Passenger Type) with a dropdown menu showing '儿童'. A red 'LOGIN' button, an orange 'RESET' button, and a green 'SUBMIT' button are at the bottom.

Reported by Wanen Gao

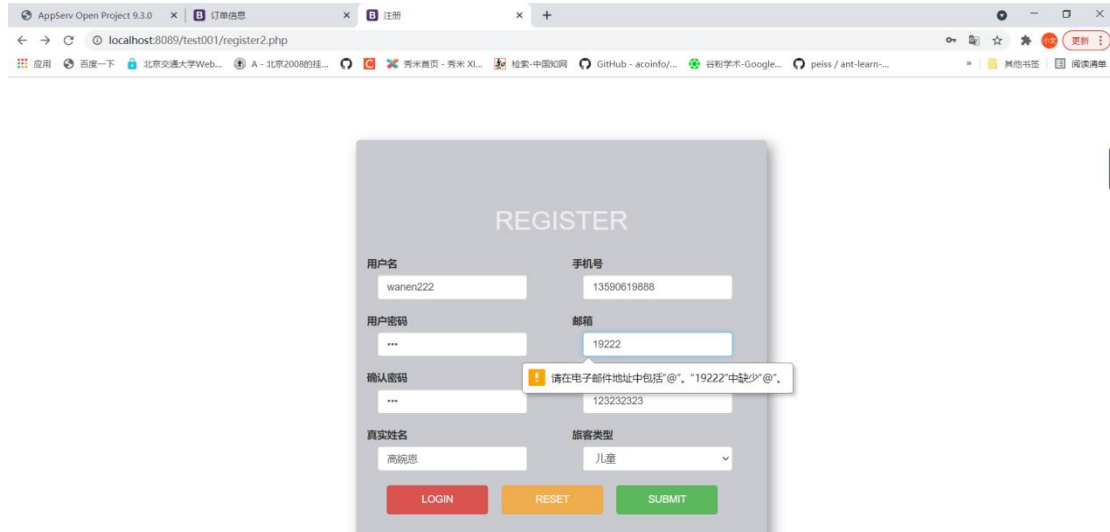
系统显示如下：



当两次输入的密码不一样时，系统显示如下：



当填入邮箱处缺少”@”时，系统显示如下：



当所有信息填写正确时，系统显示“注册成功”

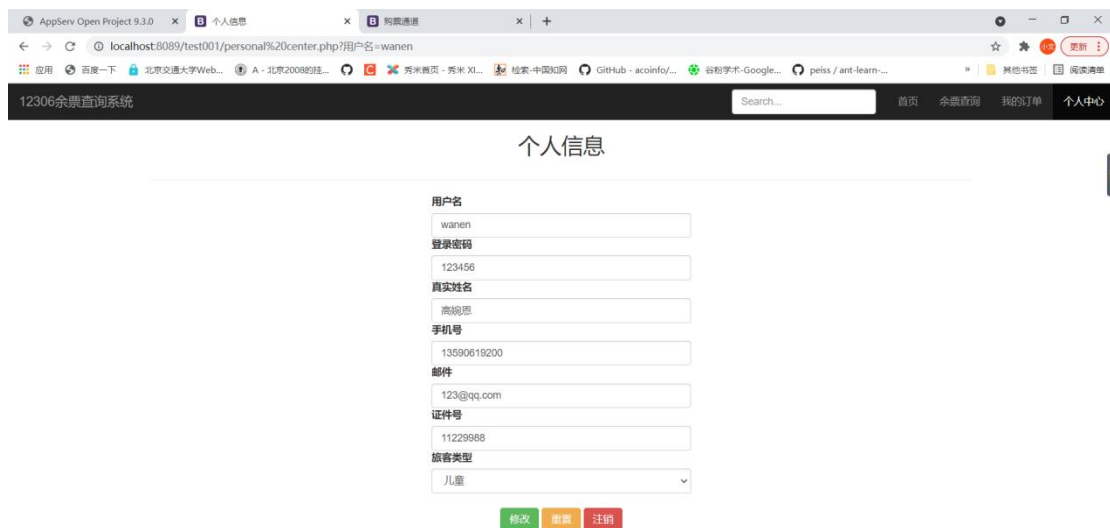
Reported by Wanen Gao



2.1.7 个人中心界面

文件名称: test001/personal center.php

页面展示:



功能: 该界面展示了当前用户的所有个人信息, 同时该界面可以对用户信息进行修改、注销。

对个人信息进行修改:

若修改后的用户名已被其他用户使用:

当前 wanen 改为 bingyan:

系统显示:

Reported by Wanen Gao



结果：修改失败

若当前手机号已被使用时：

当前 13590619200 改为 13590619213

系统显示：



结果：修改失败

若当前证件号已被使用时：

当前 11229988 改为 12 时：

系统显示：



结果：修改失败。

2.2 WEB 服务器

服务器介绍:

本项目使用的后端语言为 php, 后端框架为 Apache, 通过 AppServ 软件来搭建本地服务器。

使用的 AppServ 为 9.3.0 版本, 其中, Apache 为 2.4.41 版本, PHP 为 7.3.10 版本, MySQL 为 8.0.17 版本, phpMyAdmin 为 4.9.1 版本。

AppServ 是一个软件集合, 集成了 Apache(服务器软件), php(网页程序设计语言), Mysql(数据库系统管理软件), 和 phpMyAdmin(数据库图形界面管理软件)。使用它能够快速在本地搭建测试服务器, 省去了许多繁琐的安装过程。

Apacheweb 服务器软件拥有以下特性:

- 1.支持最新的 HTTP/1.1 通信协议
- 2.拥有简单而强有力的基于文件的配置过程
- 3.支持通用网关接口
- 4.支持基于 IP 和基于域名的虚拟主机
- 5.支持多种方式的 HTTP 认证
- 6.集成 Perl 处理模块
- 7.集成代理服务模块
- 8.支持实时监视服务器状态和定制服务器日志
- 9.支持服务器端包含指令(SSl)
- 10.支持安全 Socket 层(SSL)
- 11.提供用户会话过程的跟踪
- 12.支持 FastCGI
- 13.通过第三方模块可以支持 JavaServlets

