

北京邮电大学



物流信息系统课程设计

题目：药品库存管理系统的设计与开发

姓 名	李想
学 院	自动化学院
专 业	物流工程
班 级	2016211408
学 号	2016211876
班内序号	04
指导教师	苏志远

2019 年 6 月

药品库存管理系统的设计与开发

摘要: 库存管理系统是供应链管理中的重要一环，对控制供应链成本、货物管理有着不可或缺的作用。

该系统为药品库存管理系统，采用 B/S 结构，以 django+html 作为前后端技术，模拟了在药品库存管理中的单据流转情况。该系统简单设计了调拨、出库、采购、入库等业务流程，其中流程操作简单，不易出错。该项目在开发过程中，充分考虑了并发控制，利用 django 的并发式设计，为后人跳出 JVM 架构提供了良好的借鉴。该项目版本由 git 指令进行推进，拥有良好的回滚和拓展性。

该系统的测试数据皆提取自网上真实数据，最后证明药品库存管理系统具有一定的可行性。

关键词: 药品管理 库存管理系统 Django

Abstract: Inventory management system is an important part of supply chain management, which plays an indispensable role in controlling supply chain cost and goods management.

The system is a drug inventory management system, which adopts B/S structure and uses Django and HTML as front and back end technology to simulate the document flow in drug inventory management. This system simply designs the business process of dispatching, out of storage, purchasing and warehousing, among which the operation process is simple and not easy to make mistakes. In the development process, this project has fully considered the concurrency control and taken advantage of Django's concurrent design, which provides a good reference for future generations to jump out of the JVM architecture. The project version is pushed by git instructions with good rollback and extensibility.

The test data of this system are all extracted from real data on the Internet. Finally, the feasibility of drug inventory management system is proved.

Keyword: drug management, inventory management system, Django

目录

一、前言与自我评价.....	5
1.1 前言	5
1.2 自我评价	5
二、系统概况与国内外相关研究.....	6
2.1 系统概况.....	6
2.2 国内外相关研究.....	6
三、系统可行性分析.....	7
3.1 投资必要性.....	7
3.2 技术可行性.....	7
3.3 组织可行性.....	7
3.4 经济可行性.....	7
3.5 社会可行性.....	8
四、系统分析.....	9
4.1 需求分析.....	9
4.2 组织结构分析.....	10
4.3 业务流程分析.....	10
4.4 数据与数据流程分析.....	15
五、系统设计.....	17
5.1 总体结构设计.....	17
5.1.1 系统结构设计.....	17
5.1.2 服务器 url 设计.....	18
5.2 代码设计.....	19
5.2.1 代码结构设计.....	19
5.2.2 代码文件内容设计.....	20
5.3 数据结构设计.....	21
5.4 数据库设计.....	21

5.5 模块功能与处理过程设计.....	23
5.5.1 模块功能设计.....	23
5.5.2 处理过程设计.....	23
5.6 输入输出设计.....	24
5.7 部分程序设计.....	27
六、总结与展望.....	50
七、致谢.....	50
八、参考文献.....	50
九、项目进度计划.....	51
十、自我评价.....	52
十一、周期报告.....	53

一. 前言与自我评价

1.1 前言

本药品库存管理系统系北京邮电大学物流工程物流信息系统设计与开发课程之作业，代码保管于 https://github.com/lx1374327576/drug_management_system，如果有任何单位或个人想对该项目进行改进或者进一步开发，欢迎提出 pull request，并联系 1374327576@qq.com。该系统开发周期为两个月，完全由作者本人独立完成，采用 html+Django 架构，充分考虑了多并发因素以及数据库的可移植性和浏览器的兼容性。

作者用一己之力，使用 python 语言，利用能跟上时代潮流的技术，为后人跳出 JVM 架构提供了一个良好的借鉴。本作品为该作者的 web 开山之作，如有不足之处，请多见谅。

1.2 自我评价

该作品耗费时间较长，作者尽自己所能，花费九周余，终于勉强完成了该作业的基本要求。可以说在该项目中采用的物流信息技术可以跟得上时代的步伐，但是在业务流程上与生产实际差距较大，这使得该项目的实用价值下降。综合评价：良。

二、系统概况与国内外相关研究

2.1 系统概况

以患者为中心的科学化、专业化、精细化的现代医院运营管理中，保障患者安全、持续改善患者就医体验已成为医院各项建设关注的焦点，药品的发放在患者就医过程中是必不可少的，药品的管理也是至关重要的。随着计算机技术的普及，各行各业意识到信息化的重要性，在医院信息化中，药库占据重要地位，起到了重要枢纽的作用。一甲以上的医院基本完成了药库的信息化，信息化的药库在药品管理、药品分发、人力资源、资金利用等方面带来了可观的效益。

本系统是药品库存管理系统，不仅能降低医院的管理成本，确保患者的用药安全，还可以实时监控库存水平，保证药品充足，减少资金占用的浪费。通过美观简洁的界面和高并发设计，提高系统的自动化程度和效率。还收集了大量的数据，供高层作出重要的科学性决策，例如根据药品的使用率以及价格估测当地居民的水平。

2.2 国内外相关研究

1. T. W. Whitin《库存管理理论》分析了供应商批量和顾客采购批量的关系。

2. 发达国家例如美国、日本，已经逐步建立起了相对成熟的医院信息系统，药品库存管理系统的发展也在日趋完善。

3. 90年代末，温筱煦等采用了动态库存分析的方法，根据药库的物资流动曲线，获得药品在特定时间段的平均库存量、周转时间、流动速度、动态分析参数等信息，能够更为全面、准确的反应药库库存的变化趋势，为医院药库管理提供了有效的决策支持手段。

4. 国内也有较成熟的仓库管理系统，但大多是针对某一特定单位而设计研发的，系统功能的通用性仍然有待提高。已有系统大多未设置预留的功能接口，药品仓库管理功能发生变化后无法提供扩展功能，直接使用原有系统存在很多不方便因素。

三、系统可行性分析

3.1 投资必要性

该项目有较好的投资回报率，有投资的必要。

3.2 技术可行性

医院药品库存管理系统的主要使用者是药库的管理人员以及各科室、病区的管理人员，用户的数量是有限的，系统采用的体系结构为 C/S。系统的开发工具采用的是 pycharm，使用的数据库是 sqlite3，开发语言采用的是 python，系统中拥有多个可调参数的配置，这些可调参数可以实现各个分系统的可调参数的设置，从而将系统的灵活性大幅度的提高。系统除了提供众多的可调参数配置，也提供了各种接口，并能及时适应各种政策的改变，能随时适应医疗卫生体制改革政策的需要，开发所采用的技术是成熟的，因此，基于智能算法的医院药品库存管理系统在技术方面是具有可行性的。

3.3 组织可行性

该项目由一人完成，具有良好的计划和协调性，该人员经验丰富，能够保证项目顺利执行。

3.4 经济可行性

医院药品库存管理系统具有低成本的系统维护和管理，系统的整体架构先进，管理工具完善，对于非法用户的入侵行为能有效的防止，同时也保障了系统不会因为人为的操作失误对系统造成危害，而致使系统崩溃。系统中存储的数据庞大，数据的安全性在系统中也是至关重要，系统对于这些存储的数据，提供了冗余的保护措施，对于容错以及冗余都有相应的安全保护机制，系统在安全方面有强大的保障。系统的投入应用，可以使药库管理人员对于药品库存的管理变的简单、便捷、高效，提高了管理人员的工作效率，节约了人力。财力，同时，可以对药库药品的储存情况进行实时的监控，并能够提供灵活的调整空间。因系统的安全性能高，后期维护成本低，管理简便，开发成本低等优点。所以，医院药品库存管理系统在经济方面来说是具有可行性的。

3.5 社会可行性

该项目可以造福社会，提高人民群众的看病效率，切实解决人民群众的困难，履行我党为人民服务的宗旨。开发这个系统完全符合社会主义核心价值观，符合社会的公序良俗，缓和民族之间的矛盾，有利于社会又快又好的发展。

四、系统分析

4.1 需求分析

对于药品库存管理系统，简单地分析一下系统的功能，需要满足客户对药品的需求以及药品的采购和补货。所以对于该系统，需要入库管理员、出库管理员、采购管理员、库存管理员和系统管理员，以及给他们对应的职能。详见图 1：

