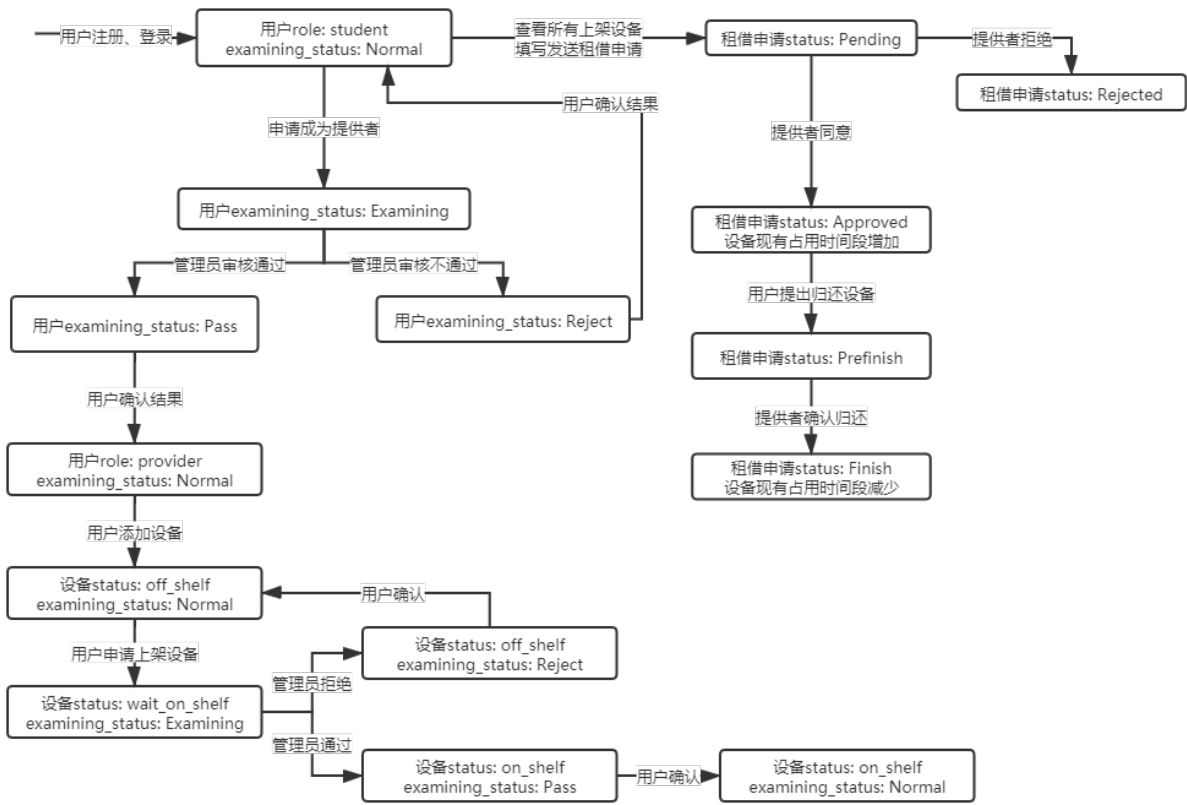


# 设备租赁管理平台：系统设计

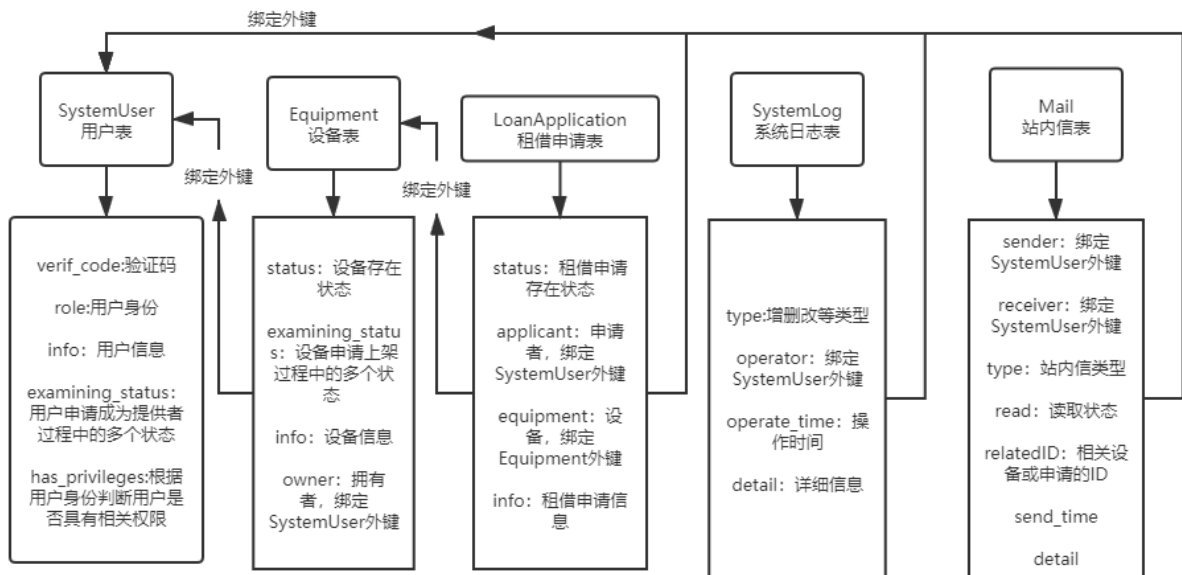
## 系统的设计逻辑

核心功能的基本流程如下：



## 数据库表设计

构建了用户、设备、租借申请、系统日志和站内信5个模型，模型的具体内容如下：



## 技术方案

- 采用前后端分离的策略，
  - 前端采用Vue.js框架及页面布局组件库bootstrap和ant-design-vue。
  - 后端采用Django框架及其自带的数据库sqlite。
  - 后端实现一组 RESTful API；客户端所有的请求都发送到前端服务上，前端将所有以 /apis 开头的 URL 转发到后端。

## 核心设计思路

- 在用户model的设计中添加了role来实现用户角色和权限的区分。
  - 在后端接口通过对role的判断来实现限制不同用户对接口的调用。
  - 在前端通过对role的判断实现了不同用户类型不同页面的展示。
- 在前端实现了根据当前状态对某些操作按钮的隐藏，限制了用户的操作逻辑，加上了异常信息的处理和发出，优化体验。
- 实现了展示用户感兴趣的数据，如在租借申请页面可以看到当前租借状态、剩余时间；管理者可以在平台数据一览直观的看到重要数据统计；用户在前端查询可以根据用户、设备状态进行分类查询。

## 实现的功能

## 后端

按照后端接口分为用户、设备、租借申请、系统日志和站内信五类，

### 用户接口

- **SystemUser数据库的增**

- 用户注册  
添加了邮箱验证的功能。

- **SystemUse数据库的改**

- 用户申请成为设备提供者  
更改了User的examining\_status便于管理员部分筛选、
- 管理员审核用户申请  
考虑到毫无理由的更改用户类型不符合逻辑，因此将管理员对用户类型的更改放置在审核用户申请中。若审核通过，则更改User的role，提升权限。同时将User的examining\_status改为Pass或Reject
- 用户确认审核申请  
将User的examining\_status恢复为Normal

- **SystemUser数据库的查**

- 管理员查询用户信息  
查询所有用户信息，利用examining参数判断是否筛选出申请成为提供者的用户。
- 普通用户查询自己的信息  
非admin身份的普通用户查询自己的信息