函数功能:

1、查询功能实现:

用户点击查询按钮提交查询信息之后, 后端会从 request 中接收浏览器发送的 GET 请求, 根据请求中的信息读取到出发地, 到达地与出发时间, 之后调用如下的存储过程从数据中查询

```
1.  $host      = 'localhost';
2.      $username = 'root';
3.      $password = false;
4.      $dbname   = '12306system2';
5.      $link     = @mysqli_connect($host,$username,$password,$dbname);
        // 连接到数据库
6.
7.      if (!empty($start) && !empty($end) && !empty($date)){
8.          if($link){
9.              mysqli_set_charset($link,'UTF-8');        // 设置数据库字符集
10.             $sql    = " CALL SELECTN('".$start."','".$end."','".$date."')
                ;";        // SQL 语句
11.             $result = mysqli_query($link, $sql);        // 执行 SQL 语句, 并返回结果
12.             $data   = mysqli_fetch_all($result);        // 从结果集中获取所有数据
13.             $num    = mysqli_num_rows($result);
14.             mysqli_close($link);
15.         }else{
16.             die('数据库连接失败! ');
17.         }
18.     }
```

2、订票功能实现:

用户点击订票按钮之后, 后端会接收到浏览器发送的订票信息, 之后后端调用数据库执行插入语句, 将订单信息插入到用户信息表中

```
1.  $host      = 'localhost';
2.      $username = 'root';
3.      $password = false;
4.      $dbname   = '12306system2';
5.      $link     = @mysqli_connect($host,$username,$password,$dbname);
        // 连接到数据库
```


Reported by Wanen Gao

```
6.
7.     if (!empty($start) && !empty($end) && !empty(date)){
8.         if($link){
9.             mysqli_set_charset($link,'UTF-8');          // 设置数据库字符集
10.            $sql = "insert into orders values ('".$other."', '". $_POST["train"]."', '". $_POST["date"]."', '". $_POST["time1"]."', '". $_POST["time2"]."', '". $_POST["src"]."', '". $_POST["dest"]."', '". $_POST["carriage"]."', '". $_POST["seat"]."');";
11.            $result = mysqli_query($link, $sql);          // 执行 SQL 语句, 并返回结果
12.            $data    = mysqli_fetch_all($result);         // 从结果集中获取所有数据
13.            $num     = mysqli_num_rows($result);
14.            mysqli_close($link);
15.        }else{
16.            die('数据库连接失败! ');
17.        }
18.    }
```

同时会调用更新余票表的存储过程使得相应的座位票数减一。

```
1.     if ($_POST["carriage"] == 1)
2.     {
3.         $sql = "update remainall set 商务座特等座余票=商务座特等座余票-1 where 车次= '". $_POST["train"]."' and 日期= '". $_POST["date"]."' and 出发站 = '". $_POST["src"]."' and 到达站 = '". $_POST["dest"]."'";";
4.         $link      = @mysqli_connect($host,$username,$password,$dbname);
           // 连接到数据库
5.         $result = mysqli_query($link, $sql);
6.         $data    = mysqli_fetch_all($result);         // 从结果集中获取所有数据
7.     }
8.     else if ($_POST["carriage"] >1 && $_POST["carriage"] <6)
9.     {
10.        $sql = "update remainall set 一等座余票=一等座余票-1 where 车次= '". $_POST["train"]."' and 日期= '". $_POST["date"]."' and 出发站 = '". $_POST["src"]."' and 到达站= '". $_POST["dest"]."'";";
11.        $link      = @mysqli_connect($host,$username,$password,$dbname);
           // 连接到数据库
12.        $result = mysqli_query($link, $sql);
13.        $data    = mysqli_fetch_all($result);         // 从结果集中获取所有数据
14.    }
15.    else{
```

Reported by Wanen Gao

```
16.      $sql = "update remainall set 二等座余票=二等座余票-1 where 车次
      = '$_POST['train']."' and 日期= '$_POST['date']."' and 出发
      站 = '$_POST['src']."' and 到达站= '$_POST['dest']."'";
17.      $link      = @mysqli_connect($host,$username,$password,$dbname);
      // 连接到数据库
18.      $result = mysqli_query($link, $sql);
19.      $data    = mysqli_fetch_all($result);    // 从结果集中获取所有数
      据
20.  }
21.
```

3、异常判断:

当用户产生以下情况时, 会发生冲突, 此时后端界面会捕获异常返回提示信息。

- ①用户名没有填写时
- ②用户名不存在时
- ③车厢号没有填写时
- ④车厢号不在 0~20 范围内时
- ⑤座位号没有填写时
- ⑥座位号不在 1~100 范围内时

判断代码如下所示:

```
1.  $link      = @mysqli_connect($host,$username,$password,$dbname);    //
    连接到数据库
2.  mysqli_set_charset($link,'UTF-8');
3.  $sql       = "select * from users where 用户
    名 = '$_POST['name']."'";      // SQL 语句
4.  $result = mysqli_query($link, $sql);    // 执行 SQL 语句, 并返回结果
5.  $data    = mysqli_fetch_all($result);  // 从结果集中获取所有数据
6.  $row      = mysqli_num_rows($result);
7.  mysqli_close($link);
8.  if (empty($_POST['name']))
9.  {
10.     $nameErr = "用户名是必须的";
11.  }
12.  else if (empty($row) ){
13.     $nameErr = "该用户名不存在";
14.  }
15.  else if (empty($_POST['carriage']))
```

