C:\Users\44622\AppData\Local\Temp\ksohtml\wpsF2DD.tmp.png

**本科毕业论文（设计）**

**题 目 基于Django框架的电脑维修管理系统的设计与实现**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**学生姓名 柯林平**

**学 号 2016141462067 年级 2016**

**指导教师 蒋玉明**

**教务处制表**

**年 月 日**

基于Django框架的电脑维修管理系统的设计与实现

专业 计算机科学与技术

学生姓名 柯林平 指导教师 蒋玉明

**[摘要]** 随着科学技术的发展，电脑普及率的提高，互联网对每一个现代人来说都不再陌生。互联网与现代人的生活息息相关。互联网的普及，极大程度的提高了各行各业的效率，促进了产业升级。电脑已经成为人们生活娱乐和办公不可或缺的存在。但对于电脑的维修，仍是专业人员才能解决的问题。传统的客户到店维修模式，有着交流不便，定价麻烦等问题，维修人员和客户不能及时有效的对产生的问题进行沟通。对于维修的电脑和员工，门店也不易进行统计和管理。所以在互联网极度普及的现在，市场上相应的管理系统应运而生。电脑维修管理系统可以很好的解决客户与技术员的交流问题，也可以方便公司对维修人员的管理。本电脑维修系统采用现在比较流行的B/S模式，数据库采用较为成熟的mysql数据库，前端采用vue.js，后端则使用Django，进行前后端分离开发。本系统实现了客户报修，维修单分配，维修单更新跟踪，维修员请假等功能。

**[关键词]** Django；Web；电脑维修；管理系统；

**Design and implementation of computer maintenance management system based on Django framework**

Specialty xxx

Student xxx Adviser xxx

**[Abstract]**：With the development of science and technology, the improvement of computer penetration, the Internet for every modern person is no longer unfamiliar. The Internet is closely related to the lives of modern people. The popularity of the Internet has greatly improved the efficiency of all walks of life and promoted industrial upgrading.

Computers have become an indispensable part of people's life and entertainment and office. But for the maintenance of computers, is still a professional can solve the problem. The traditional customer-to-store maintenance mode, there are communication inconvenience, pricing trouble and other issues, maintenance personnel and customers can not communicate the problems in a timely and effective manner. Stores are also not easy to count and manage for repaired computers and employees. So in the Internet is extremely popular now, the market for the corresponding management system came into being.

Computer maintenance management system can be a good solution to the exchange of customers and technicians, but also to facilitate the company's maintenance personnel management. This computer maintenance system adopts the now popular B/S mode, the database adopts the more mature mysql database, the front end uses vue.js, and the back end uses Django, the front back end is developed separately from the front back end. The system realizes the functions of customer repair, repair order assignment, repair order update tracking, maintenance staff leave and so on.

**[Key words]**：Django；Web；Computer repair；Management system；

**绪论**

**研发背景**

现今电脑越来越普及，但大多数人对电脑只是停留在日常使用上，对于电脑的维修和专业软件的安装并不了解，当出现问题时无法解决。这时需要一个网上报修系统方便的解决客户和维修电脑人员的问题。电脑维修管理系统，使得电脑维修更加方便。

通过电脑维修管理系统，维修人员和报修人员之间可以更好的交流，维修人员可以更加清楚的了解报修人员的具体需要，进行特定的服务，加速维修进度和精度。报修人员也能实时了解自己电脑状况，不至于时刻担心电脑可能发生的各种情况。

基于Django框架设计针的电脑维修管理系统，对于提高电脑维修人员的工作效率，加强客户与电脑维修人员的交流具有较大的实践意义。

**国内外研究现状**

随着互联网技术的发展，互联网在社会生活的各个领域得到了广泛的应用，互联网+的思想深入人心，不仅仅在大学校园内，在各行各业，都有着互联网与对应行业的结合。基于互联网的管理系统，在国内外都有着广泛的应用，例如常见的图书馆管理系统，学生管理系统。

近几年，随着Python这种新的编程语言的快速发展，基于Python的web框架Django也得到了快速的发展，在一些常见的业务需求上，有了成熟的解决方案。Django的前后端数据传输技术，对数据库的连接，对Nginx服务器部署的兼容，都有着成熟稳定的方案。前端的迅猛发展，随之产生的三大前端框架（vue.js、react.js、angular.js），都在原来前端三件套（HTML，CSS，JavaScript）的基础之上，有着长足的进步。这些新兴框架和技术的发展，足以支撑电脑维修管理系统的设计与开发。

随着互联网技术和web技术的发展，互联网+已经出现在各个领域。今后现代社会的发展，离不开互联网。互联网会以任何一种可能的形式，出现在我们的生活中，帮助我们的生活，变得越来越便利，电脑维修管理系统，只是其中一个很小的应用

**系统的使用价值与现实意义**

本电脑维修管理系统，提供了电脑报修，维修单管理，维修人员管理等常用功能，解决了客户与维修员、维修员与公司之间的需求。本系统帮助客户与维修员更好的沟通，也为使用本系统的公司提供了方便有效的员工管理方式。在互联网发达的现在，手机和电脑普及的现在，客户可以随时通过本系统进行报修，查看维修状态，与维修员沟通，大大节省了电脑维修过程中用于交流的时间，提高了电脑维修的效率。本系统还有助于对报修的电脑进行统一的管理，减少了管理人员的工作强度。

本系统主要的现实意义：

1. 提供电脑报修的方便途径；
2. 节省客户与维修人员的交流成本；
3. 节省公司对维修人员和维修电脑的管理成本；

**第一章 Django Web框架技术介绍**

**1.1 系统Web前端技术**

本系统使用的Web前端技术是目前市场上流行的三大前端框架（vue.js、react.js、angular.js）之一。Vue.js是一套用构建用户界面的JavaScript框架，它的核心库只关心视图层，使得前端的开发与后端的业务逻辑分离开来。使用vue.js，使得前端项目可以作为一个独立项目运行，而非依托于后端的服务器。vue.js使用的是MVVM（Model-View-ViewModel）的架构模式，Model即为数据模型，View即为视图。MVVM的架构模式，实现了数据双向绑定，解决了传统前端（JavaScript）中处理繁琐、代码冗余、渲染性能低、加载速度慢的问题。使用vue.js作为前端开发技术，可以加大程度的加速开发效率和提供用户体验。

**1.2 系统Web后端技术**

本系统使用的Web后端技术是Python的后端开发框架Django。Django作为轻量型的框架，具有开箱即用、配置简单、功能齐全的特点。Django框架采用的是MTV（Model-Template-View）设计模式，Model即为模型，Template即为模板，View即为视图。Django的核心组件包括路由设计（URL）、对象关系映射（ORM）、用户友好的模板语言、admin系统等。Django相比其他后端框架，其最明显的优势就是开箱即用。Django自带有开发环境使用的服务器，不需要开发人员在开发时进行服务器的配置。Django还自带对象关系映射框架，也已经事先配置好，开发者只需要使用默认配置即可。通过对象关系映射（ORM）框架，开发人员可以直接调用接口操作数据库，而不需要在代码中写繁复的sql语句，保持了代码的简洁性，也大大提高了开发效率。Django还提供了多种模板语言，可以更方便的进行视图的开发。除此之外，Django还有自带的数据库系统，但此数据库系统性能不强，一般只用于开发阶段。

