

分类号: TP311.5

UDC: D10621-408-(2019)2496-0

密 级: 公开

编号: 2015102004

成 都 信 息 工 程 大 学

学 位 论 文

基于网络爬虫的 ACM 预算系统的设计与实现

论 文 作 者 姓 名 :	龙雅容
申 请 学 位 专 业 :	软件工程
申 请 学 位 类 别 :	工学学士
指导教师姓名 (职称) :	朱毅 (教授)
论 文 提 交 日 期 :	2019 年 5 月 20 日

基于网络爬虫的 ACM 预算系统的设计与实现

摘 要

ACM 即国际大学生程序设计竞赛，是计算机领域最重要的比赛之一。ACM 教练在团队需要外出比赛的前几个月，将会对参加这次比赛队伍和老师进行初步的安排，通过上网查询火车、机票、酒店等信息，初步进行财务预算，估计这次比赛所需要产生最大费用。外出比赛前几天，ACM 教练将会确定队伍和带队老师等相关人员以及其交通和住宿费用，对比赛费用进行最终的预算。比赛结束后，教练会对此次比赛实际产生的费用，进行记录，形成报表，以便报销。毕业设计针对 ACM 经费管理，设计开发了 ACM 预算系统。ACM 教练只需输入比赛地点、时间、参赛人数和带队教师数等几个关键信息，系统便自动在网上爬取相关费用信息，进而自动生成相关初始预算、最终预算以及最后的实际账单，不必再浪费大量的时间上网查询，信息分析，统计比较以及账目整理上。

ACM 预算系统是基于 B/S 模式，采用 Django 框架，实现 ACM 竞赛预算的网站系统。它通过 12306、携程及美团爬取相关信息，并存储在 MySQL 数据库中，以 python 函数形式，在 Django 中的 views 下调用。views 分析并处理数据，自动生成相关的预算与账目信息，在网页中显示。网页端采用了 html+css+jQuery 技术，部分使用了 bootstrap 设计。

关键词：网络爬虫；ACM 竞赛；财务预算；财务报表；Django；html+css+jQuery

The Design and Implementation of a ACM Budget System based on Crawler

Abstract

The ACM, or ACM International Collegiate Programming Contest, is one of the most important competitions in the field of computing. The ACM coach will make preliminary arrangements for the teams and teachers to participate in the tournament a few months before the ACM team needs to be away. They will through the Internet access to train, air tickets, hotels and other information, preliminary financial budget, estimated that the competition needs to produce the maximum cost. A few days before the game, the ACM coach will determine the costs of the team, the instructor and other relevant personnel, as well as their transportation and accommodation costs, and make the final budget for the game costs. After the game, the coach will record the actual expenses incurred in the game, and form a report form for reimbursement. For the ACM financial management, I design and develop of the ACM budget system. The ACM coach only needs to input some key information such as the competition location, time, number of participants and number of teachers leading the team, and the system automatically crawls the relevant cost information on the Web. The related of initial budget, final budget and final actual bill is generated automatically. In this way the ACM coach does not have to waste a lot of time on the Internet search, information analysis, statistical comparison and account collation.

ACM Budget System is based on B/S model, using Django Framework, and achieving ACM the website system of competition budget. It uses 12306, CTRIP and Meituan to crawl relevant information and store it in a MySQL database as a python function called in Django views. Views analyzes and processes data, automatically generating relevant budget and account information, and displaying it on a web page. HTML + CSS + jQuery technology is used in the web side, and bootstrap design is used in part.

Key words:Crawler;ACM;Financial Budget;Financial Statements;Django;html+css+jQuery

目 录

论文总页数：32页

1 引言.....	1
1.1 课题背景.....	1
1.2 本课题研究的意义.....	1
1.3 本课题的研究方法.....	1
2 系统总体设计方案.....	2
2.1 系统总体设计与分析.....	2
2.2 系统功能设计.....	2
2.3 开发运行平台选择及分析.....	6
2.4 系统关键技术及技术难点的解决方案.....	6
2.4.1 网络爬虫的实现与接口.....	6
2.4.2 自动生成预算.....	7
2.4.3 前端不规则表格的可编辑性.....	7
2.5 系统优点.....	7
3 ACM 预算系统实施方案.....	8
3.1 ACM 预算系统实施方案.....	8
3.2 功能详细定义.....	8
3.2.1 用户信息管理.....	8
3.2.2 初始预算管理.....	9
3.2.3 最终预算管理.....	10
3.2.4 实际账单管理.....	10
3.2.5 其它信息管理.....	11
3.3 ACM 预算系统数据库设计.....	12
3.3.1 数据库概念设计.....	12
3.3.2 数据库逻辑设计.....	14
3.3.3 数据库表结构设计.....	15
4 ACM 预算系统模块具体实现.....	20
4.1 用户信息功能具体实现.....	20
4.2 初始预算功能具体实现.....	22
4.3 最终预算功能的具体实现.....	23
4.4 实际账单功能的具体实现.....	24
4.5 其它信息功能的具体实现.....	25
5 ACM 预算系统模块运行及测试结果.....	26

5.1 系统测试方法.....	26
5.2 测试用例及结果.....	27
5.3 测试总结.....	28
结 束 语.....	29
参考文献.....	30
致 谢.....	31
声 明.....	32

1 引言

1.1 课题背景

ACM 国际大学生程序设计竞赛 (ACM International Collegiate Programming Contest, 简称 ACM-ICPC 或 ICPC) 是由国际计算机协会 (ACM) 举行的一项聚焦在展现大学生创新能力、团队精神、在有限的时间内用程序解决问题的年度竞赛。近 40 年的发展, 使 ACM 国际大学生程序设计竞赛成为全球最有影响力的大学生程序设计大赛[1,2,3]。

ACM 外出比赛之前, 教练需要通过上网查询, 人员估算, 对外出比赛的费用进行初步和进一步的预算。比赛结束后也要记录此次比赛的费用, 用于报销。年终时, 还需对一年的账单进行总结。ACM 教练在进行这些工作时, 需要从网上查询火车、飞机、酒店信息, 并对这些信息分析, 进行比较, 计算停留天数和外出天数。报销和总结时需要创建表格, 寻找之前的账单。这些工作, 大大浪费教练的时间和精力。因此, 建立一个 ACM 预算系统, 就成为了迫切的需要。

国内外存在着预算系统和账目系统, 但这些系统都依托与某个公司管理系统的大框架之下。无法独立出来, 针对某一项目, 而针对 ACM 的就更没有了。

1.2 本课题研究的意义

ACM 预算系统是一个自动化的系统。它将使教练有更多的精力花费在训练中, 而不用在继续花费大量的时间和精力在预算和账目整理中。给与学生更好的训练计划, 更加关注学生。

本系统本着节省教练精力, 让教练只需输入相关信息便可自动化获取预算以及比赛结束后的账目管理。使这些账目整合透明, 账单清晰。无论相隔多久都可以查看某场比赛的预算以及相关的账目。

本系统可推广性强, 只需修改相关信息, 便可作为一种其它项目的预算系统。如本系统在改变输入信息和创建预算规则的情况下, 可根据不同事业单位不同需求, 稍作修改, 便可应用于一般事业单位中。

1.3 本课题的研究方法

本系统通过网络爬虫获取相关信息存储在 MySQL 数据库中。数据库命名为 budg, 网络爬虫通过 ajax 抓取包, 简单使用 get/post 获取 api 中的信息^[4,5], 再对信息进行删减。以 Django 为框架建立自动化的 ACM 预算管理系统。Django 是 Python 语言下建立的一个网页开发框架, 对后台的管理使其特点之一。本系统在 pycharm 中进行 Django 的开发, 前端采用 HTML+CSS+jQuery。Python 版本为 python3。

2 系统总体设计方案

2.1 系统总体设计与分析

ACM 预算系统的用户拥有教练和总教练两种身份。只有用户登录以后才能进入系统。进入系统后，用户可以创建初始预算、最终预算及实际账单并对它们进行删除，修改，查询，还可查看火车车次，飞机航班，酒店等信息，并对这些信息进行添加，删除。总教练进入系统后，可对系统中的教练进行日常管理。如可以注册用户，可以修改用户信息，尤其是用户的类型，可以注销用户，但在系统中存在创建的预算或账单的用户，不可注销。本系统的架构图如图 2-1 所示：

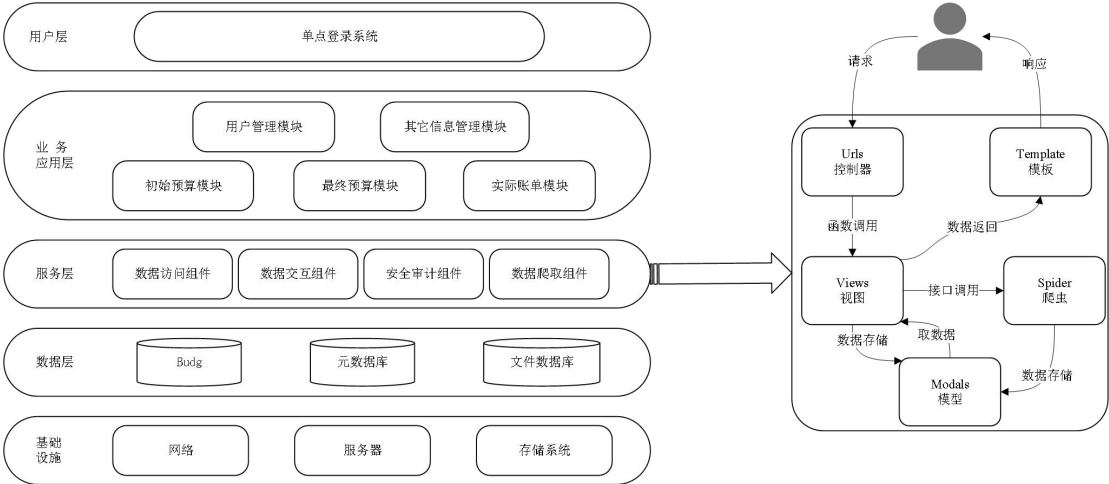


图 2-1 系统架构图

本系统的用户层采用了单点登录系统的模式，即任何页面都可以获取登录用户的信息。业务管理层拥有 5 个模块：用户管理模块、初始预算管理模块、最终预算管理模块、账单管理模块和其它管理模块。服务层含有数据访问、数据交互、安全审计、数据爬取组件，以 MVC 即浏览器-视图-模型的方式进行架构。用户在浏览器上发送请求传入给 urls，urls 调用 views 中相关的方法，views 进行数据处理之后，将数据返回到模板中进行渲染，模板对用户的请求进行响应。数据层采用了 MySQL 数据库，它含有 MySQL 自带的元数据库、文件数据库及本系统设计的 Budg 数据库。本系统需要爬取信息存储在数据库中，用户需要在服务器中获取数据库的信息。因此基础设施需要网络、服务器和存储系统。

2.2 系统功能设计

ACM 预算系统功能主要分为五大模块，用户管理模块、初始预算模块、最终预算模块、实际账单模块、其它信息管理模块。用户管理模块除用户登录外，总教练需对用户进行增删改查；初始预算管理模块查、最终预算模块、实际账单模块、其它管理模块都是用户对相关模块进行增删改查。其中最终预算模块的创

建依托与初始预算模块，实际账单模块依托于最终预算模块，即无相关的初始预算无法建立相关的最终预算，无相关的最终预算无法建立相关的实际账单。系统功能结构图如图 2-2 所示：

