

文章编号: 2095-2163(2020)12-0174-04

中图分类号: TP389.1

文献标志码: A

# 基于 Django 的机票预订信息系统设计

熊春艳, 龚元明

(上海工程技术大学 机械与汽车工程学院, 上海 201620)

**摘要:** 随着人民生活水平的提高,使用飞机出行的客户数量日益增加,客户种类趋向多样化,所需的服务和客户期望也逐渐升高,因此引入高效的机票预订系统是处理机票预定工作的必然趋势。本文在对航空公司管理信息系统调查分析的基础上,结合 WEB 开发技术,设计了一款基于 Django 的机票预订信息系统,并在开发完成后把该系统部署在云服务器上。该系统遵循 MVC 开发模式,采用 Django 框架,使用 Mysql 开源数据库,实现了用户注册和登录、机票的查询、预订等功能。该系统的实现提高了航空公司的服务质量和工作效率,同时也极大地方便了广大乘客。

**关键词:** 机票预订; Django; 云服务器

## Design of Air Tickets Booking System based on Django

XIONG Chunyan, GONG Yuanming

(College of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

**[Abstract]** With the improvement of people's living standards, the number of customers who travel by airplane is increasing. As more and more types of customers emerge, there come higher service demands and customers' expectations. Therefore, the introduction of an efficient air ticket reservation system is an inevitable trend for handling air ticket reservations. By conducting an investigation and analysis of the airline management information system, this article refers to WEB development technologies to design a flight ticket booking information system based on Django. After development, the system is deployed on the Cloud server. The system adopts an MVC development and Django framework, alongside MySQL, to fulfill the functions of sign-up, log-in, ticket inquiry, and ticket booking, etc. The fulfillment of the system allows airlines to rely on networks to provide premium services and maximize working efficiency, while making things extremely convenient for passengers.

**[Key words]** Flight ticket booking; Django; Cloud server

### 0 引言

现代社会中人们的生活节奏越来越快,方便快捷的飞机出行逐渐得到了人们的青睐。随着飞机出行的乘客数量日益增加,对各大航空公司的服务提出了更高要求,各航空公司亟待提高服务质量来提高在民航市场中的竞争力。面对乘客数量的剧增,过去的机票预订方式不仅严重影响了航空公司的工作效率,还使用户的体验感下降。拥有一个高效的信息管理系统,不仅方便用户查询航班、购买机票等,还能极大地提高航空公司的核心竞争力。因此,本文利用 Django 技术,开发了一个高效的机票预订信息系统,提高了用户体验感,改善了航空公司服务质量。

### 1 基于 MVC 模式的 Django 框架

Django 是用 Python 语言编写的一个快速 Web 开发框架,Django 是个重量级的框架,包括了表单、模板、Admin 站点、认证权限、Session 机制、缓存、数据库 OMD 支持(对象关系映射)、项目工程管理的

自动化脚本等<sup>[1]</sup>。此外,Django 本身还自带了一个轻量级的小型服务器,开发者无需部署到其他服务器就可以进行 Web 开发和测试。

Django 采用了模型-视图-控制器(Model-View-Controller, MVC)设计模式。MVC 模式分为模型层、视图层和控制层 3 个层次<sup>[1]</sup>,示意图如图 1 所示。其中模型层(Model)负责业务对象与数据库的映射(ORM),视图层(View)实现与用户的交互,控制层(controller)负责接受用户的输入调用模型<sup>[2]</sup>。