**1.3 系统数据库技术**

本系统使用的数据库技术是MySQL。MySQL是Oracle旗下产品，是最流行的关系型数据库之一。MySQL作为开源软件，具有体积小、速度快、成本低等优点。MySQL数据库在Web开发中被广泛使用，对各种流行语言（c++、Java、Python、PHP等）和框架的支持也非常成熟。MySQL除了控制台以外，还提供了更加方便的图形客户端，对于新手来说非常友好。其在Web开发中成熟的体系，试错成本较低，能极大提高开发效率。MySQL支持多种存储引擎，用户可以根据需要，选择更加合适的存储引擎。MySQL的事务处理系统，保证了ACID（原子性、一致性、隔离性、持久性）四大特性。MySQL还有着系统的数据备份和数据回滚方案，防止了数据的丢失。MySQL对于千万条记录级别的数据，也可以做到很好的快速处理，具有稳定、快速、可靠的优点。

**第二章 需求分析**

**2.1 系统可行性分析**

系统可行性分析是指在开发系统之前，对当前技术和市场的调查分析。保证当前所具备的条件能够保证系统的开发和使用。系统可行性分析包括技术可行性和经济可行性两个方面。

技术可行性：现在处于互联网时代，各类基于web开发的网站层出不穷，web网站开发技术十分成熟，各类管理系统工具也在各行业得到广泛的使用。近年来，Django框架的发展也越来越完善，前端框架技术也已经成为主流。经过详细的技术分析，现有的框架技术足以支持电脑维修管理系统的开发。

经济可行性：现在是一个互联网时代，每个人基本都拥有个人电脑。但大部分人对于电脑维修仍是一窍不通，所以现在市场上存在着许多的电脑维修门店。我们处在一个互联网+时代，这些电脑维修门店都在使用着各种管理系统。这些管理系统为电脑维修门店减轻了工作负担，提高了工作效率，降低了失误率，所以大部分电脑维修门店愿意购买一套完善的管理系统。因此，电脑维修管理系统，在经济上是可行的。

**2.2 系统总体功能需求分析**

本系统主要实现对维修单和维修人员的管理。电脑维修的过程中，会涉及到客户（报修者）、维修人员（员工）、管理人员（门店老板）三者的交互。本系统要满足这三类人员各自的需求，每类人员拥有不同的权限。不同的用户登录本系统，所要获得的功能也是不同的。因此，要合理的满足这三类用户不同的需求，本系统才具有使用价值和现实意义。总的来说，本系统需要实现的需求有：

1. 要实现客户的电脑报修需求；
2. 要实现客户对于电脑维修状态监测的需求；
3. 要实现维修人员对维修电脑管理的需求；
4. 要实现维修人员与客户交流的需求；
5. 要实现维修人员请假的需求；
6. 要实现管理员对于客户管理的功能；
7. 要实现管理员对维修电脑管理的功能；
8. 要实现管理员对维修员管理的功能；

**2.3 系统具体用例需求分析**

本系统的使用者为电脑使用者，电脑维修员和管理人员，这三类人对于计算机都有基本的使用基础，可以熟练的使用计算机软件。所以本系统会参考常用的计算机软件，设计出符合用户使用习惯的，方便快捷，便于使用的电脑维修管理系统。下面，对不同场景下使用的用例进行用例分析。

1. 用户登录。

电脑维修管理系统的用户分为三类：客户、维修员、管理员。不同用户登录后拥有不同权限，也会进入不同的主页。首先是用户在登录页面进行登录，系统对账号和密码进行匹配，验证账号和密码的正确性，再根据不同的身份进行页面跳转。用户登录用例图如图2-1所示。

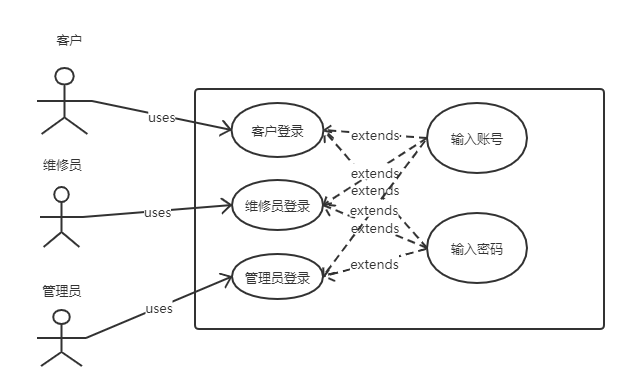


图2-1 系统登录用例图

1. 客户模块

客户管理模块包括客户报修和客户监测维修进度功能，在报修功能下，还包含报修单确认、取消报修单、支付维修费用功能等。客户模块用例图如图2-2所示。

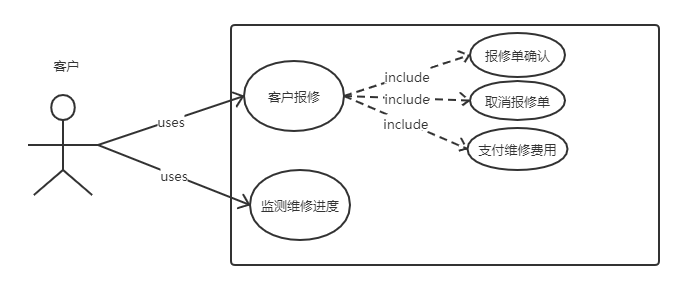


图2-2 客户模块用例图

1. 客户登录后能填写维修单进行报修，维修单提交后，等待维修员接单和完善维修单。客户还可以对完善后的维修单（维修时间和维修费用）进行确认，此后维修过程开始。在维修过程中，客户可以随时在本系统中取消维修单。客户也可在维修完成后，在本系统进行维修费用支付。
2. 客户在电脑维修过程中，可以随时通过本系统监测维修进度，了解电脑维修详情。
3. 维修员模块

维修员模块拥有维修单管理和维修员请假的功能。维修员的维修单管理功能包括完善维修单、申请维修单重新分配、更新维修进度等功能。维修员模块用例图如图2-3所示。

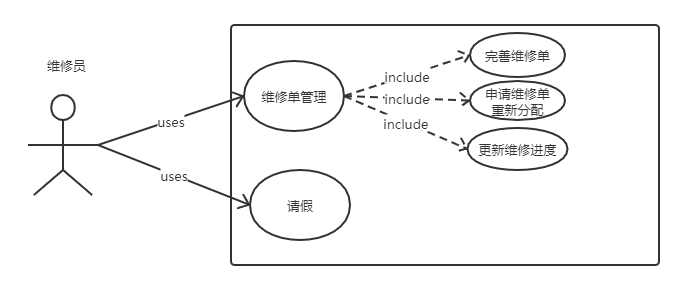


图2-3 维修员模块用例图

1. 维修员登录后，可以查看自己被分配到的维修单，对维修单进行完善（根据电脑存在问题，填写预计维修时间和维修费用），然后等待客户进行确认后进行维修工作。如果维修员认为某个维修单问题自己不能解决，可申请维修单重新分配。在维修过程中，维修员可以随时在本系统中更新维修进度。
2. 维修员可以在本系统中进行请假，等待管理员批准后假期生效。休假的维修员不会再接到维修单。
3. 管理员模块

管理员模块拥有客户信息管理、维修员信息管理、维修单信息管理功能。客户信息管理功能包括对客户信息的增、删、改、查；维修信息管理功能包括对维修员信息的增、删、改、查，还有对维修员请假申请的批准；维修单信息管理功能包括对维修单信息的增、删、改、查，还有对维修单进行重新分配。管理员模块用例图如图2-4所示。