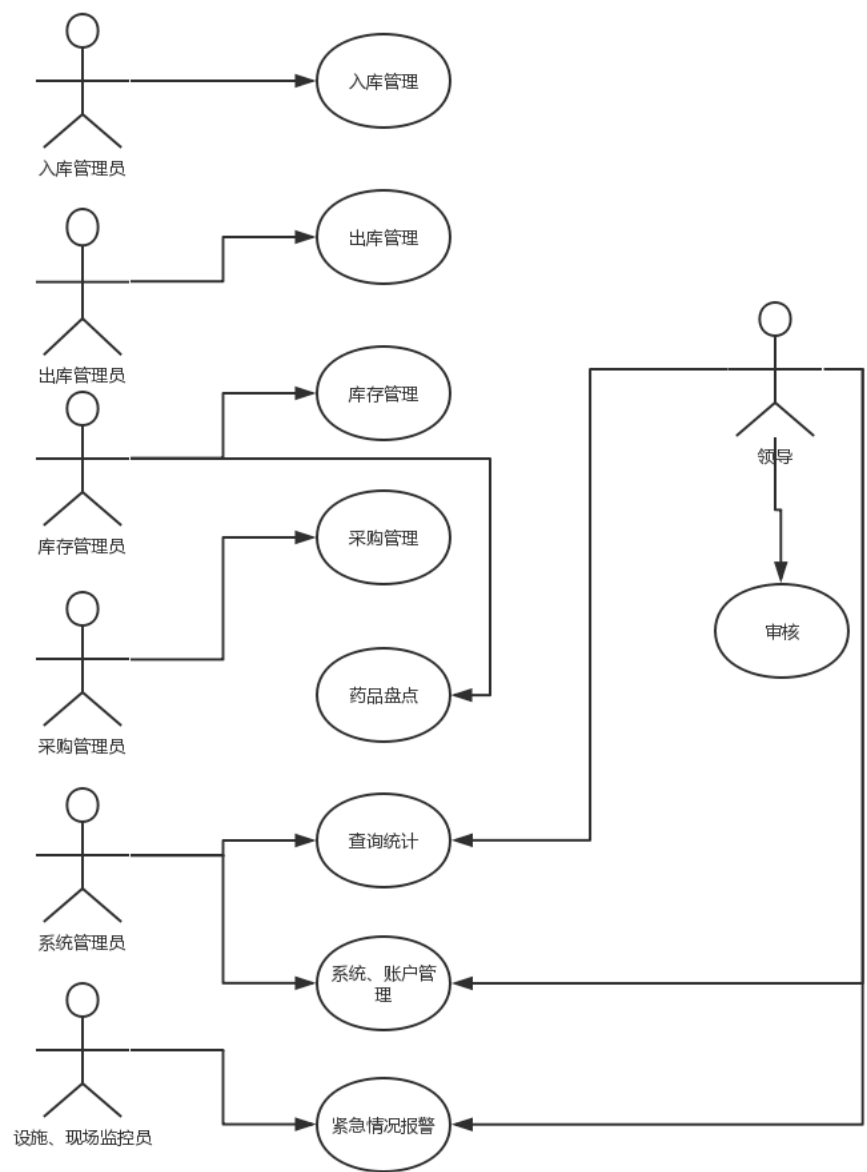


图 1

4.2 组织结构分析

该系统的组织结构由管理员/操作人员、客户和供应商组成，他们之间是并列的关系，详见图 2：

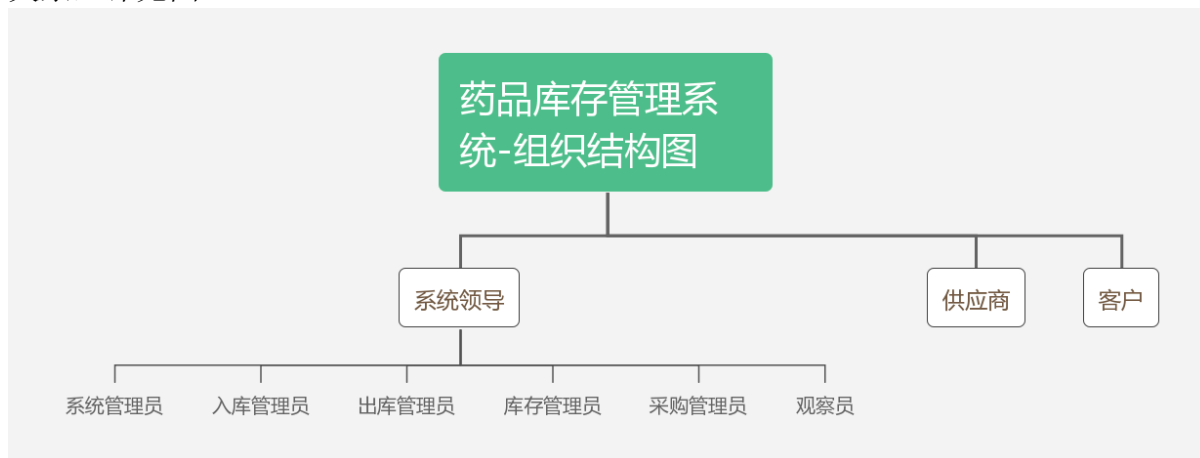


图 2

4.3 业务流程分析

在简单分析了系统业务流程之后，根据基本需求，做出了图 3：

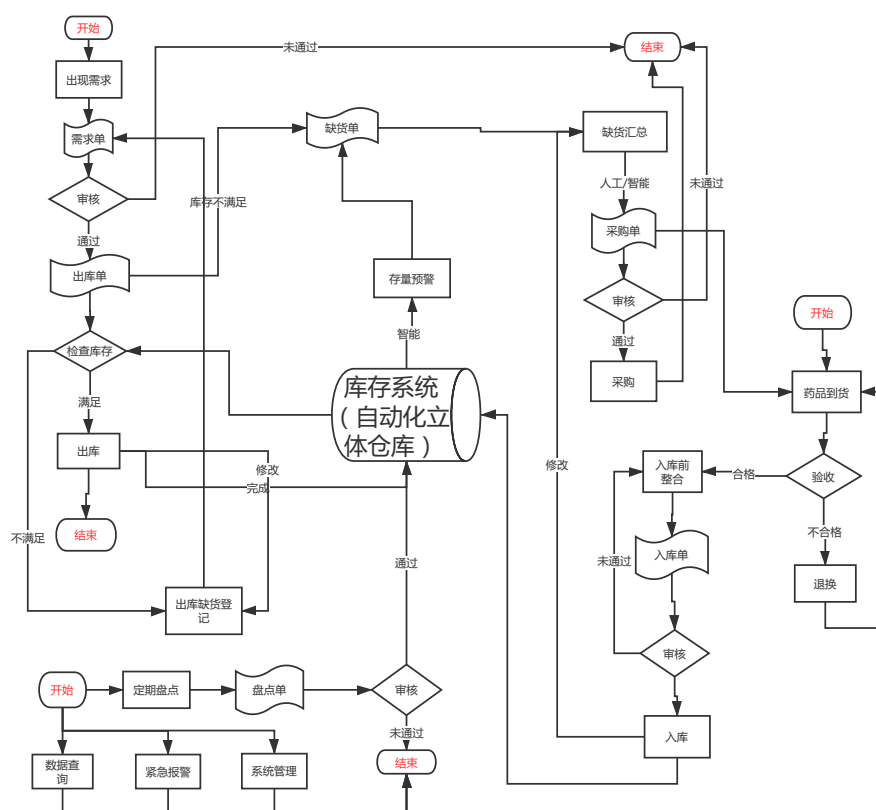


图 3

接着设计每种单据的样式，详见图 4，图 5，图 6，图 7，图 8：

调拨单

单号	demand-100				
发起单位	校本部医院				
发起人	李想				
联系方式	13757578780				
发起时间	2019年5月31日 13:25				
序号	药品编号	药品类别	药品名称	药品数量	价格
1	medicine-2	西药	板蓝根	5.0	5.0
2	medicine-3	中药	六味地黄丸	6.0	24.0
3	medicine-2	西药	板蓝根	8.0	48.0
发起人签字			收到人签字		

确认

图 4

出库单

单号	checkout-39			
仓库编号	1			
发起人	李想			
联系方式	13757578780			
发起时间	2019年5月12日 09:02			
序号	药品编号	药品名称	药品数量	价格
1	medicine-1	盘尼西林	1.0	5.0
2	medicine-4	啊米卡星	4.0	28.0
3	medicine-2	板蓝根	5.0	5.0
发起人签字		执行人签字		

确认

图 5

缺货单

单号	lack-52			
发起人	李想			
联系方式	13757578780			
发起时间	2019年5月31日 13:44			
序号	药品编号	药品名称	药品数量	价格
1	medicine-1	盘尼西林	1.0	1.0
2	medicine-2	板蓝根	5.0	10.0
3	medicine-4	啊米卡星	4.0	16.0
发起人签字		收到人签字		

确认

图 6

采购单

单号	purchase-31			
发起人	李想			
发起人联系方式	13757578780			
订货日期	2019年5月29日 18:46			
供应商	中国第一制药厂			
供应商地址	北京市海淀区西土城路10号			
供应商联系方式	13261519797			
序号	药品编号	药品名称	药品数量	价格
1	medicine-3	六味地黄丸	1.0	3.0
2	medicine-4	啊米卡星	4.0	16.0
3	medicine-7	谷精草	9.0	63.0
发起人签字		供应商收单签字		
到货验收人签字		到货供应商签字		

确认

图 7

入库单

单号	storage-22			
仓库编号	1			
发起人	李想			
联系方式	13757578780			
发起时间	2019年5月29日 18:48			
序号	药品编号	药品名称	药品数量	价格
1	medicine-2	板蓝根	8.0	16.0
2	medicine-3	六味地黄丸	8.0	24.0
发起人签字		执行人签字		

