河北工业大学

软件工程课程实验

实验报告

题目： 设备管理系统

专业： 计算机科学与技术

班级： 中法计172

组： 第十组

组长： 张心宇

成员： 祁惟明；张瑞林；刘俊池；成昭；刘维

指导教师： 张健楠

完成日期： 2019.6.7

**目 录**

1 软件需求规格说明书………………………………………………（3 ）

2 设计规格说明书……………………………………………………（13）

3 源程序清单………… ……………………………………………（31）

4 测试报告……………………………………………………………（40）

**需求规格说明书**

**1.概述**

1.1项目的目的与目标

实现公司设备的有效管理、维护记录。

本管理系统通过计算机对实验设备进行自动管理，设备管理员可以直接在计算机实现实验设备的信息管理，并在一定程度上实现自动化，改善了工作运行效率和效果。

1.2术语定义

1．开发软件：软件开发是根据用户要求建造出软件系统或者系统中的软件部分的过程

2．漏洞：漏洞是在硬件、软件协议的具体实现或系统安全策略上存在的缺陷，从而可以使攻击者能够在未授权的情况下访问或破坏系统。

3．硬件环境：计算机及其外围设备组成的计算机物理系统。 即硬件设施，是指由传播活动所需要的那些物质条件、有形条件之和构筑而成的环境。

1.3 相关文档

1.需求规格说明书

2.设计规格说明书

3.测试报告

**2.问题初始分析**

2.1用户需求描述

（1）购入新设备，登记设备购入时间、名称、型号、种类、内部编号、所属部门、存放位置等。  
（2）设备转借时，记录借用部门、经手人员、用途、拟还日期等。  
（3）设备故障时，记录故障情况、维修情况（时间、人员、结果）等。  
（4）设备报废时，记录报废原因、批准人员、经手人员、处理方式等。  
（5）对设备管理人员和使用人员提供各种设备处理的按部门、设备类型、处理日期等查询和修改

2.2初始功能提取

功能需求列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编 号 | 功 能 名 称 | 使 用 人 | 功 能 描 述 | 输 入 内 容 | 输 出 内 容 |
| 1 | 设备采购管理 | 采购人员与管理人员 | 登记设备购入时间、名称、时间、型号、种类、内部编号、所属部门、存放位置等 | 新设备信息 | 新设备编号 |
| 2 | 设备转借管理 | 使用人员与管理人员 | 记录借用部门、经手人员、用途、拟还日期等 | 转借设备信息 | 转借设备信息 |
| 3 | 设备故障管理 | 维修人员与管理人员 | 记录故障情况、维修情况等 | 设备故障与维修信息 | 设备故障维修信息 |
| 4 | 设备报废管理 | 管理人员 | 记录报废原因、批准人员、经手人员、处理方式等 | 报废设备信息 | 报废设备信息 |
| 5 | 设备查询和修改管理 | 管理人员和使用人员 | 提供各种设备处理的按部门、类型、处理日期等的查询和修改情况 | 设备信息 | 查询或修改后信息 |

**3.系统用例分析**

3.1角色分析

1. 角色列表

|  |  |  |  |
| --- | --- | --- | --- |
| 编 号 | 角色名 | 所属部门 | 简要描述 |
| 1 | 采购人员 | 采购部 | 购入新设备并登记设备购入时间、名称、型号、种类、内部编号、所属部门、存放位置 |
| 2 | 管理人员 | 管理部 | 记录借用部门、经手人员、用途、拟还日期 |
| 3 | 维修人员 | 维修部 | 进行故障维修，记录报废原因、批准人员、经手人员、处理方式 |
| 4 | 使用人员 | 相关部门 | 进行设备的使用，记录相关信息 |
| 5 | 经理 | 领导部门 | 进行登陆，密码修改，分配权限 |

3.2用例分析

1. 用例列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 用例名称 | 角色 | 优先级 | 描述 |
| 1 | 登陆 | 经理 | 高 | 经理登陆系统 |
| 2 | 分配权限 | 经理 | 中 | 登陆系统更改权限 |
| 3 | 更改密码 | 经理 | 高 | 登陆系统更改密码 |
| 4 | 记录采购信息 | 采购管理人员 | 中 | 采购管理人员记录采购信息，数量，时间，价格 |
| 5 | 修改密码 | 采购管理员 | 中 | 采购管理员登陆修改密码 |
| 6 | 修改密码 | 库存管理人员 | 低 | 库存管理人员登陆系统修改密码 |
| 7 | 登陆系统 | 库存管理人员 | 高 | 库存管理人员登陆系统 |
| 8 | 库存管理 | 库存管理人员 | 中 | 库存管理人员检查库存状态和库存数量 |
| 9 | 修改密码 | 设备维修人员 | 低 | 设备维修人员登陆系统修改密码 |
| 10 | 登陆 | 设备维修人员 | 高 | 设备维修人员登陆系统 |
| 11 | 增修信息 | 设备维修人员 | 中 | 设备维修人员增加维修信息 |