- **SystemUser数据库的删**

- 管理员可以删除用户，实现对应租借申请和设备的级联删除

- **其他功能：用户的登录和登出**

- 利用request.session实现set\_cookie

### 设备接口

- **Equipment数据库的增**

- 提供者添加设备  
输入设备的名称及描述即可增加，名称可重复

- **Equipment数据库的删**

- 提供者删除自己的设备，实现对应租借申请的级联删除
- 管理员删除上架的设备，实现对应租借申请的级联删除

- **Equipment数据库的改**

- 提供者修改自己的设备  
可修改设备的name和info，若提交的参数为空则默认不作修改
- 提供者申请上架自己的设备  
更改了Equipment的examining\_status便于管理员的部分筛选
- 管理员审核设备上架的申请  
若审核通过，则更改Equipment的status为已上架状态，对其他用户变为可见。同时将Equipment的examining\_status改为Pass或Reject
- 提供者确认审核结果  
将Equipment的examining\_status恢复为Normal，避免重复通知
- 提供者下架自己已上架的设备  
更改Equipment的status为未上架，对其他用户变为不可见

- **Equipment数据库的查**

- 用户查看当前所有上架设备  
利用status参数进行筛选，便于用户搜索和租借
- 提供者查看己方所有设备  
利用owner参数进行筛选，便于提供者进行管理
- 管理员查看当前所有上架设备  
返回所有上架设备，便于统计和管理

## 租借申请接口

- **LoanApplication数据库的增**

- 用户发送租借请求，预约某台设备未来的一个时间段，并填写理由
- 需要检查要求时间段与此设备已通过的租借申请是否重叠

- **LoanApplication数据库的改**

- 提供者审核租借申请，改动请求的状态为 approved 或 rejected，以及设置审核回复

- 若通过，则需要再次检查申请的时间段是否重叠
- 用户声明归还设备，改动请求的状态为 prefinish
- 提供者确认归还设备，改动请求的状态为 finished

- **LoanApplication数据库的查**

- 提供者查询自己所有设备的租借申请
- 用户查询自己发出的租借申请

## 系统日志接口

理论上只记录对数据库的增删改

- **SystemLog数据库的增**

- 后端直接实现了每个操作时增加SystemLog

- **SystemLog数据库的查询**

- 只在管理员页面的展示了对SystemLog数据库的展示

## 站内信接口

站内信除用于用户交流外，还用于系统通知，将设备归还、确认归还，设备审核上架结果通知，用户租借申请结果通知都放在了站内信中。

- **Mail数据库的增**

- 用户发送站内信

- **Mail数据库的改**

- 用户对站内信的确认已读，此处通过站内信的不同type实现了不同通知类型的确认操作（如确认归还）

- **Mail数据库的查**

- 查询到接受者为自己的所有站内信并展示

- **Mail数据库的删**

- 只有在确认已读后，才能够实现对站内信的删除

## 前端

### 开发框架

- 采用**Vue.js**前端框架
- 使用页面布局组件库**Ant Design**及**Bootstrap**美化UI外观
- 引入**Font Awesome**图标增加页面信息易读性。