Reported by Wanen Gao

```
16.  {
17.    $carriageErr = "车厢号是必须的";
18.  }
19.  else if ($_POST["carriage"] > 20 || $_POST["carriage"] < 0)
20.  {
21.    $carriageErr = "请选择 1~20 号车厢.车厢规格详见最左侧列表";
22.  }
23.  else if (empty($_POST["seat"]))
24.  {
25.    $seatErr = "座位号是必须的";
26.  }
27.
28.  else if ($_POST["seat"] > 100 || $_POST["seat"] <= 0)
29.  {
30.    $seatErr = "请选择 1~100 号座位";
31.  }
32.  else
33.  {
34.    $link      = @mysqli_connect($host,$username,$password,$dbname);
        // 连接到数据库
35.    $sql = "select * from orders where 车次
= '$_POST["train"].' and 日期= '$_POST["date"].' and 车厢号
= '$_POST["carriage"].' and 座位号= '$_POST["seat"].'";
36.    $result = mysqli_query($link, $sql);    // 执行 SQL 语句，并返回结
        果
37.    $num    = mysqli_num_rows($result);
38.    $link    = @mysqli_connect($host,$username,$password,$dbname);
        // 连接到数据库
39.    $sql = "select * from orders where 用户名= '$_other.'" and 车次
= '$_POST["train"].' and 日期= '$_POST["date"].' and 出发
地 = '$_POST["src"].' and 目的地= '$_POST["dest"].'";
40.
41.    $result = mysqli_query($link, $sql);    // 执行 SQL 语句，并返回结
        果
42.    $data    = mysqli_fetch_all($result);    // 从结果集中获取所有数据
43.    $num3    = mysqli_num_rows($result);
44.    // location.href='$_SERVER["HTTP_SELF"].'";
45.    if ($num3 != 0){
46.        echo "<script type='text/javascript'>
47.            alert('{$_other} 已购买过这趟旅程的车票，一人只能购买同一趟旅程的
            一张票');
48.
49.            </script>"
50.    }
```

4、登录功能

用户在点击登录之后，程序会执行如下语句，调用 login 函数：

```
1.  $res = login($username, $passwd);
```

login 函数的内容如下所示：

```
1.  function login($username, $password) {
2.      $link = db_connect();
3.      mysqli_set_charset($link, 'UTF-8');    // 设置数据库字符集
4.      // $sql = "select * from users where 用户名
        = '$username.'" and 登录密码 = sha1('$password.')";
5.      $sql = "call login('$username.', '$password.')";
6.      $result = mysqli_query($link, $sql);    // 执行 SQL 语句，并返回结果
7.      $data    = mysqli_fetch_all($result);    // 从结果集中获取所有数据
8.      mysqli_close($link);
9.      if (!$result) {
10.         return false;
11.     }
12.     if ($result->num_rows>0) {
13.         return true;
14.     } else {
15.         return false;
16.     }
17. }
```

功能介绍：

用户在前端界面点击登录之后后端会接收到用户名与密码，之后会调用查询语句在数据库中进行查询，如果用户存在且密码正确会正确登录，否则会返回相应的错误信息。

5、注册功能响应

后端代码：

```
1.  $link    = @mysqli_connect($host,$username,$password,$dbname);    /
    / 连接到数据库
```

Reported by Wanen Gao

```
2. $sql = "CALL ADDUSER('".$name."', '".$password0."', '".$trueName."',  
   ".$mobile.", '".$email."', '".$idNo."', '".$type.'');";  
3. $result = mysqli_query($link, $sql);    // 执行 SQL 语句, 并返回结果  
4. $data   = mysqli_fetch_all($result);    // 从结果集中获取所有数据  
5. echo "<script>alert('注册成功! ')</script>";
```

功能介绍:

当用户提交注册信息并且经过程序检测通过之后会传递到后端, 后端代码调用注册存储过程 CALL ADDUSER('".\$name."', '".\$password0."', '".\$trueName."', ".\$mobile.", ".\$email.", ".\$idNo.", ".\$type.") 将用户信息插入到用户表中, 如果用户名发生重复会返回错误提示信息。

6、修改信息功能响应:

当接收到前端提交的修改信息后会调用更新信息存储过程,

```
1. $sql = "CALL ALTERUSER('".$name."', '".$password0."', '".$trueName."',  
   ".$mobile.", '".$email."', '".$idNo."', '".$type."', '".$oldname.'');  
   ";
```

将用户信息更新。

7、注销功能响应:

后端代码:

```
1. $sql = "CALL DELETEUSER('".$id.'');";
```

功能介绍:

当用户点击注销按钮时, 后端会调用删除用户的存储过程将用户信息删除, 同时会激活触发器将订单表中此用户的订单清空。

8、个人订单查询响应

后端代码:

```
1. $sql = "select * from orders where 用户名 = '".$id.'";    // SQL 语  
   句
```

功能介绍：

当用户点击导航栏中的“我的订单”，后端就会执行以上代码，在订单表中查询当前用户的所有订单。

9、改票功能

后端代码：

```
1. $sql = "update orders set 出发时间='".$time1.':00', 到达时间
   = '".$time2.':00', 出发地='".$src."', 目的地='".$dest."', 车厢号
   = ".$carriage.", 座位号=".$seat." where 用户名='".$id.'" and 车次
   = '".$train.'" and 日期='".$date.'"; // SQL 语句
```

功能介绍：

当用户点击“修改”订单时，后端程序就会执行以上 update 语句，修改该订单。

10、退票功能

后端代码：

```
1. $sql = "delete from orders where 用户名 = '".$id.'" and 车
   次 = '".$train.'" and 日期 = '".$time.'" and 出发地 = '".$src.'" and 目
   的地 = '".$dest.'"; // SQL 语句
```

