北京理工大学

继续教育暨现代远程教育学院

毕业设计（论文）

毕业论文题目：基于 Python 语言 Django 框架的商品管理系统

指导教师姓名：吴素研

类　别：夜大．专升本

专　业：计算机科学与技术

班　级：2016 年计算机科学与技术 A 班（012）

姓　名：何慧君

2017 年 5 月 20 日

摘要

本文是以学习和研究网页设计与Web系统开发为目标，在课堂学习的基础上，综合运用HTML、CSS、JS、B/S系统等知识，探索利用目前流行的Python语言，开发出可用的Web系统。经过学习发现，可以使用Django框架，来快速开发Web系统。

本论文将会简要阐述Pyhton的部署，Django安装、配置，以及商品管理系统主要功能的实现，以及遇到的各种问题。

关键词：Python，Django，商品管理

目录

前言

应用程序有两种模式C/S、B/S。C/S是客户端/服务器端程序，也就是说这类程序一般独立运行。而B/S就是浏览器端/服务器端应用程序，这类应用程序一般借助Chrome等浏览器来运行。WEB应用程序一般是B/S模式。Web应用程序首先是“应用程序”，和用标准的程序语言，如C、C++等编写出来的程序没有什么本质上的不同。然而Web应用程序又有自己独特的地方，就是它是基于Web的，而不是采用传统方法运行的。换句话说，它是典型的浏览器/服务器架构的产物。B/S结构能够很好地应用在广域网上，成为越来越多的企业的选择。

一个Web应用程序是由完成特定任务的各种Web组件（web components）构成的并通过Web将服务展示给外界。在实际应用中，Web应用程序是由多个模型(model)－视图(view)－控制器(controller)等组织起来的代码组成，这些组件相互协调为用户提供一组完整的服务。

Web应用程序对企业的重要用途是对数据进行处理，管理信息系统（Management Information System,简称MIS）就是这种架构最典型的应用。MIS可以应用于局域网，也可以应用于广域网。基于Internet的MIS系统以其成本低廉、维护简便、覆盖范围广、功能易实现等诸多特性，得到越来越多的应用。

本文就是尝试利用Python语言实现商品管理系统，增强对Web应用系统需求、设计、开发的知识和能力。

1 绪论

Python 已经有将近 30 年的历史，在过去 30 年中，Python 在运维工程师和数据科学家群体中受到广泛欢迎，然而却极少有企业将 Python 作为生产环境的首选语言。在最近几年，这一情况有所改变。随着云计算、大数据以及人工智能技术的快速发展，Python 及其开发生态环境正在受到越来越多的关注。

互联网时代来临后，Python被用来在Web开发领域进行尝试，涌现出了一批基于 Python 开发一些WEB的网站，还有不少大型的、基于Python 的网站，比如 Youtube、豆瓣等网站。使得一些Web开源框架迅速成长，如Django、Flask等，为程序员高效开发Web程序提供了巨大的帮助。

进入了云计算时代，基于过去一段时间 Python 在系统管理工具的积累，以及其本身具备了非常好的系统集成能力，Python 在云计算领域可以说大放异彩。最著名的是Python开发的Openstack。不仅在私有云领域，在公有云领域，包括 AWS，包括 Google 云，当这些公有云提供出 SDK 的时候，它们首选的技术路线依然是 Python。

最近两年又火起来的人工智能领域，Python靠着过去多年在科学计算等方面的积累出现了大爆发。比如图像识别用的都是 Python OpenCV库。在深度学习领域几乎没有任何其他语言可以跟 Python 相提并论的，比如 Caffe，Theano，TesnorFlow，Keras 这些非常流行的深度学习框架，都是以 Python 为主要开发语言。事实证明了在深度学习领域目前 Python 是处于非常主导的地位。

如上所述，为了能跟上人工智能的潮流，从用户体验角度，从开发者角度来讲，Python 是更好的语言，也是更好的接口语言，值得我们学习和掌握它。另一方面，考虑到可以用 Python 集成各种各样的服务，这样能有效降低成本，同时也能够减轻自己开发团队的压力，让开发团队能够减少一些学习成本。

2 Django

Django是Python中目前风靡的Web Framework,框架能够帮助我们把程序的整体架构搭建好，而我们所需要做的工作就是填写逻辑，而框架能够在合适的时候调用你写的逻辑，而不需要我们自己去调用逻辑，让Web开发变的更敏捷.

