

基于 Django 的留学生信息管理系统的设计与实现

葛宇航

(西南林业大学 大数据与智能工程学院 昆明 650051)

摘要: 随着本校师资力量扩大, 教育改革不断深化, 使得留学生数量不断攀升, 在日常学生管理工作当中, 一个高效、方便、安全的平台显得十分尤为重要。本文基于 Django 这一 Web 开发框架, 构建了一个较为完善的留学生信息管理系统, 实现的功能包括师生基本信息管理、班级管理、成绩录入与查询、考试管理等等, 借助 Django 在 Web 开发中的强大优势, 完善的 ORM 操作、丰富的功能模块、强大的数据处理、方便的 url 路由功能, 快速高效地完成了项目开发, 并上线使用。

关键词: Django; Python; 留学生管理系统; B/S 架构; MVT

中图分类号: 99999999

文献标识码: 88888888

The design and implementation of a Django based oversea student information management system

GE Yuhang

(College of Big Data and Intelligence Engineering, Southwest Forestry University, Kunming 650224, Yunnan, China)

Abstract: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Key words: Python; Django; MIS

引言

Web2.0 时代, 移动平台同时大为盛行, 一个可适应多平台, 并且能够快速进行开发的技术才能适应潮流。Python 是一个简单的、

解释性、可交互、可移植、面向对象的高级编程语言, 用于 Web 开发尤为合适, 它在软件开发、维护、调试、优化、部署等各个生命周期中都有十分高的效率。Django 作为 Python Web 开发中最为流行的应用框架, 更

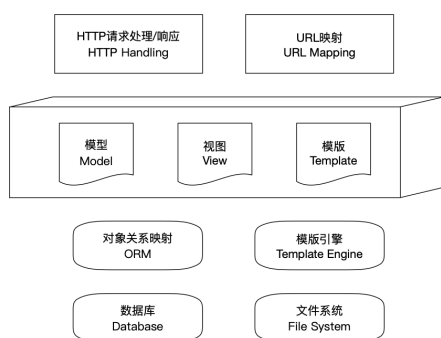


图 1 Django 架构总览图

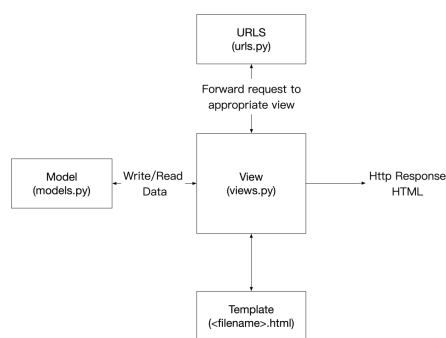


图 2 Django MVT

是将 Python 的优势发挥到极致，它安装简单且灵活，使用方便，能够开箱即用。遵循 MVC 开发模式，Django 中内置了很多 Web 开发直接能使用的模块，同时集成了一个轻量级的 Webserver，能够方便地在本地进行调试。当下有许多著名的站点使用 django 进行开发，解释型语言开发应用也越来越流行，基于 Django 这一高效框架，本文实现了一个功能较为完善的留学生信息管理系统。

1 关键技术介绍

1.1 Django 框架

Django^[1,2] 是 Python 中使用率最高的 Web 框架，它拥有良好的社区环境，完善的版本迭代机制。遵循 MVC 的软件设计模式，用它可以快速、方便地开发出一个完整的 Web 应用。Django 框架的核心，如图 1 所示，包括一个轻量级的 Web 服务器，用于接受 HTTP 请求，一个基于正则表达式的 URL 分发器，一个数据库模型用于建立数据模型与数据库相映射，一个视图系统用于处理请求，以及一个模版系统。这种层次明晰的框架设计，在实际生产中极大地有利于应用软件的设计与开发。