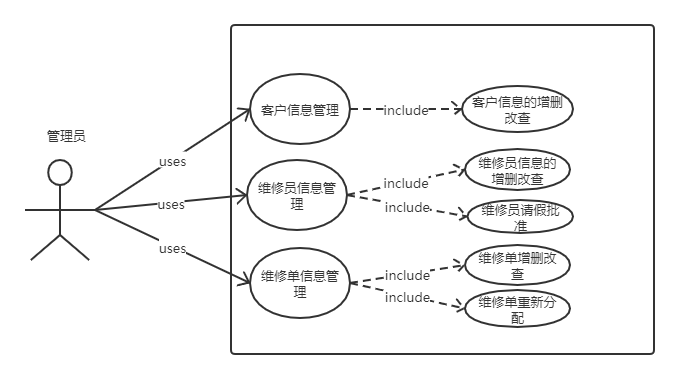


图2-4 管理员模块用例图

1. 管理员可以根据需要对客户信息进行管理。
2. 管理员可以根据需要对客户信息进行管理。除此之外，管理员可以根据维修员当月内的维修时长和创造收入等信息，对维修员的请假申请进行批准或者驳回。
3. 管理员可以根据需要对维修单信息进行管理。除此之外，管理员可以对维修单进行重新分配，把维修单分配给有能力解决的维修员。

**2.4 本章小结**

本章通过对电脑维修管理系统的需求进行分析，从技术可行性和经济可信性方面进行了可信性分析。同时，描述的本系统的大致总体需求，然后通过具体的用例，使用用例图这一UML图对本系统的功能和流程进行了分析。为后面系统的流程设计和详细设计打下基础。

**第三章 系统架构设计**

**3.1 系统软件架构设计**

本系统使用的是前后端分离的开发模式，前端通过统一登录系统，包含客户客户端、维修员客户端、管理员客户端三个。三个客户端发起的请求，中间经由Nginx反向代理服务器转发，最终到达真正的后端服务器，再经由业务层进行业务处理，向数据库中并发写入数据。后端服务器处理请求，从数据库中获取数据返回给前端客户端，由前端自主负责显示逻辑，进行最终的页面显示。本系统的登录由统一系统进行管理，保证了系统的安全性，在进行安全校验时，更加方便快捷，管理也更加方便。系统软件架构如图3-1所示。

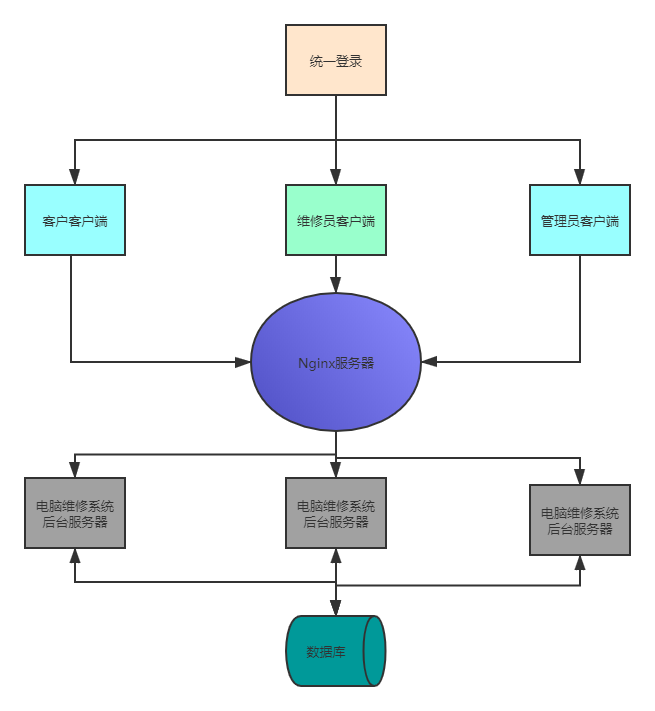


图3-1 系统软件架构图

本系统软件架构的优点：

1. 在登陆系统上进行统一管理。三类用户的注册登录都使用同一系统，有利于保证数据的可靠性和唯一性。同时，还提高了系统的安全性。
2. 使用前后端完全分析的架构，在开发时更加自由。前端的视图逻辑完全由前端负责，在进行开发时只要事先规定好接口，前端可以按照自己的进度和想法进行独立开发，而不用考虑后端的进度、逻辑和兼容性。后端只负责数据和业务逻辑的处理，可以更加专注于专业问题的解决，而不用考虑前端的视图逻辑，减轻后端开发压力。
3. 使用Nginx反向代理服务器大大提高了本系统的并发能力。当大量用户同时使用本系统时，Nginx反向代理服务器会进行负载均衡，将用户请求转发到服务压力最小的服务器上，避免了整个系统因为压力过大而崩溃的情况。
4. 本系统架构还保持了软件的可扩展性。使用前后端分离的开发方式和三个前端分离的架构，当想要进行软件功能扩展或者软件优化时，只需要对特定部分进行优化，而不需要对整个系统进行修改，可以在保持原有功能的基础上进行扩展，大大节省了维护成本。

**3.2 系统技术架构设计**

本系统的技术架构按层次分为四层，由上层到底层分别是视图层、业务层、数据对象关系映射层、数据层。视图层主要使用vue.js作为前端框架，负责视图渲染和ajax请求发起；业务层主要使用Django框架作为RESTful API接口，负责数据的提供，响应ajax请求；数据对象关系映射层使用Django框架自带的ORM框架，实现业务层到数据层的数据对象映射；数据层使用MySQL数据库，进行数据保存和持久化。系统技术架构图如图3-2所示。

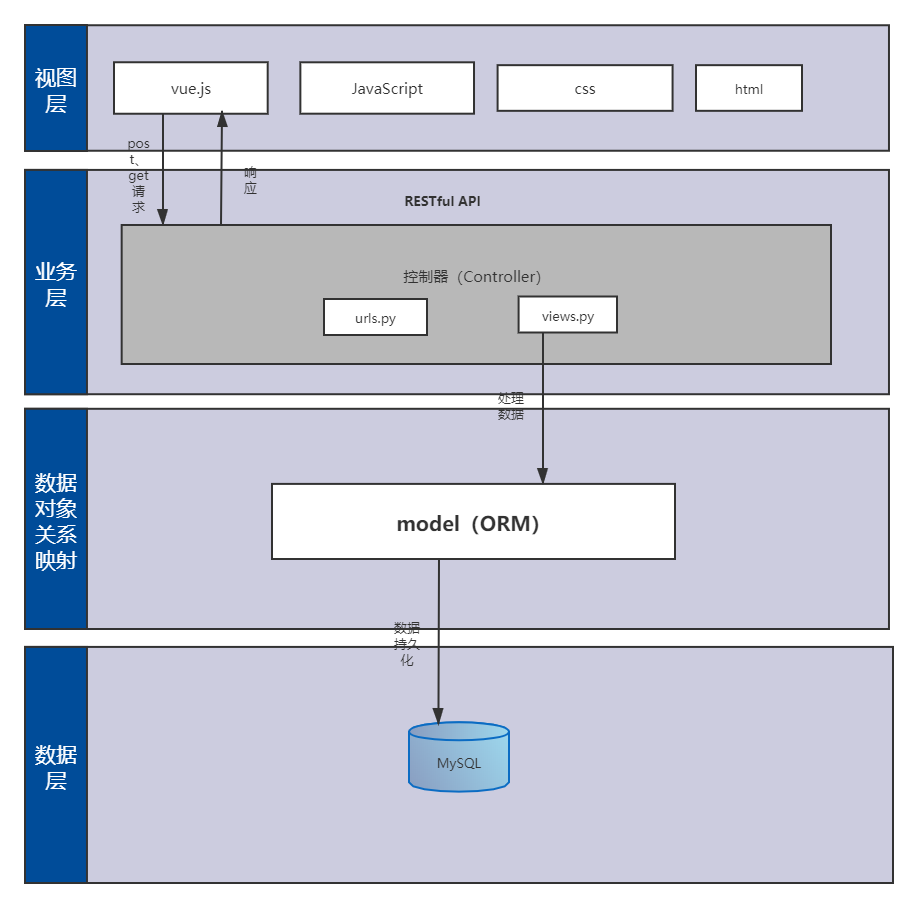


图3-2 系统技术架构图

视图层：视图层就是广义上的前端，使用的技术主要是vue.js框架，vue.js框架的底层其实就是传统的前端三件套（JavaScript、CSS、HTML）。视图层对上主要负责视图的布局、渲染、显示，对下使用ajax协议发送请求和获取数据。