功能介绍：

将当前订单删除。

2.3 数据库端

数据库表内容展示：

订单表：

Reported by Wanen Gao

用户名	车次	日期	出发时间	到达时间	出发地	目的地	车厢号	座位号
鲍萱伟	C1024	2021-10-13	18:55:00	20:15:00	安图西	吉林	3	20
陆山	C1019	2021-10-18	13:30:00	15:06:00	长春	敦化	80	17
陶晴	C1005	2021-11-02	21:49:00	23:49:00	北京西	上海虹桥	2	3
齐曼	C1011	2021-11-09	21:47:00	20:47:00	上海虹桥	广州南	8	33
齐曼	C1011	2021-11-22	15:55:00	17:58:00	上海虹桥	广州南	14	2
高高	C1003	2021-11-27	07:12:00	08:12:00	吉林	敦化	6	9
齐曼	C7002	2021-12-17	07:40:00	09:06:00	深圳	广州	1	7
贺鸿	C7002	2021-12-17	07:40:00	09:06:00	深圳	广州	1	11
陶晴	C7002	2021-12-17	07:40:00	09:06:00	深圳	广州	1	19
bingyan	C7002	2021-12-17	07:40:00	09:06:00	深圳	广州	4	1
wanen	C7002	2021-12-17	07:40:00	09:06:00	深圳	广州	7	7

余票表:

车次	日期	出发站	到达站	商务座特等座余票	商务座特等座票价	一等座余票	一等座票价	二等座余票	二等座票价
C1001	2021-12-17	吉林	延吉西	212	546	35	291.2	110	182
C1001	2021-12-17	吉林	敦化	14	273	39	145.6	102	91
C1001	2021-12-17	敦化	延吉西	110	273	200	145.6	202	91
C1001	2021-12-17	长春	吉林	291	273	35	145.6	126	91
C1001	2021-12-17	长春	延吉西	225	819	70	436.8	24	273
C1001	2021-12-17	长春	敦化	204	546	297	291.2	362	182
C1001	2021-12-18	吉林	延吉西	87	546	14	291.2	194	182
C1001	2021-12-18	吉林	敦化	247	273	160	145.6	179	91
C1001	2021-12-18	敦化	延吉西	8	273	283	145.6	282	91
C1001	2021-12-18	长春	吉林	9	273	114	145.6	187	91
C1001	2021-12-18	长春	延吉西	95	819	149	436.8	122	273
C1001	2021-12-18	长春	敦化	30	546	178	291.2	10	182
C1001	2021-12-19	吉林	延吉西	141	546	43	291.2	371	182
C1001	2021-12-19	吉林	敦化	234	273	148	145.6	168	91
C1001	2021-12-19	敦化	延吉西	213	273	242	145.6	343	91

列车表:

Reported by Wanen Gao

车次	出发站	到达站	发车时间	时间	车型	到达时间
C1001	长春	延吉西	05:47:00	02:17:00	D	08:04:00
C1002	延吉西	长春	06:20:00	02:05:00	D	08:25:00
C1003	长春	珲春	06:30:00	03:10:00	D	09:40:00
C1004	延吉西	长春	07:42:00	02:17:00	D	09:59:00
C1005	长春	延吉西	06:52:00	02:17:00	D	09:09:00
C1006	延吉西	长春	10:01:00	02:23:00	D	12:24:00
C1007	长春	延吉西	07:30:00	02:11:00	D	09:41:00
C1008	珲春	长春	10:00:00	03:05:00	D	13:05:00
C1009	长春	珲春	08:20:00	02:59:00	D	11:19:00
C1010	延吉西	长春	11:25:00	02:17:00	D	13:42:00

列车经停站表:

车次	站序	站名	到达时间	出发时间	停留时间	运营日期
C1001	1	长春	00:00:00	05:47:00	NULL	all
C1001	2	吉林	06:27:00	06:29:00	2分钟	ALL
C1001	3	敦化	07:23:00	07:25:00	2分钟	ALL
C1001	4	延吉西	08:04:00	08:04:00		ALL
C1002	1	延吉西	00:00:00	06:20:00		ALL
C1002	2	长春	08:25:00	08:25:00		ALL
C1002	3	蛟河西	07:26:00	07:27:00	1分钟	ALL
C1002	4	长春	08:30:00	08:30:00		ALL
C1003	1	长春	00:00:00	06:30:00		ALL
C1003	2	吉林	07:10:00	07:12:00	2分钟	ALL

用户表:

Reported by Wanen Gao

用户名	登录密码	姓名	手机号	邮箱	证件号	旅客类型
bingyan	123456	黄冰妍	18800192203	192811111@bjtu.edu.c	12345678	成人
wanen	123	高婉恩	13590619200	123@qq.com	11229988	儿童
wanen222	111	高婉恩	13590619888	19222@qq.com	123232323	儿童
乐鹤尔	100308	诸千	17647153016	13678937804@163.com	1003867777	成人
云立柏	100607	方坚安	15628134080	15001905510@163.com	100843	成人
何敏以	100165	孟枫	17627337578	13643210093@163.com	100252	儿童
何浩康	100350	郎南	13647495917	18252637601@163.com	100771	儿童
俞霖	100153	庞群伟	17616624524	15084740892@163.com	100489	成人
倪凌伟	100537	成强	15627811127	18201676344@163.com	100627	成人
元烨从	100766	强珍	17640083851	13619433808@163.com	100171	成人
凤沛	100715	华以博	15683342984	13688488866@163.com	100230	儿童

存储过程：

1) 单表查询：

实现登录：

```
1. DELIMITER // -- 声明存储过程的结束符
2. CREATE PROCEDURE LOGIN(in username varchar(10), in passwd varchar(
  20))
3. BEGIN
4. select * from users where 用户名=username and 登录密码 = passwd;
5. END //
```

多表查询：

实现余票查询

```
1. DELIMITER // -- 声明存储过程的结束符
2. CREATE PROCEDURE SELECTN(in start varchar(10), in end varchar(10))
  -- 存储过程的名称
3.
4. BEGIN -- 开始
5. -- sql 语句+流程控制
6. create table starttable as select 站名 from stop where 城
  市 = start;
7. create table endtable as select 站名 from stop where 城市 = end;
```

Reported by Wanen Gao

```
8.      select * from trains,starttable,endtable
9.      where starttable.站名 = trains.出发站 and endtable.站名 = trains.
      到达站;
10.     drop table starttable;
11.     drop table endtable;
12.     END //
13.
14. DELIMITER ;
```

