

# ACM Class-TicketSystem 使用手册

Dirty Deeds Done Dirt Cheap, Love Train!

## 0 简介

本项目为 ACM 班 CS147 数据结构课程大作业。

项目可分解为前端部分与后端部分。

其中，前端部分基于 Python 的 Flask 框架与 Jinja2 渲染模板开发。前端部分架构详见下文的**管理人员需知部分**。

后端部分为 C++ 编写的，借助 BPlusTree 实现数据管理的一个火车票逻辑系统。其已通过严格的正确性检验并且打包为 `code` 可执行文件，如无必要请勿修改。

**如果您为此系统的普通用户**，请跳过 **1、2 部分**，从 **3 部分** 及以后获得你想要的相关使用教程指导。

**如果您为此系统的管理人员**，且需要获得项目部署、维护等相关信息，请仔细阅读 **1、2 部分**。

## 1 前端文件与维护指南

正常途径获取本项目的部署包应该是一个 `FrontEnd` 文件夹，结构如下：

```
├── FrontEnd
│   ├── static
│   │   ├── css
│   │   ├── doc
│   │   ├── fonts
│   │   ├── images
│   │   └── js
│   ├── templates
│   │   ├── index.html
│   │   └── result.html
│   ├── code
│   ├── app.py
│   └── core.py
```

其中 `static` 为资源文件，其中子文件的内容较多所以在下面详述：

### CSS

```
bootstrap.css
bootstrap-select.css
creditly.css
easy-responsive-tabs.css
flowChart.css
font-awesome.css
```

```
jquery.seat-charts.css  
jquery-ui.css  
layout.css  
style.css  
xtiper.css  
xui.css
```

## doc

帮助文档路径位置

## fonts

```
fontawesome-webfont  
glyphicons-halflings-regular  
FontAwesome  
Montserrat-Regular  
OpenSans-Regular  
Oxygen-Regular
```

## images

```
1.jpg & 2.jpg & 3.jpg  
favicon.ico  
top-arrow.png  
train.jpg
```

## js

```
bootstrap.js  
bootstrap-select.js  
contact_me.js  
creditly.js  
dynamic_timeline.js  
easing.js  
easyResponsiveTabs.js  
jqBootstrapValidation.js  
jquery.seat-charts.js  
jquery-2.2.3.min.js  
jquery-ui.js  
move-top.js  
responsiveslides.min.js  
xtiper.min.js
```

可能出现的问题？

- 网页显示异常：请仔细查看两个 `html` 文件，其中 `index.html` 为主页面，`result.html` 为所有操作结果的呈现页，通过 Jinja2 传入参数 `query_name` 来选择性渲染。
- 数据丢失、异常：请仔细检查本地的数据文件是否损坏。`code` 将会在它的目录下生成对应的数据文件，格式均为 `*.bin`。此外，本系统还有设置登陆状态的 `cookies`，这点也需多多注意。
- 文件丢失：请重新下载完整部署包。
- 其它可能的异常：本系统在你的服务器终端会输出比较详细的调试信息，以及各个命令的格式化显示，可以借助这个更好地进行维护。

为了管理员更方便地维护，我们提供了一个类似后端部分的**命令行终端**。格式化命令详情参见 README.md。

# ACM Class Ticket System

👤 用户信息

👤 欢迎您, 奥斯卡

🔒 注销

🚆 列车信息

🚶 出行助手

🛒 订单管理

</> 控制台

</>

欢迎使用终端控制台，请输入对应的格式化命令

add\_user -c cur -u I\_am\_the\_admin -p aws! -n 奥斯卡 -m foo@bar.com -g 10

执行

離れ離れの街を  
将四散分离的街道

繋ぐ列車は行ってしまったねを  
连接起来的列车开走了呢

失くした言葉を知らないなら  
如果不知道那些失去了的言语

ポケットで握りしめて  
就在口袋里握紧双手吧

あがいた息を捨てて  
舍弃拼命挣扎的呼吸

延びる今日は眠って誤魔化せ  
动摇的今天就这样睡过去吧

失くした言葉を知らないなら  
如果不知道那些失去了的言语

各駅停車に乗り込んで  
就乘坐上每站停靠的列车吧

ラグトレイン(延误列车), 福葉曇

点击“执行”后会展示运行结果，呈现方式为一个仿 Linux 终端界面。

 / 控制台命令

## 命令执行结果



```
Ticket System Terminal,  
A simple terminal for testing this stupid system.  
ACM Class, PingPong Doubles Smash Locomotive, 2021.  
  
<<< query_profile -c I_am_the_admin -u I_am_the_admin  
>>>  
  
I_am_the_admin 奥斯卡 foo@bar.com 10  
  
...  
Complete.Please check the output.
```