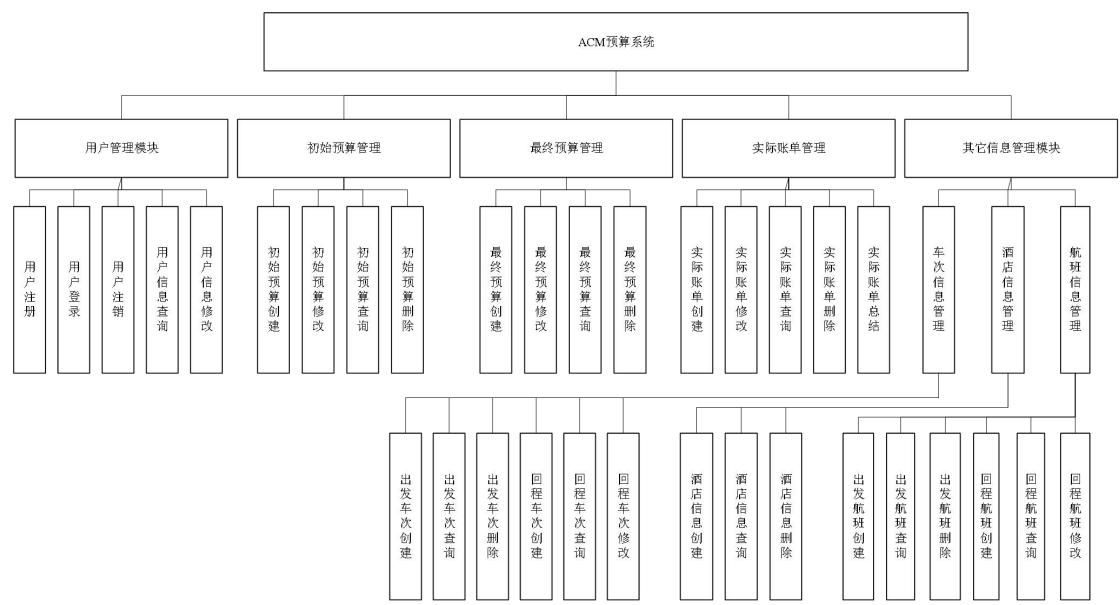


图 2-2 系统功能结构图

系统各模块功能概述如下：

(1) 用户信息管理：用户分为两种，分别为教练和总教练，有登录和查询用户信息权限。但教练查询用户信息只可以看见用户名和用户类型，而总教练还可看见密码。只有总教练可以注册和修改用户信息。本模块用例图如图 2-3 所示：

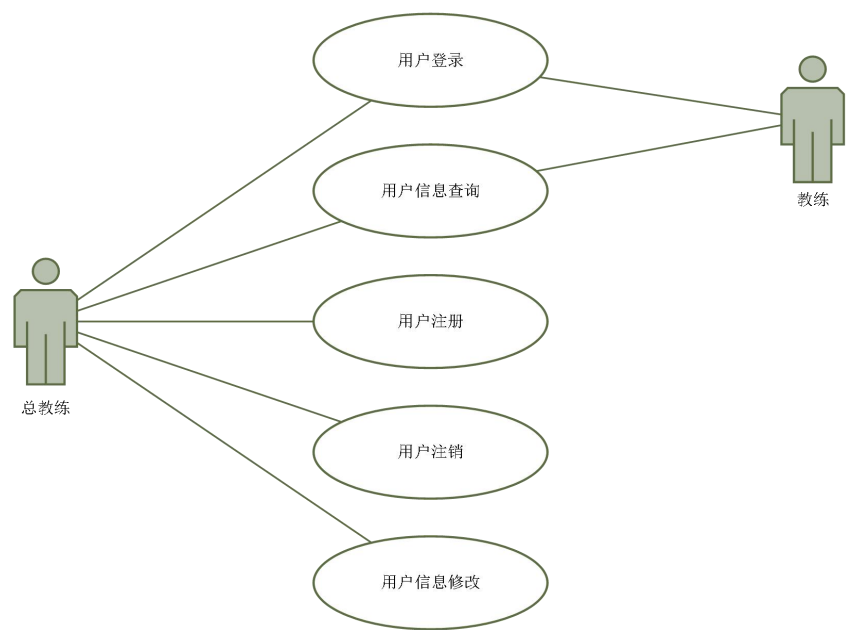


图 2-3 用户信息管理用例图

（2）初始预算管理：用户可以通过输入相关信息如竞赛名称，竞赛队伍数量等建立初始预算表。用户可修改，删除该初始预算，但删除的同时，将删除竞赛账目相关的表格。本模块的用例图如图 2-4 所示：

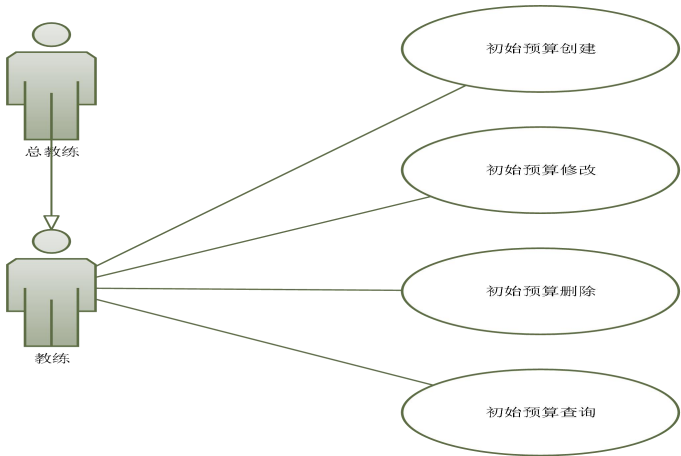


图 2-4 初始预算管理用例图

（3）最终预算管理：最终预算管理是用户在初始预算的基础上修改和确定初始预算的相关信息，进一步进行预算，相比初始预算，它将更加准确。用户建立最终预算时，将会选择一个初始预算才能进入创建页面。用户也可对最终预算进行删改查。本模块的用例图如图 2-5 所示：

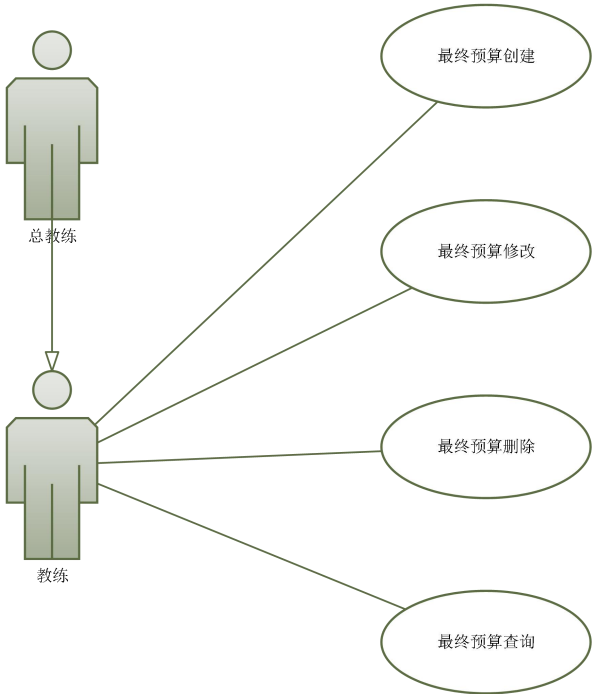


图 2-5 最终预算用例图

（4）实际账单管理：用户可以通过最终预算表创建账单，对实际账单进行查询、查看年度总结、修改数据、删除数据。本模块的用例图如图 2-6 所示：

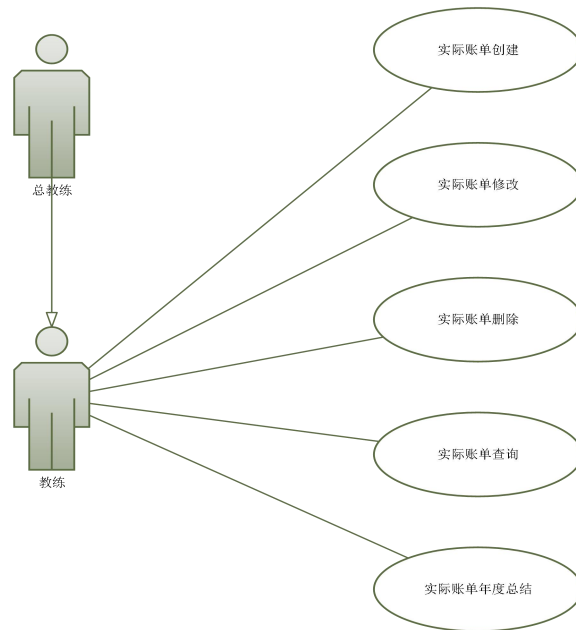


图 2-6 实际账单管理用例图

(5) 其它信息管理：其它信息是酒店，火车，航班信息管理。用户可对其它信息进行增删查操作。本模块的用例图如图 2-7 所示：

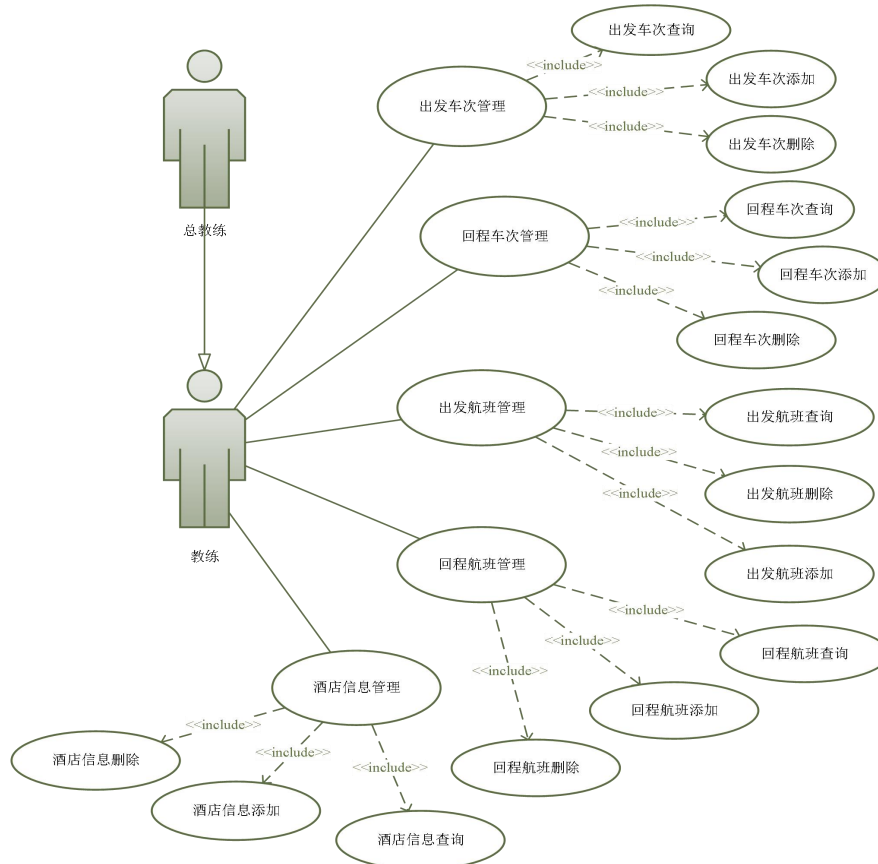


图 2-7 其它信息用例图

2.3 开发运行平台选择及分析

开发工具：本系统的开发工具为 pycharm。Pycharm 是一种 python IDE，具有一般 IDE 的功能，如调试、语法高亮、project 管理、代码跳转等功能。同时它通过了一些很好的功能用于开发 Django，自带 HTML、css 和 JavaScript 编辑器，可以更快速的通过 Django 框架进行 web 开发。