2) 数据插入

添加用户:

```
1.  -- 添加用户
2.  DELIMITER //
3.  CREATE PROCEDURE ADDUSER(in 用户名 varchar(10), in 登陆密
      码 varchar(20), in 姓名 varchar(20), in 手机号 bigint(20), in 邮
      箱 varchar(20), in 证件号 varchar(20), in 旅客类型 varchar(20))
4.  BEGIN
5.      insert into users values ( 用户名, 登陆密码, 姓名, 手机号, 邮箱, 证
      件号, 旅客类型);
6.  END //
7.  DELIMITER ;
8.  -- 执行存储过程
```

3) 数据删除

注销用户

```
1.  DELIMITER //
2.  CREATE PROCEDURE DELETEUSER(in 用户名 0 varchar(10))
3.  BEGIN
4.      delete from users where 用户名=用户名 0;
5.  END//
6.  DELIMITER ;
```

4) 数据修改

修改用户信息

```
1.  -- 修改用户信息
2.  DELIMITER //
```

Reported by Wanen Gao

```
3. CREATE PROCEDURE ALTERUSER(in 用户名 0 varchar(10), in 登陆密码
0 varchar(20), in 姓名 0 varchar(20), in 手机号 0 bigint(20), in 邮箱
0 varchar(20), in 证件号 0 varchar(20), in 旅客类型 0 varchar(20))
4. BEGIN
5. update users set 登录密码=登陆密码 0, 姓名=姓名 0, 手机号=手机号 0, 邮箱=邮箱
0, 证件号=证件号 0, 旅客类型=旅客类型 0 where 用户名=用户名 0;
6. END //
7. DELIMITER ;
```

触发器：

1、数据插入

```
1. DELIMITER //
2. create trigger user_insert_orders
3. before insert on orders
4. for each row
5. begin
6. declare a char(20);
7. select count(*) into a from users where 用户名=new.用户名;
8. if a = 0
9. then insert into users(用户名, 登录密码, 姓名, 手机号, 邮箱, 证件号, 旅客类
型) values
10. (new.用户名, '111', new.用户名, 13825551919, '111@qq.com', 654321, 成人);
11. end if;
12. end//
```

功能：

每当在订单表里增加一个信息之前都要检查一下在用户表中是否存在该用户，
否则将该用户的信息插入到用户表中

2. 数据删除

```
1. -- 删除users 中用户时，同时删除orders 表中的用户订单
2. DELIMITER //
3. create trigger user_delete_orders
4. after delete on users
5. for each row
6. begin
7. delete from orders where old.用户名=用户名;
```

```
8. end //
9. DELIMITER ;
```

功能:

将用户表中数据删除之后需要将订单表中相应的信息清空

3. 数据更新

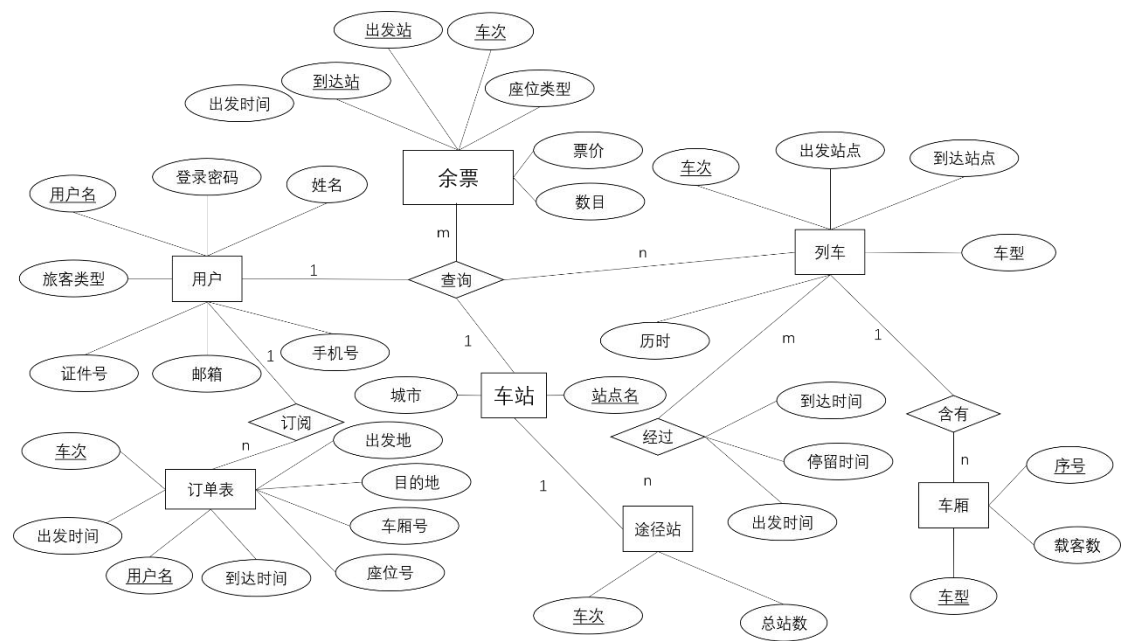
```
1. -- 数据更新(当用户更新用户名时, 新用户名已经有, 则在新用户名后添加编号)
2. DELIMITER //
3. create trigger users
4. before update on users
5. for each row
6. begin
7. declare a int;
8. select count(*) into a from users where 用户名=new.用户名;
9. if a!=0 and old.用户名 != new.用户名 then set new.用户名=concat(new.用
   户名,a);
10. end if;
11. end//
12. DELIMITER ;
```

功能:

当用户更改用户名与用户信息表中已有用户名相同时, 为其尾部增加编号, 同时监测时候名称新表与旧表相同, 增加鲁棒性。

3、系统物理模型

ER 图:



3.1 表

列车表:

表名		trains			
数据库用户		root			
主键		车次			
外键		出发站、到达站			
排序字段		车次			
索引字段		车次			
字段名称	数据类型	允许为空	唯一	默认值	约束条件
车次	Varchar(10)	N	Y	null	主键
出发站	Varchar(10)	N	N	null	外键
到达站	Varchar(10)	N	N	null	外键
历时	time	N	N	null	