## 2 项目部署指南

首先，你需要获得上述的完整部署用文件夹 **FontEnd**。

- 安装 Flask 包

```
python3 -m pip install flask
```

详细可查看 Flask 官方文档 <https://dormousehole.readthedocs.io/en/latest/>

- 本地启动

在 **FontEnd** 文件夹下，运行

```
python3 app.py
```

部署成功

```
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
```

- 进行测试

初始化系统时，会自动创建一个权限为 10 的超级管理员，初始化信息在 `core.py` 内修改：

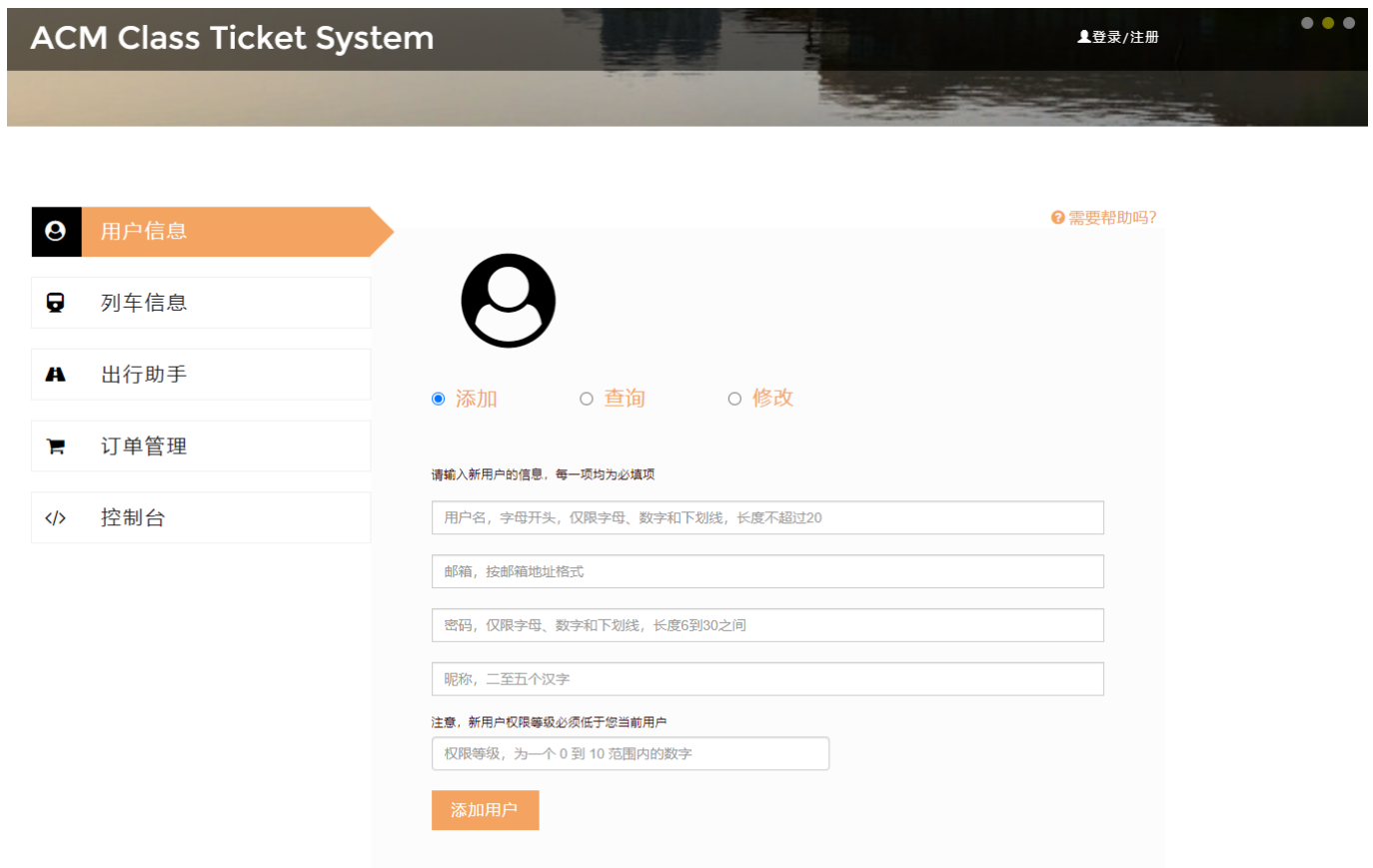
```
courier.add_user("root", "root", "root", "root", "root", "10")
```