数据库：本系统的数据库为 MySQL。MySQL 是一个关系型数据库管理系统，拥有体积小、速度快优点。

2.4 系统关键技术及技术难点的解决方案

本系统的关键技术和技术难点主要有以下几点：

2.4.1 网络爬虫的实现与接口

调度器、URL 管理器、下载器、网页解析器、应用程序构成了 python 的网络爬虫。爬虫系统的中枢系统是调度器，它管理了其它几个部分，其中 URL 管理器包含了所有已经爬取和未爬取的地址；网页下载器从 URL 管理器中获取为爬取的地址进行下载网页，它的部分功能已在 urllib2 中实现；网页解析器对从网页下载器中得到网页信息进行解析；应用程序是用户利用正则表达式通过需求从网页上获取相关信息的应用程序[6,7,8,9,10]。其框架图如图 2-8 所示：

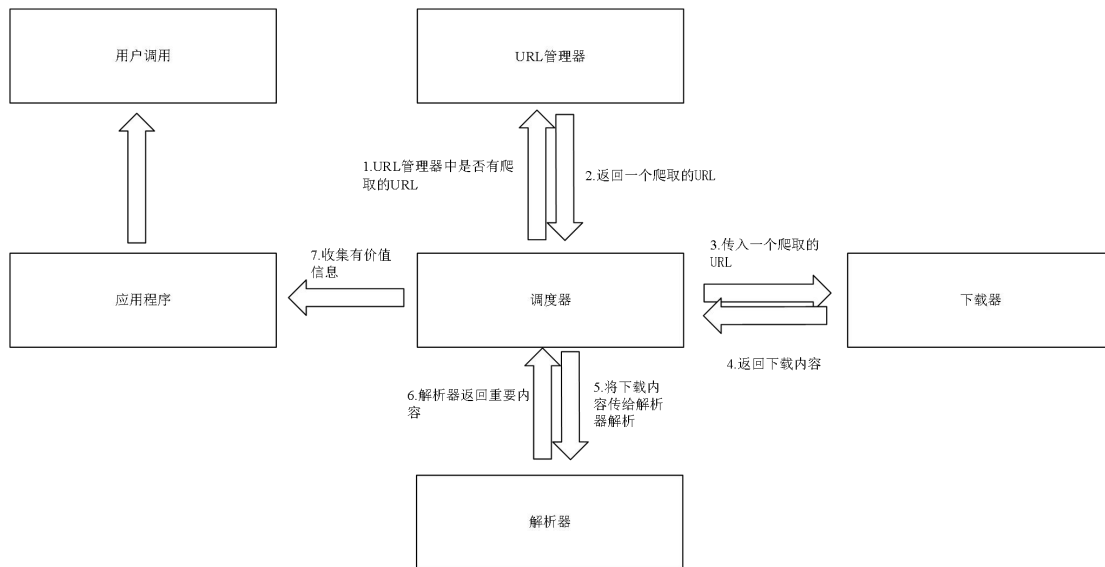


图 2-8 爬虫框架图

本系统采用的 Get/Post 方法来自于 request 库^[11,12]。Get/Post 是通过 HTTP 协议中的 GET/POST 请求方式对目标网页发起请求，获取网页中的信息。为了假装浏览器浏览网页，需设置头信息。在浏览器的开发者工具中找到 network 中获取网页的 heads 信息。Get 和 Post 区别，Get 获取全部信息而 Post 可以通过表单的方式获取特定信息。

为了爬取需要的信息，如携程上的机票信息，需要分析页面，寻找 api^[13]，携程上的机票 api 挺难找，不能直接通过浏览器查找，只能在 network^[14,15,16]下查询，层层分析，其 api 为 https://flights.ctrip.com/itinerary/api/12808/products。必须将爬虫写成一个简单的函数，在 django 下单独建立 py 文件，由 view 来调用。

2.4.2 自动生成预算

为了选取最合适的预算，程序将根据最值生成不同的方案，并且还要考虑一些突发方案。比如回程时间太晚需要延一天才有车次等情况，情况复杂，导致条件较多，实现较为麻烦。本系统采取分步处理，在爬虫期间需爬取相隔一天的情况，进行预处理。并且将火车车次和航班车次分为出发和回程两部分，系统采用 ORM 的方式创建相关数据表与生成方案。

ORM 是后端框架 Django 的核心思想，即“Object Relational Mapping”（对象--关系映射）。针对的也就是数据库、模型、视图这三个板块了，即在模型中创建一个实例对象，该实例对象对应为数据库中的数据表名，类属性对应数据表的字段，在 view 中对这个实例对象进行操作时，对应相关 MySQL 操纵语句。模型到数据库转换时，调用 Django 下已经写好的 python 方法，将对应 python 语句转换为 MySQL 语句^[17]。其体系框架如图 2-9 所示：

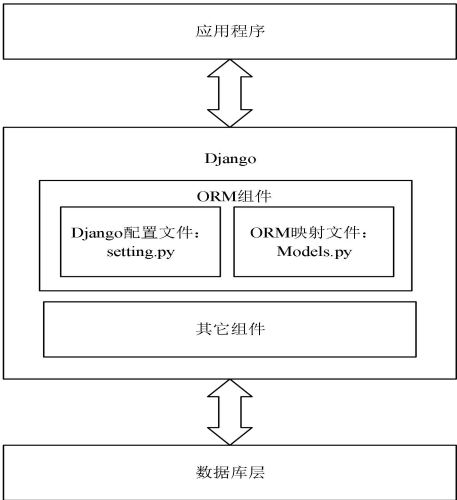


图 2-9 Django ORM 体系结构图

2.4.3 前端不规则表格的可编辑性

系统采用了 Bootstrap 的框架对前端进行编码。Bootstrap 是一种基于 HTML、CSS、JavaScript 开发的简洁、直观、强悍的 web 开发框架。它提供了一个带有网格系统、链接样式、背景的基本结构。用于网页开发非常快捷、方便并且实用。

为了实现不规则表格，系统以 bootstrap 表格样式为基础，设计不规则表格。

2.5 系统优点

(1) 用 Django 框架设计系统：用户直接在浏览器浏览网页，对其进行操作，

无需提前下载安装。

(2) 实时预算：用户通过输入相关信息进行预算，由于加入了爬虫，因此可以对于输入的信息进行实时查询。

(3) 操作简单：导航栏简单，增删改查基于大多数人的习惯，上手容易。

3 ACM 预算系统实施方案

3.1 ACM 预算系统实施方案

ACM 预算系统主要是针对校内 ACM 团队外出比赛的预算以及实际账目的管理。它使 ACM 预算进行最大限度的保守估计，让教练心中有数，揣有足够的资金，以防突发状况，同时比赛结束后，也让本次比赛的账目情况清晰明了。

3.2 功能详细定义

ACM 预算系统的用户分为教练与总教练。总教练享有最高权限，教练拥有用户信息管理部分权限及其它所有权限。

3.2.1 用户信息管理

在用户信息管理中，教练只能登录并修改自己的信息（除用户类型），可查询不带密码的用户信息。总教练可登录，查看所有人信息并修改所有人信息，也可注册和注销用户，用户管理流程图如图 3-1 所示：