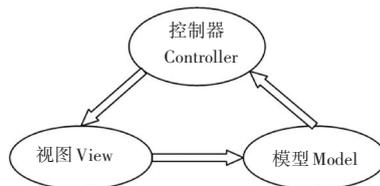


图 1 MVC 模式示意图

Fig. 1 MVC Schema diagram

Django 沿用了 MVC 模式的系统框架。然而

**作者简介:** 熊春艳(1995-),女,硕士研究生,主要研究方向:汽车电子控制、物联网技术;龚元明(1969-),男,博士,教授,主要研究方向:汽车电子控制、物联网技术。

收稿日期: 2020-09-01

Django 中采用的 MVC 模式与其它的 MVC 模式有所区别。从逻辑上看, Django 整个框架为 MTV 模式, 其模式示意图如图 2 所示, 分别为视图 (View)、模板 (Template) 和模型 (Model) 3 部分, MTV 模式将 MVC 的 V 模式分为 2 部分展现, 即分别为视图和模板<sup>[3]</sup>。至于 MVC 模式中的 C 则集成到了 Django 框架本身, 由框架完成对应部分工作, 开发者只需要把 (带正则表达式的) URL 和视图匹配就可以了。此外, Django 还具有模板引擎和对象关系映射两大特色模块。模板引擎中的模板是一个纯文本文件, 一个模板一般包含变量和区块标签两部分<sup>[4]</sup>, 模板引擎处理相关模板的操作, 使源代码更加简捷。对象关系映射 (ORM) 与关系数据库连接起来得到一个方便使用的数据库 API。

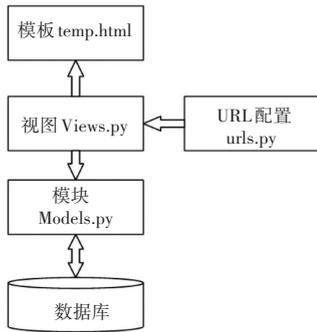


图 2 Django 框架的 MTV 模式示意图

Fig. 2 The MTV mode of the Django framework

## 2 机票预定系统设计方案

本文的机预订系统主要分为乘客和系统管理员

两大功能模块。乘客模块分为个人信息模块和机票信息模块。个人信息模块包括用户的注册和登录等功能, 用户可以修改个人信息和对个人信息查询。机票信息管理模块主要有订票、修改订单信息与查询订单信息等功能, 用户可以随时查询自己已订购的机票信息, 也可以根据需要修改和退票。系统管理员需要实时更新系统, 根据用户所做的修改信息做出相应的调整和更新。系统管理员模块包括系统管理员个人信息模块和航班信息模块, 航班信息主要有两大功能, 分别是修改航班信息和查询航班信息, 系统管理员拥有对航班信息进行增加、删除、修改等操作的权限, 用户根据自身需要查询航班信息。系统管理员也会在用户对自己的基本信息做出修改后做出相应的调整。系统结构组成图如图 3 所示。

系统采用了针对嵌入式平台应用的关系型数据 MySQL, MySQL 本身占用空间小、整体运行效率高。本系统需要用 7 张表来存放数据信息, 分别是: 航班基本信息表、订单信息表、用户信息表、管理员信息表、机场表、飞机型号表、航班计划表。系统主要信息见表 1-3。

- (1) 航班基本信息表, 包括航班编号、票价、剩余票数、航班名称、出发城市、出发时间、到达城市和到达时间;
- (2) 订单信息表, 包括机票编号、航班编号、订购时间、价格、座位类型、用户 ID 等;
- (3) 用户信息表, 包括姓名、用户名、密码、性别、身份证号、联系电话等。