### 3.3 用例建模（Use Case Modeling）

**3.3.1 \*\*\*子系统（模块）用例模型**

**（1）用例图**

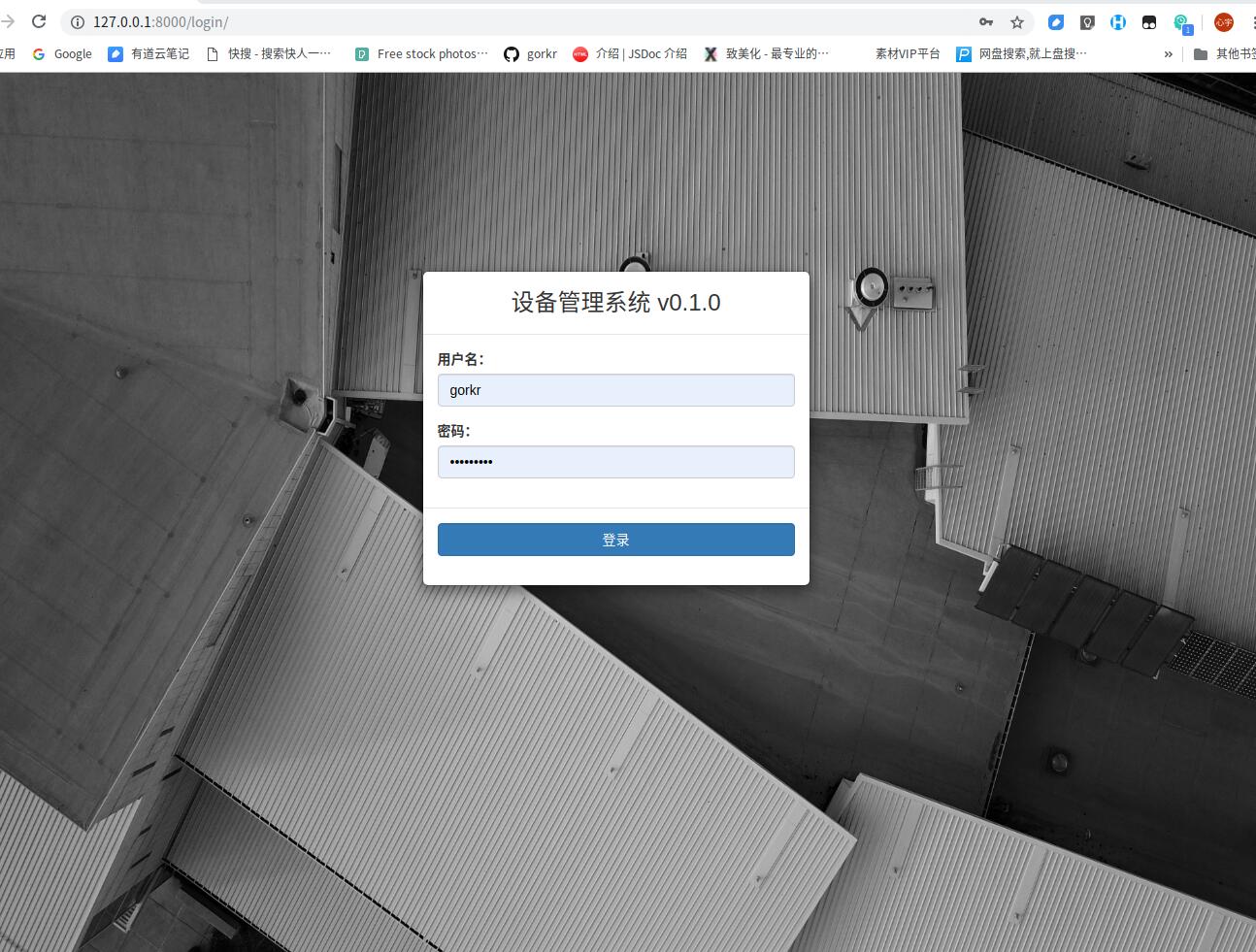
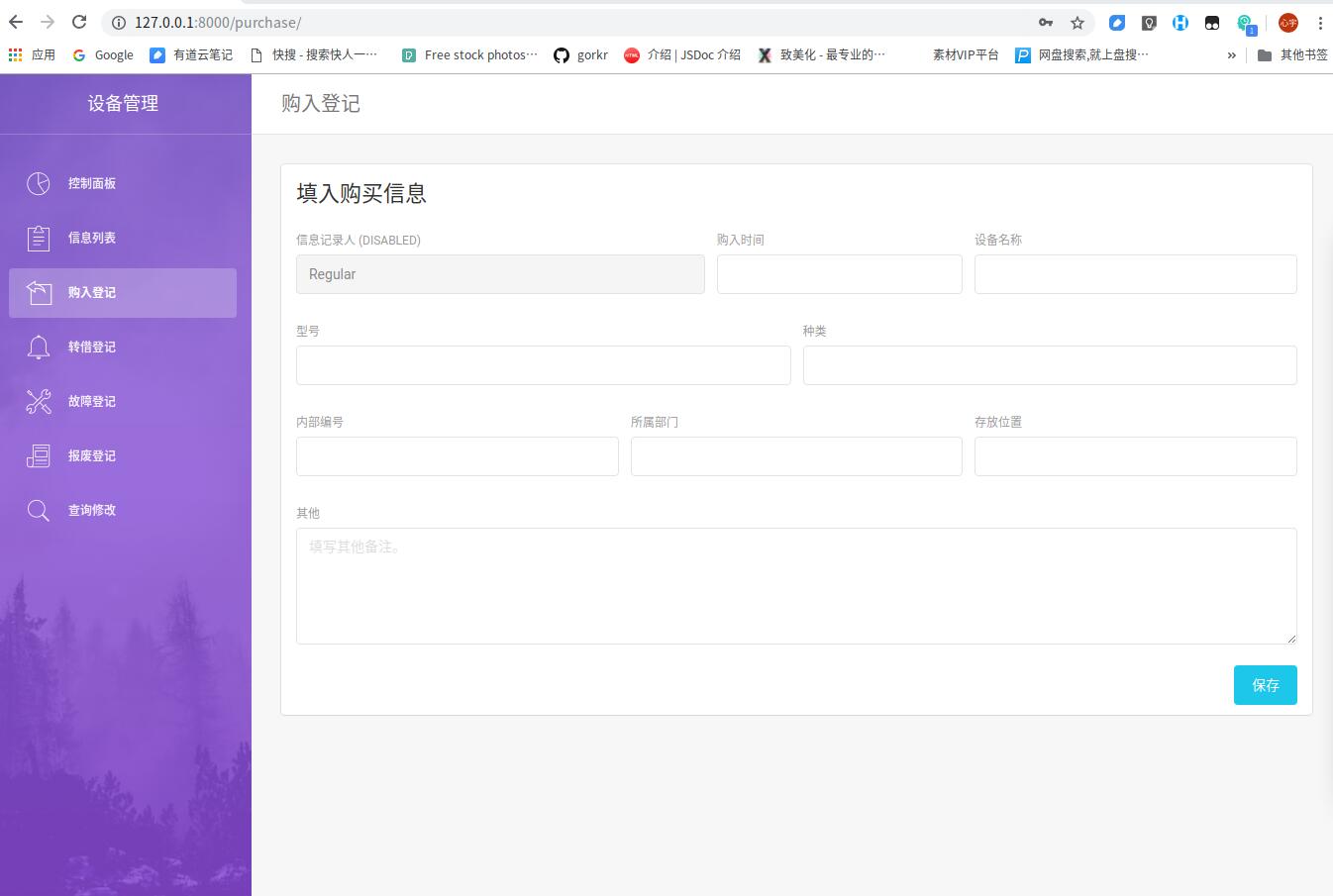
****

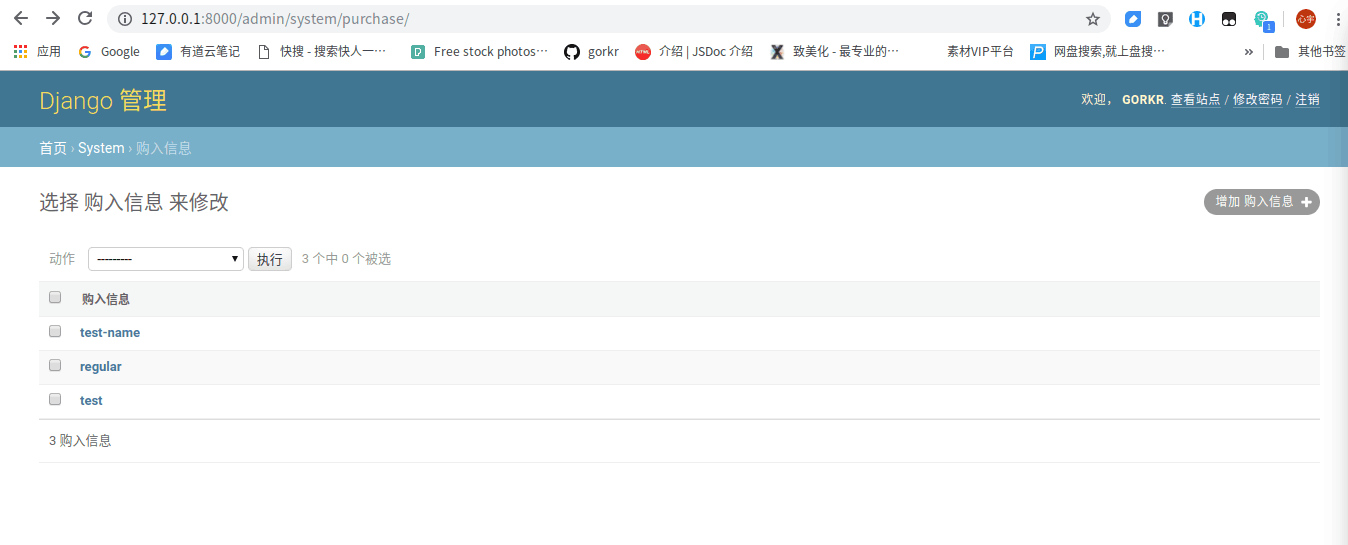
**（2）用例表**

|  |  |
| --- | --- |
| 用例名称 | 登陆 |
| 描述 | 此用例用于描述相关人员登陆系统  参与者 使用人员和管理人员 |
| 优先级 | 基本的 |
| 前置条件 | 相关人员打开系统 |
| 后置条件 | 使用者登入系统，开始操作 |
| 主事件流 | 1. 相关人员打开系统  2. 输入密码登陆 |

|  |  |
| --- | --- |
| 用例名称 | 采购 |
| 描述 | 采购人员填写采购信息 |
| 参与者 | 采购人员，管理人员 |
| 优先级 | 基本的 |
| 前置条件 | 采购人员采购完成 |
| 后置条件 | 新设备信息录入系统 |
| 主事件流 | 1. 采购人员采购完成 2. 管理人员开始输入采购信息 |