确认

图 8

接着分析每个单据的状态转移过程，详见图 9：

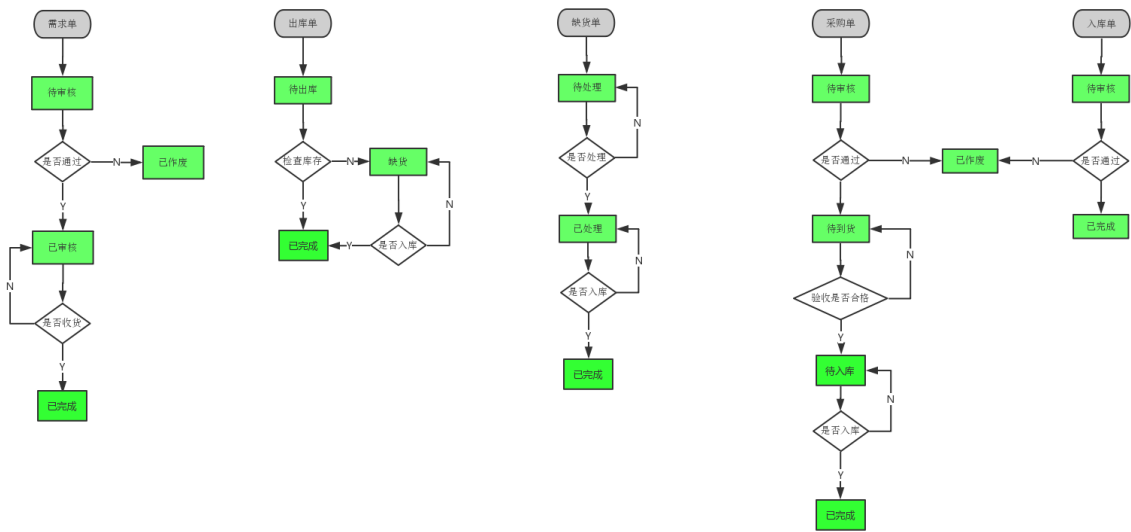


图 9

单据之间的流转情况如下图所示，见图 10：

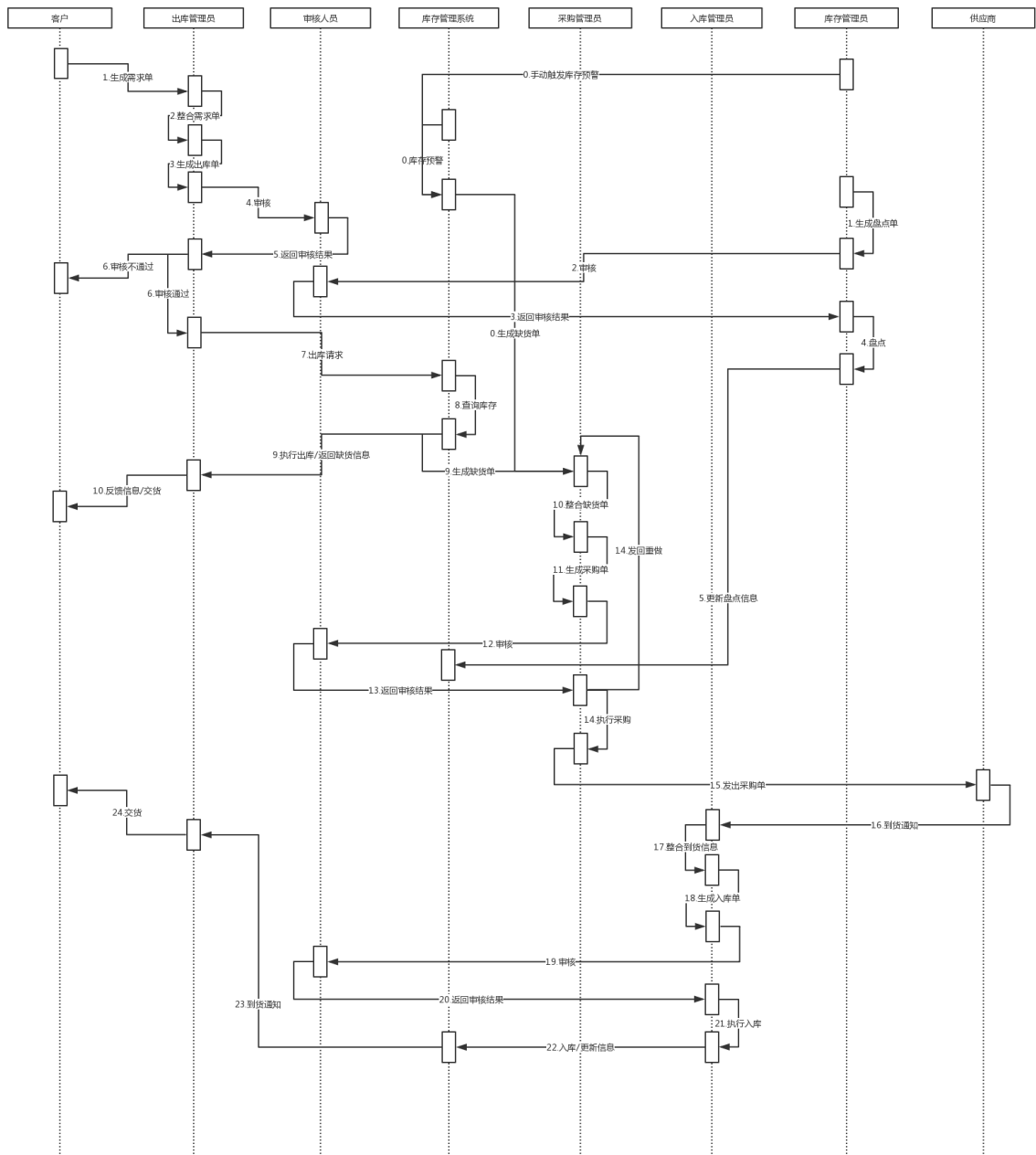


图 10

4.4 数据与数据流程分析

数据定义如下：

基本数据类

```
class Medicine_type(models.Model):  
    id = models.IntegerField('id', primary_key=True)  
    name = models.CharField('name', blank=False, max_length=100)  
  
    def __str__(self):  
        return self.name  
  
class Medicine(models.Model):  
    id = models.IntegerField('id', primary_key=True)  
    name = models.CharField('name', blank=False, max_length=100)  
    total_num = models.FloatField('total_num', default=0)  
    now_num = models.FloatField('now_num', default=0)  
    medicine_type = models.ForeignKey(Medicine_type, on_delete=models.CASCADE)  
  
    def __str__(self):  
        return self.name  
class Medicine_type(models.Model):  
    id = models.IntegerField('id', primary_key=True)  
    name = models.CharField('name', blank=False, max_length=100)  
  
    def __str__(self):  
        return self.name  
  
class Medicine(models.Model):  
    id = models.IntegerField('id', primary_key=True)  
    name = models.CharField('name', blank=False, max_length=100)  
    total_num = models.FloatField('total_num', default=0)  
    now_num = models.FloatField('now_num', default=0)  
    medicine_type = models.ForeignKey(Medicine_type, on_delete=models.CASCADE)  
  
    def __str__(self):  
        return self.name
```

Id:	<input type="text" value="8"/>
Name:	<input type="text" value="紫药水"/>
Total_num:	<input type="text" value="5.0"/>
Now_num:	<input type="text" value="0.0"/>
Medicine type:	<div>中药   </div>

图 11

数据流程如图 11:

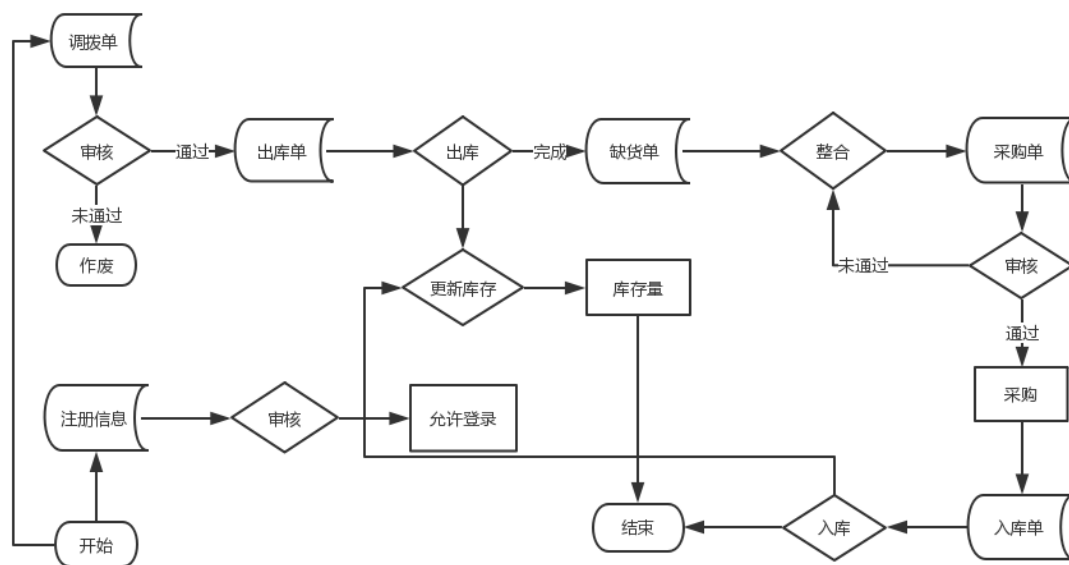


图 12

五、系统设计

5.1 总体结构设计

5.1.1 系统结构设计

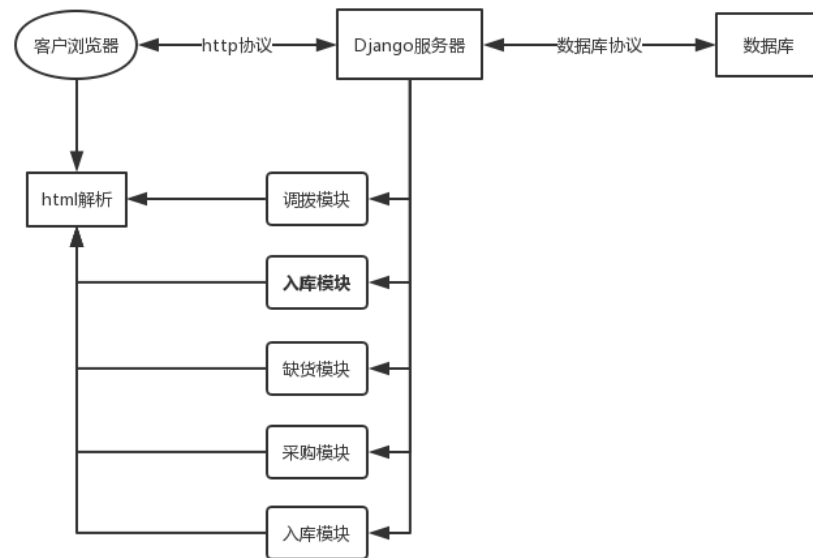


图 13

5.1.2 服务器 url 设计

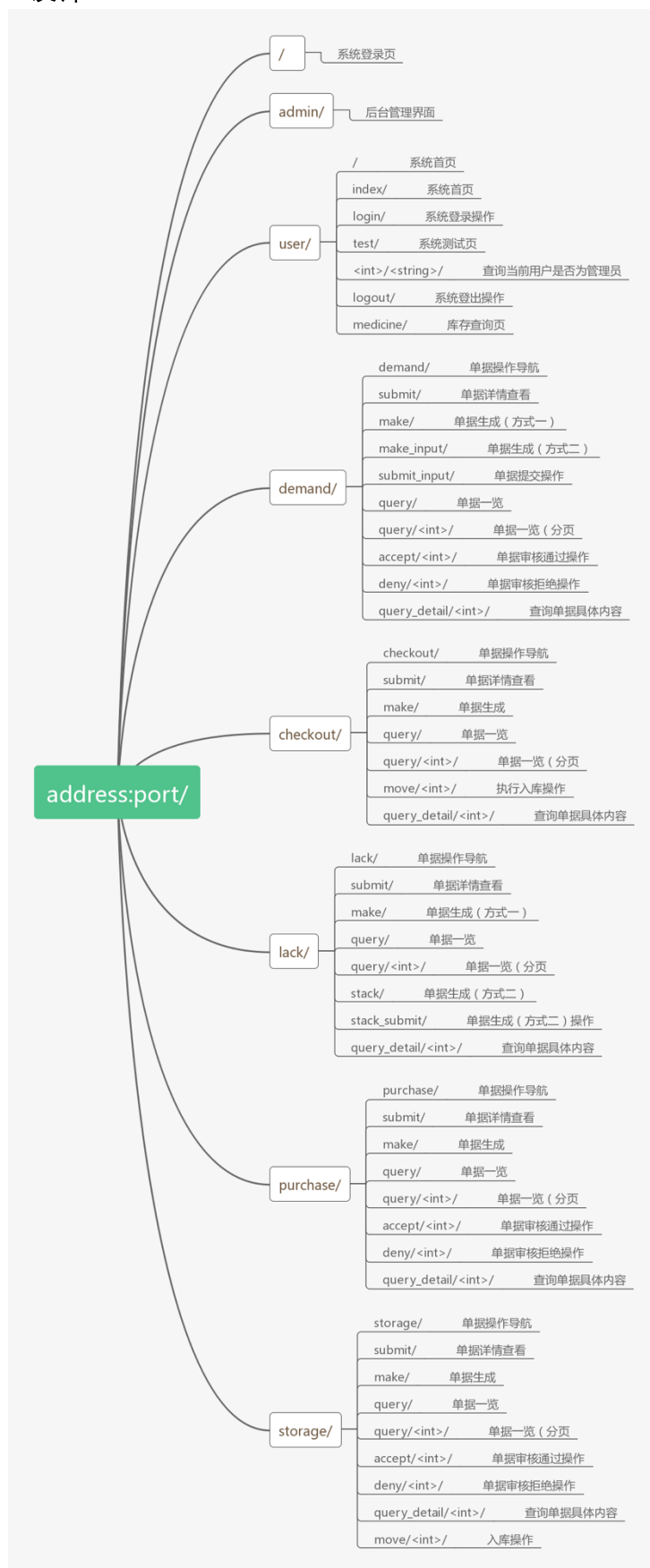


图 14

5.2 代码设计

5.2.1 代码结构设计

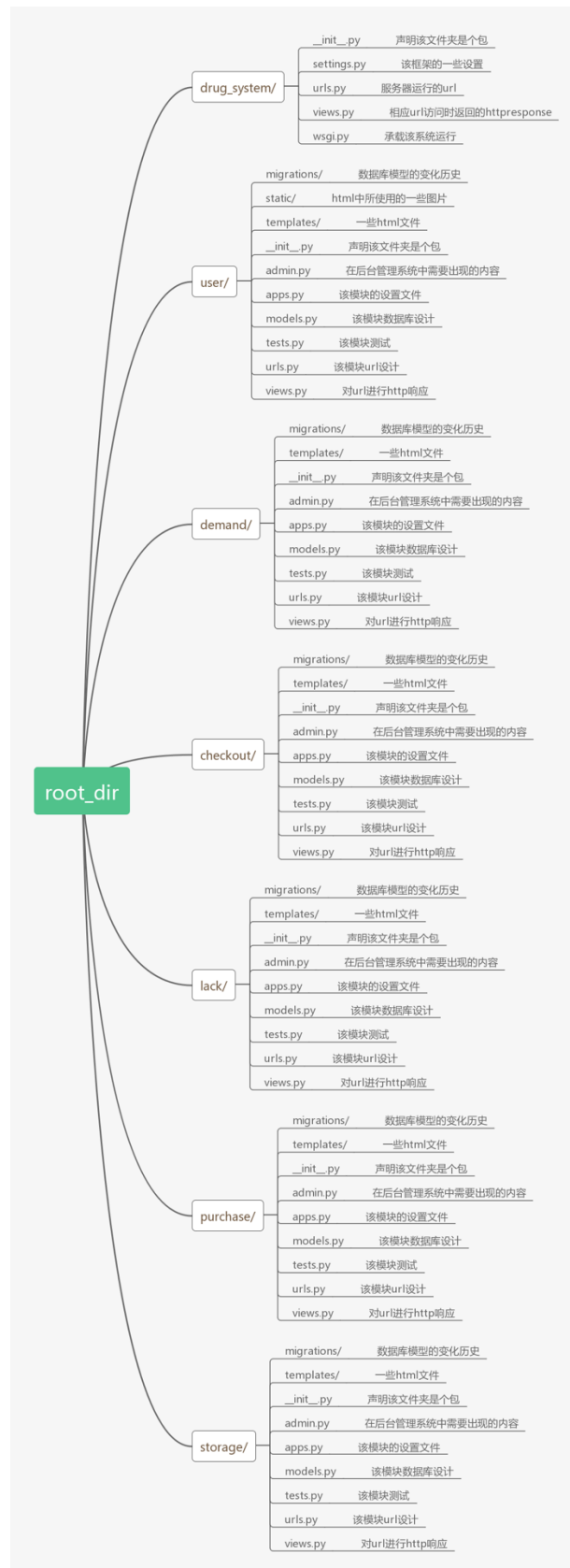


图 15

5.2.2 代码文件内容设计

我们以 demand/views.py 为例，只要我们写个每个函数与 url.py 一一对应即可。

Views.py:

```
from django.shortcuts import render
from .models import Demander_Demand, Detail, Demand, Demander, Medicine
from django.http import Http404
from django.http import HttpResponseRedirect
from django.urls import reverse
from user.models import User

# Create your views here.
def demand(request)

def submit(request)

def make(request)

def query(request)

def query_page(request, page_id)

def query_detail(request, demand_id)

def deny(request, demand_id):

def accept(request, demand_id)

def make_input(request):

def submit_input(request)
```

urls.py:

```
from django.urls import path

from . import views

app_name = 'demand'

urlpatterns = [
    path('demand/', views.demand, name='demand'),
    path('submit/', views.submit, name='submit'),
    path('make/', views.make, name='make'),
    path('make_input/', views.make_input, name='make_input'),
    path('submit_input/', views.submit_input, name='submit_input'),
    path('query/', views.query, name='query'),
    path('query/<int:page_id>', views.query_page, name='query_page'),
    path('accept/<int:demand_id>', views.accept, name='accept'),
    path('deny/<int:demand_id>', views.deny, name='deny'),
    path('query_detail/<int:demand_id>', views.query_detail, name='query_detail'),
]
```

5.3 数据结构设计

本项目完全采用线性数据结构，所有数据按序号排列。

5.4 数据库设计

数据库 E-R 图，详见图 16:

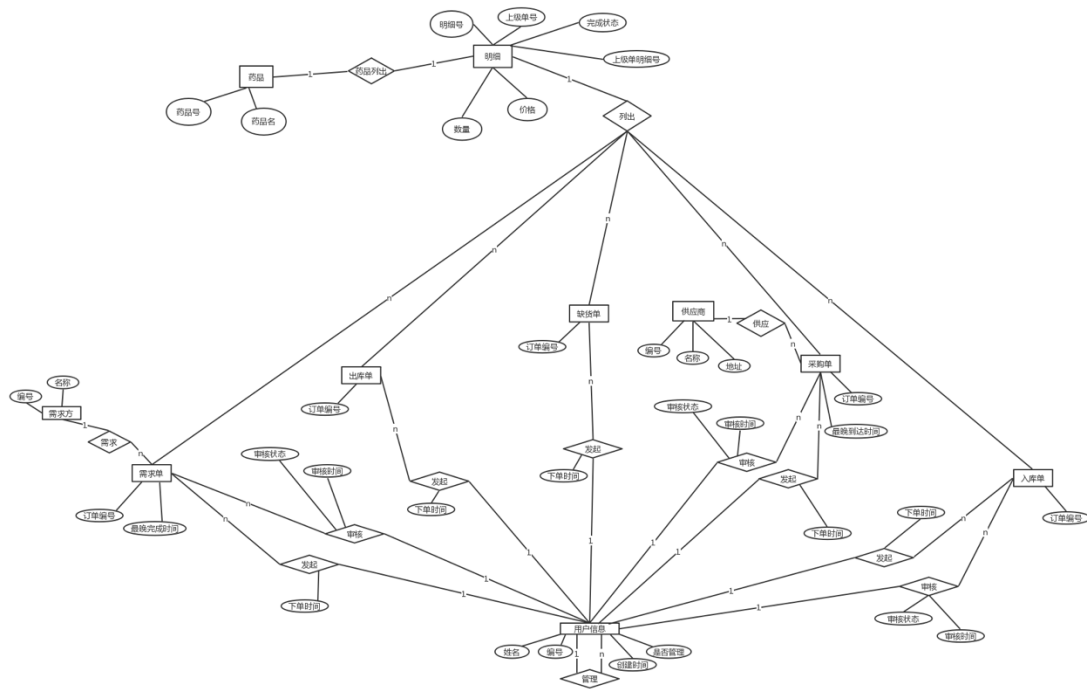
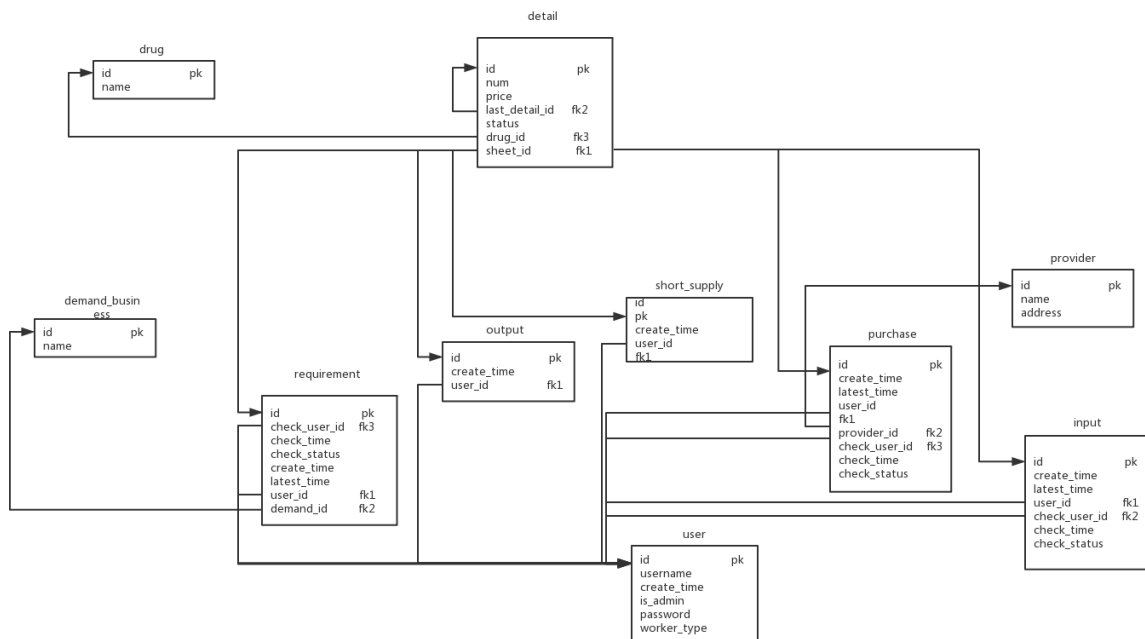


图 16

数据库 pdm 图，详见图 17:



Django 自带数据库部署设计，代码如下（举例）：

```

class Detail(models.Model):
    id = models.IntegerField('id', primary_key=True)
    father_id = models.IntegerField('father_id', default=0)
    father_detail_id = models.IntegerField('father_detail_id', default=0)
    medicine = models.ForeignKey(Medicine, blank=False, on_delete=models.CASCADE)
    create_time = models.DateTimeField('create_time', default=timezone.now)
    num = models.FloatField('num')
    price = models.FloatField('price')
    status = models.IntegerField('status', default=0) # 0 未处理 1 处理中 2 已完成
    form_type = models.IntegerField('form_type') # 1 需求单 2 出库单 3 缺货单 4 采购单
    5 入库单
    form_id = models.IntegerField('form_id')
  
```

5.5 模块功能与处理过程设计

5.5.1 模块功能设计

名称	功能
user	登录，登出，库存查询
demand	生成、查询、审核调拨单
checkout	生成、查询出库单，出库
lack	生成、查询缺货单
purchase	生成、查询、审核采购单
storage	生成、查询、审核入库单，入库