Reported by Wanen Gao

车型	varchar	N	N	D	外键
----	---------	---	---	---	----

其他附加说明:

1) 时间信息的数据类型包括:

time data datetime timestamp year

time 类型数据为 h:m:s 符合历时要求, 因而这里的历时选择了 time 类型

2) 车型信息包括

C (城际) D (动车) G (高铁)

车站表:

表名	stop				
数据库用户	root				
主键	站名				
外键	无				
排序字段	站名				
索引字段	站名、城市				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
站名	Varchar(10)	N	Y	null	主键
城市	Varchar(10)	N	N	null	

附加说明:

1) 本表主要用于查询车站于城市之间的对应关系, 用户可能输入城市名, 我们需要将其转化为站点名

途径站表:

表名	passstop				
数据库用户	root				
主键	车次, 站序, 运营日期				

Reported by Wanen Gao

外键		无			
排序字段		车次			
索引字段		车次			
字段名称	数据类型	允许为空	唯一	默认值	约束条件
车次	Varchar(10)	N	Y	null	主键/外键
站序	int	N	N	null	主键
站点	Varchar(10)	N	N	null	外键
到达时间	time	N	N	null	
出发时间	time	N	N	null	
运营日期	date	N	N	null	主键

设置途径站表是为了方便的将列车途径站显示给用户，方便用户查询规划自己的路线

余票表：

表名		remain			
数据库用户		root			
主键		车次、出发站、到达站			
外键		车次、出发站、到达站			
排序字段		车次			
索引字段		车次、出发站、到达站			
字段名称	数据类型	允许为空	唯一	默认值	约束条件
车次	Varchar(10)	N	Y	null	主键/外键
出发站	Varchar(10)	N	N	null	主键/外键

Reported by Wanen Gao

到达站	Varchar(10)	N	N	null	主键/外键
商务座特等 座余票	int	N	N	null	
商务座特等 座票价	float	N	N	null	
一等座余票	int	N	N	null	
一等座票价	float	N	N	null	
二等座余票	int	N	N	null	
二等座票价	float	N	N	null	

用户表：

表名	users				
数据库用户	root				
主键	用户名				
外键	无				
排序字段	用户名				
索引字段	用户名				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
用户名	Varchar(10)	N	Y	null	主键
登陆密码	Varchar(20)	N	N	null	
姓名	Varchar(20)	N	N	null	
手机号	bigint	N	Y	null	
邮箱	Varchar(20)	N	Y	null	

Reported by Wanen Gao

证件号	Varchar(20)	N	N	null	
旅客类型	Varchar(20)	N	N	成人	

其他附加说明:

1)旅客类型的取值范围为 (儿童, 成人, 学生)

2)由于用户名, 手机号, 邮箱号都是唯一的, 其都为候选码, 这里我们选择用户名作为主键

订单表:

表名	orders				
数据库用户	root				
主键	用户名, 车次, 日期				
外键	车次、出发地、目的地				
排序字段	用户名				
索引字段	用户名、日期				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
用户名	Varchar(10)	N	Y	null	主键/外键
车次	Varchar(10)	N	N	null	主键/外键
日期	date	N	N	null	主键
出发日期	time	N	N	null	
到达日期	time	N	N	null	
出发地	Varchar(10)	N	N	null	外键
目的地	Varchar(10)	N	N	null	外键
车厢号	int	N	N	null	

Reported by Wanen Gao

座位号	int	N	N	null	
-----	-----	---	---	------	--

附加说明：

对于订单而言，一个用户可能有多个订单，可以通过车次号与用户名组合作为主键来区分

车厢表：

表名	carriage				
数据库用户	root				
主键	(车型, 序号)				
外键	无				
排序字段	车型				
索引字段	无				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
车型	varchar	N	N	1	主键
序号	int	N	N	1	主键
载客数	varchar	N	N	100	

三种完整性约束验证

约束一：实体完整性约束，用户表的主属性用户名，列车表的主键车次等不能为空。

尝试插入空值到具有非空约束的用户表的用户名属性：

```
1. insert into orders
2. values (null, 'C1024', '2021-10-10', '18:55:00', '20:15:00', '安图西', '吉林', 3, 30);
```

出现报错:

ERROR 1048 (23000): Column '用户名' cannot be null

约束二: 参照完整性约束, 外键的对应关系, 订单表中的外键用户名必须在用户表中对应用户名。

尝试插入不存在的用户到订单表中:

```
1. mysql> insert into orders
2.      -> values ('ggg','C1024','2021-10-10','18:55:00','20:15:00','安图西','吉林',3,30);
```

出现报错:

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`12306system2`.`orders`, CONSTRAINT `orders_FK1` FOREIGN KEY (`用户名`) REFERENCES `users` (`用户名`))

约束三: 用户定义完整性, 由列车订单的性质可知, 订单的目的地不可能为空。

尝试插入空值到订单的目的地中:

```
1. mysql> insert into orders
2.      -> values ('鲍萱伟','C1024','2021-10-10','18:55:00','20:15:00','安图西',null,3,30);
```

出现报错:

ERROR 1048 (23000): Column '目的地' cannot be null

4、系统安全体系设计

1) 用户口令采用密文存储，采用不可逆加密算法加密，查资料了解现有的不可逆算法。

不可逆加密算法介绍：

不可逆加密算法的特征是加密过程中不需要使用密钥，输入明文后由系统直接经过加密算法处理成密文，这种加密后的数据是无法被解密的，只有重新输入明文，并再次经过同样不可逆的加密算法处理，得到相同的加密密文并被系统重新识别后，才能真正解密。

(1) MD5(Message Digest algorithm 5, 信息摘要算法)

MD5 是计算机广泛使用的的杂凑算法之一（又译摘要算法、哈希算法），广泛应用于加密和解密技术，常用于文件校验。不管校验文件有多大，经过 MD5 后都能生成唯一的 MD5 值。