**（3）界面原型**

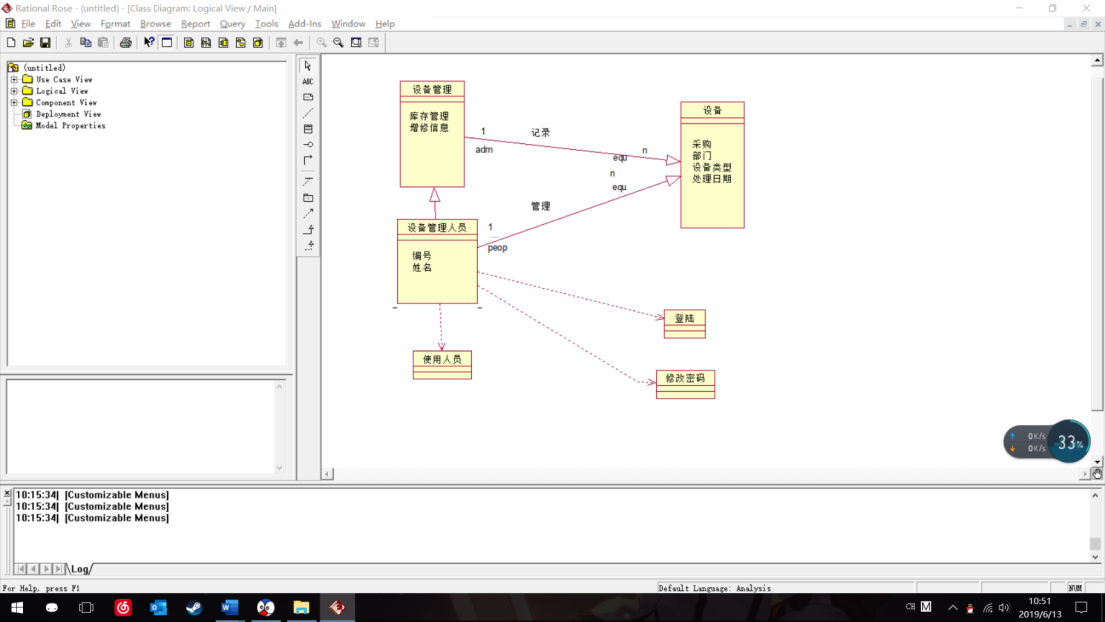
1. 



### 3.4 数据建模（Data Modeling）

|  |  |  |
| --- | --- | --- |
| 候选关键抽象表格 | | |
| 候选的关键抽象 | 排除的原因 | 选定的原因 |
| 设备 |  | 设备 |
| 修改密码 |  | 修改密码 |
| 登陆 |  | 登陆 |
| 库存管理 | 设备管理的属性 |  |
| 增修信息 | 设备管理的属性 |  |
| 采购 | 设备属性 |  |
| 部门 | 设备属性 |  |
| 设备类型 | 设备属性 |  |
| 处理日期 | 设备属性 |  |
| 设备管理人员 |  | 设备管理人员 |
| 使用人员 |  | 使用人员 |
| 设备管理 |  | 设备管理 |

|  |  |
| --- | --- |
| 管理设备 | |
| 职责 | 协作者 |
| 登陆系统  修改密码  添加购买信息  删除购买信息  分配权限 | 管理人员  设备管理系统  采购人员  库存管理员 |



**4.非功能需求**

### 4.1 系统性能目标

1. 1.付出较低的开发成本  
   2.达到要求的软件功能  
   3取得较好的软件性能  
   4.开发的软件易于移植  
   5.需要较低的维护费用  
   6.能按时完成开发任务，及时完成开发任务，及时交付费用  
   7.开发的软件可靠性高

### 4.2 安全性

### 4.3 可靠性

### 4.4 灵活性

### 4.5 特殊需求

1．是否功能齐全，运行稳定；

2．在网络功能方面是否方便管理；

3．是否具有多任务，高效率的特点；

4．开发时间及费用支出是否超出预期；

## 4.6 目标系统假设与约束条件

1．建议开发软件运行的最短寿命：五年

2．进行系统方案选择比较的期限：一周

2．经费来源和使用限制 ：使用时应定期维护，防止系统出现高漏洞而带来些复杂问题。

3．法律和政策方面的限制：此软件设计合法，不存在于法律、政策相违背的现象。

5．硬件环境：若干台计算机

**二、设计规格说明书**

## 1．引言（Introduction）

### 1.1 目的（Purpose）

本文档根据设备管理系统的需求规格说明书，定义了软件系统结构和系统的主要功能模块以及相互之间的联系。

### 1.2 命名规则（Naming Rule）

(1) Pascal规范:所有单词首字母大写.

例如: UserNameTable

Came1规范:除了第一个单词,所有单词首字母大写.

例如: userNameTable

全局变量:使用Came1规范,并以g-开头.

例如: g-failedFlag

局部变量、方法参数:使用Came1规范.

例如: 1ineCount

类名:使用Pascal规范

例如: MyClass

成员方法:使用Pascal规范

例如: MyMethod O

普通成员变量: Pascal并且以m-开头.

例如: m-MemberVarible

控件变量: Control类型使用Pascal规范,以m-开头,控件名结尾

例如:m-SubmitButton

Value类型使用Pascal规范,以m-开头.

例如: m-Datal

(2)数据库命名规则:

表中字段名均不加前缀

表名均以“dt-table-”为前缀

索引名均以"dt-index-”为前级

视图名均以"dt-view.”为前缀

### 1.3 术语定义（Terms Glossary）

表3-1 术语定义或解释表

|  |  |  |
| --- | --- | --- |
| 序 号 | 术 语 名 称 | 术 语 定 义 |
| 1 | 总体结构 | 软件系统的总体逻辑结构。按照不同的设计方法，有不同的总体逻辑结构。若采用面向功能或面向数据的设计方法，则总体逻辑结构为一树形的功能模块结构图。若采用面向对象或面向部件（构件）的设计方法，则总体逻辑结构为部件（构件）的组装图 |
| 2 | 逻辑架构 | 它的目的是将系统分解成多个独立的子系统，本质上是遵循分而治之的理念。 |
| 3 | 子系统 | 具有相对独立功能的小系统叫做子系统。一个大的软件系统可以划分为多个子系统，每个子系统可由多个模块或多个部件组成 |
| 4 | 程序逻辑 | 程序逻辑是描述和论证程序行为的逻辑，又称[霍尔逻辑](https://baike.baidu.com/item/%E9%9C%8D%E5%B0%94%E9%80%BB%E8%BE%91/10007845" \t "_blank)。程序和逻辑有着本质的联系。如果把程序看成一个执行过程，程序逻辑的基本方法是先给出建立程序和逻辑间联系的[形式化方法](https://baike.baidu.com/item/%E5%BD%A2%E5%BC%8F%E5%8C%96%E6%96%B9%E6%B3%95/4298196" \t "_blank)，然后建立程序逻辑系统，并在此系统中研究程序的各种性质。 |
| 5 | 数据库 | 而言之可视为[电子化](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E5%8C%96" \t "_blank)的[文件柜](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6%E6%9F%9C" \t "_blank)——存储电子[文件](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6" \t "_blank)的处所，用户可以对文件中的数据进行新增、截取、更新、删除等操作。 |
| 6 | 参考资料 | 参考资料是指本文件书写时用到的其他资料。如各种有关规范、模板、标准、准则 |