表 1

5.5.2 处理过程设计

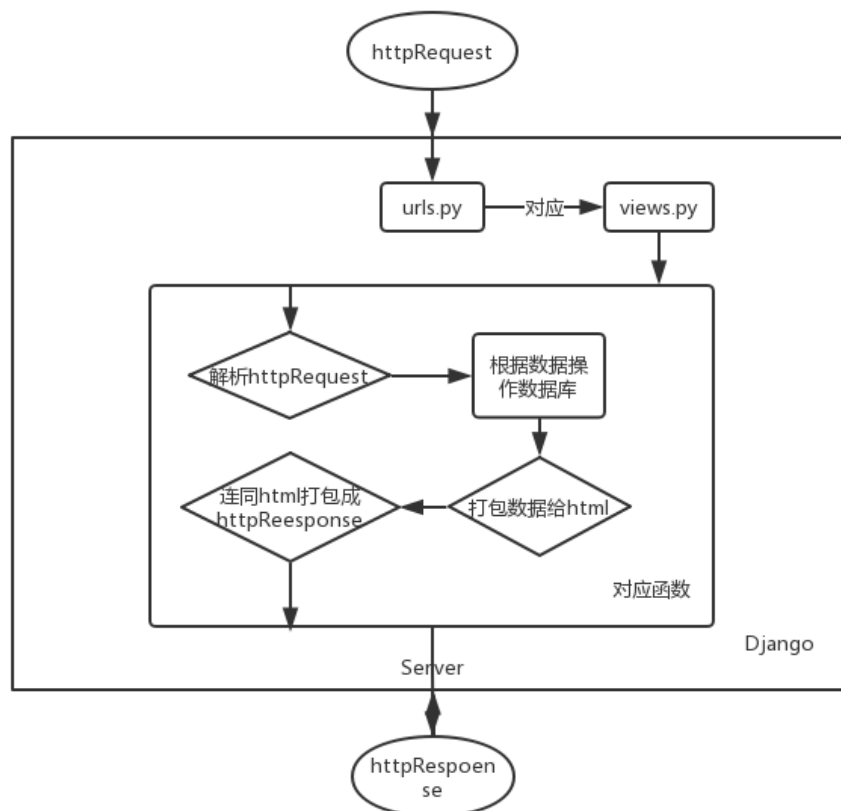


图 17

5.6 输入输出设计

登录界面：



图 18

库存查询界面：

药品库存管理系统			
返回首页 注销			
药品数量查询			
药品序号	药品类别	药品名称	药品数量
medicine-1	西药	盘尼西林	204
medicine-2	西药	板蓝根	163
medicine-3	中药	六味地黄丸	154
medicine-4	西药	啊米卡星	168
medicine-5	西药	麻黄碱	153
medicine-6	西药	咖啡因	155
medicine-7	中药	谷精草	154
medicine-8	中药	紫药水	155

图 19

单据生成界面：

药品库存管理系统

返回首页

注销

生成调拨单

当前调拨单号	demand-103			
选择调拨方	校本部医院			
序号	药品编号	药品类别	药名	数量
1	medicine-1	西药	盘尼西林	
2	medicine-2	西药	板蓝根	
3	medicine-3	中药	六味地黄丸	
4	medicine-4	西药	啊米卡星	
5	medicine-5	西药	麻黄碱	
6	medicine-6	西药	咖啡因	
7	medicine-7	中药	谷精草	
8	medicine-8	中药	紫药水	

提交

图 20

单据查询界面：

药品库存管理系统

入库 返回首页 注销

调拨单一览

序号	调拨单编号	发起时间	发起人	详情	审核状态	按钮
1	demand-100	2019年5月31日 13:25	李想	详情	待审核	接受 拒绝
2	demand-99	2019年5月31日 13:25	李想	详情	待审核	接受 拒绝
3	demand-98	2019年5月31日 13:25	李想	详情	待审核	接受 拒绝
4	demand-97	2019年5月31日 13:25	李想	详情	待审核	接受 拒绝
5	demand-96	2019年5月31日 13:24	李想	详情	待审核	接受 拒绝
6	demand-95	2019年5月31日 13:24	李想	详情	待审核	接受 拒绝
7	demand-94	2019年5月31日 13:24	李想	详情	待审核	接受 拒绝
8	demand-93	2019年5月31日 13:24	李想	详情	待审核	接受 拒绝
9	demand-92	2019年5月31日 13:24	李想	详情	待审核	接受 拒绝
10	demand-91	2019年5月31日 13:24	李想	详情	待审核	接受 拒绝

« 1 2 3 4 5 6 7 »

图 21

详情查询页：

调拨单

单号	demand-98				
发起单位	校本部医院				
发起人	李想				
联系方式	13757578780				
发起时间	2019年5月31日 13:25				
序号	药品编号	药品类别	药品名称	药品数量	价格
1	medicine-2	西药	板蓝根	5.0	5.0
2	medicine-3	中药	六味地黄丸	6.0	24.0
3	medicine-4	西药	啊米卡星	10.0	80.0
发起人签字			收到人签字		

确认

图 22

5.7 部分程序设计

展示 demand 模块程序，其他部分程序详见

https://github.com/lx1374327576/drug_management_system

admin.py:

```
from django.contrib import admin

# Register your models here.
from .models import Medicine, Demander, Detail, Demand, Demander_Demand,
Medicine_type
admin.site.register(Medicine)
admin.site.register(Demander)
admin.site.register(Detail)
admin.site.register(Demand)
admin.site.register(Demander_Demand)
admin.site.register(Medicine_type)
```

apps.py:

```
from django.apps import AppConfig

class DemandConfig(AppConfig):
    name = 'demand'
```

models.py

```
from django.db import models
from django.utils import timezone
from user.models import User

class Medicine_type(models.Model):

    id = models.IntegerField('id', primary_key=True)
    name = models.CharField('name', blank=False, max_length=100)

    def __str__(self):
        return self.name

class Medicine(models.Model):

    id = models.IntegerField('id', primary_key=True)
    name = models.CharField('name', blank=False, max_length=100)
    total_num = models.FloatField('total_num', default=0)
    now_num = models.FloatField('now_num', default=0)
    medicine_type = models.ForeignKey(Medicine_type, on_delete=models.CASCADE)

    def __str__(self):
        return self.name

class Detail(models.Model):

    id = models.IntegerField('id', primary_key=True)
    father_id = models.IntegerField('father_id', default=0)
    father_detail_id = models.IntegerField('father_detail_id', default=0)
    medicine = models.ForeignKey(Medicine, blank=False, on_delete=models.CASCADE)
    create_time = models.DateTimeField('create_time', default=timezone.now)
    num = models.FloatField('num')
    price = models.FloatField('price')
    status = models.IntegerField('status', default=0) # 0 未处理 1 处理中 2 已完成
```

```

    form_type = models.IntegerField('form_type') # 1 需求单 2 出库单 3 缺货单 4 采购单
5 入库单
    form_id = models.IntegerField('form_id')

class Demander(models.Model):

    id = models.IntegerField('id', primary_key=True)
    name = models.CharField('name', blank=False, max_length=100)

    def __str__(self):
        return self.name

class Demand(models.Model):

    id = models.IntegerField('id', primary_key=True)
    due_time = models.DateTimeField('due_time', default=timezone.now)
    demander = models.ForeignKey(Demander, on_delete=models.CASCADE)
    create_time = models.DateTimeField('create_time', default=timezone.now)
    create_user = models.ForeignKey(User, on_delete=models.CASCADE)
    status = models.IntegerField('status', default=0) # 0 未处理 1 审核通过 2 单子完
成 3 审核未通过
    check_time = models.DateTimeField('check_time', default=timezone.now)

class Demander_Demand(models.Model):

    id = models.IntegerField('id', primary_key=True)
    demander = models.ForeignKey(Demander, on_delete=models.CASCADE)
    num = models.FloatField('num')
    medicine = models.ForeignKey(Medicine, on_delete=models.CASCADE)

```

urls.py:

```

from django.urls import path

from . import views

app_name = 'demand'

urlpatterns = [
    path('demand/', views.demand, name='demand'),
    path('submit/', views.submit, name='submit'),
    path('make/', views.make, name='make'),
    path('make_input/', views.make_input, name='make_input'),
    path('submit_input/', views.submit_input, name='submit_input'),
    path('query/', views.query, name='query'),
    path('query/<int:page_id>/', views.query_page, name='query_page'),
    path('accept/<int:demand_id>/', views.accept, name='accept'),
    path('deny/<int:demand_id>/', views.deny, name='deny'),
    path('query_detail/<int:demand_id>/', views.query_detail, name='query_detail'),
]

```

views.py:

```

from django.shortcuts import render
from .models import Demander_Demand, Detail, Demand, Demander, Medicine
from django.http import Http404
from django.http import HttpResponseRedirect
from django.urls import reverse
from user.models import User

# Create your views here.

```

```

def demand(request):
    context = {}
    return render(request, 'demand/index.html', context)

def submit(request):
    demand_map = request.POST
    lx = User.objects.get(id=1)
    demander = Demander.objects.get(id=1)
    demand_form = Demand(create_user=lx, demander=demander)
    demand_form.save()
    demand_form_list = Demand.objects.all()
    length = demand_form_list.count()
    demand_form = demand_form_list[length-1]
    detail_list = []
    flag = False
    for key in demand_map:
        if not flag:
            flag = True
            continue
        single = Demander_Demand.objects.get(id=key)
        detail = Detail(price=single.id*single.num, num=single.num,
medicine=single.medicine,
                        form_type=1, form_id=demand_form.id)
        detail.save()
        de_list = Detail.objects.all()
        length = de_list.count()
        detail = de_list[length - 1]
        detail_list.append(detail)
    context = {
        "demand_form": demand_form,
        "detail_list": detail_list,
    }
    return render(request, 'demand/submit.html', context)

def make(request):
    demands = Demander_Demand.objects.all()
    demand_list = Demand.objects.all()
    demand_num = demand_list.count() + 1
    context = {
        "demand_list": demands,
        "demand_num": demand_num,
    }
    # print('context', context)
    return render(request, 'demand/demand.html', context)

def query(request):
    return HttpResponseRedirect(reverse('demand:query')+'1')

def query_page(request, page_id):
    demand_list = Demand.objects.all().order_by('status', '-id')
    list_length = demand_list.count()
    if page_id*10 >= list_length+10:
        return HttpResponseRedirect(reverse('demand:query') + str((list_length-1)
// 10+1))
    demand_list = demand_list[(page_id-1)*10:min(page_id*10, list_length)]
    status_dict = {
        1: '审核通过',
        2: '已完成',
        0: '待审核',
        3: '审核未通过',
    }