业务层：业务层主要负责业务逻辑的处理，属于后端重要的一部分。本系统的业务层只负责提供RESTful API接口，对上层合法的请求进行响应并返回数据。对下一层，通过数据对象映射，获取所需的数据和保存所需要的数据。

数据对映射层：数据对象映射层（ORM）使用的技术是Django框架自带的ORM框架，负责业务层对象和数据层对象的映射。

数据层：数据层使用MySQL数据库，负责数据的保存、查询、更新、持久化。

**3.3 系统网络架构设计**

本系统的网络架构设计属于常见的web网络架构。在客户端（浏览器）中输入域名，经过DNS域名服务器解析得到真正的服务器IP地址，再经由Nginx服务器进行负载均衡，将到达的请求转发到后端服务器，多个后端服务器再向唯一的数据库服务器进行连接。最终，完成整个系统的网络连接流程。系统网络架构图如图3-3所示。

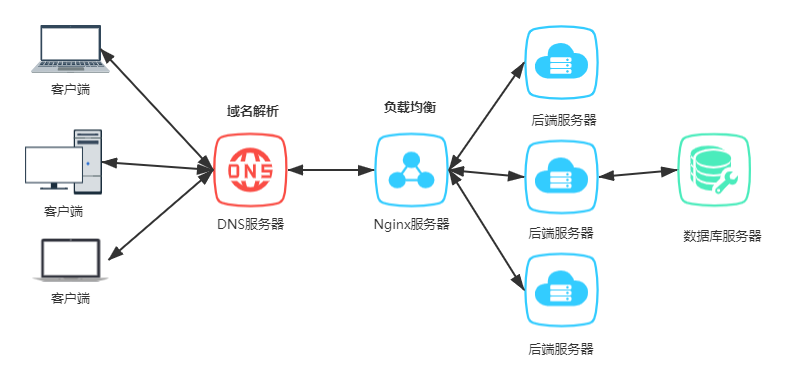


图3-3 系统网络架构图

**3.4 本章小结**

本章主要从软件、技术、网络三方面实现了对电脑维修系统的架构设计。其中，软件架构设计大体决定了系统开发的整体架构；技术架构设计确定了系统开发使用的技术工具和优点；网络架构设计走通了整个系统的流程，保证了系统在网络上的可运行性。本章的架构设计，为系统详细设计打下了基础和大体框架。

**第四章 系统流程设计**

本章主要对电脑维修管理系统的几个主要流程进行设计，包括登录流程、客户报修流程、维修单重新分配流程、维修员请假流程。

**4.1 登录流程设计**

登录流程如图4-1所示。

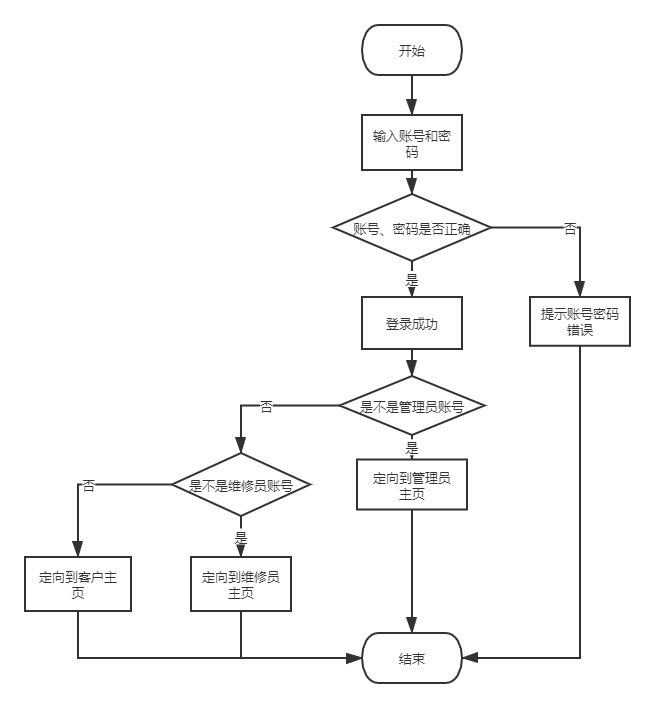


图4-1 登录流程图

用户在系统统一登录界面进行登录。用户首先输入账号和密码，前端将用户输入的账号密码发送给后端，后端在数据库中进行账号密码匹配。如果账号存不存在，则后端返回值为-2，前端提示用户“账号不存在”；如果账号存在但密码错误，则返回值为-1，前端提示用户“密码错误”；如果账号密码匹配则登录成功，在后端记录用户的SessionID，用户登录成功。用户登录成功后进行页面跳转，先判断账号是不是管理员账号，如果是后端返回值为1，前端页面跳转到管理员主页。如果不是管理员账号，接着进行判断，如果是维修员账号，后端返回值为2，前端页面跳转到维修主页。如果既不是管理员也不是维修员，后端返回值为3，前端跳转到客户主页。整个统一登录流程结束。

**4.2 客户报修流程设计**

客户报修流程如图4-2所示。

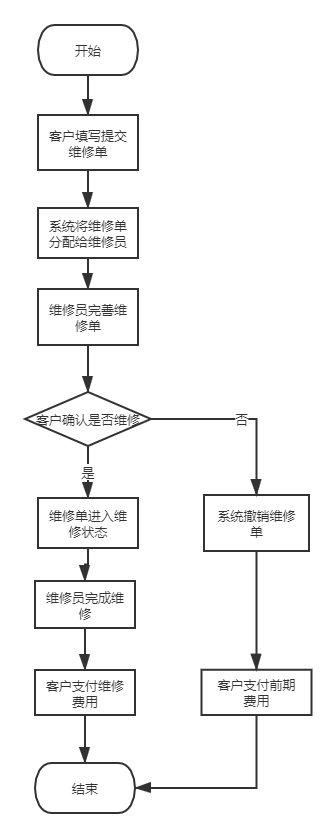


图4-2 客户报修流程图

客户登录后可以在相应页面填写维修单（姓名、手机、邮箱、地址、电脑问题描述），然后提交维修单。客户提交的维修单最开始为未分配状态，系统检测到未分配的维修单然后对维修单进行分配。维修单被分配给对应维修员，此时维修单状态变为已分配。被分配到维修单的维修员对维修单进行完善，填写预计维修时间和预计维修费用，然后提交系统，维修单状态变为已完善。系统对已经完善的维修单的客户发送信息提示（手机短信、电子邮件）。客户收到系统提示以后进入系统查看完善的维修单，进行确认维修或者放弃维修。客户如果放弃维修，则在收到电脑以后支付前期费用，然后系统撤销维修单，客户报修流程结束。客户如果确定维修，则维修单进入维修状态。维修员将电脑维修完成以后，修改维修单状态为维修完成，系统向客户发送信息提示（手机短信、电子邮件）。客户接到提示和收回电脑以后，支付维修费用。客户报修流程结束。