1.2 Django MVT

Django 的 MTV 设计模式（图 2）类似于标准的 MVC，包括四个模块：

- 1) **Models.py** 用于创建数据库模型，是对数据库的上层封装，大大简化了编码过程中对数据库的增删改查操作，与 MVC 中的 Model 功能类似。
- 2) **Views.py** 是主要的功能模块，负责业务逻辑处理，与 Template 进行数据交换，与 MVC 中的 Control 功能类似。
- 3) **Templates** 文件夹中的保存的模版文件，用于生成最终 HTML 页面，与 MVC 中的 View 功能类似。
- 4) **url.py** 则用于定义整个系统或某个子模块的路由表，指定了 URL 与 **views.py** 的映射关系。

urls.py 根据用户发起的请求，调用 **views.py** 中对应的函数，与数据模型以及模版进行交互，响应用户请求。

1.3 Bootstrap

Bootstrap^[3] 是当下最流行的开源前端开发框架，有响应式布局、移动平台优先的特

性,为此十分适合本项目的前端构建。用户可以直接使用 Bootstrap 提供的 CSS 样式表,方便、快捷地构建出自己想要的布局,更重要的是 Bootstrap 同时拥有大量的插件,设计开发的系统大部分业务需要进行表格处理,Bootstrap table 插件刚好可以满足系统的需求,bootstrap table 基于 jquery^[4] 对表格处理的基本逻辑进行上层封装,开发者只需要对相应的 javascript 代码进行简单的配置与参数传递,就可以完成单选、多选、排序、分页、编辑、过滤、导入导出等常见功能,无需进行二次开发,大大减少的开发者的实际工作量。

1.4 MySQL

MySQL^[5] 有着高性能、成本低、可靠性好,在过去成为最流行的关系型数据库。MySQL 使用 C 和 C++ 语言编写,具有可移植性,为包括 Python 在内的大量编程语言提供了 API, Linux 下运行极为流畅,支持多线程。开源免费的特点使得其受到中小企业的欢迎,它的这些优点,使其对于本系统的开发尤为合适。

2 系统架构分析

本系统主要开发目的是以班级为单位的留学生个人信息、课程和成绩等信息的管理,增加工作效率,提高管理能力。同时让学生可以更便捷地查询自己的成绩,本文吸取了大量相关学生信息管理系统开发的经验,采用了 B/S 架构的设计思想,基于强大开源的 Python 的 Web 应用框架 Django 进行二次开发,采用 MVT 的软件设计模式,即模型 Model,视图 View 和模板 Template。使得维护起来较为方便,高内聚低耦合。实现了相对完善的功能,其内容不仅涉及对学生资料

进行管理分发,还包括对学生成绩信息,考试信息,教师授课,成绩录入,综合评价等等,已投入实际使用,使得留学生管理工作趋于科学化、高效化、规范化。

3 系统架构设计分析

3.1 系统用户设计

本系统面向的用户分为学生、教师、管理员等不同角色,每个角色权限与功能各不相同,用户输入用户 ID 和密码,进行系统登录,后台通过用户 ID 对应的用户类型匹配返回的页面,实现角色分离。

3.1.1 管理员

管理员拥有最高的权限,包括班级管理、教师管理、课程管理、学生管理、考试管理、学生名单的导入和学生成绩的导出(以班级为单位)等,管理员在考试管理页面考试科目相关设置,教师页面才能显示出考试科目,完成考试后,对成绩进行录入。管理员也可以对成绩进行查询和统计等操作。

3.1.2 教师

教师对考试成绩的录入,点击对应考试最后一栏的成绩录入,即可获取参加该考试的所有学生,首先按照平时、期中、期末分配成绩比例,然后再依次进行成绩录入,录入完成后,点击提交按钮即可保存,此后提交按钮将变成无法点击状态,无法进行第二次提交操作。

3.1.3 学生

学生页面直接进行成绩查询,选择要查询的学年和学期,点击 Search 按钮,即可查

询到自己的成绩数据。当然，如果需要导出，也可通过工具栏上的导出工具进行成绩导出操作。

3.2 主要功能模块

如图 3 所示, 系统的主要功能模块包括:

登录功能： 系统入口，按角色进入不同页面，密码加密保存于数据库，登录时采用同样的加密算法，将用户输入的密码加密后与数据库比对，验证合法即可完成认证，成功跳转；

角色管理：分为管理员，教师，学生等三个角色，便于控制用户各自权限；

- 1) 管理员: 拥有最大权限, 班级管理, 成绩管理, 学生管理, 教师管理, 考试管理, 考试成绩管理等, 同时管理员可以给普通用户分配 ID;
- 2) 教师: 对考试成绩进行录入, 查看与修改个人信息, 管理学生信息, 查看、统计学生成绩;
- 3) 学生: 对考试成绩进行查询, 查看与修改个人信息;