Django是一个高级Python Web框架，鼓励快速,简洁，以程序设计的思想进行开发. 通过使用这个框架，可以减少很多开发麻烦，使你更专注于编写自己的app，而不需要重复造轮子. Django免费并且开源.

2.1 Django 特点

1) 完全免费并开源源代码。
2) 快速高效开发。
3) 使用MTV架构(熟悉Web开发的应该会说是MVC架构)。
4) 强大的可扩展性。

2.2 Django 工作方式

用户在浏览器中输入URL后的回车，浏览器会对URL进行检查，首先判断协议,如果是http就按照 Web 来处理，然互调用DNS查询，将域名转换为IP地址，然后经过网络传输到达对应Web服务器，服务器对url进行解析后，调用View中的逻辑(MTV中的V)，其中又涉及到Model(MTV中的M)，与数据库的进行交互，将数据发到Template(MTV中的T)进行渲染，然后发送到浏览器中，浏览器以合适的方式呈现给用户。

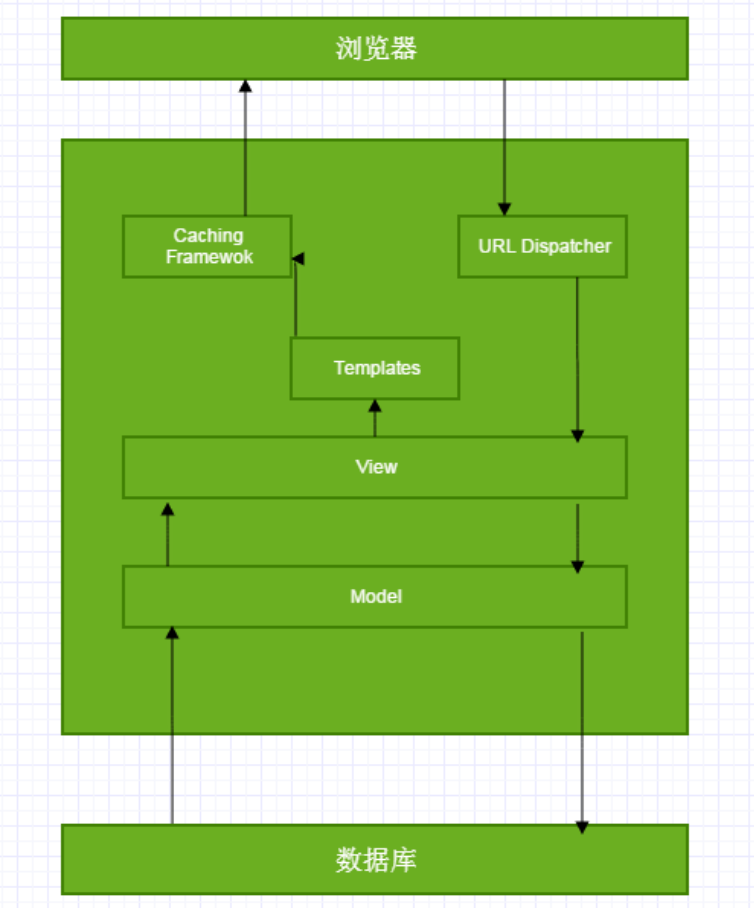


图 2-1 工作方式

2.3 开发环境

本文使用的主要开发环境为：

1) 操作系统：macOS Sierra 10.12.4
2) 开发语言：Python 3.6.1
3) Web 框架：Django1.11.1
4) 数据库：SQLite3
5) 前端框架：Bootstrap3.3.7
6) 开发 IDE：Pycharm CE 2017.1
7) 虚拟环境：Anaconda 4.3.17
8) 版本管理：GitHub