## 核心框架：

- **两大页面显示组件：Admin, CommonUser**

- 用户模型中的用户类型有三种

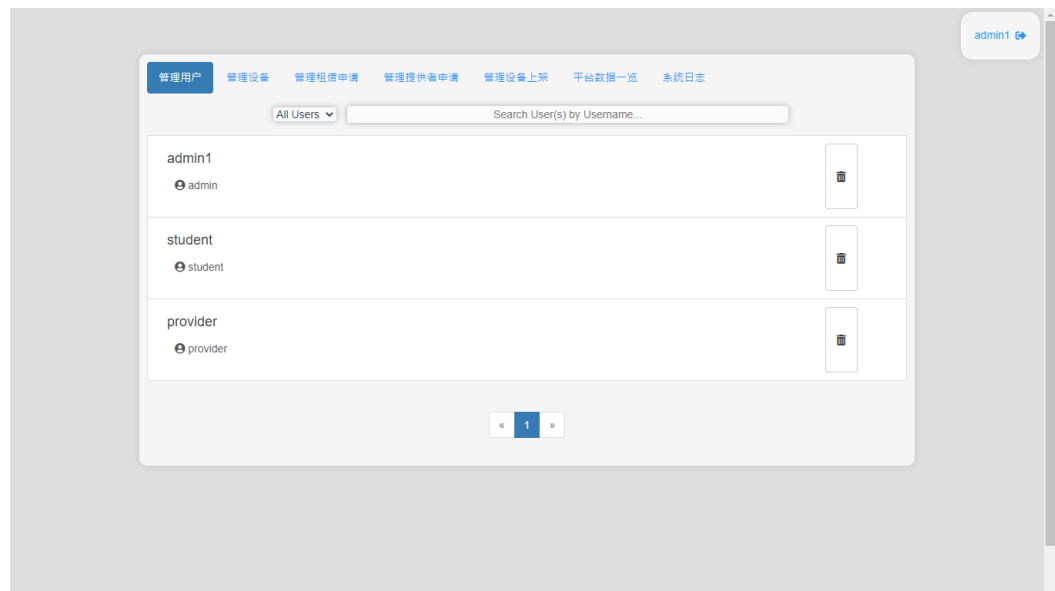
1. admin
2. provider
3. student

而provider的操作只是在student原有的操作上增加以下三项操作：

1. 管理自己的设备
2. 租借申请与借出历史
3. 增加设备

因此设计两大页面：**Admin**和**CommonUser**作为主要列表显示页面

- **Admin: 学院工作助理（平台管理员）**



- **CommonUser:**



CommonUser中显示的栏目会根据用户的类型（student或provider）隐藏、显示部分功能，例如对student隐藏上述三种provider功能，对provider隐藏申请成为设备提供者的页签。

### ● 五个用于单组数据显示的组件：

根据取得的数据列表进行显示渲染，因此我们设计了几个子部件并使用Vue中提供的 v-for 功能迭代渲染：

1. DeviceInfo：用于显示设备相关数据及功能按钮
2. LoanInfo：用于显示租借相关数据及功能按钮
3. UserInfo：用于显示用户相关数据及功能按钮
4. Mail：用于显示站内信发信用户及信件内容
5. Log：用于显示系统日志数据

其中功能按钮包含删除、修改、审核、设备上下架等，以及点击后可以弹出表单的按钮

### ● 两个用于用户注册及登录的页面组件：Login, Logon

#### 实现功能：

- 对于所有后端接口操作提供图形界面。
- 增加数据筛选函数，包含：
  1. 分页
  2. 资料筛选搜寻
  3. 提示框提示正确或错误信息
  4. 部分页签添加小标显示数据数量

## 扩展功能

统整一下我们额外完成了两项扩展功能，包含：

1. 系统操作日志的记录和查询
2. 站内信的系统可以支持用户之间的在线聊天

## 非功能需求

---

本项目由于预期使用环境中的数据量不大（千至万数量级），接口响应速度可以保持相当快。另外，前端只在刷新页面时重新加载所有数据，而一些页面上只在需要时重新请求数据，这样便保证了页面的整体响应速度。

使用的框架 Django 在 TechEmpower Framework Benchmarks 的综合得分中排名 98/104，不够理想，但是本项目避免了单独在内存中记录的数据，因此具有潜在的横向扩展性，理论上可以部署至服务器集群，实现更高的并发。在实际测试中，可以保证在预期数据量下访问速度保持基本不变。

项目附有部署文档与使用说明，前端也包含了一定的提示信息，确保即使界面的打磨不足，也能让用户能够相对顺利地使用。

## 小组成员信息及分工

---

### 后端

余齐齐：用户相关的模型搭建与接口设计  
彭维方：设备相关的模型搭建与接口设计  
吕时清：租借申请相关的模型搭建与接口设计

### 前端

周彦如：设计组件与页面布局、前端数据展示接口  
其余三人：与自己的后端接口相关的页面与逻辑设计

### 文档

余齐齐、彭维方：系统设计文档  
吕时清：部署文档  
周彦如：操作使用说明文档及前端功能