### 1.4 参考资料（References）

[1] 用户需求报告

[2] 数据库设计规范

[3] 命名规范

### 1.5 相关文档（Related Documents）

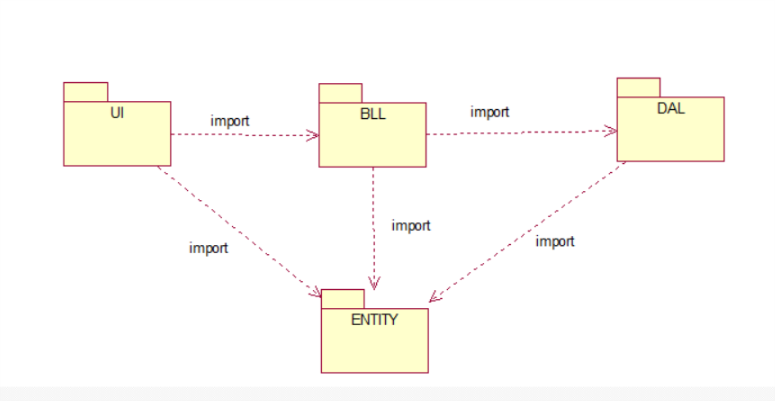
[1] 源程序清单

[2] 测试计划及报告

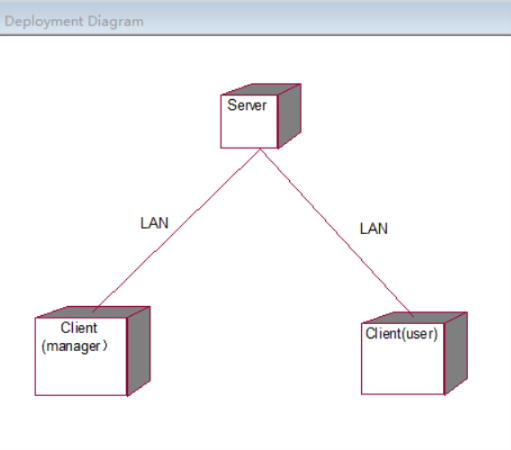
## 2．总体设计（Design of Collective）

### 2.1 体系结构设计（Design of Architecture）

**2.1.1 系统逻辑架构**



**2.1.2 系统物理架构**

****

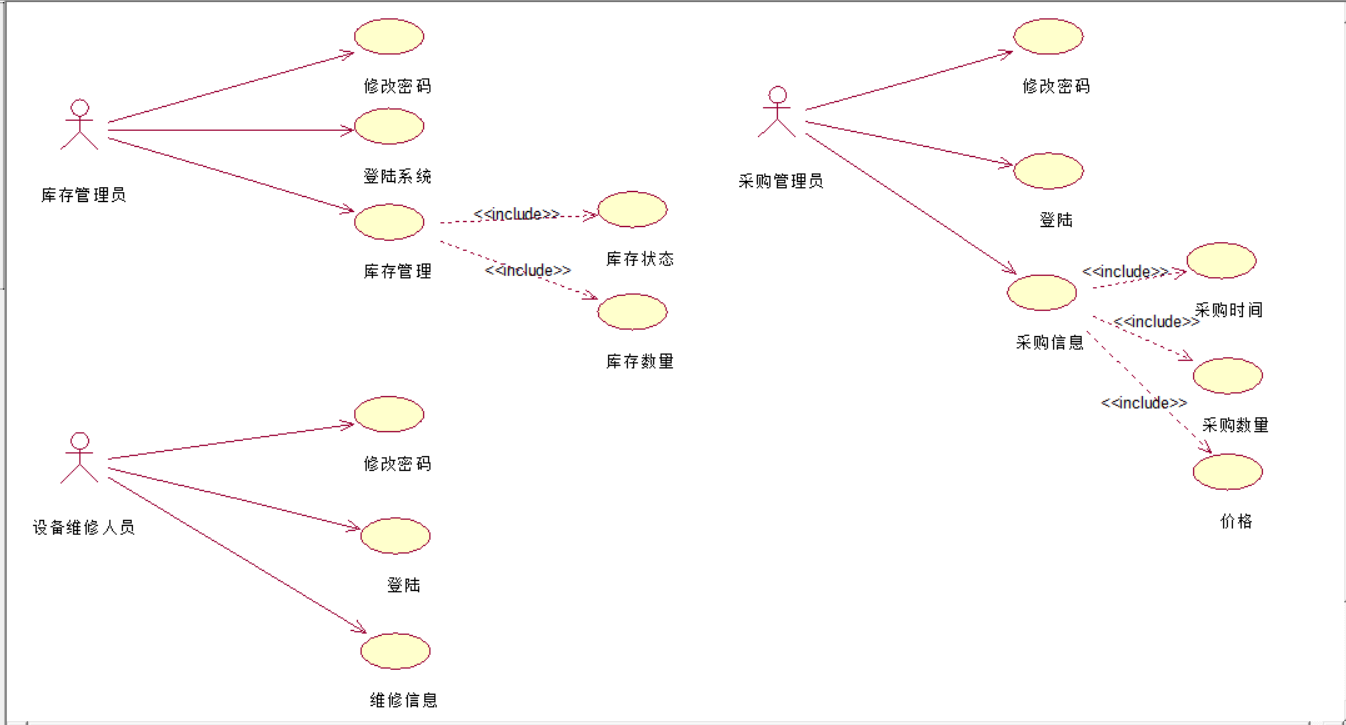
### 2.2 子系统清单（Subsystem List）

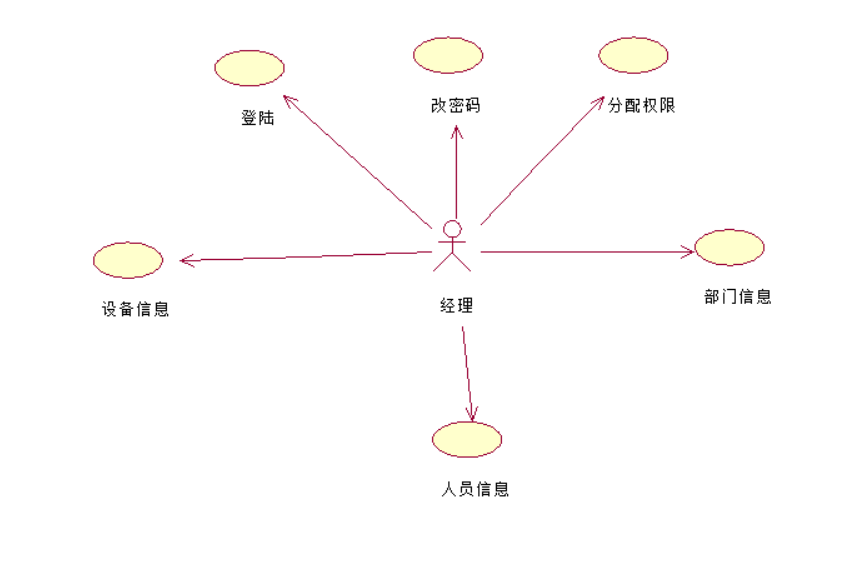
表3-2 子系统清单

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号 | 子系统英文名 | 子系统功能简述 | 子系统之间的关系 |
| SS1 | UI | [用户](https://baike.baidu.com/item/%E7%94%A8%E6%88%B7" \t "_blank)界面(User Interface)是指对[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053" \t "_blank)的人机交互、操作逻辑、界面美观的整体设计。好的UI设计不仅是让[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053" \t "_blank)变得有个性有品味，还要让软件的操作变得舒适、简单、自由、充分体现软件的定位和特点。 | 表现层依赖业务逻辑层以实现业务数据处理.  编写代码传递数据给BLL层。 |
| SS2 | BLL | 即业务逻辑层，主要是针对具体的问题的操作，也可以理解成对数据层的操作，对数据业务逻辑处理。如果说数据层是积木，那逻辑层就是对这些积木的搭建。 | 业务逻辑层依赖数据访问层来读取数据库中的数据或将处理结果存入数据库中。（添加对DAL层的引用） |
| SS3 | DAL | 数据访问层的英文缩写，即为数据访问层（Data Access Layer）。其功能主要是负责数据库的访问。 |  |
| SS4 | ENTITY | 实体就是一个特定的软件模块。 |  |