### 3 用户相关部分

当你需要查询、管理用户信息时，请访问此页面。

接下来的部分为普通用户使用该系统的指南。

通常，第一次访问时你应该是未登录状态，界面大概如下：



ACM Class Ticket System

登录/注册

需要帮助吗?

用户信息

列车信息

出行助手

订单管理

控制台

添加 查询 修改

请输入新用户的信息，每一项均为必填项

用户名，字母开头，仅限字母、数字和下划线，长度不超过20

邮箱，按邮箱地址格式

密码，仅限字母、数字和下划线，长度6到30之间

昵称，二至五个汉字

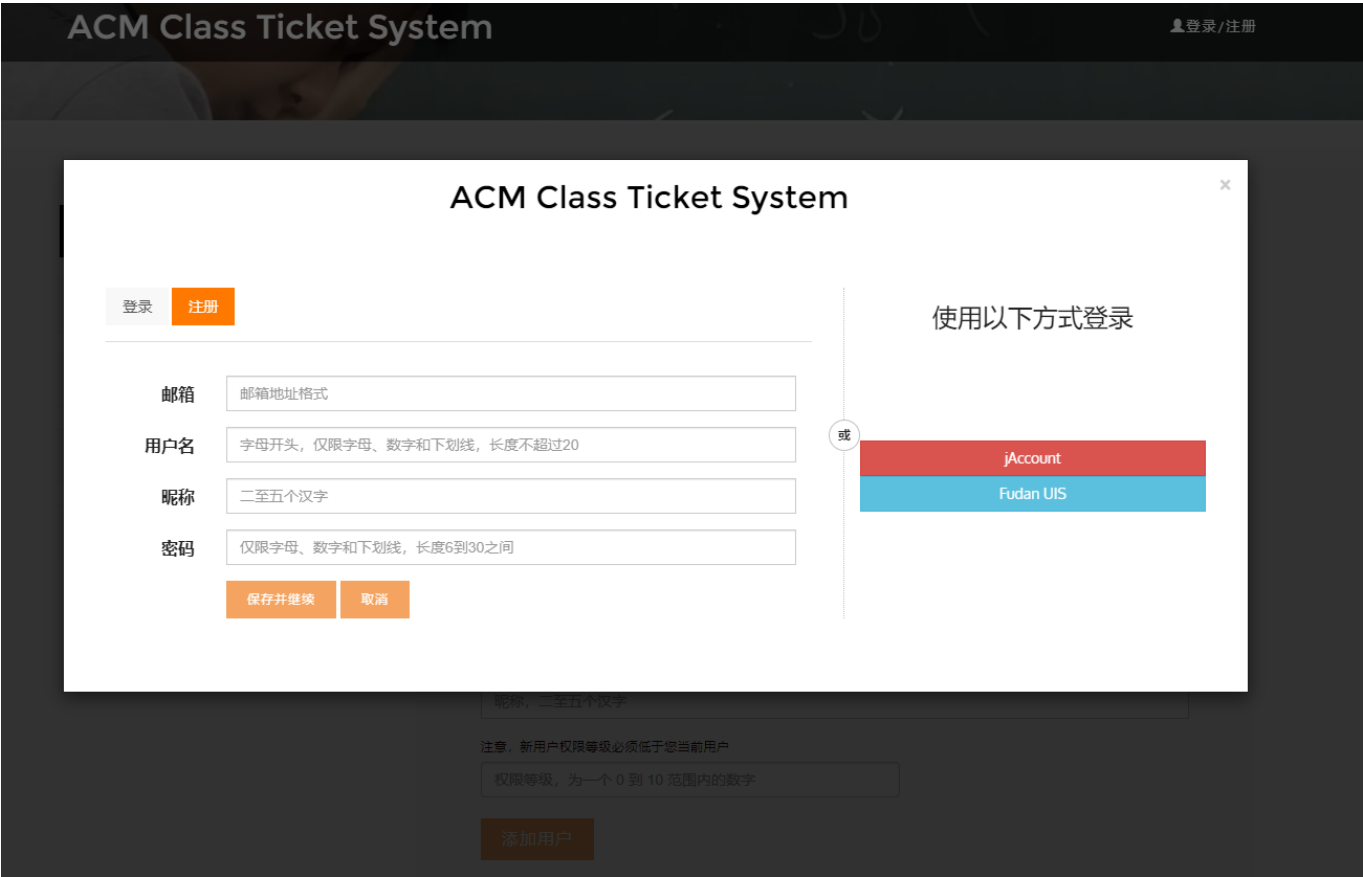
注意，新用户权限等级必须低于您当前用户

权限等级，为一个 0 到 10 范围内的数字

添加用户

你可以通过“需要帮助吗？”这一入口随时找到我们。

你应该首先进行右上角的“注册”操作。注册默认创建一个**权限为 1**的账号。



注意用户名重名或注册信息填写不要求都会导致注册失败。

注册结束后，请在同样的窗口进行登录。

- Q：用户的添加、查询、修改对普通用户开放吗？  
A：暂时开放。请留心系统给出的所有注释。

4 列车相关部分

**当你需要查询、管理火车信息时，请访问此页面。**

对于一个列车，我们需要添加其经过的站数以及各个站的信息。

在该系统，可以看到填写表单里有一处站数的填写处，下方的路线图会根据站数即时拉长。

请输入该火车的行程路线

4

起

站名

到下站历时

起点站无停靠时长

到下站票价

S2

站名

到下站历时

停靠时长

到下站票价

S3

站名

到下站历时

停靠时长

到下站票价

终

站名

终点站无下一站

终点站无停靠时长

终点站无下一站

添加火车

起点站、终点站的部分参数是无意义的，不必也无法填写。

查询一个火车得到的界面大致如下：

车次行程🚆



车次编号

Whythesearethech

列车类型