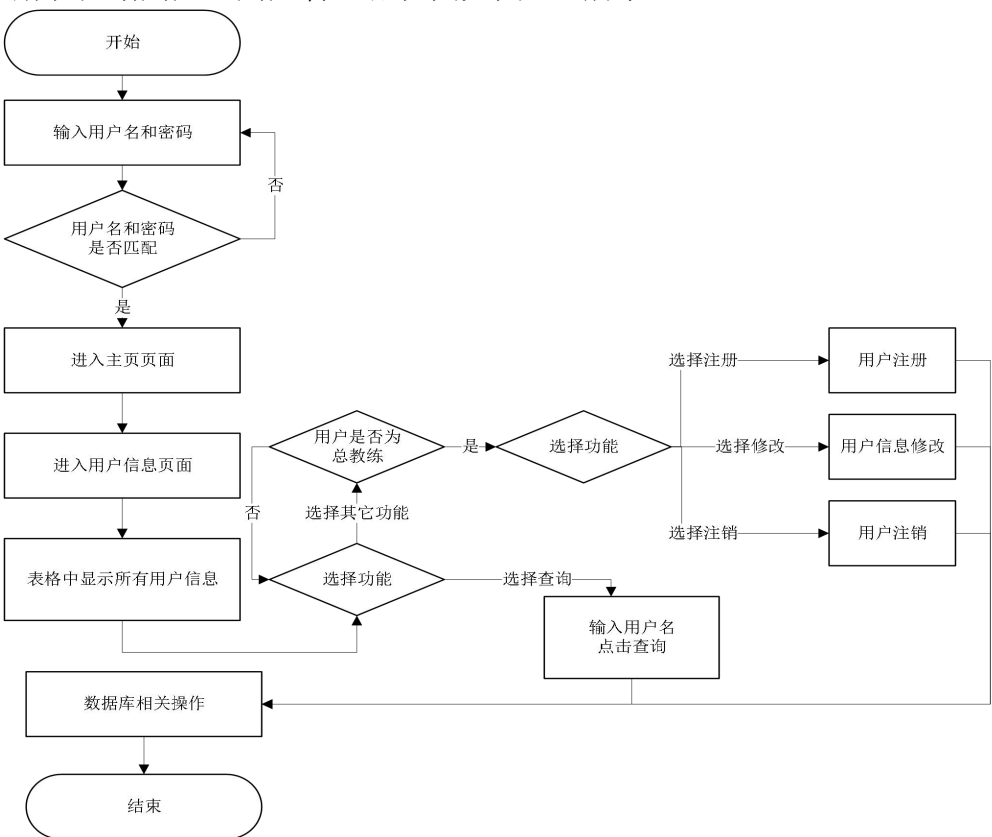


图 3-1 用户管理流程图

（1）用户登录：本系统为单点登录系统，并且只有用户只有输入正确的信息和密码才可进入系统。

（2）用户注册：总教练可以对用户进行注册。注册用户的信息字段有用户名、密码和用户类型，用户类型只需选择是否是总教练即可。

（3）用户查询：用户可以查看所有用户的信息，也可查询某一个用户的用户信息。但只有总教练才可以查看用户的密码信息。

（4）用户修改：总教练可以修改所有用户的所有信息。教练只能修改自己的密码。

（5）用户注销：总教练可以对用户进行注销操作，但不可注销已创建预算或账单的用户。

3.2.2 初始预算管理

在初始预算管理中，所有用户均可创建、删除、查询、修改初始预算信息。初始预算流程图如图 3-2 所示：

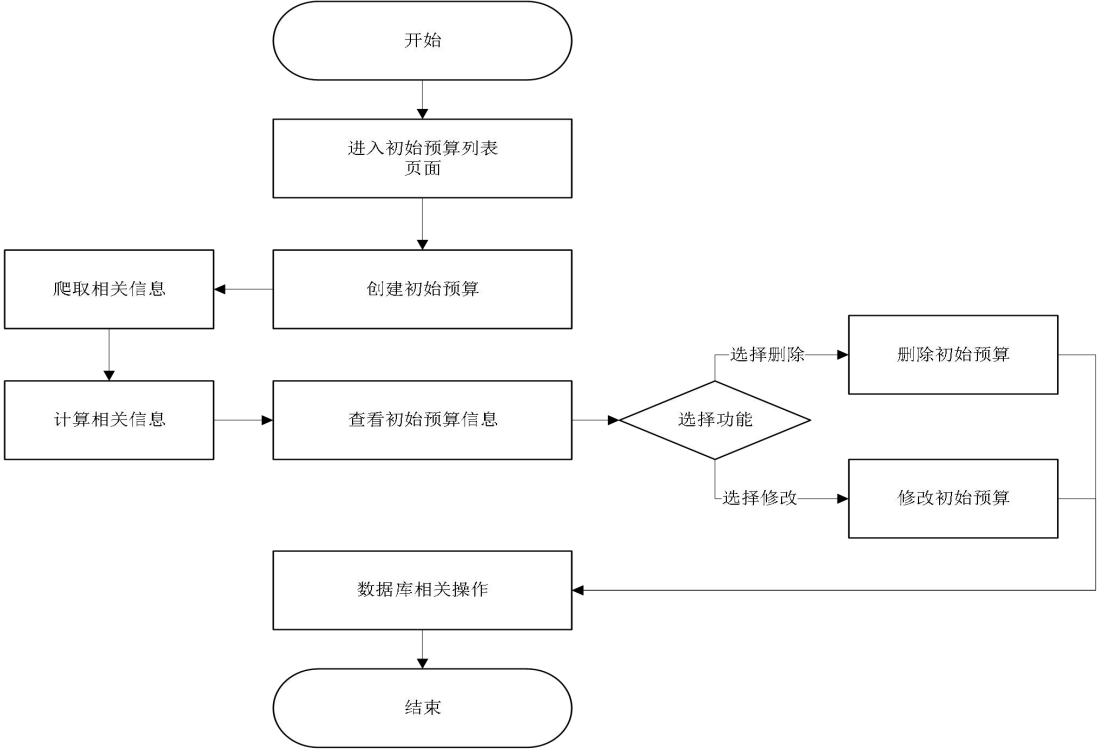


图 3-2 初始预算流程图

（1）初始预算查询：所有用户均可查询所有的初始预算，也可根据年份，城市，比赛名信息查询相关的初始预算。

（2）初始预算删除：所有用户均可删除初始预算，但不可删除已根据该预算创建最终预算的初始预算。

（3）初始预算修改：所有用户均可修改初始预算，但只可修改初始预算的

部分信息。

(4) 初始预算创建：所有用户均可输入相关信息创建初始预算。

3.2.3 最终预算管理

最终预算管理是在初始预算管理的基础下，进一步预算。用户可对最终预算信息进行增删改查操作。最终预算流程图如图 3-3 所示：

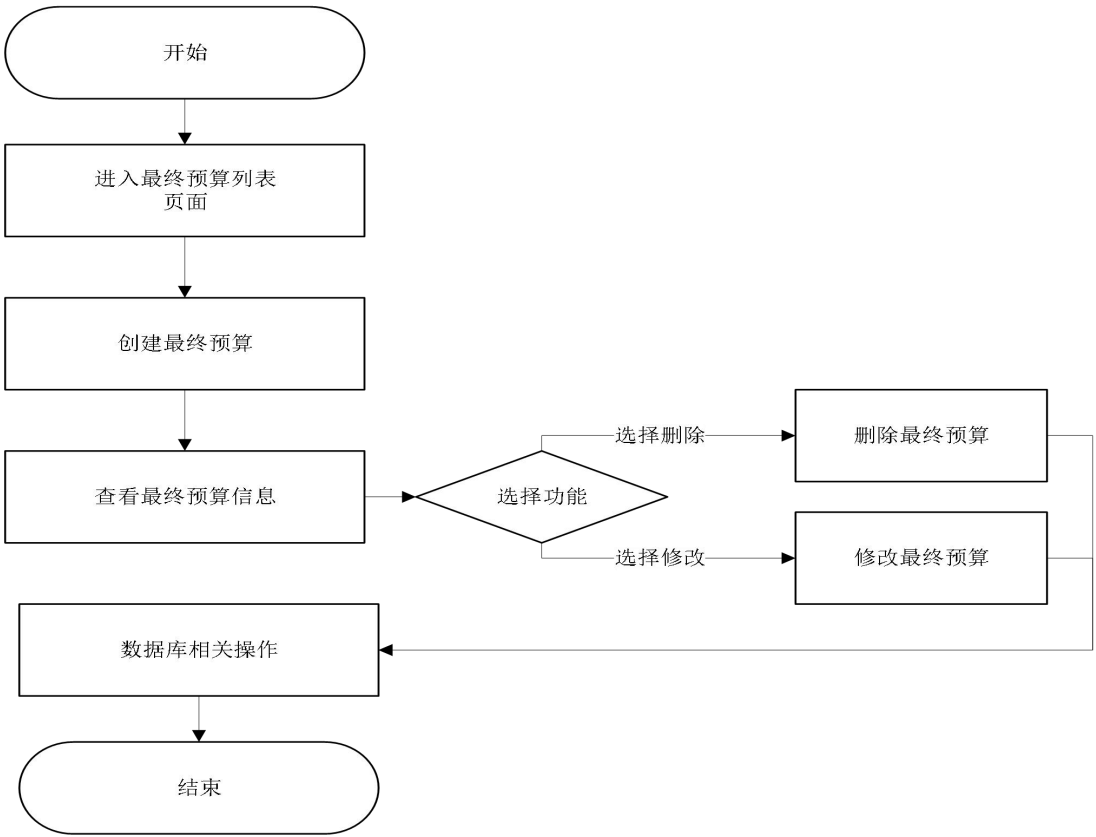


图 3-3 最终预算流程图

(1) 最终预算查询：所有用户均可查询所有的最终预算，也可根据年份、城市、比赛名查询相关的最终预算。

(2) 最终预算创建：所有用户可以根据创建好的初始预算进一步创建最终预算。

(3) 最终预算修改：所有用户可对最终预算信息进行修改。

(4) 最终预算删除：所有用户都可删除最终预算，但不可删除已根据该预算创建实际账单的最终预算。

3.2.4 实际账单管理

实际账单管理为竞赛结束之后，用户与最终预算进行核对，将所有与实际不符的信息进行修改，并增添相关信息如备注等。系统也有实际账单删、改、查功能。实际账单程图如图 3-4 所示：

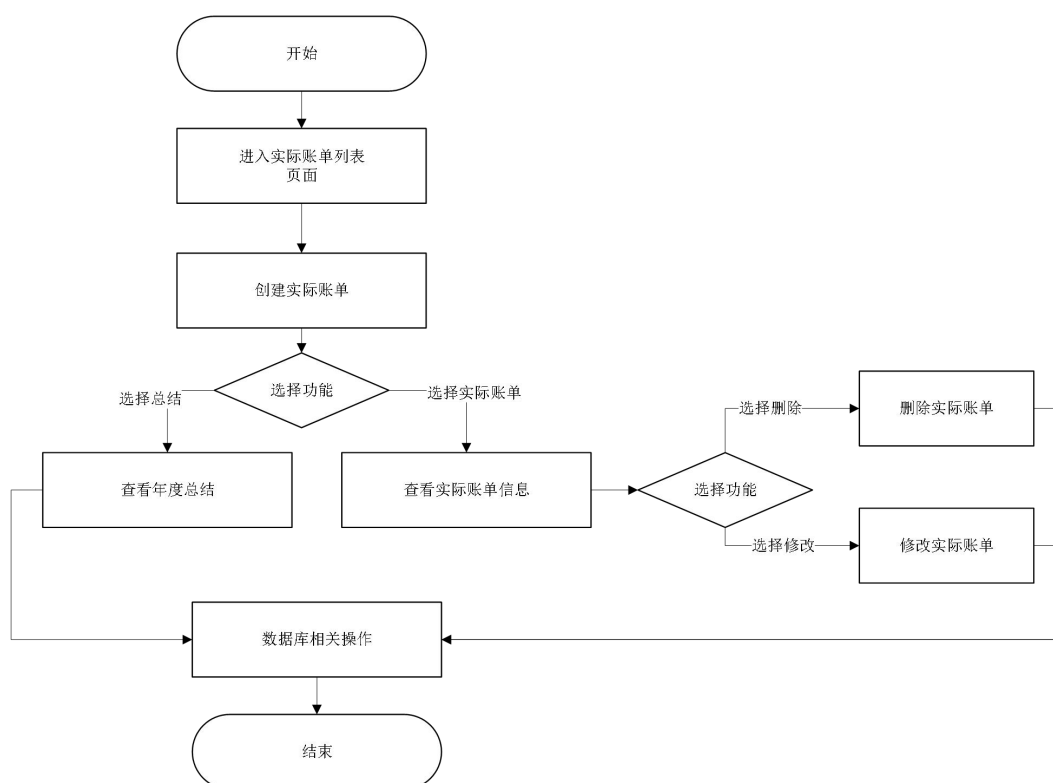


图 3-4 实际账单流程图

- (1) 实际账单查询：用户可查询所有的实际账单，也可根据年份、城市、比赛名查询相关的实际账单。
- (2) 实际账单建立：用户可根据最终预算进一步建立实际账单。
- (3) 实际账单修改：用户可修改所有的实际账单的信息。
- (4) 实际账单删除：用户可对实际账单进行删除。
- (5) 实际账单总结：用户可查看实际账单的年度总结，查看某一年总结发费和这一年每次比赛的发费。

3.2.5 其它信息管理

其它信息是指通过爬虫爬取的酒店，出发车次、回程车次、出发航班、回程航班的信息。用户查看任意一项的所有信息。用户可对这项信息进行查询，删除和添加操作。注意，这些信息不允许修改。其它信息流程图如图 3-5 所示：

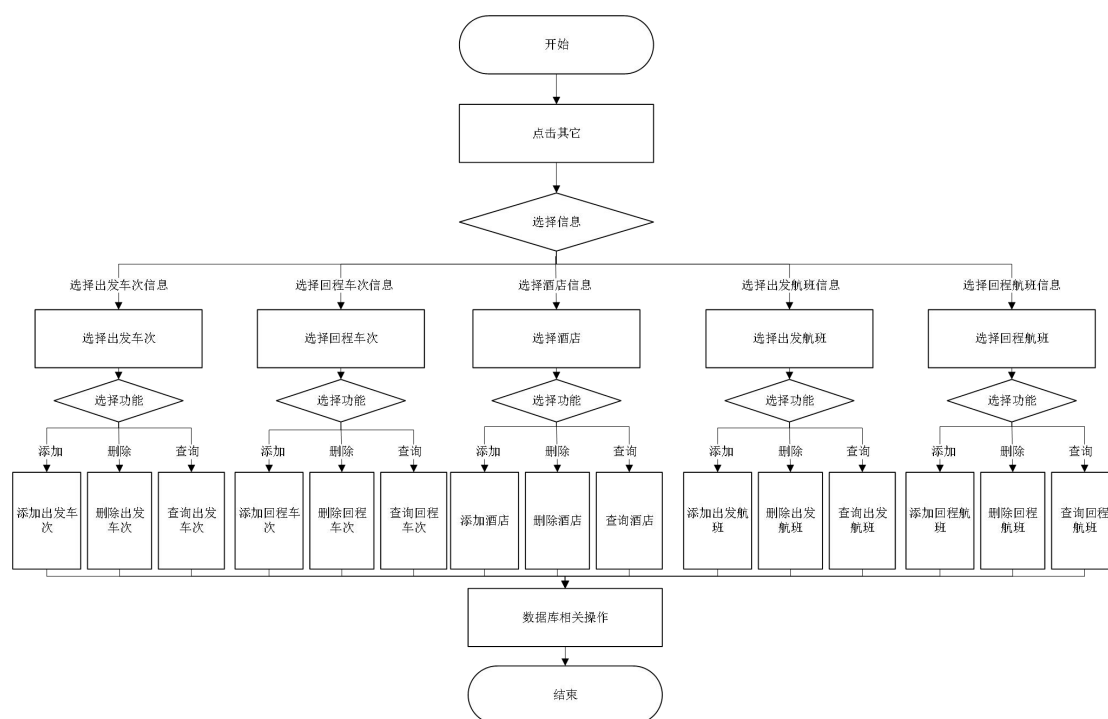


图 3-5 其它信息流程

(1) 酒店/车次/航班/信息查询：用户可对酒店、出发车次和回程车次、出发航班和回程航班的所有信息进行查询。如若查询酒店信息，也可根据酒店名称和地点查询相关信息；如若查询出发车次，可根据出发时间，车次，地点查询相关信息；如若查询回程车次，可根据到达时间，车次，地点查询相关信息；如若查询出发航班，可根据出发时间，航班号，地点查询相关信息；如若查询回程航班，可根据到达时间，航班号，地点查询相关信息。