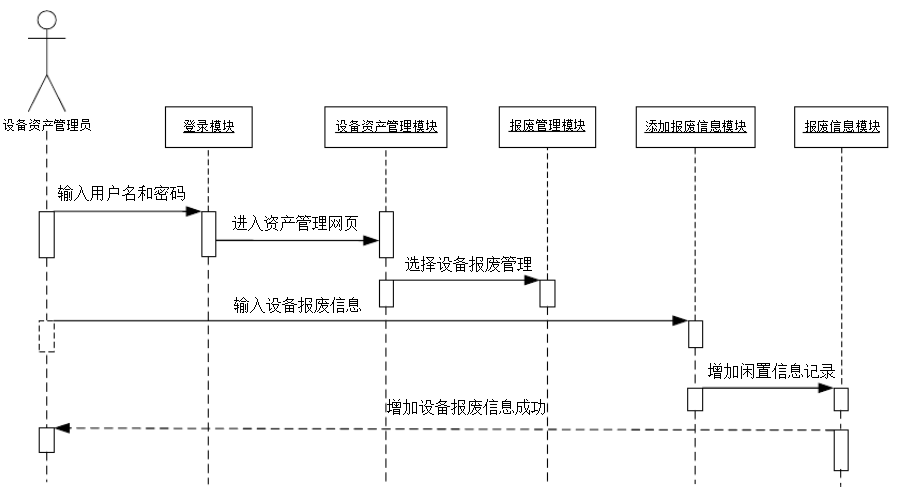
### 2.3 模块设计（Module Design）

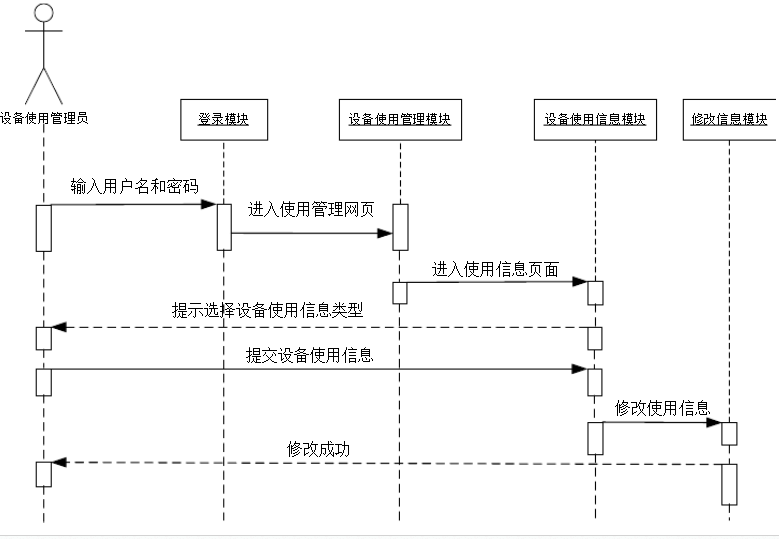
**2.3.1 用例实现**

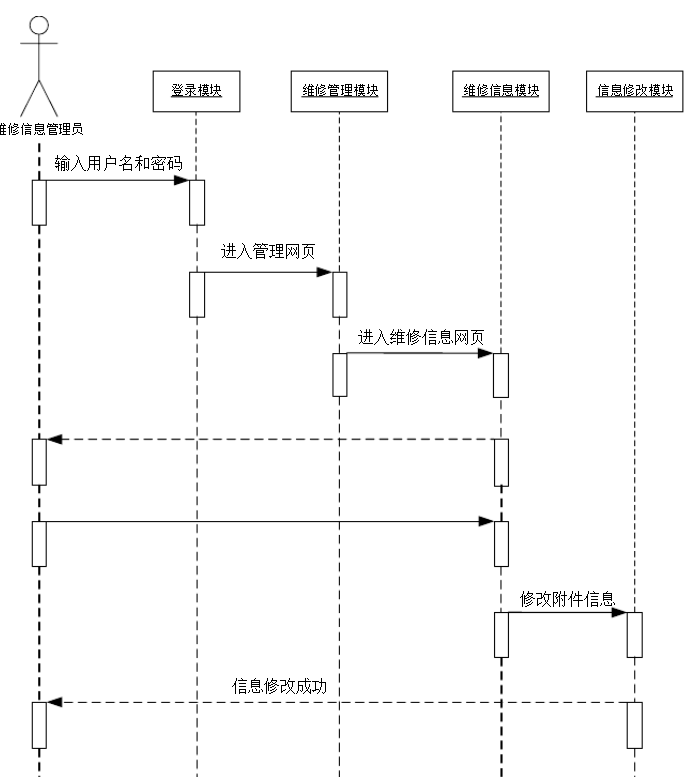
****

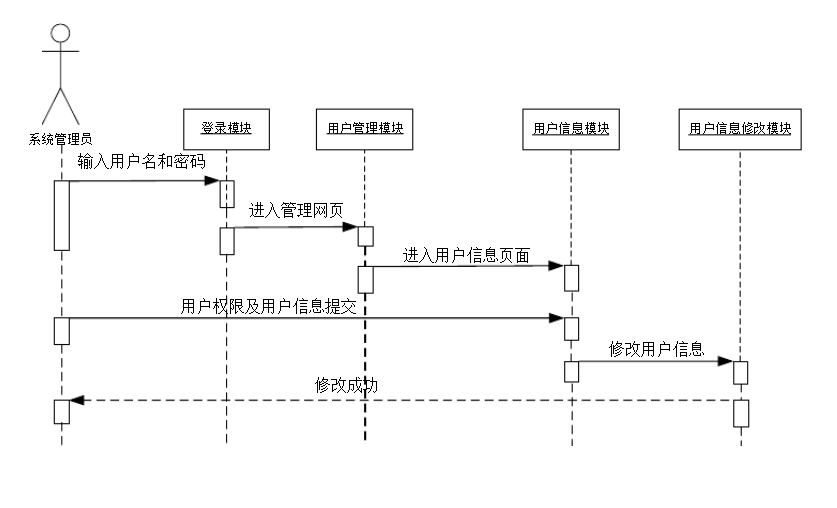
****

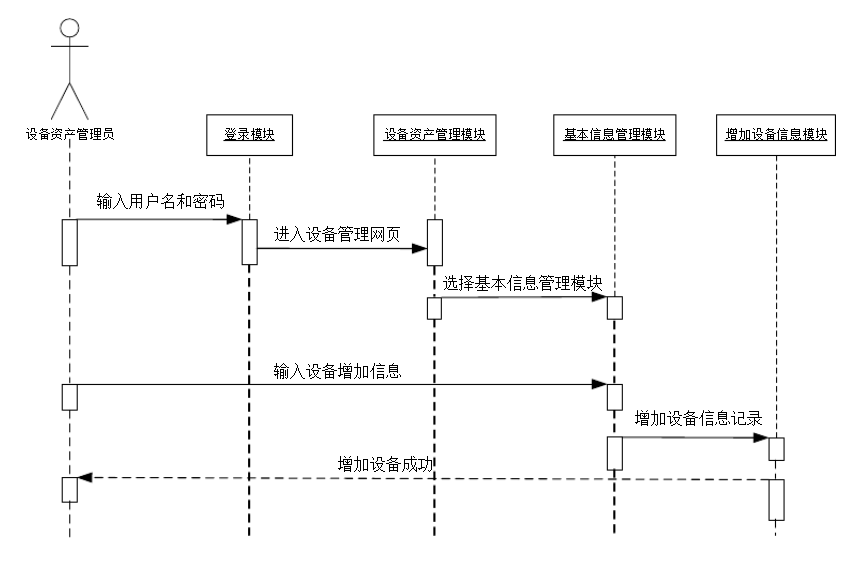
**序列图：**

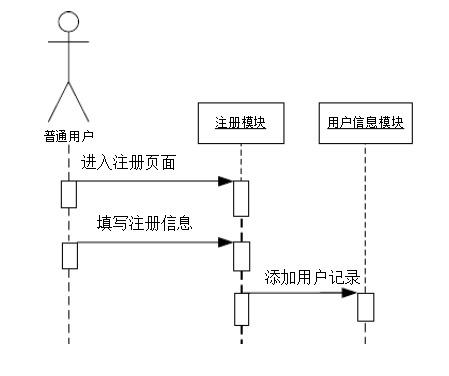
****

****

****

****

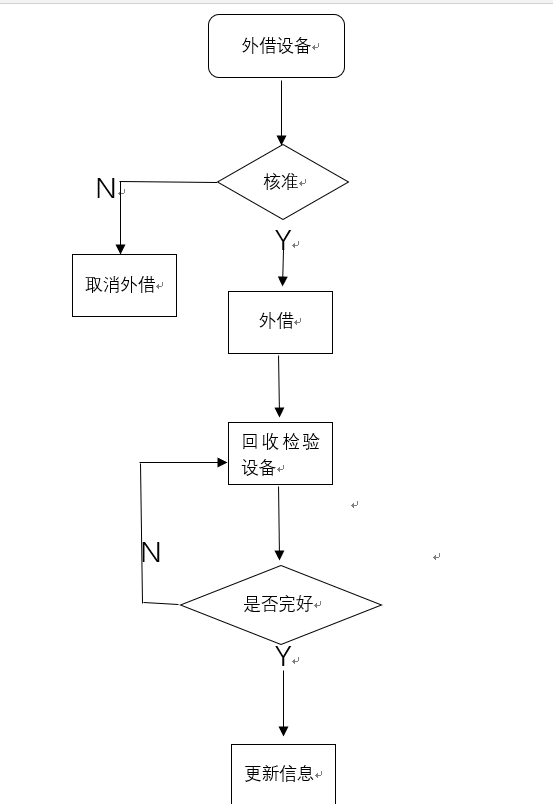
****

****

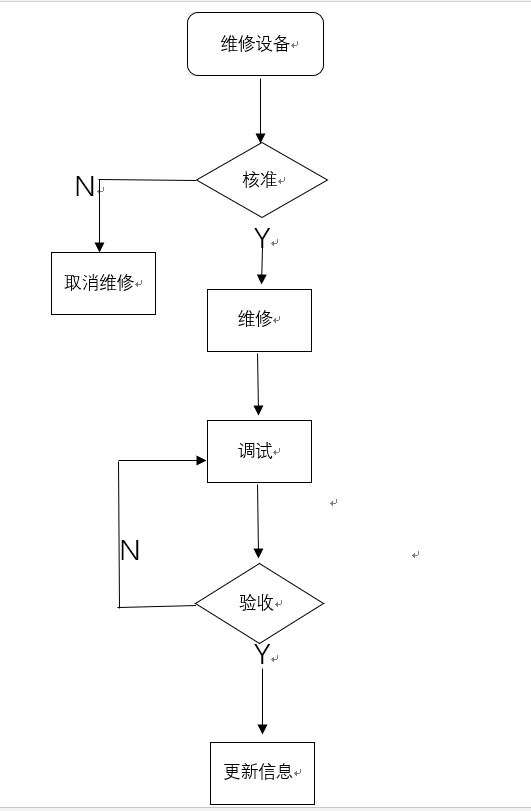
### D:\软件\QQ\QQ的数据\840686724\FileRecv\流程1.PNG2.3.2 关键程序逻辑设计

（1）设备采购

（2）外借设备



（3）维修设备



## 3．数据结构设计（Design of Data Structure）

### 3.1 数据库表名清单（DB Table List）

表3-4 数据库表名清单

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 中文表名 | 英文表名 | 表功能说明 |
| 1 | 用户 | User | 记录信息 |
| 2 | 购入设备表 | Equipment purchase list | 记录购入设备的信息 |
| 3 | 设备转借表 | Equipment borrowing table | 记录设备转借的信息 |
| 4 | 设备故障表 | Equipment fault list | 记录设备故障的信息 |
| 5 | 设备报废表 | Equipment scrap list | 记录设备报废的信息 |
| 6 | 设备一览表 | [list](javascript:;) [of](javascript:;) [the](javascript:;) [equipment](javascript:;) | 记录设备信息 |
| 7 | 设备维修表 | Equipment maintenance list | 记录设备维修信息 |