**4.3 维修单重新分配流程设计**

维修单重新分配流程如图4-3所示。

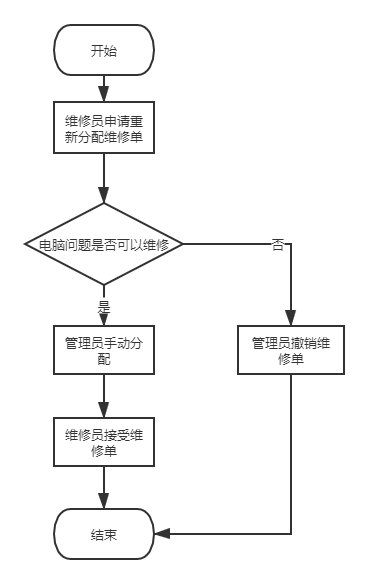


图4-3 维修单重新分配流程图

维修员对分配给自己的维修单进行查看，对维修单问题描述和维修电脑进行检查，判断自己无法维修。维修员填写维修单重新分配申请，写明申请理由（电脑存在问题自己能力不足无法维修或者电脑问题已经不能维修），提交申请给管理员，维修单状态变为重新分配。管理员查看维修单重新分配申请，判断如果是电脑损坏已经无法维修，则寄回客户电脑，撤销维修单，维修单重新分配流程结束。如果电脑损坏问题还可以维修，则手动将维修单重新分配给有能力维修的维修员，维修单状态变为已分配。被分配的维修员查看维修单问题描述和电脑，接受维修单，维修单重新分配流程结束。

**4.4 维修员请假流程设计**

维修员请假流程如图4-4所示。

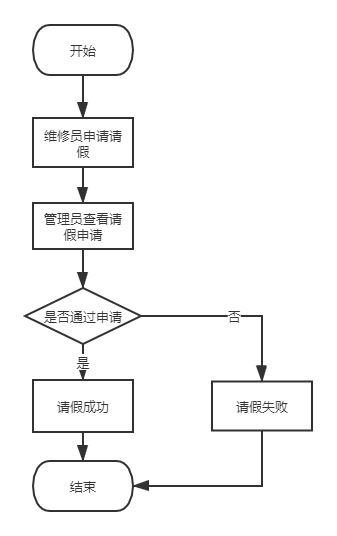


图4-4 维修员请假流程图

维修员填写请假申请表（请假原因、请假时间），提交申请到管理员。管理员查看维修员请假申请，根据维修员本月内维修时长、维修创造收入、维修单数量和请假天数进行综合考量，符合请假条件，管理员则通过该维修员的请假申请，系统发送通知（手机短信和电子邮件）给维修员，维修员的状态变为休假，维修员请假流程结束。如果维修员不符合请假条件，管理员则驳回请假申请，填写驳回申请理由，系统发送通知（手机短信和电子邮件）给维修员，请假失败，请假流程结束。

**4.5 本章小结**

本章进行了系统主要的流程设计，包括了登录流程、客户报修流程、维修单重新分配流程、维修员请假流程。这四个流程，是本电脑维修管理系统的主要功能体现，满足了客户、维修员和管理员的核心需求。本章的流程设计，为系统详细设计建立了大致框架和方向，是系统设计中不可缺少的一步。

**第五章 系统详细设计**

系统模块设计可以明确系统各模块的完整功能和数据信息。本章从客户模块、维修员模块、管理员模块、数据库模块四个模块对电脑维修管理系统进行详细设计，完善前期设计的缺漏和不足，对系统模块进行更加详细的设计。

**5.1 客户模块详细设计**

客户模块主要包括以下功能：

1. 客户登录功能：客户进入系统统一登录页面，输入账号密码进行登录。如果账号不存在，则系统提示账号不存在；如果账号存在但密码不匹配，则系统提示密码错误；如果账号密码正确，系统在后端记录客户端的SessionID，前端跳转到客户主页。
2. 客户报修功能：客户登录后，在主页点击“电脑报修”选项，进入报修页面，报修页面为一张表单，需要填写的信息有姓名、手机号码、电子邮箱、地址、电脑问题描述，填写完成后提交，前端提示提交成功。客户提交的维修单由系统进行维修单分配，将维修单分配给当前不是休假状态且本月维修时长最短的维修员。
3. 客户监测维修单功能：客户登录后，在主页点击“我的维修单管理”，前端显示客户维修单详细信息（维修单状态、预计维修时长、预计维修费用），预计维修时长和预计维修费用两项信息需要维修员完善维修单后才会显示，否则是空白状态。
4. 客户取消维修单功能：客户登录后，在主页点击“我的维修单管理”，进入维修单详细信息页面。点击维修单后面的“放弃维修”，进行维修单取消，此操作后系统将删除对应维修单，并对应支付信息，客户在“维修付款”页面可以查看。
5. 客户支付功能：客户登录后，在主页点击“维修付款”选项，进入支付页面，支付页面显示需要支付的信息（维修单号、维修结果、支付费用）。客户点击“付款”使用支付宝或微信支付进行付款。支付完成以后提示“完成支付”。

**5.2 维修员模块详细设计**

维修员模块主要包括以下功能：

1. 维修单查看功能：维修员登录后，在主页点击“我的维修单”，查看被分配到的维修单列表，每条维修单会显示详细信息（维修单号、维修单状态、电脑问题描述、预计维修时长、预计维修费用），预计维修时长和预计维修费用两项需要维修员完善维修单以后才会有，否则为空白。
2. 维修单完善功能：维修员登录后，在主页点击“我的维修单”，在每条未完善的维修单后面有“完善”选项，点击“完善”，进入维修单完善页面。填写维修单预计维修时长、预计维修费用两项信息，提交系统，系统提示“维修单已完善”，跳转回“我的维修单”页面，更新维修单信息。
3. 申请维修单重新分配：维修员登录后，在主页点击“我的维修单”，在每条维修单后与“申请重新分配”选项，点击后跳转到维修单重新分配申请页面，填写申请理由，提交系统，系统提示“重新分配申请已提交管理员”，跳转回维修员主页。
4. 维修完成功能：维修员登录后，在主页点击“我的维修单”，在对应的维修单后点击“维修完成”，“我的维修单”页面刷新，对应维修状态修改为“已完成”。
5. 申请休假功能：维修员登录后，在主页点击“申请休假”，跳转到申请休假页面。维修员填写请假时间、请假理由，提交系统，系统提示“休假申请已提交管理员”，跳转回维修员主页。

**5.3 管理员模块详细设计**

管理员模块主要包括以下功能：

1. 客户管理：管理员登录以后，在主页点击“客户管理”，进入客户管理页面。客户管理页面显示客户信息列表，点击对应客户，进入客户信息修改页面，可以对客户信息进行修改。
2. 维修单管理：管理员登录以后，在主页点击“维修单管理”，进入维修单管理页面。维修单管理页面有两个选项“维修单基本信息管理”、“维修单重新分配”。点击“维修单基本信息管理”，可以对维修单信息进行修改。点击“维修单重新分配”，可以查看维修员申请重新分配的维修单，点击对应维修单，可以对其进行重新分配。重新分配后，提示“重新分配成功”，“维修单重新分配”页面更新，已被分配的申请不再显示。
3. 维修员管理：管理员登录以后，在主页点击“维修员管理”，进入维修员管理页面。维修员管理页面有两个选项“维修员基本信息管理”、“维修员休假申请审批”、点击“维修员基本信息管理”，可以对维修员基本信息进行修改。点击“维修员休假申请审批”，可以查看维修员休假申请列表，在对应申请后面可以选择通过或者驳回，驳回时需要填写驳回理由，系统提示“休假申请已处理”，页面刷新，以被处理的申请不再显示。