(2) 酒店/车次/航班信息删除：用户可对酒店、出发车次和回程车次、出发航班和回程航班的信息进行删除。用户对酒点添加的信息有酒店名称、地点和价格；对出发车次和回程车次添加的信息有出发/回程车次、出发时间、到达时间、地点和价格；对出发航班和回程航班添加的信息有出发/回程航班号、出发时间、到达时间、地点和价格。

(3) 酒店/车次/航班信息添加：用户可以添加酒店、出发车次和回程车次、出发航班和回程航班的信息进行删除。

3.3 ACM 预算系统数据库设计

3.3.1 数据库概念设计

根据 ACM 预算系统的需求设计，其数据库 E-R 图如图 3-6 所示：

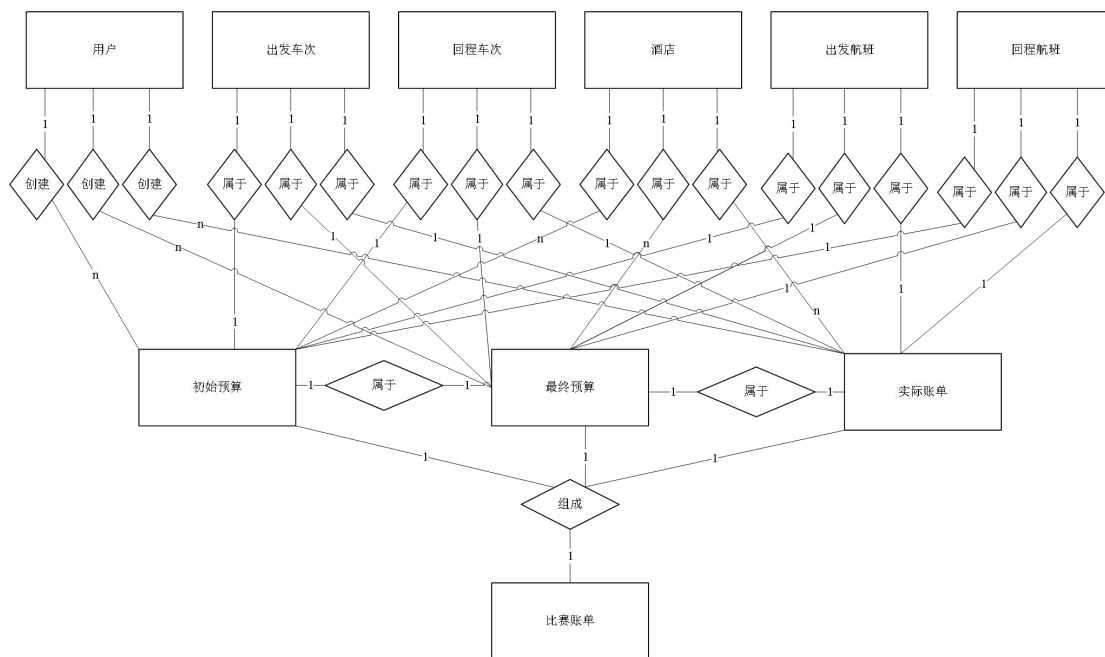


图 3-6 系统 E-R 图

(1) 一个用户可以创建 n 个初始预算，一个初始初始预算只能由一个用户创建；一个用户可以创建 n 个最终预算，一个最终预算只能由一个用户创建；一个用户可以创建 n 个实际账单，一个实际账单只能由一个用户创建。

(2) 一个出发车次只能属于一个初始预算，一个初始预算只能包含一个出发车次；一个出发车次只能属于一个最终预算，一个最终预算只能包含一个出发车次；一个出发车次只能属于一个实际账单，一个实际账单只能包含一个出发车次。

(3) 一个回程车次只能属于一个初始预算，一个初始预算只能包含一个回程车次；一个回程车次只能属于一个最终预算，一个最终预算只能包含一个回程车次；一个回程车次只能属于一个实际账单，一个实际账单只能包含一个回程车次。

(4) 一个酒店可以属于 n 个初始预算，一个初始预算只能包含一个酒店；一个酒店可以属于 n 个最终预算，一个最终预算只能包含一个酒店；一个酒店可以属于 n 个实际账单，一个实际账单只能包含一个酒店。

(5) 出发航班代表一个出发航班只能属于一个初始预算，一个初始预算只能包含一个出发航班；一个出发航班只能属于一个最终预算，一个最终预算只能包含一个出发航班；一个出发航班只能属于一个实际账单，一个实际账单只能包含一个出发航班。

(6) 一个回程航班只能属于一个初始预算，一个初始预算只能包含一个回程航班；一个回程航班只能属于一个最终预算，一个最终预算只能包含一个回程航班。

航班；一个回程航班只能属于一个实际账单，一个实际账单只能包含一个回程航班。

(7) 一个初始预算只能属于一个最终预算，一个最终预算只能包含一个初始预算。

(8) 一个最终预算只能属于一个实际账单，一个实际账单只能包含一个最终预算。

(9) 一个初始预算、一个最终预算和一个实际账单只能组成一个比赛账单，一个比赛账单只能由一个初始预算、一个最终预算和一个实际账单组成。

3.3.2 数据库逻辑设计

在 ACM 预算系统的数据库物理模型中，字母 pk 表示的是该字段是该表的主键，字母 fk 表示该字段是该表的外键，字母 null 表示该字段在该表中可以为空。实心箭头用来表示表与表之间的关系。ACM 预算系统的数据库物理模型图如图 3-7 下所示：

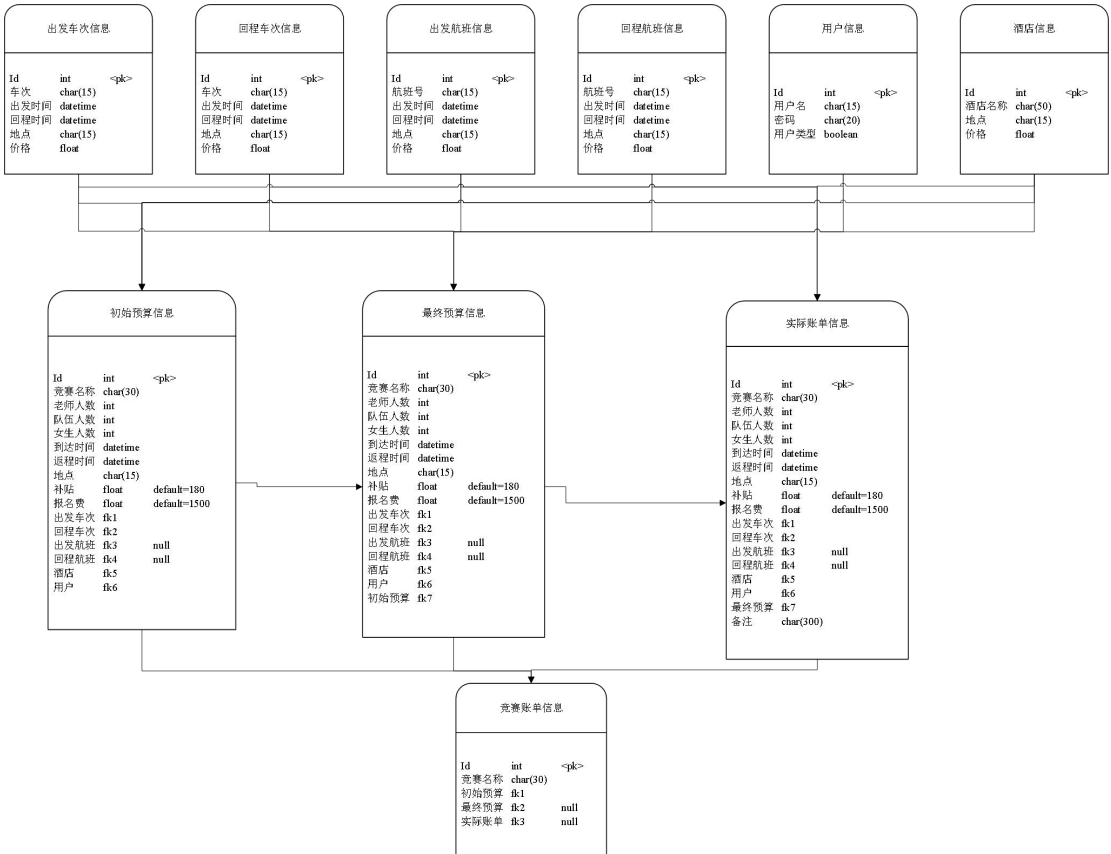


图 3-7 数据库物理模型图

本系统主要涉及的数据共有十张分别是：

- (1) 用户信息表：通过用户信息表，存储所有用户的信息。
- (2) 出发车次信息和回程车次信息表：通过出发车次信息表和回程车次信

息表，存储所有经过网络爬虫爬取的车次信息。

(3) 出发航班信息和回程航班信息表：通过出发航班信息表和回程航班信息表，存储所有经过网络爬虫爬取的航班信息。

(4) 酒店信息表：通过酒店信息表，存储所有经过网络爬虫爬取的酒店信息。

(5) 初始预算信息表：通过初始预算表，存储所有创建的初始预算相关信息。

(6) 最终预算信息表：通过最终预算表，存储所有创建的最终预算相关信息。

(7) 实际账单信息表：通过实际账单信息表，存储所有创建的实际账单相关信息。

(8) 竞赛账单信息表：通过竞赛账单表，存储所有创建的初始预算、最终预算和实际账单的相关信息。

其中，出发车次信息表、回程车次信息表、出发航班信息表、回程航班信息表、酒店信息表用户信息表均和初始预算信息表、最终预算信息表、实际账单信息表关联，初始预算信息表、最终预算信息表，实际账单信息表和竞赛账单信息表有关联。

3.3.3 数据库表结构设计

本系统主要涉及的数据表总共有十张，分别是：用户信息表、出发车次和回程车次表、出发航班和回程航班表、酒店信息表、初始预算表、最终预算表、实际账目表、竞赛账目表。

(一) 用户信息表

用户信息表存储的是本系统所有的用户信息。其含有 id，用户名，密码，用户类型 4 个字段。其中 id 为主键，整型；用户名字符串型，最长为 15，不为空；密码为字符串型，最长为 20，不为空；用户类型为布尔型，True 值为总教练，False 为教练，不为空。表结构如表 3-1 所示：

表 3-1 用户信息表 (Admi)