### 3.2 数据库表之间关系说明（Relation of DB Table）

用户图：

使用年限

报废日期

折现率

维修信息

价格

状态

分类号

类型号

设备种类

设备号

设备信息实体

用户信息实体

密码

等级

姓名

设备图：

设备分类表：

种类id

设备分类表

名称

分类id

设备种类表：

设备种类表

名称

种类id

编号

设备外借表：

外借id

部门id

设备编号

备注

使用时间

外借日期

使用人

设备外借表

购买表：

型号

购买日期

名称

存放位置

设备编号

设备购买表

使用部门

种类

部门信息：

部门实体信息

父部门

部门名称

部门ID

用户权限表：

用户权限表

权限名称

权限ID

描述

设备维修图：

维修时间

设备ID

设备维修实体信息

维修结果

维修人员

### 3.3 数据库表的详细清单（Particular List of DB Table）

表3-5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段中文名 | 字段英文名 | 类型、宽度、精度 | 取值约束 | 空否 | 默认值 | 主键/外键 | 索引否 |
| 1 | 用户性别 | User.Sex | 字符/5 | Man/Woman | 否 | Man | 外键 | 索引 |
| 2 | 用户工号 | User.num | 整型/10 | 大于零  小于10位 | 否 | 0000000000 | 外键 | 索引 |
| 3 | 操作时间 | User.time | 整型/8 | 月份大于零小于12  日期大于0小于31 | 否 | 1999/04/02 | 主键 | 索引 |
| 4 | 机器编号 | Number | 整型/10 | 大于零  不超10位数 | 否 | 0000000000 | 外键 | 索引 |
| 5 | 购入时间 | N．time | 整型/8 | 月份大于零小于12  日期大于0小于31 | 否 | 2000/00/00 | 外键 | 否 |
| 6 | 名称 | N．name | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 7 | 型号 | N．type | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 8 | 种类 | N．tag | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 9 | 内部编号 | N．num | 整型/10 | 大于零  不超10位数 | 否 |  | 外键 | 否 |
| 10 | 所属部门 | N．depart | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 11 | 存放位置 | N．location | 整型/10 | 大于零  不超10位数 | 否 |  | 外键 | 否 |
| 12 | 借用部门 | S．depart | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 13 | 经手人员 | S．user | 整型/10 | 大于零  不超10位数 | 否 |  | 外键 | 否 |
| 14 | 用途 | S．way | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 15 | 拟换日期 | S．date | 整型/8 | 月份大于零小于12  日期大于0小于31 | 否 | 2011/00/00 | 外键 | 否 |
| 16 | 故障情况 | F．con | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 17 | 维修情况 | F．repair | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 18 | 报废原因 | D．scarp | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |
| 19 | 批准人员 | D．puser | 整型/10 | 大于零  不超10位数 | 否 |  | 外键 | 否 |
| 20 | 经手人员 | D．juser | 整型/10 | 大于零  不超10位数 | 否 |  | 外键 | 否 |
| 21 | 处理方式 | D．way | 字符/10 | 小于10字符 | 否 |  | 外键 | 否 |

# 三、源程序清单

## 1 用户登录模块

### 1.1 描述 (Description)

实现用户登录系统，登陆后根据用户类型获得相应权限。

### 1.2 代码(Program)

#### 数据模型 models.py

# models.py  
# 链接mysql，数据模型  
  
class User(models.Model):  
 '''用户表'''  
  
 gender = (  
 ('male', '男'),  
 ('female', '女'),  
 )  
  
 POS = (  
 ('U', '使用人员'),  
 ('A', '管理人员')  
 )  
  
 name = models.CharField(max\_length=128, unique=True, primary\_key=True)  
 password = models.CharField(max\_length=256)  
 email = models.EmailField(unique=True)  
 sex = models.CharField(max\_length=32, choices=gender, default='男')  
 c\_time = models.DateTimeField(auto\_now\_add=True)  
 position = models.CharField(max\_length=1, choices=POS)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 ordering = ['c\_time']  
 verbose\_name = '用户'  
 verbose\_name\_plural = '用户'

#### 逻辑视图 views.py

@csrf\_exempt  
def login(request):  
  
 """  
 登录审查  
 """  
  
 if request.method == "POST":  
 username = request.POST.get('username', None) # 根据name属性获取  
 password = request.POST.get('password', None)  
 message = ""  
 if username and password:  
 username = username.strip()  
 try:  
 user = models.User.objects.get(name=username)  
 if user.password == password:  
 return redirect('/purchase/') # 如果在这里传参 user  
 else:  
 message = "密码错误，请重新输入。"  
 except:  
 message = "用户名错误，请重新输入"  
 return render(request, 'system/login.html', {"message": message})  
 return render(request, 'system/login.html')

#### 管理模块注册 admin.py

admin.site.register(models.User)

#### 地址映射 urls.py

urlpatterns = [  
 url(r'^login/', views.login),  
]

#### 登录页面 login.html

{% load static %}  
{% block content %}  
<div class="modal-dialog" style="margin-top: 15%;width: 400px">  
 <div class="modal-content">  
 <form class='form-login' action="/login/" method="post">  
 {% if message %}  
 <div class="alert alert-warning">{{ message }}</div>  
 {% endif %}  
  
  
 {% csrf\_token %}  
 <div class="modal-header">  
 <h3 class="modal-title text-center" id="myModalLabel">设备管理系统 v0.1.0</h3>  
 </div>  
  
 <div class="modal-body" id = "model-body">  
 <div class="form-group">  
 <label for="id\_username">用户名：</label>  
 <input type="text" name="username" class="form-control" placeholder="Username" autocomplete="off" id="id\_username" autofocus required>  
 </div>  
 <div class="form-group">  
 <label for="id\_password">密码：</label>  
 <input type="password" name="password" class="form-control" placeholder="Password" autocomplete="off" id="id\_password" required>  
 </div>  
 </div>  
 <div class="modal-footer">  
 <div class="form-group" >  
 <button type="submit" class="btn btn-primary form-control" id="login">登录</button>  
 </div>  
  
 </div>  
 </form>  
 </div><!-- /.modal-content -->  
</div><!-- /.modal -->  
{% endblock %}

## 2 购入登记模块

### 2.1 描述(Description)

记录购买信息，存入数据库，并更新table页面。

### 2.2 代码(Program)

#### 数据模型 models.py

class Purchase(models.Model):  
 '''  
 被谁购入信息记录表  
 '''  
  
 recoder = models.CharField(max\_length=128, verbose\_name="记录人") # 应该使用外键值对  
 date = models.CharField(max\_length=128, verbose\_name="时间")  
 name = models.CharField(max\_length=128, verbose\_name="设备名称")  
 type = models.CharField(max\_length=128, verbose\_name="型号")  
 category = models.CharField(max\_length=128, verbose\_name="种类")  
 number = models.BigIntegerField(unique=True, primary\_key=True, verbose\_name="编号 ")  
 part = models.CharField(max\_length=128, verbose\_name="所属部门")  
 location = models.CharField(max\_length=128, verbose\_name="存放位置")  
 c\_time = models.DateTimeField(auto\_now\_add=True)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 class Meta:  
 ordering = ['number']  
 verbose\_name = '购入信息'  
 verbose\_name\_plural = '购入信息'

#### 逻辑试图 views.py

def purchase(request):  
  
 '''  
 购入信息存入数据库  
 :param request:   
 :return:   
 '''  
 # 购入信息存入数据库  
 if request.method == "POST":  
 recoder = request.POST.get("recorder", None)  
 name = request.POST.get("name", None)  
 number = request.POST.get("number", None)  
 date = request.POST.get("date", None)  
 type = request.POST.get("type", None)  
 category = request.POST.get("category", None)  
 part = request.POST.get("part", None)  
 location = request.POST.get("location", None)  
  
 tmp = models.Purchase(name=name, number=number, date=date  
 , type=type, category=category, part=part,  
 location=location, recorder=recoder)  
  
 tmp.save()  
 return redirect('/table/')  
 return render(request, 'system/purchase.html')

#### 管理模块注册 admin.py

admin.site.register(models.Purchase)

#### 地址映射 urls.py

urlpatterns = [  
 url(r'^purchase/', views.purchase),  
]