**5.4 数据库详细设计**

数据库在系统中承担着数据存储、更新、查询、备份的重要任务，一个合理的数据组织结构设计可以极大优化数据存储时间和查询时间，减少冗余数据和不必要的代码。

**5.4.1 数据库E-R图设计**

数据库E-R图将数据实体化，使各个数据对象之间的关系显而易见。系统数据E-R如图5-1所示。

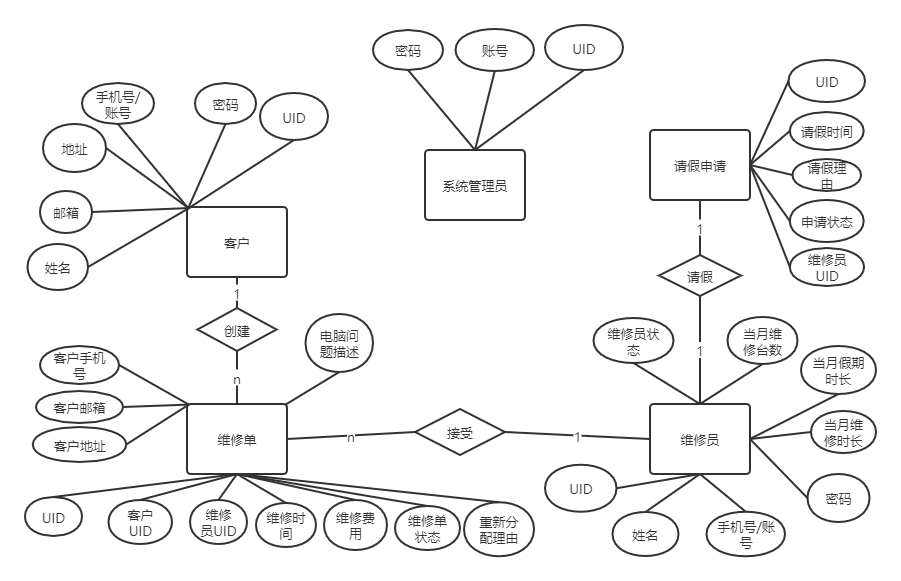


图5-1 数据E-R图

数据包括系统管理员、客户、维修单、维修员、请假申请五个实体。

客户实体包含属性：UID、姓名、账号/手机号、密码、地址、邮箱。其中UID为每个实体都有的属性，是实体的唯一性标识。

维修员实体包含属性：UID、姓名、手机号/账号、密码、当月维修时长、当月假期时长、当月维修台数、维修员状态。其中UID为每个实体都有的属性，是实体的唯一性标识。

维修单实体包括属性：UID、客户UID、维修员UID、维修时长、维修费用、维修单状态、重新分配理由、客户手机、客户地址、客户邮箱、电脑问题描述。其中UID为每个实体都有的属性，是实体的唯一性标识。客户UID和维修员UID作为外键，用于表示实体之间的联系。

请假申请实体包含属性：UID、维修员UID、请假时间、请假理由、申请状态。其中UID为每个实体都有的属性，是实体的唯一性标识。维修员UID作为外键，用于表示实体之间的关系。

管理员实体包含属性：UID、账号、密码。其中UID为每个实体都有的属性，是实体的唯一性标识。

客户实体与维修单实体之间是一对多关系，一个客户可以创建多张维修单。维修员实体与维修单实体之间是一对多关系，一个维修员可以接受多张维修单。维修员实体与请假申请实体之间是一对一关系，一个维修员同时只有一个请假申请。

**5.4.2数据库表设计**

本系统数据库包括统管理员、客户、维修单、维修员、请假申请五张表。本系统使用MySQL数据库。

1. 系统管理员表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 备注 |
| admin\_id | int | 否 | 管理员唯一标识，主键 |
| admin\_account | string | 否 | 管理员账号 |
| admin\_password | string | 否 | 管理员密码 |

表5-1 admin表

1. 客户表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 备注 |
| customer\_id | int | 否 | 客户唯一标识，主键 |
| customer\_account | string | 否 | 客户账号 |
| customer\_password | string | 否 | 客户密码 |
| customer\_address | string | 否 | 客户地址 |
| customer\_email | string | 是 | 客户电子邮箱 |
| customer\_name | string | 否 | 客户姓名 |

表5-2 customer表

customer\_email字段允许为空，为空时联系通过手机短信即账号。

1. 维修单表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 备注 |
| order\_id | int | 否 | 维修单唯一标识，主键 |
| customer\_id | string | 否 | 客户id，外键 |
| staff\_id | string | 否 | 维修员id，外键 |
| order\_time | int | 是 | 维修时长 |
| order\_money | int | 是 | 维修费用 |
| order\_status | int | 否 | 维修单状态 |
| re\_dist\_reason | string | 是 | 申请重新分配维修单理由 |
| computer\_problem | string | 否 | 电脑问题描述 |
| customer\_phone | string | 否 | 客户手机 |
| customer\_email | string | 是 | 客户电子邮箱 |
| customer\_address | string | 否 | 客户地址 |

表5-3 order表

order\_time、order\_money字段在维修单没被维修员完善之前为空。re\_dist\_reason字段在维修员没有申请重新分配维修单之前为空。customer\_email字段允许为空，为空时联系通过手机短信即账号。

1. 维修员表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 备注 |
| staff\_id | int | 否 | 维修员唯一标识，主键 |
| staff\_name | string | 否 | 维修员姓名 |
| staff\_account | string | 否 | 维修员账号 |
| staff\_password | string | 否1 | 维修员密码 |
| staff\_month\_time | int | 否 | 维修员当月维修时长 |
| staff\_month\_holiday | int | 否 | 维修员当月假期 |
| staff\_month\_repair | int | 否 | 维修员当月维修电脑数 |
| staff\_status | int | 否 | 维修员状态 |

1. 请假申请表。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否允许为空 | 备注 |
| apply\_id | int | 否 | 请假申请唯一标识，主键 |
| apply\_time | string | 否 | 请假时间 |
| apply\_reason | string | 否 | 请假理由 |
| apply\_status | int | 否 | 请假申请状态 |
| staff\_id | int | 否 | 维修员id，外键 |