序号	逻辑字段	物理字段	类型 (长度)	规则	字段说明
1	Id	Id	INT	PK	主键
2	用户名	name	VARCHAR(15)	not null	必填项
3	密码	password	VARCHAR(20)	not null	必填项
4	用户类型	ty	BOOLEAN	not null	必填项, false 教练, true 总教练

(二) 出发车次信息表

出发车次信息表存储的是基于网络爬虫的 ACM 预算系统在 12306 网站爬取的所有出发时火车车次的信息。其含有 id, 车次, 出发时间, 到达时间, 地点, 价格 6 个字段。Id 为主键, 整型; 车次为字符串型, 最长为 15, 不能为空; 出发时间和到达时间为时间型, 不能为空; 地点为字符串型, 最长为 15, 不能为空; 价格为浮点型, 不能为空。该表的表结构如表 3-2 所示:

表 3-2 出发车次信息表 (STra)

序号	逻辑字段	物理字段	类型 (长度)	规则	字段说明
1	Id	Id	INT	PK	主键
2	车次	T_id	VARCHAR(15)	not null	必填项
3	出发时间	stime	DATETIME	not null	必填项
4	到达时间	etime	DATETIME	not null	必填项
5	地点	site	VARCHAR(15)	not null	必填项
6	价格	price	FLOAT	Not null	必填项

(三) 回程车次表

回程车次信息表存储的是基于网络爬虫的 ACM 预算系统在 12306 网站上爬取的所有回程时火车车次的信息。其含有 id, 车次, 出发时间, 到达时间, 地点, 价格 6 个字段。Id 为主键, 整型; 车次为字符串型, 最长为 15, 不能为空; 出发时间和到达时间为时间型, 不能为空; 地点为字符串型, 最长为 15, 不能为空; 价格为浮点型, 不能为空。该表的表结构如表 3-3 所示:

表 3-3 回程车次信息表 (ETra)

序号	逻辑字段	物理字段	类型 (长度)	规则	字段说明
1	Id	Id	INT	PK	主键
2	车次	T_id	VARCHAR(15)	not null	必填项
3	出发时间	stime	DATETIME	not null	必填项
4	到达时间	etime	DATETIME	not null	必填项
5	地点	site	VARCHAR(15)	not null	必填项
6	价格	price	FLOAT	Not null	必填项

(四) 酒店信息表

酒店信息表存储的是基于网络爬虫的 ACM 系统在美团网上爬取的所有相关酒店的信息。其含有 id, 酒店名称, 地点, 价格 4 个字段。Id 为主键, 整型; 酒店名称为字符串型, 长度最大值为 50, 不能为空; 地点为字符串型, 长度最大值为 15, 不能为空; 价格为浮点型, 不能为空。该表的表结构如表 3-4 所示:

表 3-4 酒店信息表 (Acc)

序号	逻辑字段	物理字段	类型 (长度)	规则	字段说明
1	Id	Id	INT	PK	主键
2	酒店名称	name	VARCHAR(50)	not null	必填项
3	地点	site	VARCHAR(15)	not null	必填项
4	价格	price	FLOAT	not null	必填项

(五) 出发航班信息表

出发航班信息表存储的是基于网络爬虫的 ACM 预算系统在携程网站上爬取的所有出发时飞机航班的信息。其含有 id, 航班号, 出发时间, 到达时间, 地点, 价格 6 个字段。Id 为主键, 整型; 航班号为字符串型, 最长为 15, 不能为空; 出发时间和到达时间为时间型, 不能为空; 地点为字符串型, 最长为 15, 不能为空; 价格为浮点型, 不能为空。该表的表结构如表 3-5 所示:

表 3-5 出发航班信息表 (SAir)

序号	逻辑字段	物理字段	类型 (长度)	规则	字段说明
1	Id	Id	INT	PK	主键
2	航班号	A_id	VARCHAR(15)	not null	必填项
3	出发时间	stime	DATETIME	not null	必填项
4	到达时间	etime	DATETIME	not null	必填项
5	地点	site	VARCHAR(15)	not null	必填项
6	价格	price	FLOAT	not null	必填项

(六) 回程航班信息表

回程航班信息表存储的是基于网络爬虫的 ACM 预算系统在携程网站上爬取的所有回程时飞机航班的信息。其含有 id, 航班号, 出发时间, 到达时间, 地点, 价格 6 个字段。其中 id 为主键, 并且除主键外的其余字段均不能为空。该表的表结构如表 3-6 所示:

表 3-6 回程航班信息表 (EAir)

序号	逻辑字段	物理字段	类型 (长度)	规则	字段说明
1	Id	Id	INT	PK	主键
2	航班号	A_id	VARCHAR(15)	not null	必填项
3	出发时间	stime	DATETIME	not null	必填项
4	到达时间	etime	DATETIME	not null	必填项
5	地点	site	VARCHAR(15)	not null	必填项
6	价格	price	FLOAT	not null	必填项

(七) 初始预算信息表

初始预算信息表存储的在是 ACM 预算系统中创建的所有初始预算相关信息。它的字段有 id，初始预算的表 id，为主键；比赛名称、老师数量、队伍数量、女性数量、到达时间、回程时间、地点、补贴费都不能为空；出发车次、回程车次、出发航班、回程航班、酒店、创建人为外键，其中回程航班和出发航班可以为空。出发车次和回程车次分别关联出发车次表和回程车次表；出发航班和回程航班分别关联出发航班表和回程航班表；酒店和创建人分别关联酒店信息表和用户信息表。该表的表结构如表 3-7 所示：

表 3-7 初始预算信息表（Eleb）

序号	逻辑字段	物理字段	类型（长度）	规则	和报名字段说明
1	Id	Id	INT	PK	主键
2	比赛名称	name	VARCHAR(30)	not null	必填项
3	老师数量	tecnum	INT	not null	必填项
4	队伍数量	teanum	INT	not null	必填项
5	女性数量	fnum	INT	not null	必填项
6	到达时间	stime	DATETIME	not null	必填项
7	回程时间	etime	DATETIME	not null	必填项
8	地点	site	VARCHAR(15)	not null	必填项
9	补贴费	allo	FLOAT	default=100	默认值为 100
10	报名费	apply	FLOAT	default=1500	默认值为 1500
11	出发车次	Str_id	INT	FK not null	关联出发车次信息表、必填
12	回程车次	Etr_id	INT	FK not null	关联回程车次信息表、必填
13	出发航班	Sai_id	INT	FK	关联出发航班信息表
14	回程航班	Eai_id	INT	FK	关联回程航班信息表
15	酒店	Ac_id	INT	FK not null	关联酒店信息表、必填
16	创建人	Ad_id	INT	FK not null	关联用户信息表、必填

（八）最终预算信息表

设计说明：最终预算信息表存储的在是 ACM 预算系统中创建的所有最终预算相关信息。它的字段有 id，为最终预算表 id，是主键；比赛名称、老师数量、队伍数量、女性数量、到达时间、回程时间、地点、补贴费和报名费都不能为空；出发车次和回程车次是外键，分别关联出发车次表和回程车次表，且不能为空；出发航班和回程航班是外键，分别关联出发航班表和回程航班表，可以为空；酒店为外键，关联酒店信息表，不能为空；创建人为外键，关联用户信息表，不能为空。初始预算为外键，关联初始预算表，不能为空关联。最终预算表的表结构如表 3-8 所示：

表 3-8 最终预算信息表（Ultb）

序号	逻辑字段	物理字段	类型（长度）	规则	字段说明
1	Id	Id	INT	PK	主键
2	比赛名称	name	VARCHAR(30)	not null	必填项
3	老师数量	tecnum	INT	not null	必填项
4	队伍数量	teanum	INT	not null	必填项
5	女性数量	fnum	INT	not null	必填项
6	到达时间	stime	DATETIME	not null	必填项
7	回程时间	etime	DATETIME	not null	必填项
8	地点	site	VARCHAR(15)	not null	必填项
9	补贴费	allo	FLOAT	default=100	默认值为 100
10	报名费	apply	FLOAT	default=1500	默认值为 1500
11	出发车次	Str_id	INT	FK not null	关联出发车次信息表、必填
12	回程车次	Etr_id	INT	FK not null	关联回程车次信息表、必填
13	出发航班	Sai_id	INT	FK	关联出发航班信息表
14	回程航班	Eai_id	INT	FK	关联回程航班信息表
15	酒店	Ac_id	INT	FK not null	关联酒店信息表、必填
16	创建人	Ad_id	INT	FK not null	关联用户信息表、必填
17	初始预算	El_id	INT	FK not null	关联初始预算信息表、必填

（九）实际账单信息表

设计说明：实际单目信息表存储的在是 ACM 预算系统中创建的所有实际账单相关信息。该表的字段有 id，实际账单表 id，为主键；比赛名称，由年份和 CCPC 或者 ICPC 加上地点组成，不能为空；老师数量，必须大于零，不能为空；队伍数量，必须大于零，不能为空；女性数量，正整数，不能为空；到达时间和回程时间为到达比赛地点的时间和结束比赛准备返回的时间，由日期和时间组成，精确到分，不能为空；地点，比赛城市，不能为空；补贴费和报名费，分别默认为 100 和 1500，精确到小数点后一位，不能为空；出发车次和回程车次是外键，分别关联出发车次表和回程车次表，表示出发的火车信息和结束比赛的火车信息，不能为空；出发航班和回程航班是外键，分别关联出发航班表和回程航班表，表示老师有可能乘坐飞机时的来回飞机信息，因此可为空；酒店，外键，关联酒店信息表，表示比赛中居住的酒店信息，不能为空；创建人，外键，关联用户信息表，表示创建该实际账单的创建人，可与初始预算和最终预算不同，不能为空；最终预算，外键，关联最终预算表，不能为空；备注，为比赛意外项，不能为空，若无意外，可写无。该表的表结构如表 3-9 所示：

表 3-9 实际账单信息表（Account）

序号	逻辑字段	物理字段	类型（长度）	规则	字段说明
1	Id	Id	INT	PK	主键
2	比赛名称	name	VARCHAR(30)	not null	必填项
3	老师数量	tecnum	INT	not null	必填项
4	队伍数量	teanum	INT	not null	必填项
5	女性数量	fnum	INT	not null	必填项
6	到达时间	stime	DATETIME	not null	必填项
7	回程时间	etime	DATETIME	not null	必填项
8	地点	site	VARCHAR(15)	not null	必填项
9	补贴费	allo	FLOAT	default=100	默认值为 100
10	报名费	apply	FLOAT	default=1500	默认值为 1500
11	出发车次	Str_id	INT	FK not null	关联出发车次信息表、必填
12	回程车次	Etr_id	INT	FK not null	关联回程车次信息表、必填
13	出发航班	Sai_id	INT	FK	关联出发航班信息表
14	回程航班	Eai_id	INT	FK	关联回程航班信息表
15	酒店	Ac_id	INT	FK not null	关联酒店信息表、必填
16	创建人	Ad_id	INT	FK not null	关联用户信息表、必填
17	最终预算	Ul_id	INT	FK not null	关联初始预算信息表、必填
18	备注	comm	VARCHAR(300)	not null	必填项

（十）竞赛账单信息表

设计说明：竞赛账单信息表存储的在是 ACM 预算系统中创建的所有预算和账单相关信息。该表的表结构如表 3-10 所示：

表 3-10 竞赛账单信息表（CName）

序号	逻辑字段	物理字段	类型（长度）	规则	字段说明
1	Id	Id	INT	PK	主键
2	比赛名称	name	VARCHAR(30)	not null	必填项
3	初始预算	el_id	INT	FK not null	关联初始预算信息表、必填
4	最终预算	ul_id	INT	FK	关联最终预算信息表
5	实际账目	ac_id	INT	FK	关联实际账目信息表