2.4 安装 Django

使用终端安装最新版的Django：

```
conda install Django
```

2.5 Bootstrap 安装

Bootstrap，来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。

从官网下载所需资源，稍后复制到开发目录中：

http://getbootstrap.com/getting-started/#download

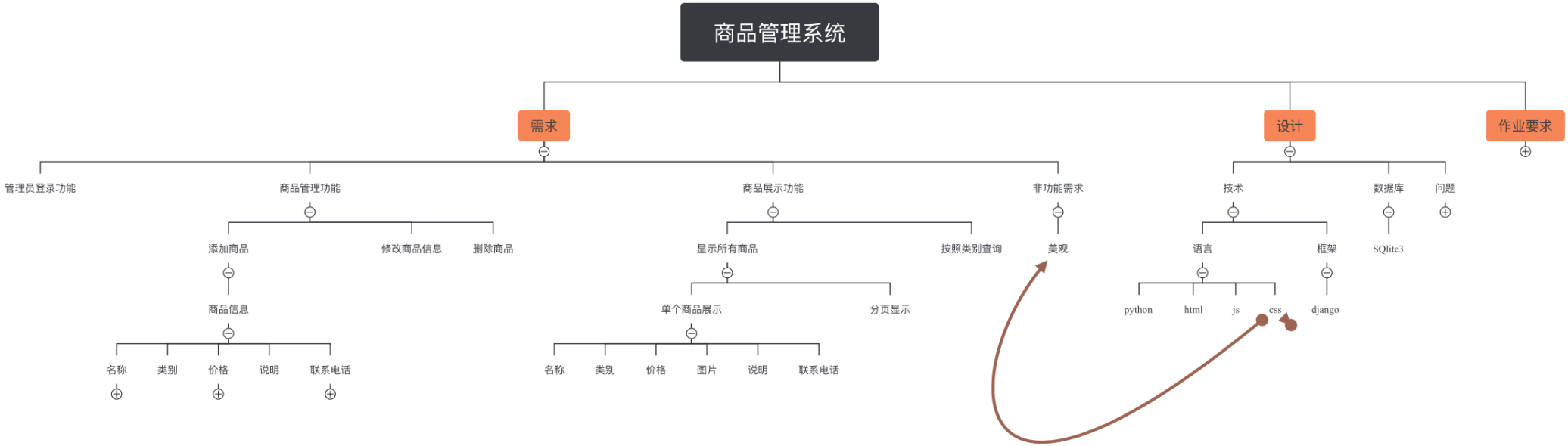
```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   └── bootstrap-theme.min.css.map
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphicons-halflings-regular.eot
    ├── glyphicons-halflings-regular.svg
    ├── glyphicons-halflings-regular.ttf
    ├── glyphicons-halflings-regular.woff
    └── glyphicons-halflings-regular.woff2
```

图2-2 BootStrap目录

3 商品管理系统主要功能及系统设计

3.1 功能设计

主要功能和设计考虑如图3-1所示：



3.2 主要界面

4 代码实现

从现在开始正式的进入代码阶段，简述开发过程。

4.1 创建项目

我们创建一个名为WebStore的Django项目，创建项目的指令[1]（终端）如下:

```
django-admin.py startproject WebStore
```

文件结构如下：

```
WebStore
├── manage.py
└── WebStore
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

4.2 设置数据库

Django项目建成后，就可以设置数据库了。本文默认使用SQLite数据库，在WebStore/WebStore/setting.py中可以查看和修改数据库设置:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

还可以设置其他数据库，如MySQL，PostgreSQL，现在为了便于发布和提供老师审核，使用默认数据库设置。

[1] 在 windows 下可以使用 django-admin startproject WebStore 的命令，无 py 后缀。

4.3 设置国际化

Django 支持国际化，多语言。Django的国际化是默认开启的，如果不需要国际化支持，可以在设置文件中设置 USE_I18N = False，那么Django会进行一些优化，不加载国际化支持机制。在WebStore/WebStore/setting.py修改默认的语言和时区。

```
LANGUAGE_CODE = 'zh-hans'
TIME_ZONE = 'Asia/Shanghai'
USE_I18N = True
```

4.4 配置静态文件

静态文件是指网站中的js,css,图片，视频等文件，通常在WebStore/WebStore/setting.py中已经设置好了。在WebStore根目录下创建static文件夹，在static下再创建upload、css、js、image等目录。

```
STATIC_URL = '/static/'
```

4.5 创建 Django app

在Django中的app被认为是一个功能模块，与其他的web框架可能有很大的区别，将不通功能放在不同的app中，方便代码的复用。在WebStore目录下，终端输入创建app指令：

```
python manage.py startapp commodity
```

WebStore根目录下，文件结构如下：

```
── commodity
|  ├── __init__.py
|  ├── admin.py
|  ├── migrations
|  |  └── __init__.py
|  ├── models.py
|  ├── tests.py
|  └── views.py
├──db.sqlite3
├──manage.py
├──WebStore
├── __init__.py
├── settings.py
├── urls.py
```

必须在WebStore/WebStore/setting.py下添加新建app