O

发车日期

07-05

发行情况

已发行

现在发行它

📍 重庆市江津市	🟢 xx-xx xx:xx	🔴 07-05 00:32	余 61895 票	到此累计花费 ¥ 0
📍 吉林省长春市长春市	🟢 07-05 03:09	🔴 07-05 03:18	余 61895 票	到此累计花费 ¥ 233
📍 江西省九江市	🟢 07-05 05:14	🔴 07-05 05:21	余 61895 票	到此累计花费 ¥ 319
📍 山东省荣成市	🟢 07-05 06:40	🔴 07-05 06:43	余 61895 票	到此累计花费 ¥ 510
📍 四川省自贡市	🟢 07-05 08:24	🔴 07-05 08:29	余 61895 票	到此累计花费 ¥ 838

左侧为该车次的基本信息，你可以在此页面发行该火车，也可以选择列车信息的“操作”栏里发行它。已发行的火车按钮显示为灰色。

右侧为该火车当天的行程表，其中标为绿色的时间表示其**到达该站点的时刻**，红色的时间表示其**离开该站点的时刻**。无意义的数据显示为 x。

• Q：火车的图片是该火车的预览图吗？

A：否，只是装饰图片。

5 出行相关部分

当你需要拟定出行计划时，请访问此页面。

本系统目前只支持不换乘的车票查询，以及**只换乘一次**的最优方案查询。

最优方案大致呈现方式如下。



## 最优换乘方案

根据您的要求，我们提供给您的最优换乘方案如下。建议您：

### 先乘坐

车次编号	起始站	出发时间	终点站	到达时间	座位数	价格
 Idarenotproceedt	 上海市	 07-05 19:04	 云南省曲靖市	 07-05 22:29	 92042	¥3648-¥364 

### 再换乘


车次编号	起始站	出发时间	终点站	到达时间	座位数	价格
 Underfootthedivine	 云南省曲靖市	 07-05 22:45	 北京市	 07-06 02:03	 74536	¥8488-¥848 