4 ACM 预算系统模块具体实现

4.1 用户信息功能具体实现

用户通过账号登录系统后，进入主页面，点击侧导航栏上的用户信息，进入用户信息页面，系统将会以表格的形式展示所有用户信息。教练只可看见用户用

户名和密码，且只能查询，不能做其它操作。总教练可看见所有信息，可对用户进行注册，修改，注销等功能。用户信息主页面如图 4-1、4-2 所示：

+ 注册

× 注销

Q 查询

<input type="checkbox"/>	用户名	密码	用户类型
<input type="checkbox"/>	lyr	123	总教练
<input type="checkbox"/>	lsm	123	教练
<input type="checkbox"/>	ldd	123	教练
<input type="checkbox"/>	lsls	321	总教练

图 4-1 用户信息页面--总教练

Q 查询

用户名	用户类型
lyr	总教练
lsm	教练
ldd	教练
lsls	总教练

图 4-2 用户信息页面--教练

用户登录：系统初始页面为用户登录页面，只有用户登录以后，才能进入系统。用户登录页面中，用户具体信息位于正中，含有用户密码，用户名输入框。用户名输入框含有不为空，最小长度为 3，最大长度为 15 的限制。密码输入框含有不为空，最小长度为 3 最大长度为 20 的限制。用户输入相关信息，点击登录，若登录失败，将提示错误信息。错误信息含有“用户名不存在”和“密码错误”两种。在登录按钮左边含有重置按钮，点击重置，可清空用户名和密码输入框。用户登录成功后，进入系统主页。系统主页中含有导航栏和侧导航栏，导航栏左边写有 ACM 预算系统，点击它即可进入主页。导航栏右边，则为初始预算、最终预算、实际账单、其它菜单。主页侧导航栏上方，显示用户头像，下方有“欢迎您”的信息，再下方有修改和登出信息。侧导航栏的下边位置依次为用户信息、出发车次、回程车次、酒店、出发航班、回程航班菜单。主页主要内容为竞赛账单列表。只要存在竞赛账单，将会列出相关的预算和实际账单名称。

用户注册：总教练登录以后，进入主页，点击侧导航栏上的用户信息，进入用户信息页面。用户信息页面含有一张表格，表格里显示了所有用户的所有信息，

在表格左上方，含有注册按钮。点击注册按钮，系统弹出用户注册模态框。其中含有用户名、密码输入框，总教练单选框。其中用户名含有长度最小为 3 最大为 15 的条件；密码含有长度最小为 3 最大为 20 的条件。填写信息，点击提交，若用户名已存在，将会返回用户注册模态框，并警告该用户名已存在。成功注册后注册模态框消失，用户信息页面刷新。

用户修改：总教练登录以后，进入主页，点击侧导航栏上的用户信息，进入用户信息页面。用户信息页面含有一张表格，表格里显示了所有用户的所有信息。点击表格中需要修改的信息，修改后，回车，页面将刷新，相关信息已被修改。用户信息修改成功。

用户注销：总教练登录以后，进入主页，点击侧导航栏上的用户信息，进入用户信息页面。用户信息页面含有一张表格，表格里显示了所有用户的所有信息。表格前方含有单选框，用户可点击表头的单选框，实现用户的全选或取消全选。点击所有需要注销的用户的单选框，然后点击注销按钮，页面刷新，表格中不再含有相关页面信息，用户注销成功。但不可注销自己和已创建预算和账单的用户。

4.2 初始预算功能具体实现

初始预算模块含有初始预算页面，其以列表的形式展示了所有的初始预算信息，点击其中的某一条，即可查询该初始预算表。在初始预算信息列表上方添有查询和创建的选项。用户创建新的初始预算，后台将会生成相关竞赛账目表。

用户可进入某个初始预算表中修改信息或删除这条初始预算，但不能删除和最终预算关联的预算表。

初始预算查询：用户点击进入初始预算页面，初始预算页面主要是以列表的形式展示了所有初始预算。在初始预算列表含有创建按钮，创建按钮右边，则为年份下拉框、城市和比赛名称输入框。输入框右边有查询按钮。通过年份的选择、城市和比赛名称的输入，可查询所有符合情况的初始预算，其中，比赛名称的优先级别最高，即若比赛名称不为空，查询的初始预算一定是该比赛的初始预算。

初始预算删除：用户进入主页后，点击需要删除的初始预算，进入初始预算信息页面。初始预算信息页面以表格的形式显示了该初始预算的具体信息。表格上方，含有创建按钮，在表格下方，含有删除和提交按钮。用户点击删除按钮，若页面返回初始预算主页面，初始预算列表中不再含有该初始预算，即用户删除该预算成功。

初始预算创建：用户在初始预算页面中点击创建，进入初始预算创建页面中。初始预算创建页面含有比赛类型 ICPC 和 CCPC 的选择，含有老师、队伍、女性数量、补贴及报名费的输入，含有出发时间和回程时间及地点的下拉框。点击提交，若初始预算创建成功，将进入建立的初始预算具体信息页面。否则，停留在初始预算创建页面，并含有错误提示。其关键代码如下：

```

def cre(request):
    if request.method == 'POST':
        elebform = CreForm(request.POST)    #前端数据渲染与获取前端数据
        if elebform.is_valid():
            ele = Eleb.objects.model()        #创建初始预算实例
            ele.tecnum = elebform.cleaned_data['teachernum']    #老师数量赋值
            ele.teanum = elebform.cleaned_data['teamnum']        #队伍数量赋值
            ele.fnum = elebform.cleaned_data['girlnum']    #女性数量赋值
            stime = elebform.cleaned_data['starttime']    #到达时间赋值
            etime = elebform.cleaned_data['endtime']    #回程时间赋值
            ele.allo = elebform.cleaned_data['allo']    #补贴费赋值
            ele.apply = elebform.cleaned_data['apply']    #报名费赋值
            (ele.name, site, stime, etime) = get_name(name, site, stime, etime)
                                #获取比赛名、地点、时间

            ele.site = site
            creTra(stime[0], etime[0], site)    #12306 火车车票爬虫
            creAir(stime[0], etime[0], site)    #携程机票爬虫
            ceAcc(stime[0], etime[0], site)    #美团酒店爬虫
            stime = stime[0] + " " + stime[1]    #日期时间写在一起
            etime = etime[0] + " " + etime[1]    #日期时间写在一起
            ele.stime = datetime.strptime(stime, "%Y-%m-%d %H:%M") #到达时间 str->datetime
            ele.etime = datetime.strptime(etime, "%Y-%m-%d %H:%M") #回程时间 str->datetime
            (train, price1, price2, accc) = get_price(ele.stime, ele.etime, site, ele.allo)
                                #获取交通信息，不同交通价格，酒店信息

            ele.ai_id = LoginAdmi.objects.all()[0].user    #登录用户
            ele.str_id = train['st_train']    #出发车次信息
            ele.etr_id = train['end_train']    #回程车次信息
            ele.ac_id = accc    #酒店信息
            if(price1 > price2):    #不同交通工具价格比较
                ele.sai_id = sai_id, ele.eai_id = eai_id
            ele.save()    #创建初始预算
            ele = gt_ele(ele.name, ele)
            return render_to_response('e_select.html', {'ele': ele})    #转到初始预算具体页面
        else:
            elebform = CreForm()
            return render_to_response('create.html', {'elebform': elebform})    #停留本页面

```

4.3 最终预算功能的具体实现

用户通过初始预算在表格中修改或输入相关数据进一步进行预算。每个最终预算都可以进行修改，删除，查询等操作。但修改只可修改创建时可输入的对象，同时在创建的时只能根据已存在并且无最终预算的比赛的最终预算，删除时，不

能删除已关联实际账目的最终预算。最终预算主页面如图 4-3 所示：

初始预算

最终预算

实际账单

创建

年份: 没有选中任何项

城市:

比赛名称:

查询

2019CCPC北京站最终预算表

2019ICPC乐山站最终预算表

图 4-3 最终预算主页面

最终预算创建：用户进入最终预算主页面，点击创建下拉框按钮，将弹出所有没通过它创建过的最终预算的初始预算。若无该种初始预算，将会弹出“请先创建初始预算哦”的信息。点击某一项初始预算，进入最终预算创建页面，根据实际情况，修改老师数量、队伍数量、女性数量、队伍车次选择、老师车次或航班选择、队伍或老师酒店选择、报名费、补贴费。然后点击表格右下方的提交按钮，进入最终预算具体信息页面，即创建最终预算成功。

4.4 实际账单功能的具体实现

实际账单是依托与最终预算，通过选择最终预算，进行实际账单的创建。在实际账单模块中，用户可对实际账单进行增删改查操作。实际账单模块还拥有对实际账单通过年份进行年度总结。实际账单的主页面如图 4-4 所示：

创建

年度总结

年份: 没有选中任何项

城市:

比赛名称:

查询

2019CCPC北京站实际账单

图 4-4 实际账单主页面

查看实际账单具体信息：用户进入实际账单主页面，点击需要查看的实际账单，即可进入该实际账单信息页面。实际账单信息页面以表格的形式显示了实际账单的具体信息，在表格右上方含有创建按钮，表格右下方有提交和删除按钮。可以使用户对实际账单进行修改或删除。表格中将老师和队伍分开，显示了他们分别的交通费用，酒店费用以及总费用。

修改实际账单具体信息：用户进入实际账单主页面，点击需要修改的实际账单，即可进入该实际账单相关信息页面。可修改的信息有：老师数量、队伍数量、女性数量、队伍的出发车次和回程车次、老师的交通工具选择，若选择航班，则需选择相关的航班信息，若选择车次，则需选择相关的车次信息、酒店信息、报名费、补贴费和备注。修改这些信息，点击提交。页面刷新，相关信息已被修改，

表示实际账单信息修改成功。

查看年度总结：用户进入实际账单主页面，点击年度总结，弹出所有实际账单的年份。若无时间账单，将会弹出“请先创建实际账单哦的”的信息。点击任意一项年份，进入年度总结页面。其中含有这一年所有比赛的队伍每场的交通费用、酒店费用、报名费用、总费用、老师的交通费用、酒店费用、补贴费用、总费用、比赛总费和这一年的总费用。其关键代码为：

```
def accountyear(request,year):
    elelist = Account.objects.all(),eles = [], prices = 0
    for ele in elelist:
        if(ele.name[0:4] == year):
            price = {}
            pri_acc = (ele.etr_id.stime - ele.str_id.etime).days * ele.ac_id.price      #酒店价格
            price['stu_accpri'] = ((ele.fnum + 1) // 2 + (ele.teanum*3 - ele.fnum + 1) // 2) * pri_acc
            price['stu_traionpri'] = (ele.str_id.price + ele.etr_id.price) * ele.teanum * 3
            price['stu_apply'] = ele.apply * ele.teanum      #学生报名费
            price['te_traionpri'] = (ele.str_id.price + ele.etr_id.price) * ele.tecnum
            te_acc = ele.tecnum // 2, te_day = (ele.etr_id.etime - ele.str_id.stime).days + 1
            if ele.tecnum % 2:
                if (ele.teanum * 3 - ele.fnum) % 2 == 0:
                    te_acc += 1      #老师需要的酒店房间数
            if ele.sai_id != None:
                pri_acc = (ele.eai_id.stime - ele.sai_id.etime).days * ele.ac_id.price
                te_day = (ele.eai_id.etime - ele.sai_id.stime).days + 1      #老师外出天数
                price['te_traionpri'] = (ele.eai_id.price + ele.sai_id.price) * ele.tecnum * 0.5
            price['te_accpri'] = te_acc * pri_acc      #老师酒店费用
            price['te_allo'] = ele.allo * te_day * ele.tecnum      #老师补贴费
            price['te_price'] = price['te_traionpri'] + price['te_accpri'] + price['te_allo']
            price['stu_price'] = price['stu_traionpri'] + price['stu_accpri'] + price['stu_apply']
            price['pri'] = price['te_price'] + price['stu_price']      #比赛总费用
            eles.append({'name', ele.name, 'price', price})
            prices += price['pri']
    return render_to_response('accyear.html', {'eles':eles, 'prices':prices})
```