#### 界面 purchase.html

{% load static %}  
  
  
  
  
 <div class="content">  
 <div class="container-fluid">  
 <div class="row">  
 <div class="col-md-12">  
 <div class="card">  
 <div class="header">  
 <h4 class="title">填入购买信息</h4>  
 </div>  
 <div class="content">  
 <form action="/purchase/" method="post">  
 <div class="row">  
 <div class="col-md-5">  
 <div class="form-group">  
 <label>信息记录人 (disabled)</label>  
 <input type="text" class="form-control" name="recoder" disabled value="Regular">  
 </div>  
 </div>  
 <div class="col-md-3">  
 <div class="form-group">  
 <label>购入时间</label>  
 <input type="text" name="date" class="form-control" >  
 </div>  
 </div>  
 <div class="col-md-4">  
 <div class="form-group">  
 <label >设备名称</label>  
 <input type="text" name="name" class="form-control" >  
 </div>  
 </div>  
 </div>  
  
 <div class="row">  
 <div class="col-md-6">  
 <div class="form-group">  
 <label>型号</label>  
 <input type="text" name="type" class="form-control" >  
 </div>  
 </div>  
 <div class="col-md-6">  
 <div class="form-group">  
 <label>种类</label>  
 <input type="text" name="category" class="form-control" >  
 </div>  
 </div>  
 </div>  
<!--  
 <div class="row">  
 <div class="col-md-12">  
 <div class="form-group">  
 <label>Address</label>  
 <input type="text" class="form-control" placeholder="Home Address" value="Bld Mihail Kogalniceanu, nr. 8 Bl 1, Sc 1, Ap 09">  
 </div>  
 </div>  
 </div>  
-->  
 <div class="row">  
 <div class="col-md-4">  
 <div class="form-group">  
 <label>内部编号</label>  
 <input type="text" name="number" class="form-control" >  
 </div>  
 </div>  
 <div class="col-md-4">  
 <div class="form-group">  
 <label>所属部门</label>  
 <input type="text" name="part" class="form-control" >  
 </div>  
 </div>  
 <div class="col-md-4">  
 <div class="form-group">  
 <label>存放位置</label>  
 <input type="text" name="location" class="form-control" value="">  
 </div>  
 </div>  
 </div>  
  
 <div class="row">  
 <div class="col-md-12">  
 <div class="form-group">  
 <label>其他</label>  
 <textarea rows="5" class="form-control" placeholder="填写其他备注。" value="other"></textarea>  
 </div>  
 </div>  
 </div>  
  
 <button type="submit" class="btn btn-info btn-fill pull-right">保存</button>  
 <div class="clearfix"></div>  
 </form>  
 </div>  
 </div>  
 </div>  
  
 </div>  
 </div>  
 </div>  
  
  
  
 </div>

# 四、测试报告

## 1. 概述(Summary)

### 1.1 项目简介(Project Synopsis)

实现公司设备的有效管理、维护记录。

购入新设备，登记设备购入时间、名称、型号、种类、内部编号、所属部门、存放位置等。

设备转借时，记录借用部门、经手人员、用途、拟还日期等。

设备故障时，记录故障情况、维修情况（时间、人员、结果）等。

设备报废时，记录报废原因、批准人员、经手人员、处理方式等。

对设备管理人员和使用人员提供各种设备处理的按部门、设备类型、处理日期等查询和修改等功能。

### 1.2 术语定义(Terms Glossary)

E=equipment,IBM=International Business Machines

### 1.3 参考资料(References)

[1] 《需求规格说明书》

[2] 《设计规格说明书》

## 2. 组件测试(Module Test)

表5-2 模块测试记录

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 路径标识 | 输入 | 期望输出 | 输出内容 | 发现问题 | 测试结果 | 测试时间 | 测试人 |
| 1 | 语句覆盖 | 设备采购、Y、Y | 更新信息 | 更新信息 | 无 | √ | 2019年6月10日 | 成昭 |
| 2 | 判定覆盖 | 设备采购N\Y、Y | 取消采购 更新信息 | 取消采购 更新信息 | 无 | √ | 2019年6月10日 | 成昭 |
| 3 | 条件覆盖 | 设备采购N\Y、Y | 取消采购 更新信息 | 取消采购 更新信息 | 无 | √ | 2019年6月10日 | 成昭 |
| 5 | 条件组合覆盖 | 设备采购、Y、Y | 更新信息 | 更新信息 | 无 | √ | 2019年6月10日 | 成昭 |
| 6 | 路径覆盖 | 设备采购、Y、N、Y | 更新信息 | 更新信息 | 无 | √ | 2019年6月10日 | 成昭 |

## 3．功能测试(Function Test)

### 3.1 系统功能需求（Function Request of Target System）

表5-3 功能需求点列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编 号 | 功 能 名 称 | 使 用 人 | 功 能 描 述 | 输 入 内 容 | 输 出 内 容 |
| 1 | 系统登陆 | 管理人员 | 用户身份验证 | 用户名=ka 密码=123456 登陆 | 显示进入后的页面。 |
| 2 | 设备查询 | 管理人员 | 查询设备情况 | 设备型号=E0000001设备名=计算机 | 设备型号=E0000001 设备名=计算机 规格=IBMR66 单价=5000 购买日期=2009-11-2生产厂家=IBM 购买人=张 数量=1 |
| 3 | 报废记录查询 | 管理人员 | 查询设备报废情况 | 设备型号=E0000002设备名=计算机 | 设备型号=E0000002 设备名=计算机 报废日期=2015-5-5 责任人=张 |
| 4 | 设备申请 | 管理人员 | 查询所要求购买的设备信息 | 设备名=空调 规格=FS45I 数量=5 申请人=刘 | 在数据库申请表中添加记录 |

### 3.2. 功能测试报告(Report for Function Test)

表5-4 功能测试记录

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编号 | 功能名称 | 输入内容 | 期望输出 | 输出内容 | 发现问题 | 测试结果 | 测试时间 | 测试人 |
| 1 | 系统登陆 | 用户名=ka 密码=123456 登陆 | 显示进入后的页面 | 显示进入后的页面 | 无 | √ | 2019年6月10日 | 成昭 |
| 2 | 设备查询 | 设备型号=S0000001设备名=计算机 | 设备型号=E0000001 设备名=计算机 规格=IBMR66 单价=5000 购买日期=2009-11-2生产厂家=IBM 购买人=张 数量=1 | 设备型号=E0000001 设备名=计算机 规格=IBMR66 单价=5000 购买日期=2009-11-2生产厂家=IBM 购买人=张 数量=1 | 无 | √ | 2019年6月10日 | 成昭 |
| 3 | 报废记录查询 | 设备型号=S0000002设备名=计算机 | 设备型号=E0000002 设备名=计算机 报废日期=2015-5-5 责任人=张 | 设备型号=E0000002 设备名=计算机 报废日期=2015-5-5 责任人=张 | 无 | √ | 2019年6月10日 | 成昭 |
| 4 | 设备申请 | 设备名=空调 规格=FS45I 数量=5 申请人=刘 | 在数据库申请表中添加记录 | 在数据库申请表中添加记录 | 无 | √ | 2019年6月10日 | 成昭 |

## 4. 测试结论(Test Verdict)

测试日期：2019年6月10日

测试地点：专业教室

测试环境：PyCharm软件

列出系统的强项：调动数据信息

列出系统的弱项：输入添加信息

列出不符合项的统计结果：无

测试人员签字：成昭