```

```

    }
    context = {
        'demand_list': demand_list,
        'status_dict': status_dict,
        'page_id': page_id,
        'page_id_minus': page_id-1,
        'page_id_plus': page_id+1,
        'page_id_minus2': page_id-2,
        'page_id_minus3': page_id-3,
        'page_id_plus2': page_id+2,
        'page_id_plus3': page_id+3,
    }
    return render(request, 'demand/query.html', context)

def query_detail(request, demand_id):
    detail_list = Detail.objects.filter(form_type=1).filter(form_id=demand_id)
    demand_form = Demand.objects.get(id=demand_id)
    context = {
        'detail_list': detail_list,
        'demand_form': demand_form,
    }
    return render(request, 'demand/submit.html', context)

def deny(request, demand_id):
    demand_form = Demand.objects.get(id=demand_id)
    if demand_form.status != 0:
        return Http404('your operation is not correct!')
    else:
        demand_form.status = 3
        demand_form.save()
        return HttpResponseRedirect(reverse('demand:query'))

def accept(request, demand_id):
    demand_form = Demand.objects.get(id=demand_id)
    if demand_form.status != 0:
        return Http404('your operation is not correct!')
    else:
        demand_form.status = 1
        demand_form.save()
        return HttpResponseRedirect(reverse('demand:query'))

def make_input(request):
    medicine_list = Medicine.objects.all()
    demander_list = Demander.objects.all()
    demand_list = Demand.objects.all()
    demand_num = demand_list.count() + 1
    context = {
        'medicine_list': medicine_list,
        'demander_list': demander_list,
        'demand_num': demand_num,
    }
    return render(request, 'demand/make_input.html', context)

def submit_input(request):
    medicine_map = request.POST
    lx = User.objects.get(id=1)
    demander_id = medicine_map['demander_id']
    demander = Demander.objects.get(id=demander_id)
    demand_form = Demand(create_user=lx, demander=demander)
    demand_form.save()
    demand_form_list = Demand.objects.all()

```

```

length = demand_form_list.count()
demand_form = demand_form_list[length - 1]
for key in medicine_map:
    test = -1
    if key == 'demander_id':
        continue
    try:
        test = int(medicine_map[key])
    except Exception:
        print('key:', key, 'map[key]:', medicine_map[key])
    if test <= 0:
        continue
    else:
        medicine_id = int(key)
        medicine_num = test
        medicine = Medicine.objects.get(id=medicine_id)
        detail = Detail(price=medicine_id * medicine_num, num=medicine_num,
medicine=medicine,
                        form_type=1, form_id=demand_form.id)
        detail.save()
detail_list = Detail.objects.filter(form_type=1).filter(form_id=demand_form.id)
context = {
    "demand_form": demand_form,
    "detail_list": detail_list,
}
return render(request, 'demand/submit.html', context)

```

templates/demand/demand.html:

```

<html>
<head>
    <meta charset="utf-8">
    <title>药品库存管理系统</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
    * {
        box-sizing: border-box;
    }

    body {
        margin: 0;
    }

    /* 头部样式 */
    .header {
        background-color: #f1f1f1;
        padding: 20px;
        text-align: center;
    }

    /* 导航条 */
    .topnav {
        overflow: hidden;
        background-color: #333;
    }

    /* 导航链接 */
    .topnav a {
        float: left;
        display: block;
        color: #f2f2f2;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }

```

```

}

/* 链接 - 修改颜色 */
.topnav a:hover {
    background-color: #ddd;
    color: black;
}

/* 创建三个相等的列 */
.column {
    float: left;
    padding: 10px;
}

/* 左侧宽度 */
.column.left {
    width: 20%;
    background-image:url("/static/2.jpeg");
    height: 1000px;
}

/* 右侧宽度 */
.column.right {
    width: 20%;
    background-image:url("/static/2.jpeg");
    height: 1000px;
}

/* 中间区域宽度 */
.column.middle {
    width: 60%;
}

/* 列后面清除浮动 */
.row:after {
    content: "";
    display: table;
    clear: both;
}

/* 响应式布局 - 宽度小于 600px 时设置上下布局 */
@media screen and (max-width: 600px) {
    .column.side, .column.middle {
        width: 100%;
    }
}

/* 底部样式 */
.footer {
    background-color: #f1f1f1;
    padding: 10px;
    text-align: center;
}

/* 表格样式 */
table, tr, td {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
    width: 100%;
}

```



```

tr{
    height:40px
}
td{
    text-align:center;
    vertical-align:middle;
}
.checkbox{
    width:30px;
    height:30px;
}
.submit{
    width:100px;
    height:50px;
    background-color:red;
    color:white;
    font-size:20px;
}
.submit-div{
    width:100%;
    text-align: center;
}
</style>
</head>
<body>
    <div class="header">
        <h1>药品库存管理系统</h1>
    </div>

    <div class="topnav">
        <a href="{% url 'demand:demand' %}">调拨</a>
        <a href="{% url 'checkout:checkout' %}">出库</a>
        <a href="{% url 'lack:lack' %}">缺货</a>
        <a href="{% url 'purchase:purchase' %}">采购</a>
        <a href="{% url 'storage:storage' %}">入库</a>
        <a href="{% url 'user:index' %}">回到首页</a>
        <a href="{% url 'user:logout' %}">注销</a>
    </div>
    <div class="row">
        <div class="column left">
        </div>
        <div class="column middle">
            <div class="submit-div">
                <h2> 生成调拨单 </h2>
            </div>
            <form action="{% url 'demand:submit' %}" method="post">
                {% csrf_token %}
                <table>
                    <tr>
                        <td><strong>当前调拨单号</strong></td>
                        <td> demand-{{demand_num}} </td>
                    </tr>
                    <tr>
                        <td> <strong>序号</strong> </td>
                        <td> <strong>调拨需求方名称</strong> </td>
                        <td> <strong>药品类别</strong> </td>
                        <td> <strong>药品名称</strong> </td>
                        <td> <strong>药品数量</strong> </td>
                        <td> <strong>是否选择</strong> </td>

```

```

        </tr>
        {% for demand in demand_list %}
        <tr>
            <td> {{demand.id}} </td>
            <td> {{demand.demander.name}} </td>
            <td> {{demand.medicine.medicine_type.name}} </td>
            <td> {{demand.medicine.name}} </td>
            <td> {{demand.num}} </td>
            <td>
                <input name="{{demand.id}}" type="checkbox"
value="checked" class="checkbox" />
            </td>
        </tr>
        {% endfor %}
    </table>
    <br>
    <div class="submit-div">
        <input type="submit" value="提交" class="submit">
    </div>
</form>
</div>

<div class="column right">
</div>
</div>
<div class="footer">
    <p>项目地址: https://github.com/lx1374327576/drug\_management\_system</p>
    <p>Copyright © 2018-2019 1374327576@qq.com All Rights Reserved. 备案号:
bupt-2016211876</p>
</div>
</body>
</html>

```

templates/demand/index.html:

```

<html>
    <head>
        <meta charset="utf-8">
        <title>药品库存管理系统</title>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <style>
            * {
                box-sizing: border-box;
                text-decoration: none
            }

            body {
                margin: 0;
            }

            /* 头部样式 */
            .header {
                background-color: #f1f1f1;
                padding: 20px;
                text-align: center;
            }

            /* 导航条 */
            .topnav {
                overflow: hidden;
                background-color: #333;
            }

```

```

/* 导航链接 */
.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

/* 链接 - 修改颜色 */
.topnav a:hover {
    background-color: #ddd;
    color: black;
}

/* 创建三个相等的列 */
.column {
    float: left;
    padding: 10px;
}

/* 左侧宽度 */
.column.left {
    width: 20%;
    background-image: url("/static/2.jpeg");
    height: 505px;
}

/* 右侧宽度 */
.column.right {
    width: 20%;
    background-image: url("/static/2.jpeg");
    height: 505px;
}

/* 中间区域宽度 */
.column.middle {
    width: 60%;
    background-image: url("/static/2.jpeg");
    height: 505px;
}

.input_1{
    outline:none;
    margin-top:10px;
    border:0;
    width:300px;
    height:50px;
    border-radius:15px;
    padding-left:15px;
    font-size:18px;
    color:#999;
}
.input_2{
    margin:20px 0px;
    border:0;
    background:#fff;
}
.input_3{
    outline: none;
    width:50px;
    height:30px;
}

```

```

        border:0;
        padding:5px 7px;
        background:#fff;
        border-radius:20px;
        margin:0px 10px;
    }

    /* 列后面清除浮动 */
    .row:after {
        content: "";
        display: table;
        clear: both;
    }

    /* 响应式布局 - 宽度小于 600px 时设置上下布局 */
    @media screen and (max-width: 600px) {
        .column.side, .column.middle {
            width: 100%;
        }
    }

    form{
        position: relative;
        width:970px;
        height:800px;
        color:#fff;
        text-align:center;
        padding:65px;
    }

    /* 底部样式 */
    .footer {
        background-color: #f1f1f1;
        padding: 10px;
        text-align: center;
    }
</style>
</head>
<body>
    <div class="header">
        <h1>药品库存管理系统</h1>
    </div>

    <div class="topnav">
        <a href="{% url 'demand:demand' %}">调拨</a>
        <a href="{% url 'checkout:checkout' %}">出库</a>
        <a href="{% url 'lack:lack' %}">缺货</a>
        <a href="{% url 'purchase:purchase' %}">采购</a>
        <a href="{% url 'storage:storage' %}">入库</a>
        <a href="{% url 'user:index' %}">回到首页</a>
        <a href="{% url 'user:logout' %}">注销</a>
    </div>
    <div class="row">
        <div class="column left">
            <!---->
        </div>
        <div class="column middle">
            <a href="{% url 'demand:make' %}" class="input_3">生成调拨单 ( 勾选 )
</a>

```

```

        <br><br>
        <a href="{% url 'demand:make_input' %}" class="input_3">生成调拨单
( 输入 ) </a>

        <br><br>
        <a href="{% url 'demand:query' %}" class="input_3">查询调拨单</a>
    </div>

    <div class="column_right">
        <!----
->
    </div>
</div>
<div class="footer">
    <p>项目地址: https://github.com/lx1374327576/drug_management_system</p>
    <p>Copyright © 2018-2019 1374327576@qq.com All Rights Reserved. 备案号:
bupt-2016211876</p>
</div>
</body>
</html>

```

demand/templates/make_input.html

```

<html>
<head>
<meta charset="utf-8">
<title>药品库存管理系统</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
    box-sizing: border-box;
}

body {
    margin: 0;
}

/* 头部样式 */
.header {
background-color: #f1f1f1;
padding: 20px;
text-align: center;
}

/* 导航条 */
.topnav {
    overflow: hidden;
    background-color: #333;
}

/* 导航链接 */
.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

/* 链接 - 修改颜色 */
.topnav a:hover {

```

```

        background-color: #ddd;
        color: black;
    }
    /* 创建三个相等的列 */
    .column {
        float: left;
        padding: 10px;
    }

    /* 左侧宽度 */
    .column.left {
        width: 20%;
        background-image:url("/static/2.jpeg");
        height: 1000px;
    }

    /* 右侧宽度 */
    .column.right {
        width: 20%;
        background-image:url("/static/2.jpeg");
        height: 1000px;
    }

    /* 中间区域宽度 */
    .column.middle {
        width: 60%;
    }

    /* 列后面清除浮动 */
    .row:after {
        content: "";
        display: table;
        clear: both;
    }

    /* 响应式布局 - 宽度小于 600px 时设置上下布局 */
    @media screen and (max-width: 600px) {
        .column.side, .column.middle {
            width: 100%;
        }
    }

    /* 底部样式 */
    .footer {
        background-color: #f1f1f1;
        padding: 10px;
        text-align: center;
    }

    /* 表格样式 */
    table, tr, td {
        border: 1px solid black;
    }

    table {
        border-collapse: collapse;
        width: 100%;
    }
    tr {
        height: 40px
    }
    td {