4.5 其它信息功能的具体实现

其它信息包含了出发车次和回程车次，出发航班和回程航班，酒店信息。通过点击导航栏上的其它，弹出下拉框，选择某一项，将进入某一项的所有信息查看页面。也可在侧导航栏中选择出发车次和回程车次、出发航班和回程航班、酒店，并且用户进入系统后，在任意一个页面中，都可以进行选择操作比如选择出发车次，将进入以表格的形式展现所有出发车次信息的页面。在每个页面里，用

户都可以对相对的信息进行添加，删除，查询操作。其菜单选择如图 4-5 所示：

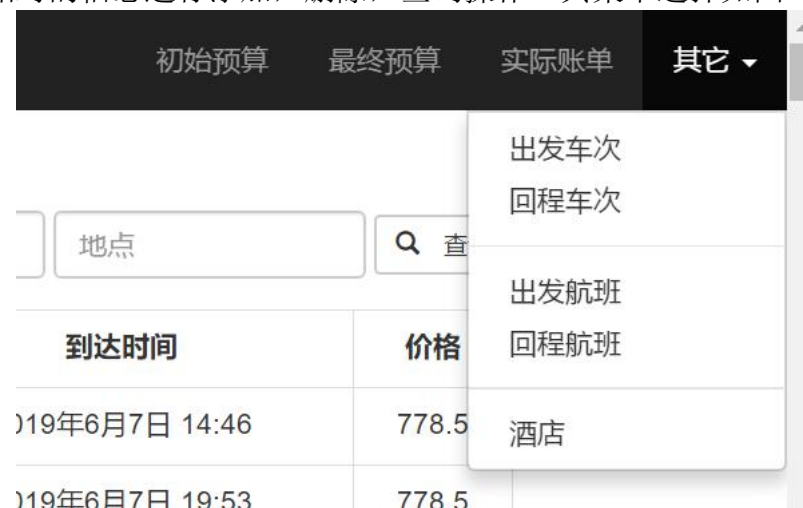


图 4-5 其它信息菜单栏选择

查看回程航班信息：用户登录系统后，进入主页，在导航栏上点击其它，弹出下拉框。点击回程航班，或点击侧导航栏中的回程航班，即可进入回程航班信息页面。回程航班信息页面上方含有到达时间、地点、航班号输入框。填写相关信息，点击查询，即可查询相关的回程航班信息。回程航班页面如图 4-6 所示：



图 4-6 回程航班信息页面

添加酒店信息：用户登录系统后，进入主页，在导航栏上点击其它，弹出下拉框。点击酒店，或点击侧导航栏中的酒店，即可进入酒店信息页面。酒店信息页面上方含有添加按钮。点击添加，弹出酒店添加模态框，填写酒店名称、地点、价格信息，点击保存。

5 ACM 预算系统模块运行及测试结果

5.1 系统测试方法

本系统在测试过程中主要是在单元测试下运用黑盒测试。

单元测试是指在系统使用过程中，对系统中的某一个小模块进行测试。黑盒

测试是指在系统完成开发后，不考虑系统中的结构与代码，将自己作为用户，直接测试完成后的功能。在测试过程中，与用户一样，只能看到输出和输入，查看输出是否是预期效果。

5.2 测试用例及结果

用 ACM 预算系统最终预算管理中的添加最终预算、删除最终预算、修改最终预算、通过城市查询最终预算为例说明本系统的测试用例及结果。

添加最终预算用例如表 5-1 所示：

表 5-1 添加最终预算信息用例

模块名	添加最终预算信息		
用例作者	龙雅容	BUG 编号	
测试员	龙雅容	测试日期	2015.5.10
测试类型	功能测试	测试工具	无
用例 ID	001		
用例描述	用户登录系统，进入主页，点击最终预算，进入最终预算页面。点击创建，弹出下拉框，选择初始预算，进入最终预算创建页面。修改相关信息，提交表格。		
前驱条件	系统正常运行，用户进入最终预算信息页面，且存在无最终预算的初始预算。		
实际结果	转到最终预算信息表页面，展示创建好的最终页面		
备注	无		

删除最终预算用例如表 5-2 所示：

表 5-2 删除最终预算信息用例

模块名	删除最终预算信息		
用例作者	龙雅容	BUG 编号	
测试员	龙雅容	测试日期	2015.5.10
测试类型	功能测试	测试工具	无
用例 ID	002		
用例描述	用户登录系统，进入主页，点击最终预算，进入最终预算页面。点击需要删除的列，进入最终预算表页面，点击删除。		
前驱条件	系统正常运行，用户已登录系统，进入最终预算信息页面，存在最终预算		
实际结果	转到最终预算页面，无以删除的最终预算的列		
备注	无		

修改最终预算用例如表 5-3 所示：

表 5-3 修改最终预算信息用例

模块名	修改最终预算信息		
用例作者	龙雅容	BUG 编号	
测试员	龙雅容	测试日期	2015.5.10
测试类型	功能测试	测试工具	无
用例 ID	003		
用例描述	用户登录系统，进入主页，点击最终预算，进入最终预算页面。点击需要修改的最终预算，进入最终预算表页面，点击需要修改的信息进行修改，然后点击提交。		
前驱条件	系统正常运行，用户已登录系统，进入最终预算信息页面，存在最终预算		
实际结果	最终预算信息页面被刷新，相关数据已被修改		
备注	无		

通过城市查询最终预算用例如表 5-4 所示：

表 5-4 通过城市查询最终预算信息用例

模块名	查询最终预算信息		
用例作者	龙雅容	BUG 编号	
测试员	龙雅容	测试日期	2015.5.10
测试类型	功能测试	测试工具	无
用例 ID	004		
用例描述	用户登录系统，进入主页，点击最终预算，进入最终预算页面。在城市输入框中，点击查询。		
前驱条件	系统正常运行，用户进入最终预算信息页面，存在需要查询的最终预算		
实际结果	最终预算页面只显示需要查询的最终预算		
备注	无		

5.3 测试总结

通过测试可知，本系统基本功能已经完成，实际运行效果与预期结果一致。因此，基于网络爬虫的 ACM 预算系统符合要求。

结 束 语

本系统根据 ACM 预算系统实际需求，以 Django 为框架，设计了一个实时的预算系统。具体实现了 ACM 负责人在此系统上输入相关信息，自动生成预算，并且可以根据赛前车票、住宿、人数等具体确定实现进一步预算。在赛后还可根据预算信息，修改和输入确定的信息，使账目清晰明了。对于 ACM 负责人来说，这是一个很好的功能，它让每一次比赛的账单信息系统化，特别是年度账目总结，不必在东凑西找之前的账目，直接点击，信息马上出现。

在开发过程中，我曾经推翻过最开始的数据库设计，重新设计新的数据库。在此过程中，我深刻的理解到数据库的合理设计对于实现相关功能的重要性。尤其是在本系统的数据库精密相关，步步紧扣的情况下，数据库的设计就更加重要了。

参考文献

- [1] 杨莉, 杨福华, 吕峰.用 ACM 促进程序设计课程教学[J].中国新通信, 2018,05(9)
- [2] 候向辉, 徐卫, 李琰琰.以 ACM/ICPC 竞赛促进计算机专业实践教学研究[J].计算机时代, 2018, 15(7)
- [3] 赵龙, 李贝贝, 王喜凤.ACM/ICPC 竞赛对培养大学生综合能力的作用[J].安徽工业大学学报(社会科学版), 2013,07(5)
- [4] 赖敬之.基于网络爬虫的新浪微博数据分析网站的建议[J].通信设计与应用, 2018,2
- [5] 唐琳, 董依萌, 何天宇.基于 Python 的网络爬虫技术的关键性问题探索[J].探索与观察, 2018。
- [6] 翟高粤.基于 Python 的数据分析概述[J].甘肃科技纵横, 2018(11):5-7+26
- [7] 严斐, 肖璞.Python 框架下基于主题的数据爬取技术研究 with 实现[J].计算机时代, 2018(11):10-13
- [8] 张敏, 李野.基于网络爬虫的少数民族文学对外传播成效研究[J].西南大学学报(自然科学版), 2019,45(02):173-177
- [9] 陈智, 梁娟, 谢兵, 傅篱.新浪微博数据爬取研究[J].智能处理与应用, 2016
- [10] 刘顺程, 岳思颖.大数据时代下基于 Python 的网络信息爬取技术.数据库技术.
- [11] Wikipedia.Web crawler[DB/OL].https://en.wikipedia.org/wiki/Web_crawler, 2018
- [12] Python Software Foundation.About Python[EB/OL].<https://www.python.org/about/>, 2018
- [13] 张文静.互联网对大众传播的影响[D].开封: 河南大学, 2015
- [14] 王红崧, 周海晏.基于百度地图 API 的旅游地理信息系统开发[J].现代计算机(专业版), 2012(23):60-6
- [15] 朱婷, 赵婷婷.基于百度地图 API 的网点成图平台的开发和应用[J].测绘地理信息, 2019,44(01),121-123+126
- [16] 徐青.JavaScrip 恶意代码检测技术研究[D].成都:西南交通大学, 2014
- [17] 向光军.基于 Django 的数据持久化研究[J].楚雄师范学院学报, 2013-06-20

致 谢

在论文完成过程中，本人还得到了朱毅老师和杜晓宇老师的热情指导。无论是选题的可行性，还是时间的安排以及具体需求的了解。这两位老师都从自身了解方面给与了我极大的支持。在遇到困难时，这两位老师都主动帮我，给予我可行的建议。由于自身自制力较差，这两位老师都在一定程度上监督我完成这次的毕业设计，我对此表示衷心的感谢。

最后向在百忙之中评审本文的各位专家、老师表示衷心的感谢！

作者简介：

姓 名： 龙雅容

性别： 女

出生年月： 1996 年 07 月 19 日

民族： 汉

E-mail: 920330289@qq.com

声 明

本论文的工作是2018年 10 月至2019年5月在成都信息工程大学软件工程学院完成的。文中除了特别加以标注地方外，不包含他人已经发表或撰写过的研究成果，也不包含为获得成都信息工程大学或其他教学机构的学位或证书而使用过的材料。

关于学位论文使用权和研究成果知识产权的说明：

本人完全了解成都信息工程大学有关保管使用学位论文的规定，其中包括：

(1) 学校有权保管并向有关部门递交学位论文的原件与复印件。

(2) 学校可以采用影印、缩印或其他复制方式保存学位论文。

(3) 学校可以学术交流为目的复制、赠送和交换学位论文。

(4) 学校可允许学位论文被查阅或借阅。

(5) 学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

除非另有科研合同和其他法律文书的制约，本论文的科研成果属于成都信息工程大学。

特此声明！

作者签名：

年 月 日