MD5 算法具有如下几个特点：

- ①压缩性：任意长度的数据，算出的 MD5 值长度都是固定的。
- ②容易计算：从原数据计算出 MD5 值很容易。
- ③抗修改性：对原数据进行任何改动，哪怕只修改 1 个字节，所得到的 MD5 值都有很大区别。
- ④弱抗碰撞：已知原数据和其 MD5 值，想找到一个具有相同 MD5 值的数据（即伪造数据）是非常困难的。
- ⑤强抗碰撞：想找到两个不同的数据，使它们具有相同的 MD5 值，是非常困难的。因此，基于 MD5 算法的特性，可知用该算法对密码加密是很有效的。

(2) SHA(Secure Hash Algorithm, 安全散列算法)

SHA 算法主要适用于数字签名标准（Digital Signature Standard DSS）里面定义的数字签名算法（Digital Signature Algorithm DSA）。该算法的思想是接收一段明文，然后以一种不可逆的方式将它转换成一段（通常更小）密文，并把它们转化为长度较短、位数固定的输出序列即散列值（也称为信息摘要或信息认证代码）的过程。

2) 设计用户修改密码流程，用 VISO 画出流程图

用户修改密码的流程：

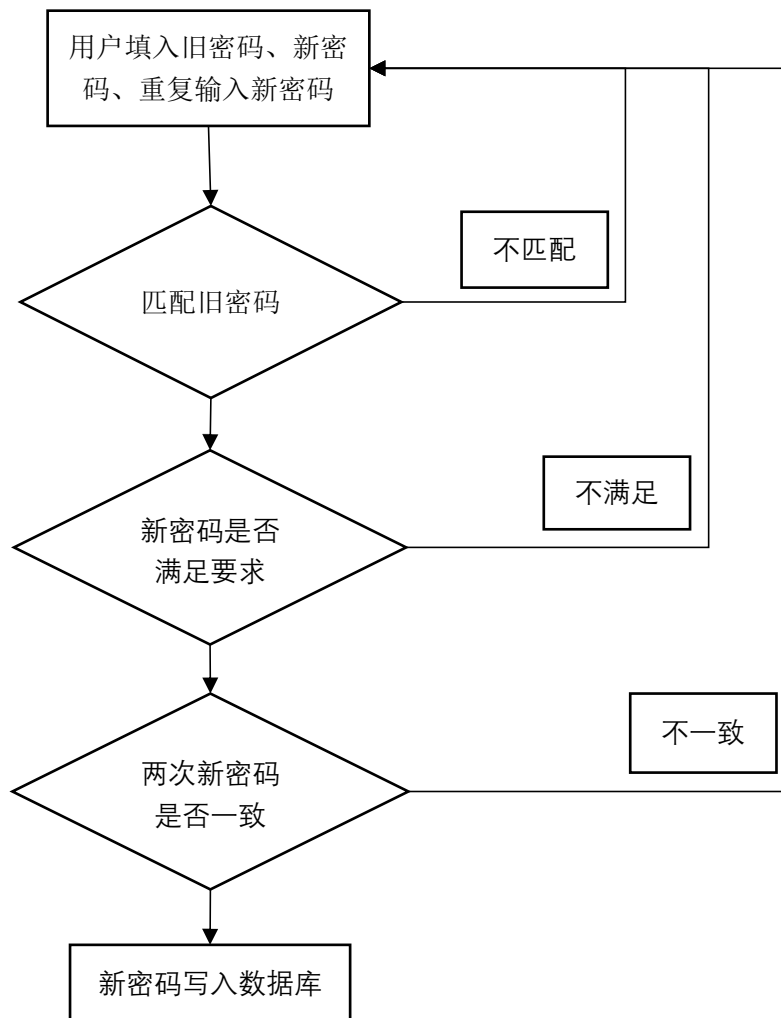
首先用户需要通过旧密码登录系统，再进入修改密码服务。

1.填入旧密码；

Reported by Wanen Gao

2. 填入新密码，重复确认新密码（要求：6-16 位的字母和数字的组合，密码区分大小写）
3. 若输入的旧密码加密后与数据库中存储的内容不符合，则提示：修改密码错误，重新输入；
4. 若新密码不符合密码约束性要求，提示输入不合要求，重新输入；
5. 匹配两次输入的新密码，若不匹配，提示两次输入不一致
6. 若数据准确无误，则更新数据库，提示：密码修改成功！

修改密码流程图：



并发机制：

1. MySQL 锁类型

在 MySQL 中主要是通过"读写锁"来实现并发控制。

****读锁(read lock): ****也叫共享锁(share lock)，多个读请求可以同时共享一把锁来读取数据，而不会造成阻塞。

****写锁(write lock): ****也叫排他锁(exclusive lock)，写锁会排斥其他所有获取锁的请求，一直阻塞，直到完成写入并释放锁。

读写锁可以做到读读并行，但是无法做到写读、写写并行。

2. MySQL 锁粒度

上面提及的读写锁是根据 MySQL 的锁类型来划分的，而读写锁能够施加的粒度在数据库中主要体现为表和行，也称为**表锁(table lock)**、**行锁(row lock)**。

表锁(table lock): 是 MySQL 中最基本的锁策略，它会锁定整张表，这样维护锁的开销最小，但是会降低表的读写效率。如果一个用户通过表锁来实现对表的写操作(插入、删除、更新)，那么先需要获得锁定该表的写锁，那么在这种情况下，其他用户对该表的读写都会被阻塞。一般情况下"alter table"之类的语句才会使用表锁。

行锁(row lock): 行锁可以最大程度地支持并发读写，但数据库维护锁的开销会比较大。行锁是我们日常使用最多的锁策略，一般情况下 MySQL 中的行级锁由具体的存储引擎实现，而不是 MySQL 服务器层面去实现(表锁 MySQL 服务器层面会实现)。




3.多版本并发控制(MVCC)