```
INSTALLED_APPS = (
...
'commodity', #这里填写的是 app 的名称
)
```

4.6 创建 models

在WebStore/commodity/models.py下编写如下程序:

```
class Commondity(models.Model):
    CommodityName = models.CharField('商品名称', max_length=100)
    CommodityCategory = models.CharField('商品类别', max_length=50, blank=True)
    CommodityPrice = models.DecimalField('商品价格', max_digits=11, decimal_places=2)
    CommonditylImage = models.ImageField('商品图片', upload_to='static', default='static/upload/None/no-img.jpg')
    CommodityDateTime = models.DateTimeField('登记日期', auto_now_add=True)
    CommodityContent = models.TextField('商品说明', blank=True, null=True)
    CommonditylContactMobile = models.CharField('联系电话', max_length=11, blank=True, null=True)

    def __str__(self):
        return self.CommodityName    # 一般系统默认使用<Commodity: Commodity object> 来表示对象，通过这个函数可以告诉系统使用 CommodityName 字段来表示这个对象

    class Meta:
        verbose_name = '商品' #给模型起个更好听的名字，这儿相当于进行了汉化。
        verbose_name_plural = '所有商品'
        # 按时间下降排序
        ordering = ['-CommodityDateTime']
```

4.7 运行程序

在WebStore根目录下，用终端命令行输入以下指令：

```
$ python manage.py runserver    #启动Django中的开发服务器
```

启动浏览器，输入http://127.0.0.1:8000/，就可以访问网站了。

4.8 同步数据库

在WebStore根目录下，用终端命令行输入以下指令：

```
python manage.py migrate #同步在model中建立的数据库
```

如果对model进行了修改，需要先执行一次makemigrations命令，再执行migrae。

```
python manage.py makemigrations #先检查更新
```

4.9 设置 Admin

Django有一个优秀的特性，内置了Django admin后台管理界面，方便管理者进行添加和删除网站的内容. 新建的项目系统已经为我们设置好了后台管理功能，可以在WebStore/WebStore/setting.py中查看。

在WebStore/commodity/admin.py中增加代码，让后台管理界面能对“商品”信息进行管理。默认管理界面中仅显示上面这是的“CommodityName”，为更方便的在后台管理信息，需要增加一些一些代码。具体的如下：

```
from django.contrib import admin
from commodity.models import Commodity


class CommonditeyAdmin(admin.ModelAdmin):
    list_display = ('CommodityName', 'CommodityCategory',
'CommodityPrice', 'CommodityDateTime',)


admin.site.register(Commondity, CommonditeyAdmin)
```

4.10创建超级用户

初次登陆后台管理界面，需要使用如下命令账号创建超级用户：

```
python manage.py createsuperuser
```

按照提示输入用户名、邮箱、密码，创建第一个超级用户。

4.11正式编写 template

在commodity目录下创建template目录，在template下增加base.html,做为本文程序的基础模版。

下一阶段开始模版、首页的编制。

5 结论

深度学习目前是一个非常热门的研究方向，利用卷积神经网络的卷积层、池化层和全连接层等基本结构，就可以让这个网络结构自己学习和提取相关特征，并加以利用。这种特性对许多研究提供了许多便利，可以省略过往非常繁杂的建模过程。此外，深度学习现在在图像分类、物体检测和纹理转换等方面都已经有了非常大的成果和进步。深度学习应用面非常广，而且通用性强，完全可以继续努力将其拓展到其它应用领域。

致谢

转眼间，本学期即将结束了。在这几个月的时间里，继续学习计算机相关课程，收获颇丰。在学习和课程中得到了老师和同学们很多指导和帮助。由于时间太紧，本计划将使用Django开发出更完整的一个知识管理系统（简易blog），发布到云主机上，很遗憾没有实现。但是不会放弃，将继续利用以后的时间，持续学习和完善，完成个人平台的搭建。

首先我要感谢我的指导教师吴素研老师。她治学严谨、知识渊博，鼓励我们对网页编程等相关内容，按照自己的兴趣学习，以便有更深入的理解，确实受益匪浅。

我还要感谢班主任景老师，是她的谆谆教导，让我打定主意，安心学习，今年的课程顺利学完。

此外，我还要感谢同学们，给予很多启发和帮助。

参考文献

Django官方网站：https://docs.djangoproject.com/en/1.11/
Django基础教程：http://code.ziqiangxuetang.com/django/django-tutorial.html