```

```

        text-align:center;
        vertical-align:middle;
    }
    .checkbox{
        width:30px;
        height:30px;
    }
    .submit{
        width:100px;
        height:50px;
        background-color:red;
        color:white;
        font-size:20px;
    }
    .submit-div{
        width:100%;
        text-align: center;
    }
}

</style>
</head>
<body>
    <div class="header">
        <h1>药品库存管理系统</h1>
    </div>

    <div class="topnav">
        <a href="{% url 'demand:demand' %}">调拨</a>
        <a href="{% url 'checkout:checkout' %}">出库</a>
        <a href="{% url 'lack:lack' %}">缺货</a>
        <a href="{% url 'purchase:purchase' %}">采购</a>
        <a href="{% url 'storage:storage' %}">入库</a>
        <a href="{% url 'user:index' %}">回到首页</a>
        <a href="{% url 'user:logout' %}">注销</a>
    </div>
    <div class="row">
        <div class="column left">
        </div>
        <div class="column middle">
            <div class="submit-div">
                <h2> 生成调拨单 </h2>
            </div>
            <form action="{% url 'demand:submit_input' %}" method="post">
                {% csrf_token %}
                <table>
                    <tr>
                        <td><strong>当前调拨单号</strong></td>
                        <td> demand-{{demand_num}} </td>
                    </tr>
                    <tr>
                        <td><strong>选择调拨方</strong></td>
                        <td>
                            <select name="demander_id">
                                {% for demander in demander_list %}
                                    <option
value="{{demander.id}}">{{demander.name}}</option>
                                {% endfor %}
                            </select>
                        </td>
                    </tr>
                </table>
            </form>
        </div>
    </div>

```

```

        <tr>
            <td> <strong>序号</strong> </td>
            <td> <strong>药品编号</strong> </td>
            <td> <strong>药品类别</strong> </td>
            <td> <strong>药名</strong> </td>
            <td> <strong>数量</strong> </td>
        </tr>
        {% for medicine in medicine_list %}
        <tr>
            <td> {{forloop.counter}} </td>
            <td> medicine-{{medicine.id}} </td>
            <td> {{medicine.medicine_type.name}} </td>
            <td> {{medicine.name}} </td>
            <td><input type="text" name="{{medicine.id}}"
class="input_1"/></td>
        </tr>
        {% endfor %}
    </table>
    <br>
    <div class="submit-div">
        <input type="submit" value="提交" class="submit">
    </div>
</form>
</div>

<div class="column right">
</div>
</div>
<div class="footer">
    <p>项目地址: https://github.com/lx1374327576/drug\_management\_system</p>
    <p>Copyright © 2018-2019 1374327576@qq.com All Rights Reserved. 备案号:
bupt-2016211876</p>
</div>
</body>
</html>

```

demand/templates/query.html:

```

<html>
    <head>
        <meta charset="utf-8">
        <title>药品库存管理系统</title>
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <style>
            * {
                box-sizing: border-box;
            }

            body {
                margin: 0;
            }

            /* 头部样式 */
            .header {
                background-color: #f1f1f1;
                padding: 20px;
                text-align: center;
            }

            a {
                text-decoration: none;
            }

```



```

}

/* 导航条 */
.topnav {
    overflow: hidden;
    background-color: #333;
}

/* 导航链接 */
.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

/* 链接 - 修改颜色 */
.topnav a:hover {
    background-color: #ddd;
    color: black;
}

/* 创建三个相等的列 */
.column {
    float: left;
    padding: 10px;
}

/* 左侧宽度 */
.column.left {
    width: 20%;
    background-image: url("/static/2.jpeg");
    height: 1000px;
}

/* 右侧宽度 */
.column.right {
    width: 20%;
    background-image: url("/static/2.jpeg");
    height: 1000px;
}

/* 中间区域宽度 */
.column.middle {
    width: 60%;
}

/* 列后面清除浮动 */
.row:after {
    content: "";
    display: table;
    clear: both;
}

/* 响应式布局 - 宽度小于 600px 时设置上下布局 */
@media screen and (max-width: 600px) {
    .column.side, .column.middle {
        width: 100%;
    }
}

```

```

/* 底部样式 */
.footer {
    background-color: #f1f1f1;
    padding: 10px;
    text-align: center;
}

/* 表格样式 */
table, tr, td {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
    width: 100%;
}
tr {
    height: 40px;
}
td {
    text-align: center;
    vertical-align: middle;
}
.checkbox {
    width: 30px;
    height: 30px;
}
.submit {
    width: 50px;
    height: 30px;
    background-color: red;
    color: white;
    font-size: 20px;
}
.submit-div {
    width: 100%;
    text-align: center;
}
.submit {
    padding: 4px 6px;
}
.submit:hover {
    background-color: #ddd;
    color: black;
}
ul.pagination {
    display: inline-block;
    padding: 0;
    margin: 0;
}

ul.pagination li {display: inline;}

ul.pagination li a {
    color: black;
    float: left;
    padding: 8px 16px;
    text-decoration: none;
    transition: background-color .3s;
    border: 1px solid #ddd;
}

.pagination li:first-child a {
    border-top-left-radius: 5px;
    border-bottom-left-radius: 5px;
}

```

```

}

.pagination li:last-child a {
    border-top-right-radius: 5px;
    border-bottom-right-radius: 5px;
}

ul.pagination li a.active {
    background-color: #4CAF50;
    color: white;
    border: 1px solid #4CAF50;
}
ul.pagination li a:hover:not(.active) {background-color: #ddd;}
</style>
</head>
<body>
    <div class="header">
        <h1>药品库存管理系统</h1>
    </div>

    <div class="topnav">
        <a href="{% url 'demand:demand' %}">调拨</a>
        <a href="{% url 'checkout:checkout' %}">出库</a>
        <a href="{% url 'lack:lack' %}">缺货</a>
        <a href="{% url 'purchase:purchase' %}">采购</a>
        <a href="{% url 'storage:storage' %}">入库</a>
        <a href="{% url 'user:index' %}">回到首页</a>
        <a href="{% url 'user:logout' %}">注销</a>
    </div>
    <div class="row">
        <div class="column left">
        </div>
        <div class="column middle">
            <div class="submit-div">
                <h2> 调拨单一览 </h2>
            </div>
            <table>
                <tr>
                    <td> <strong>序号</strong> </td>
                    <td> <strong>调拨单编号</strong> </td>
                    <td> <strong>发起时间</strong> </td>
                    <td> <strong>发起人</strong> </td>
                    <td> <strong>详情</strong> </td>
                    <td> <strong>审核状态</strong> </td>
                    <td> <strong>按钮</strong> </td>
                </tr>
                {% for demand in demand_list %}
                <tr>
                    {% widthratio page_id_minus 1 10 as page_id_tmp %}
                    <td> {{forloop.counter|add:page_id_tmp}} </td>
                    <td> demand-{{demand.id}} </td>
                    <td> {{demand.create_time}} </td>
                    <td> {{demand.create_user.user_name}} </td>
                    <td><a href="{% url 'demand:query_detail'
demand.id %}">详情</a></td>
                    {% if demand.status == 0 %}
                    <td>
                        {{status_dict.0}}

```

```

        </td>
        <td>
            <a type="button" href="{% url 'demand:accept'
demand.id %}" class="submit">接受</a>
            <a type="button" href="{% url 'demand:deny'
demand.id %}" class="submit">拒绝</a>
        </td>
    {% endif %}
    {% if demand.status == 1 %}
        <td>
            {{status_dict.1}}
        </td>
        <td></td>
    {% endif %}
    {% if demand.status == 2 %}
        <td>
            {{status_dict.2}}
        </td>
        <td></td>
    {% endif %}
    {% if demand.status == 3 %}
        <td>
            {{status_dict.3}}
        </td>
        <td></td>
    {% endif %}
</tr>
{% endfor %}
</table>
<br>
<div class="submit-div">
    <ul class="pagination">
        {% if page_id < 4 %}
            {% if page_id == 1 %}
                <li><a href="{% url 'demand:query_page'
1 %}"><</a></li>
                <li><a class='active' href="{% url
'demand:query_page' 1 %}">1</a></li>
                <li><a href="{% url 'demand:query_page'
2 %}">2</a></li>
                <li><a href="{% url 'demand:query_page'
3 %}">3</a></li>
                <li><a href="{% url 'demand:query_page'
4 %}">4</a></li>
            {% endif %}
            {% if page_id == 2 %}
                <li><a href="{% url 'demand:query_page'
page_id_minus %}"><</a></li>
                <li><a href="{% url 'demand:query_page'
1 %}">1</a></li>
                <li><a class='active' href="{% url
'demand:query_page' 2 %}">2</a></li>
                <li><a href="{% url 'demand:query_page'
3 %}">3</a></li>
                <li><a href="{% url 'demand:query_page'
4 %}">4</a></li>
            {% endif %}
            {% if page_id == 3 %}
                <li><a href="{% url 'demand:query_page'
page_id_minus %}"><</a></li>
                <li><a href="{% url 'demand:query_page'
1 %}">1</a></li>
                <li><a href="{% url 'demand:query_page'
2 %}">2</a></li>
                <li><a class='active' href="{% url

```

```

'demand:query_page' 3 %}">3</a></li>
        <li><a href="{% url 'demand:query_page'
4 %}">4</a></li>
        {% endif %}
        {% if page_id == 4 %}
        <li><a href="{% url 'demand:query_page'
page_id_minus %}"><</a></li>
        <li><a href="{% url 'demand:query_page'
1 %}">1</a></li>
        <li><a href="{% url 'demand:query_page'
2 %}">2</a></li>
        <li><a href="{% url 'demand:query_page'
3 %}">3</a></li>
        <li><a class='active' href="{% url
'demand:query_page' 4 %}">4</a></li>
        {% endif %}
        <li><a href="{% url 'demand:query_page'
5 %}">5</a></li>
        <li><a href="{% url 'demand:query_page'
6 %}">6</a></li>
        <li><a href="{% url 'demand:query_page'
7 %}">7</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_plus %}">>>/a></li>
        {% endif %}
        {% if page_id >= 4 %}
        <li><a href="{% url 'demand:query_page'
page_id_minus %}"><</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_minus3 %}">{{page_id_minus3}}</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_minus2 %}">{{page_id_minus2}}</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_minus %}">{{page_id_minus}}</a></li>
        <li><a class="active" href="{% url 'demand:query_page'
page_id %}">{{page_id}}</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_plus %}">{{page_id_plus}}</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_plus2 %}">{{page_id_plus2}}</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_plus3 %}">{{page_id_plus3}}</a></li>
        <li><a href="{% url 'demand:query_page'
page_id_plus %}">>>/a></li>
        {% endif %}
    </ul>
</div>
</div>
</div>
<div class="column right">
</div>
<div class="footer">
<p>项目地址: https://github.com/lx1374327576/drug_management_system</p>
<p>Copyright © 2018-2019 1374327576@qq.com All Rights Reserved. 备案号:
bupt-2016211876</p>
</div>
</body>
</html>