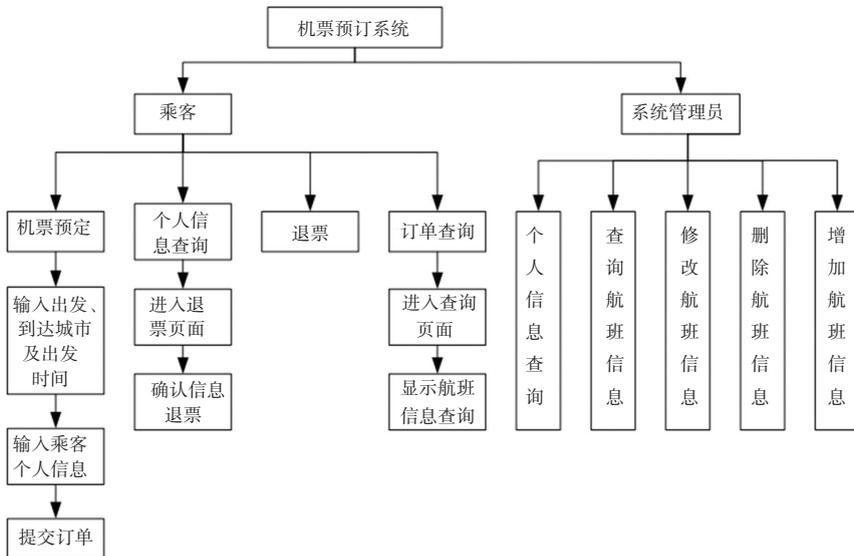


图 3 系统结构组成图

Fig. 3 System structure composition diagram

表 1 航班基本信息表

Tab. 1 Flight information table

字段名称	数据类型	字段长度	字段含义	约束
Fl_Id	Int	10	航班编号	主键
JJ_Price	Int	10	经济舱票价	
JJ_Ticket	Int	3	经济舱剩余票	
TD_Price	Int	10	头等舱票价	
TD_Ticket	Int	3	头等舱剩余票	
F1_Name	Varchar	10	航班名称	
Dep_City	Varchar	10	出发城市	
Dep_Time	Datetime	8	出发时间	
Ar_City	Varchar	10	到达城市	
Ar_time	Datetime	8	到达时间	

表 2 订单信息表

Tab. 2 Order information table

字段名称	数据类型	字段长度	字段含义	约束
Ticket_Id	Int	11	机票编号	主键
Flight_Id	Int	10	航班编号	
Order_Time	Datetime	8	订购时间	
Price	Int	10	价格	
Space_Type	Varchar	10	座位类型	
User_Id	Int	11	用户 ID	

表 3 用户信息表

Tab. 3 User information table

字段名称	数据类型	字段长度	字段含义	约束
Name	Varchar	20	真实姓名	
User_Name	Varchar	20	用户名	
Password	Varchar	20	密码	
Sex	Varchar	1	性别	
ID_Number	Varchar	10	身份证号	主键
Phone	Int	11	联系电话	

### 3 系统软件架构设计

#### 3.1 模型层设计 (M)

M 表示 Model(模型),一个模型类在数据库中对应一张表,当创建一个模型时,Django 的 ORM 框架会在数据库映射一张对应的表。再用迁移命令将模型同步到 MySQL 数据库中。机票预定系统将需要储存数据保存在 MySQL 数据库中,主要有 7 个模型,分别为航班信息模型、订单信息模型、用户信息模型、管理员信息模型、机场表信息模型、飞机型号信息模型、航班计划信息模型。

#### 3.2 视图层 (V) 和模板层 (T) 设计

编写页面视图 (view) 和模板视图 (Template) 是

整个项目的核心,为整个项目提供了逻辑框架。本系统函数文件 views.py 中包含了处理用户页面、查询页面、航班信息页面、用户付款页面的 URL 请求的逻辑函数。其本质就是接受 Http Request 对象并返回 Http Response 对象的过程。template 里面放的是前端页面的文件夹,其中包含了静态文件夹和动态文件夹。Django 采用 URLconf 模块来控制所呈现给用户的视图,urls.py 控制系统调用 views.py 中的视图函数,模块中的 urls.py 负责对与用户页面、查询页面、航班信息页面、航班信息页面、用户付款页面等页面的请求做匹配。

### 4 系统实现

查询页面需要乘客输入出发城市、到达城市、出发日期等信息,即可进行航班的查询。查询流程图如图 4 所示,查询页面效果图如图 5 所示。

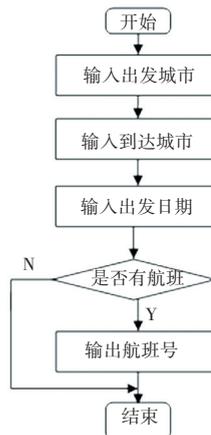


图 4 查询流程图

Fig. 4 Query flow chart

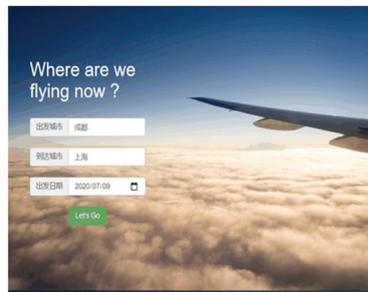


图 5 查询页面

Fig. 5 Thequery page

用户进入航班信息页面后,用户可以根据需要选择最合适的航空公司。用户在确定所需航班后需要填写真实的姓名、身份证号、还有常用的联系方式,完成相关信息的填写后跳转到付款页面,在该页面上会显示用户的基本信息及乘坐的航班信息,提交付款之后预订成功。用户付款页面效果图如图 6 所示。

(下转第 181 页)