5.5 本章小结

本章对系统的数据进行设计，实现了数据对象的实体E-R图和数据库表单的设计，为实现一个存储数据快速、查询数据快速、更新数据快速的系统打下了坚实的地基。

第六章 系统使用说明

6.1 系统统一登录

系统统一登录页面如图7-1所示。

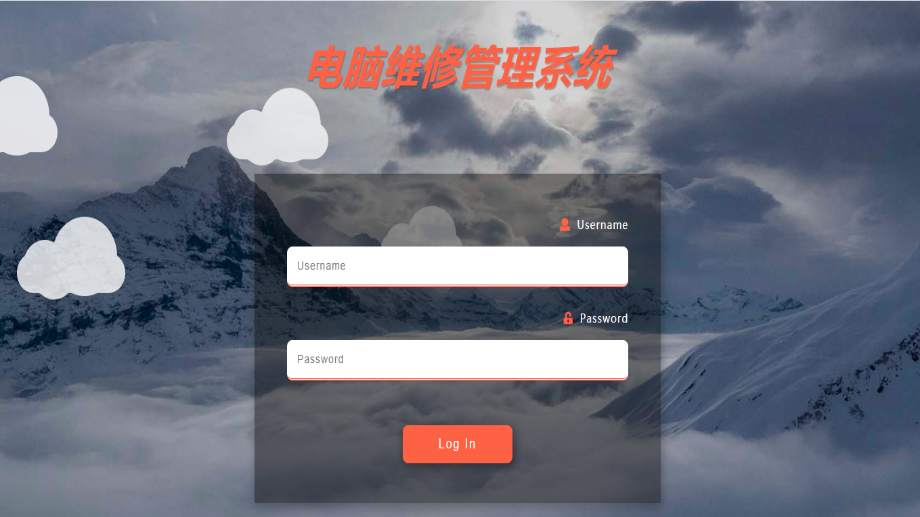


图7-1 系统统一登录页面

用户在浏览器中输入系统对应的域名，进入到统一登录页面，在统一登录页面输入账号和密码。登录成功后，系统根据账号身份的不同，跳转到不同用户的主页。登录失败，系统会提示登录失败原因，用户可以重新输入账号密码进行登录尝试。密码错误三次以后，账号将会被锁定五分钟，五分钟后解锁，可以重新尝试登录。

6.2 客户模块

系统客户主页如图7-2和图7-3所示。



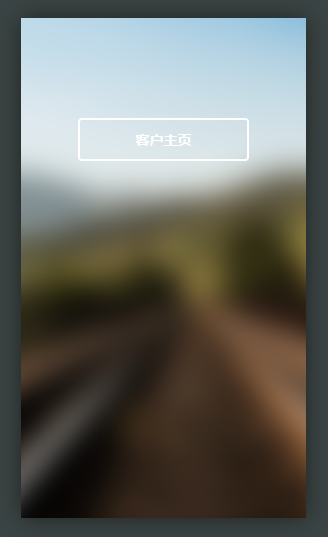


图7-2 客户主页 图7-3 客户主页

客户账号登录后进入客户主页，如图7-2。点击“客户主页”，侧边栏会现“电脑维修”、“我的维修单”、“维修付款”、“退出登录”的功能，如图7-3。客户可以根据自己需要，进入对应页面，完成填报维修单、确认维修单、查看维修单、维修单付款、退出登录等操作。填报维修单的入口在“电脑报修”；确认维修单和查看维修单的入口都在“我的维修单管理”。

6.3 维修员模块

维修员主页如图7-4和图7-5所示。

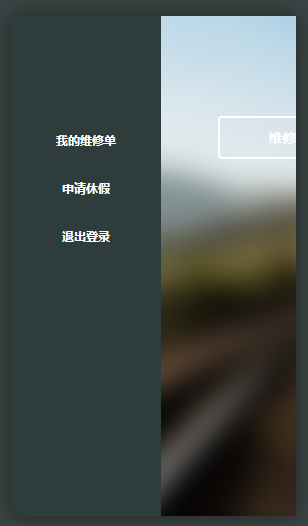




图7-4 维修员主页 图7-5 维修员主页

维修员账号登录后进入维修员主页，如图7-4。点击“维修员主页”，侧边栏会现“我的维修单”、“申请休假”、“退出登录”的功能，如图7-5。维修员可以根据自己需要，进入对应页面，完成维修单完善、申请维修单重新分配、维修单完成、申请休假、退出登录等操作。完成维修单、申请维修单重新分配、维修单完成的入口都在“我的维修单”。申请休假时，维修员需要填写休假时间和请假理由，两者都不能为空，否则请假申请将不能提交。

6.4 管理员模块

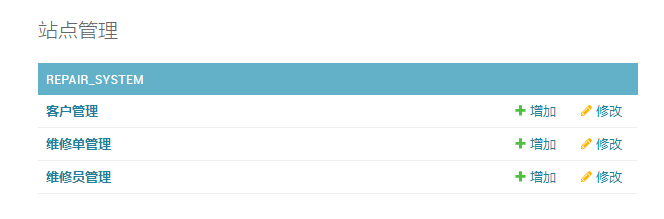
 管理员主页如图7-6所示。

图7-6 管理员主页

管理员账号登录后进入管理员主页，如图7-6所示。管理员主页有“客户管理”、“维修单管理”、“维修员管理”三个选项，点击对应选项，对客户、维修单、维修员进行操作，包括数据的增删改查、维修单的重新分配、维修员请假申请的审批等功能。点击对应选项，可以查看对应的客户、维修单、维修员列表，列表页面具有翻页功能。

6.5 本章小结

本章主要是对系统使用的说明，配合系统效果图，帮助用户更好的理解系统的功能和使用规则。系统的使用说明主要为客户、维修员、管理员三类用户服务，使得系统简易明了，容易上手。

第七章 总结

通过本次电脑维修管理系统的设计与开发，对四年来学习到的知识进行了实践与应用。本次系统的设计与开发，不仅仅是对已有知识的巩固，也是对新知识学习的过程。本次设计与开发，让我从整体上开始设计一个完整的系统，包括需求分析、架构设计、流程设计、详细设计、使用说明等部分。在开发过程中，也更加了解和熟练的使用了Django框架、vue.js框架、MySQL数据库、Nginx服务器等开发工具。最终完成了本系统最开始设计开发的目标。

1. 了解电脑维修管理系统在国内外的研究现状和研发背景。
2. 根据系统的需求分析，设计出系统的大致功能。
3. 根据系统使用的技术设计系统的技术架构。
4. 对系统进行详细设计，按模块对系统的功能进行了详细设计。

系统的设计与开发是一个不小的工程，在这个过程中资料的查询、学习新技能、按部就班的完成计划是一种在学习中进步的过程。

参考文献

[1] 王丹.  [大普通信人事管理系统的设计与实现](https://www.cnki.net/kcms/detail/detail.aspx?filename=1017062822.nh&dbcode=CMFD&dbname=CMFD2018&v=MDk3MTREbVVMektWRjI2R2JPK0hObk9yWkViUElSK0MzODR6aDRYbkQwTFRnMlgyaHN4RnJDVVJMT2VadWRvRnk=)[D]. 电子科技大学 2017

[2] 李云云.  [浅析B/S和C/S体系结构](https://www.cnki.net/kcms/detail/detail.aspx?filename=KXZB201101005&dbcode=CJFQ&dbname=CJFD2011&v=Mjk4MDVMT2VadWRvRnlEbVY3M1BMalhSYkxHNEg5RE1ybzlGWVlSK0MzODR6aDRYbkQwTFRnMlgyaHN4RnJDVVI=)[J]. 科学之友. 2011(01)

[3] 吴卫平,王丽芳,蒋泽军.  [基于ORM的数据持久层框架研究](https://www.cnki.net/kcms/detail/detail.aspx?filename=WXYJ200807050&dbcode=CJFQ&dbname=CJFD2008&v=MjQwNjVMRzRIdG5NcUk5QVpJUitDMzg0emg0WG5EMExUZzJYMmhzeEZyQ1VSTE9lWnVkb0Z5RG1WTDNQTWpYU1o=)[J]. 微电子学与计算机. 2008(07)

[4] 刘红梅.  [基于UML的图书管理系统建模](https://www.cnki.net/kcms/detail/detail.aspx?filename=JYXH200703029&dbcode=CJFQ&dbname=cjfd2007&v=MjkxNjl0Yk1ySTlIYllSK0MzODR6aDRYbkQwTFRnMlgyaHN4RnJDVVJMT2VadWRvRnlEbVZMM1BMelRUWnJHNEg=)[J]. 计算机与现代化. 2007(03)