```

demand/templates/submit.html:

```

<html>
<head>
<meta charset="utf-8">

```

```

<title>药品库存管理系统</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
    box-sizing: border-box;
}

body {
    margin: 0;
}

a{
    text-decoration:none;
}

/* 头部样式 */
.header {
background-color: #f1f1f1;
padding: 20px;
text-align: center;
}

/* 导航条 */
.topnav {
    overflow: hidden;
    background-color: #333;
}

/* 导航链接 */
.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

/* 链接 - 修改颜色 */
.topnav a:hover {
    background-color: #ddd;
    color: black;
}

/* 创建三个相等的列 */
.column {
    float: left;
    padding: 10px;
}

/* 左侧宽度 */
.column.left {
    width: 20%;
    background-image:url("/static/2.jpeg");
    height: 1000px;
}

/* 右侧宽度 */
.column.right {
    width: 20%;
    background-image:url("/static/2.jpeg");
    height: 1000px;
}

```

```

/* 中间区域宽度 */
.column.middle {
    width: 60%;
}

/* 列后面清除浮动 */
.row:after {
    content: "";
    display: table;
    clear: both;
}

/* 响应式布局 - 宽度小于 600px 时设置上下布局 */
@media screen and (max-width: 600px) {
    .column.side, .column.middle {
        width: 100%;
    }
}

/* 底部样式 */
.footer {
    background-color: #f1f1f1;
    padding: 10px;
    text-align: center;
}

/* 表格样式 */
table, tr, td {
    border: 1px solid black;
}

table {
    border-collapse: collapse;
    width: 100%;
}
tr {
    height: 40px;
}
td {
    text-align: center;
    vertical-align: middle;
}
.checkbox {
    width: 30px;
    height: 30px;
}
.submit {
    width: 100px;
    height: 50px;
    background-color: red;
    color: white;
    font-size: 20px;
}
.submit-div {
    width: 100%;
    text-align: center;
}
.submit-div a {
    padding: 14px 16px;
}
.submit-div a: hover {
    background-color: #ddd;
    color: black;
}

```

```

    }
</style>
</head>
<body>
    <div class="header">
        <h1>药品库存管理系统</h1>
    </div>

    <div class="topnav">
        <a href="{% url 'demand:demand' %}">调拨</a>
        <a href="{% url 'checkout:checkout' %}">出库</a>
        <a href="{% url 'lack:lack' %}">缺货</a>
        <a href="{% url 'purchase:purchase' %}">采购</a>
        <a href="{% url 'storage:storage' %}">入库</a>
        <a href="{% url 'user:index' %}">回到首页</a>
        <a href="{% url 'user:logout' %}">注销</a>
    </div>
    <div class="row">
        <div class="column left">
        </div>
        <div class="column middle">
            <div class="submit-div">
                <h2> 调拨单 </h2>
            </div>
            <br>
            <table>
                <tr>
                    <td> <strong>单号</strong> </td>
                    <td colspan="5"> demand-{{demand_form.id}} </td>
                </tr>
                <tr>
                    <td> <strong>发起单位</strong> </td>
                    <td colspan="5"> {{demand_form.demander.name}} </td>
                </tr>
                <tr>
                    <td> <strong>发起人</strong> </td>
                    <td colspan="5"> {{demand_form.create_user.user_name}}
</td>

                </tr>
                <tr>
                    <td> <strong>联系方式</strong> </td>
                    <td colspan="5"> 13757578780 </td>
                </tr>
                <tr>
                    <td> <strong>发起时间</strong> </td>
                    <td colspan="5"> {{demand_form.create_time}} </td>
                </tr>
                <tr>
                    <td> <strong>序号</strong> </td>
                    <td> <strong>药品编号</strong> </td>
                    <td> <strong>药品类别</strong> </td>
                    <td> <strong>药品名称</strong> </td>
                    <td> <strong>药品数量</strong> </td>
                    <td> <strong>价格</strong> </td>
                </tr>
                {% for detail in detail_list %}
                    <tr>

```



```

        <td> {{forloop.counter}} </td>
        <td> medicine-{{detail.medicine.id}} </td>
        <td> {{detail.medicine.medicine_type.name}} </td>
        <td> {{detail.medicine.name}} </td>
        <td> {{detail.num}} </td>
        <td> {{detail.price}} </td>
    </tr>
    {% endfor %}
</tr>
<td> <strong>发起人签字</strong> </td>
<td colspan="2"></td>
<td> <strong>收到人签字</strong> </td>
<td colspan="2"></td>
</tr>
</table>
<br>
<div class="submit-div">
    <a href="{% url 'demand:query' %}" class="submit">确认</a>
</div>
</div>

<div class="column right">
</div>
</div>
<div class="footer">
    <p>项目地址: https://github.com/lx1374327576/drug\_management\_system</p>
    <p>Copyright © 2018-2019 1374327576@qq.com All Rights Reserved. 备案号:
bupt-2016211876</p>
</div>
</body>
</html>

```

六、总结与展望

这是我的第一个 Django 框架实现的 web 作品，参考了官网上的部署文档，在前端没有使用任何框架，历时八个星期终于完成。于个人来看，还是比较满意的。

从技术上看，可以说该项目中采用的技术完全跟得上时代的步伐。现在很多网站都是使用 React 和 Django 进行简易开发的，效果还不错，还可以满足高并发的需求。由于时间投入的不是很多，所以在前端开发上，并没有采取 React 框架，而是采取了简易的 html 交互方式。包括分页原理，并没有使用 ajax，而是直接在 url 中作了点文章。在实现细节上没有什么难度，能够比较轻松的用数据库进行联合操作。Django 还自己提供了各种中间件和数据库简易部署操作，让程序员能够彻底从数据库管理工具中走出来，能够直接用代码和后台管理工具部署数据库、导入数据。这也是 Django 比较大的优点之一。经过了这个小学期，可以说是对 Django 有了比较多的了解，也能进行产出了，这是一个比较大的提升。

从业务流程上看，总体来讲对于大致的业务流程还是比较清楚的，但是在细节方面考虑的不是很清楚，这可能跟没有在实际生产环境实习过有关。我根据自己的想象和推理，设计出了药品的出库、采购、入库等流程，能够保证基本的单据流转，但是可能和生产实际不符。在用户体验方面，在单据流转中，采用了直接勾选的方式，总的来说是比较方便。虽然可能与传统方式不符，但不失为一种新颖的方式。在之后的工作中，面对全新的业务流程，要更加细致的分析和理解，更加贴近生产实际。

这个药品库存管理系统只是大学生涯中一个小小的作品，在之后的工作学习中还要再接再厉，争取在一个方向做到顶尖。

七、致谢

感谢苏老师为该作品提出的意见和建议！

感谢所有参与测试的同学！

感谢舍友对该作品的吐槽！

八、参考文献

本项目未参考任何文献，仅参考了 Django 官方部署文档。

官网地址 <https://www.djangoproject.com/>

九、项目进度计划

第五周：

1. 选择“药品库存管理系统”作为项目课题
2. 了解系统概况和国内外相关性研究。
3. 进行系统可行性分析。

第六周到第八周：

1. 完成需求分析。
2. 完成组织结构分析。
3. 完成业务流程分析。
4. 完成数据与数据流程分析。
5. 状态图设计。
6. 数据库 pdm 图设计。

第九周到第十二周：

1. 熟悉 django 框架和部署方法。
2. 做一个后端部署 demo。
3. 设计 url 结构和 django-app 结构。
4. 完成调拨流程。
5. 完成出库/缺货流程。
6. 完成采购/入库流程。
7. 完成 html 设计和美化。
8. 增加分页功能。
9. 增加其他功能。
10. 完成后台管理界面。

十、自我评价

这是我的第一个 Django 框架实现的 web 作品，参考了官网上的部署文档，在前端没有使用任何框架，历时八个星期终于完成。于个人来看，还是比较满意的，综合评价：良。

从技术上看，可以说该项目中采用的技术完全跟得上时代的步伐。现在很多网站都是使用 React 和 Django 进行简易开发的，效果还不错，还可以满足高并发的需求。由于时间投入的不是很够，所以在前端开发上，并没有采取 React 框架，而是采取了简易的 html 交互方式。包括分页原理，并没使用 ajax，而是直接在 url 中作了点文章。在实现细节上没有什么难度，能够比较轻松的用数据库进行联合操作。Django 还自己提供了各种中间件和数据库简易部署操作，让程序员能够彻底从数据库管理工具中走出来，能够直接用代码和后台管理工具部署数据库、导入数据。这也是 Django 比较大的优点之一。经过了这个小学期，可以说是对 Django 有了比较多的了解，也能进行产出了，这是一个比较大的提升。

从业务流程上看，总体来讲对于大致的业务流程还是比较清楚的，但是在细节方面考虑的不是很清楚，这可能跟没有在实际生产环境实习过有关。我根据自己的想象和推理，设计出了药品的出库、采购、入库等流程，能够保证基本的单据流转，但是可能和生产实际不符。在用户体验方面，在单据流转中，采用了直接勾选的方式，总的来说是比较方便。虽然可能与传统方式不符，但不失为一种新颖的方式。在之后的工作中，面对全新的业务流程，要更加细致的分析和理解，更加贴近生产实际。

这个药品库存管理系统只是大学生涯中一个小小的作品，在之后的工作学习中还要再接再厉，争取在一个方向做到顶尖。

十一、周期报告

具体可参考工程中的 `git commit history`

第五周：

1. 选择“药品库存管理系统”作为项目课题
2. 了解系统概况和国内外相关性研究。
3. 进行系统可行性分析。

第六周：

1. 完成需求分析。
2. 完成组织结构分析。
3. 完成业务流程分析。
4. 完成数据与数据流程分析。

第七周：

1. 状态图设计。
2. 数据库 pdm 图设计。

第八周：

1. 熟悉 django 框架和部署方法。
2. 做一个后端部署 demo。

第九周：

1. 设计 url 结构和 django-app 结构。
2. 完成调拨流程。

第十周：

1. 完成出库/缺货流程。
2. 完成采购/入库流程。

第十一周：

1. 完成 html 设计和美化。
2. 增加分页功能。

第十二周：

1. 增加库存查询功能。
2. 完成后台管理界面。