MVCC(MultiVersion Concurrency Control)，多版本并发控制。在 MySQL 的大多数事务引擎(如 InnoDB)中，都不只是简单地实现了行级锁，否则会出现这样的情况："数据 A 被某个用户更新期间(获取行级写锁),其他用户读取该条数据(获取读锁)都会被阻塞"。但现实情况显然不是这样，这是因为 MySQL 的存储引擎基于提升并发性能的考虑，通过 MVCC 数据多版本控制，做到了读写分离，从而实现不加锁读取数据进而做到了读写并行。

5.2 存储与恢复

故障恢复：

1. 通过数据库备份技术将数据库保存，确保了数据库端的安全。

 12306系统备份	2021/12/18 17:06	SQL 源文件	70,272 KB
 users结构及数据备份	2021/12/18 16:55	SQL 源文件	18 KB
 users数据备份	2021/12/18 16:54	SQL 源文件	17 KB

- 2.在 cmd 界面恢复系统：

如下图：

```
D:\appServ\MySQL\bin>mysqldump --no-defaults -u root -p --databases 12306system2 > 12306系统备份.sql
Enter password:
```

```
1.  mysqldump --no-defaults -u root -p --databases 12306system2 > 12306
系统备份.sql
```

5、系统运行环境设计与部署结构

系统运行环境设计：

前端：使用 **html css js** 编写，可在任何支持以上三种语言的浏览器中运行

后端：使用 **php** 语言编写，嵌入到 **html** 中

服务器：

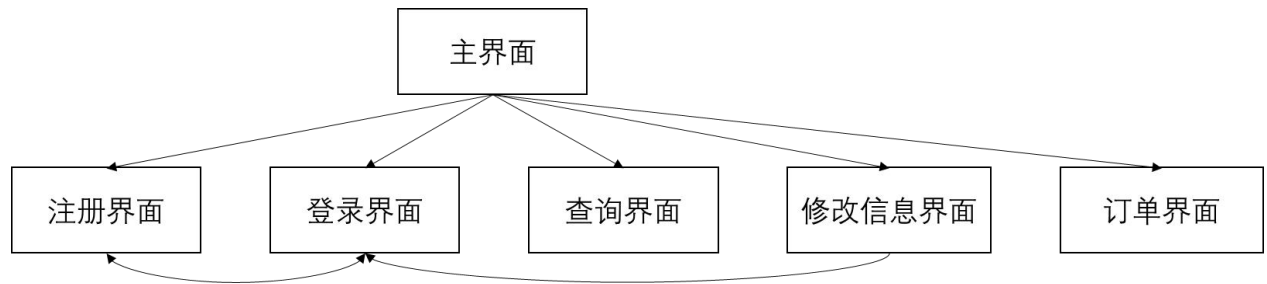
本项目使用的后端语言为 **php**，后端框架为 **Apache**，通过 **AppServ** 软件来搭建本地服务器。

使用的 **AppServ** 为 9.3.0 版本，其中，**Apache** 为 2.4.41 版本，**PHP** 为 7.3.10 版本，**MySQL** 为 8.0.17 版本，**phpMyAdmin** 为 4.9.1 版本。

数据库：

MYSQL 数据库

部署结构:



6、源代码列表及说明

本项目的前后端代码都写在了所有的.php 文件中。

test001 文件夹中:














此电脑 > 本地磁盘 (D:) > appServ > www > test001 >						
名称	修改日期	类型	大小			
.idea	2021/12/30 21:41	文件夹				
.ipynb_checkpoints	2021/12/16 23:58	文件夹				
.vs	2021/11/27 20:07	文件夹				
.vscode	2021/11/15 17:26	文件夹				
assets	2021/12/27 17:30	文件夹				
css	2021/12/27 17:30	文件夹				
fonts	2021/11/12 8:40	文件夹				
img	2021/12/27 22:23	文件夹				
js	2021/12/27 17:30	文件夹				
php	2021/12/28 10:43	文件夹				
addSuccess	2021/11/23 22:09	PHP 源文件	3 KB			
addUser	2021/12/27 12:36	PHP 源文件	2 KB			
alterUser	2021/12/28 1:20	PHP 源文件	5 KB			
deleteUser	2021/12/27 22:35	PHP 源文件	1 KB			
functions	2021/12/31 15:01	PHP 源文件	3 KB			
index	2021/12/1 23:40	PHP 源文件	2 KB			
login2	2021/12/31 14:57	PHP 源文件	8 KB			
login2_raw	2021/12/27 18:49	PHP 源文件	2 KB			
personal center	2021/12/28 0:40	PHP 源文件	10 KB			
register2	2021/12/27 21:17	PHP 源文件	18 KB			
register2_raw	2021/12/28 10:33	PHP 源文件	4 KB			
test	2021/12/27 22:00	PHP 源文件	2 KB			

其中, css、js、img 为 bootstrap 框架, 里面分别包括.css 文件、.js 文件、和图片信息。

Reported by Wanen Gao

在网页中打开 login2.php 界面即可打开登录界面。

test001/php 文件夹中：

此电脑 > 本地磁盘 (D:) > appServ > www > test001 > php					搜索"php"
名称	修改日期	类型	大小		
 add_my_orders	2021/12/28 10:21	PHP 源文件	22 KB		
 MyOrders	2021/12/27 16:20	PHP 源文件	7 KB		
 orders	2021/12/28 10:42	PHP 源文件	3 KB		
 orders_add	2021/12/26 0:34	PHP 源文件	2 KB		
 orders_add_raw	2021/11/27 20:16	PHP 源文件	2 KB		
 orders_alter	2021/12/27 17:05	PHP 源文件	25 KB		
 orders_alter_manager	2021/12/28 10:55	PHP 源文件	24 KB		
 orders_alter_raw	2021/11/12 22:48	PHP 源文件	2 KB		
 orders_del	2021/12/27 17:03	PHP 源文件	3 KB		
 orders_find	2021/12/15 23:59	PHP 源文件	4 KB		
 query	2021/12/27 22:30	PHP 源文件	7 KB		
 query_raw	2021/12/28 1:25	PHP 源文件	8 KB		
 selectTrain	2021/12/30 22:48	PHP 源文件	6 KB		