每次查询一个出行方案，你都需要选择要你关心的优先级：时间短优先还是费用低优先。

优先级高的方案会优先被呈现，在车票查询中会出现在表单更上面。

您希望的优先级是？

越快越好

越快越好

实惠优先

车票查询界面还提供了购票渠道，可以直接在此进行购票。

购票后会自动帮你填好大部分表单，你只需要选择订单数量，然后提交订单。

车票清单

符合条件的车票路线共有 50 条

车次编号	起始站	出发时间	终点站	到达时间	座位数	价格
crimeslessimporta	北京市	07-05 03:34	天津市	07-05 03:54	62266	¥450-¥45 订购
LeavesofGrass	北京市	07-05 14:47	天津市	07-05 15:36	582	¥1530-¥153 订购
face	北京市	07-05 18:23	天津市	07-05 19:40	8002	¥2160-¥216 订购
Thepennantisflyin	北京市	07-05 10:15	天津市	07-05 11:49	7230	¥2990-¥299 订购
listen	北京市	07-05 15:39	天津市	07-05 17:23	9263	¥1290-¥129 订购
rebuffsasthetree	北京市	07-05 07:44	天津市	07-05 09:31	85085	¥2840-¥284 订购
Martyrsartistsin	北京市	07-05 03:03	天津市	07-05 04:51	7958	¥2150-¥215 订购
SHUTnotyourdoors	北京市	07-05 22:15	天津市	07-06 00:06	80574	¥5270-¥527 订购
Farfromthelanko	北京市	07-05 05:56	天津市	07-05 07:55	99519	¥1700-¥170 订购
ResolutewarlikeOne	北京市	07-05 18:16	天津市	07-05 20:29	44325	¥4640-¥464 订购
Thefirststepsay	北京市	07-05 23:54	天津市	07-06 02:36	8279	¥7690-¥769 订购

- Q: 价格均打 1 折是?
- A: 是本系统独占大力度优惠。

6 订单相关部分

当你需要下单，或者查询、退订订单时，请访问此页面。

下单时，请注意自己是否接受候补购票。

是否接受候补购票？

不接受（余票不足将直接购买失败）

不接受（余票不足将直接购买失败）

接受（余票不足进入候补队列，有票时立即购买）

接受候补购票后，若余票不足，如果订单数未超过车辆总座位数，系统会将您的订单放入候补队列。

当有人退票让出足够的余票后，会按时间先后顺序第一时间从候补队列中选择合法的订单，其立即购买成功。

一个查询完毕的订单结果大致如下，其中订单状态共有三种：Success（成功购票）、Pending（位于候补队列）和 Refunded（已退票）。

订单列表

您的订单共有 72 份

订单状态	车次编号	起始站	出发时间	终点站	到达时间	购票数量	总价
Refunded	melancholyrhythm	新疆和田市	07-07 10:12	北京市	07-07 17:41	3	¥3 退票
Pending	INSCRIPTIONS	河南省禹州市	07-21 01:17	贵州省都匀市	07-21 06:08	6058	¥604 退票
Success	Orwithdrawntomuse	湖北省孝感市	07-16 01:05	云南省昆明市	07-16 06:28	1809	¥408 退票
Success	Imbuedastheypass	新疆乌苏市	07-03 02:50	湖南省永州市	07-03 04:45	6886	¥619 退票
Refunded	INcabindshipsat	福建省宁德市	08-06 01:29	北京市	08-06 17:03	6620	¥1125 退票
Success	Orwithdrawntomuse	吉林省通化市	08-07 21:41	湖北省孝感市	08-08 00:58	354	¥152 退票
Success	knownotwhitherye	辽宁省沈阳市	08-19 14:50	四川省绵竹市	08-20 05:25	11134	¥2930 退票
Pending	LeavesofGrass	湖北省荆州市	07-11 21:20	吉林省松原市	07-12 06:42	1386	¥712 退票
Refunded	INSCRIPTIONS	贵州省都匀市	07-07 06:12	北京市	07-07 12:58	1292	¥724 退票
Success	Returninguponmypo	山东省烟台市	08-20 18:31	江苏省扬中市	08-20 23:31	6888	¥970 退票

对于未退票的订单，你可以在该页面随时通过右侧按钮进行退票。

Q：退票后订单会一直保留在我的购票记录上吗？

A：会的。

7 退出系统

如果您为普通用户，直接退出网页即可。您的浏览器很可能保留了您的登录 cookie，所以下次进入系统会自动登录。清除浏览器的 cookie 可以防止这一点。

如果您为管理员，想要停止系统，可以在服务器终端结束 Flask 进程。注意不要非正常退出，以免数据丢失。