班级管理： 添加、修改、删除班级基本信息，
查看班级人员详情；

教师管理： 添加、修改、删除教师基本信息，分配教师工号，用于登录；

课程管理： 添加、修改、删除课程基本信息，
安排任课教师并分配班级；

考试管理： 添加、修改、删除一场考试信息，
设置考试时间、类型、科目、班级等；

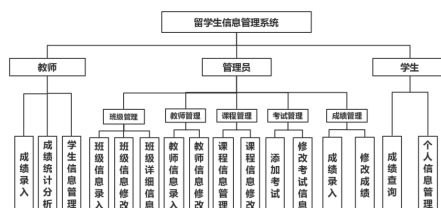


图 3 系统功能模块设计图

成绩管理： 管理员对考试成绩进行修改和录入，不限次数，教师只能对考试成绩进行录入无法修改，且只能录入一次。

3.3 数据库连接

本系统使用 Mysql 作为后台数据库，首先安装 mysqlclient 类库，作为 django 与 mysql 进行连接驱动，然后在 setting.py 中配置数据库，以下是配置：

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'studSys',
        'USER': 'root',
        'PASSWORD': '123456',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}
```

还可以设置其他数据库, 如 `SqlServer`, `PostgreSQL`, 只需修改数据库驱动 `ENGINE` 参数即可。配置后使用如下命令来测试是否连接成功。

```
python manager.py dbshell
```

根据数据库表结构创建相应的 `models.py`, 并将所属 `app` 添加入 `settings.py` 配置中。编写完成后使用如下命令检查是否存在语法错误,

```
python manage.py validate
```

若无语法错误, 使用如下命令迁移并同步数据库:

```
python manage.py makemigrations
python manage.py migrate
```

3.4 前后端数据交互

主要步骤:

- 1) 按照数据库设计, 完成 `model.py` 模型文件编写 (模型与数据库表一一映射, 每个模型都是一个 Python Class, 每个模型类属性都相当于一个字段, `model` 相当于提供一个访问数据库的 API);
- 2) 将前端页面构建好, 放入 `templates` 模版目录中;
- 3) 完成 `url` 的编写, 对应即将使用的视图 `views` 文件;
- 4) 编写 `views` 文件, 完成业务逻辑函数, 渲染模版前端文件;

下面以获取学生表数据为例分析代码:

```
urls.py:
urlpatterns = [
    url(r'data', std.getData)
]
```

前端发起查询的 Ajax^[6] 请求, `url` 路由通过匹配到视图下 `getData()` 方法:

```
views.py:
@csrf_exempt
def getData(request):
    '''获取数据'''
    if request.method == 'POST':
        '''ORM 查询获取到所有的学生数据模型对象'''
        Stud = StudTable.objects.all()
        dlen = len(Stud)
        if Stud:
            '''遍历模型对象, 获取具体数据'''
            for row in Stud:
                result = {}
                result['num'] = row.stuID_id
                result['ch_name'] = row.stuChName
                .....
            jsonData.append(result)
mydata = {
    "total": dlen,
    "rows": jsonData
}
return JsonResponse(mydata)
```

定义数据长度 `dlen`, 使用一个数组 `jsonData` 来保存数据。`getData()` 方法判断为 `POST` 方法后, 进行 `ORM` 查询操作 (`StudTable.objects.all()`) 从 `StudTable` 表中查询到所有的数据对象, 遍历对象依次查询属性, 以字典的方式保存, 再添加到 `jsonData` 中 (目的是为了与 `bootstrap table` 的数据接口一致, 代码块中 `mydata` 即为标准数据接口, 带 `total` 和 `rows` 两个属性, 前者为数据长度, 后者为实际传输的数据字典)。获取到数据后再以 `json` 格式传回前端 `bootstrap table` 中, 即可完成数据填充。

对学生数据的添加、删除、修改等操作的步骤类似, 此处不做详细叙述。

4 实现结果展示和分析

4.1 运行环境介绍

- 1) 硬件环境。使用阿里巴巴提供的云服务器 ECS；
- 2) 软件环境。操作系统是 Debian GNU/Linux 的 Testing 版。基于 Linux 的服务器平台相比 Windows Server 更加稳定，也易于配置。使用 Nginx 做前端 HTTP 服务器，uWSGI 作为中间件，通过它提供的 API 与后端 Django 应用通信，并与前端 Nginx 相链接；
- 3) 后台数据库。本系统采用 mysql5.6 作为数据库；
- 4) 系统结构。本系统采用 B/S 模式，客户端只需要一个浏览器就可以方便地使用，对用户硬件要求较低，可行性高；

4.2 主要界面展示

下面是主要界面的展示，应文章篇幅原因，只展示部分页面，图 4 是登录界面，图 5 管理员登录，左边是可以操作的所有权限列表，包括班级管理、学生管理、教师管理、课程管理、成绩管理等。

图 6 是管理员进行成绩管理的详情界面，可以进行包括筛选、修改、录入成绩等操作。图 7 是对学生个人信息进行管理的界面，可以直接通过行内编辑后自动保存新内容，图 8 是学生端登录后查询成绩的页面。

5 结束语

本系统主要解决了对留学生信息管理的问题，使得原本繁杂、重复、无序的工作变得

Login to your account

☐ Remember me

图 4 登录页面



图 5 管理员界面

成绩管理

学号: 课程: 平均分: 平均分: 平均分: 平均分:

学号	课程	平均分	平均分	平均分	平均分
201801001	Engineering Mathematics	73	0	92	94
201801002	Engineering Mathematics	69	0	21	46
201801003	Engineering Mathematics	79	0	79	79
201801004	Engineering Mathematics	73	0	97	97
201801005	Engineering Mathematics	100	0	100	100
201801006	Engineering Mathematics	69	0	66	67
201801007	Engineering Mathematics	67	0	97	95
201801008	Engineering Mathematics	64	0	53	65
201801009	Engineering Mathematics	73	0	97	97
201801010	Engineering Mathematics	0	0	0	0

Showing 1 to 10 of 49 rows | 10 rows per page

图 6 管理员成绩管理

学生基本信息

学号	姓名	出生日期	性别
201801001	男	1999-01-01	计算机科学与技术(1)
201801002	男	1999-01-01	计算机科学与技术(1)
201801003	男	1999-01-01	计算机科学与技术(1)
201801004	男	1999-01-01	计算机科学与技术(1)
201801005	男	1999-01-01	计算机科学与技术(1)
201801006	男	1999-01-01	计算机科学与技术(1)
201801007	男	1999-01-01	计算机科学与技术(1)
201801008	男	1999-01-01	计算机科学与技术(1)
201801009	男	1999-01-01	计算机科学与技术(1)
201801010	男	1999-01-01	计算机科学与技术(1)

Showing 1 to 10 of 49 rows | 10 rows per page

图 7 学生信息管理

Detail

Year: 2018-2019 Term: 1

Course	Score
Principle of Computer Component	73
Computer Networking	79
Programming Language in C++	79
Engineering Mathematics	63

Showing 1 to 4 of 4 rows

图 8 学生成绩查询

更加有条理。系统赋予不同角色相应的权限,增强了教师和教务管理的工作人员之间协同合作,信息共享更为方便快捷,也使得学生能够更加方便的对自己的学习情况有更具体的了解。同时,该系统的实现体现了 Django 在 Web 开发中的高效、敏捷等特点。

参考文献

- [1] Django 框架架构总览理解 Django 是如何运作的. <http://www.nowamagic.net/academy/detail/13281808>. 2014.
- [2] 杨武帅, 万启元, 桑贤伯. 基于 Python 和 Django 框架的物联网智能设备管理系统的设计与实现. 中国新技术新产品, 2018, No.369(11): 18-20.
- [3] 汪红宇, 刘雄辉, 乔波. 基于 Bootstrap 框架实现响应式布局. 电脑与信息技术, 2017, 25(5): 19-22.
- [4] FRANKLIN J, FERGUSON R. Ajax with jQuery. Beginning JQuery, 2017.
- [5] 苟文博, 于强. 基于 MySQL 的数据管理系统设计与实现. 电子设计工程, 2017, 25(6): 62-65.
- [6] LIU C H, WU C J, CHEN H M. Testing of AJAX-Based Web Applications Using Hierarchical State Model. in: IEEE International Conference on E-business Engineering. 2017.