[5] 吴廷焰,文繁荣.  [基于django的科研管理信息平台的研究与实现](https://www.cnki.net/kcms/detail/detail.aspx?filename=KJXX201135105&dbcode=CJFQ&dbname=CJFD2011&v=MDQyMzVYMmhzeEZyQ1VSTE9lWnVkb0Z5RG1WTHpKTGlmVGRyRzRIOURQcW81RllZUitDMzg0emg0WG5EMExUZzI=)[J]. 科技信息. 2011(35)

[6] 任丹,侯英姿,王方雄,姜来.  [基于Flask和Vue的AIS数据分析系统设计与开发](https://www.cnki.net/kcms/detail/detail.aspx?filename=RJZZ201910026&dbcode=CJFQ&dbname=CJFDTEMP&v=MjMxMDVMRzRIOWpOcjQ5SFlvUitDMzg0emg0WG5EMExUZzJYMmhzeEZyQ1VSTE9lWnVSdkZDdmtWYnZMTnlmUmQ=)[J]. 软件. 2019(10)

[7] 孙娉娉,李新,史广军.  [基于前后端分离的内容管理系统](https://www.cnki.net/kcms/detail/detail.aspx?filename=KYXH201604009&dbcode=CJFQ&dbname=CJFD2016&v=MjUzNDhuRDBMVGcyWDJoc3hGckNVUkxPZVp1UnZGQ3ZrVzcvTkxqVFRackc0SDlmTXE0OUZiWVIrQzM4NHpoNFg=)[J]. 科研信息化技术与应用. 2016(04)

[8] 易剑波.  [基于MVVM模式的WEB前端框架的研究](https://www.cnki.net/kcms/detail/detail.aspx?filename=XXDL201619041&dbcode=CJFQ&dbname=CJFD2016&v=MDA4NzQyWDJoc3hGckNVUkxPZVp1UnZGQ3ZrVzcvTFBUWFBZckc0SDlmTnBvOUJaWVIrQzM4NHpoNFhuRDBMVGc=)[J]. 信息与电脑(理论版). 2016(19)

[9] 王胜,张靖.  [基于Vue.js高速路政管理系统的设计与实现](https://www.cnki.net/kcms/detail/detail.aspx?filename=DNZS201721040&dbcode=CJFQ&dbname=CJFD2017&v=MjgyNTVrVzcvTElTUFJmYkc0SDliT3JvOUJaSVIrQzM4NHpoNFhuRDBMVGcyWDJoc3hGckNVUkxPZVp1UnZGQ3Y=)[J]. 电脑知识与技术. 2017(21)

[10] 宋永鹏.  [Web前端开发技术以及优化研究](https://www.cnki.net/kcms/detail/detail.aspx?filename=SZTJ201808034&dbcode=CJFQ&dbname=CJFD2018&v=MjY0NDNVUkxPZVp1UnZGQ3ZrVzcvS05qZmZaTEc0SDluTXA0OUdZSVIrQzM4NHpoNFhuRDBMVGcyWDJoc3hGckM=)[J]. 数字通信世界. 2018(08)

[11] 陈岩.  [轻量级响应式框架Vue.js应用分析](https://www.cnki.net/kcms/detail/detail.aspx?filename=GLXZ201803075&dbcode=CJFQ&dbname=CJFD2018&v=MjUzOTFNckk5Q1lZUitDMzg0emg0WG5EMExUZzJYMmhzeEZyQ1VSTE9lWnVSdkZDdmtXNy9MSWlIVGRMRzRIOW4=)[J]. 中国管理信息化. 2018(03)

[12] 孙琦.  [浅议MVC设计模式](https://www.cnki.net/kcms/detail/detail.aspx?filename=KJXX200716061&dbcode=CJFQ&dbname=cjfd2007&v=MDkzNjRkckc0SHRiTnFZOURaWVIrQzM4NHpoNFhuRDBMVGcyWDJoc3hGckNVUkxPZVp1UnZGQ3ZsVnIvQUxpZlQ=)[J]. 科技信息(科学教研). 2007(16)

[13] 周吉波.  [基于Django Web框架的车载管理系统的设计与应用](https://www.cnki.net/kcms/detail/detail.aspx?filename=1012513574.nh&dbcode=CMFD&dbname=CMFD2013&v=MTI1NzBSbDdFMElQbmJtd3hBMWVMdVZSTGlmWmVSdUVpanRWRjI2SExhNUhkVExxK0EwWStnT0RYUTl1Ug==)[D]. 浙江工业大学 2012

[14] 李迎辉.  [Python开发Rails框架——Django框架介绍](https://www.cnki.net/kcms/detail/detail.aspx?filename=ITSJ200611046&dbcode=CJFQ&dbname=cjfd2006&v=MTkxNTZZb1IrQzM4NHpoNFhuRDBMVGcyWDJoc3hGckNVUkxPZVp1UnZGQ3ZsVWJ2QUxEbllaTEc0SHRmTnJvOUI=)[J]. 程序员. 2006(11)

[15] 朱成.  [基于Django的安全威胁任务管理系统构建](https://www.cnki.net/kcms/detail/detail.aspx?filename=1012330703.nh&dbcode=CMFD&dbname=CMFD2012&v=MTE1MTFJUG5ibXd4QTFlTHVWUkxpZlplUnVGU3p0VkYyNkhMQzdIdGJNck9BMFkrZ09EWFE5dVJSbDdFMA==)[D]. 复旦大学 2011

[16] 范凯.  [如何选择轻量级Web服务器](https://www.cnki.net/kcms/detail/detail.aspx?filename=ITSJ200710070&dbcode=CJFQ&dbname=cjfd2007&v=MTk4MTVDWklSK0MzODR6aDRYbkQwTFRnMlgyaHN4RnJDVVJMT2VadVJ2RkN2bFVidkFMRG5ZWkxHNEh0Yk5yNDk=)[J]. 程序员. 2007(10)

[17] 刘班.  [基于Django快速开发Web应用](https://www.cnki.net/kcms/detail/detail.aspx?filename=DNZS200907033&dbcode=CJFQ&dbname=CJFD2009&v=MDc5NThzeEZyQ1VSTE9lWnVSdkZDdmxVYnZBSVNQUmZiRzRIdGpNcUk5R1o0UitDMzg0emg0WG5EMExUZzJYMmg=)[J]. 电脑知识与技术. 2009(07)

[18] (美) 杰佛 (Jeff,F.),(美) 鲍尔 (Paul,B.),(美) 陈仲才,著.Django Web开发指南[M]. 机械工业出版社, 2009

[19] 雨虹等,编著.企业级Web开发实战[M]. 电子工业出版社, 2009

[20] (美) 施瓦茨 (Schwartz,B.),(美) 扎伊采夫 (Zaitsev,P.),(美) 特卡琴科 (Tkachenko,V.),(美) 扎沃德尼 (Zawodny,J.D.),著.高性能MySQL[M]. 电子工业出版社, 2010

[21] 贾志娟.  [基于Django框架的软件自动化测试分布式部署系统的研究与实现](https://www.cnki.net/kcms/detail/detail.aspx?filename=1012333669.nh&dbcode=CMFD&dbname=CMFD2012&v=MTY4NDBGckNVUkxPZVp1UnZGQ3ZsVWJ2QVZGMjZITEM3SGRmS3BwRWJQSVIrQzM4NHpoNFhuRDBMVGcyWDJoc3g=)[D]. 北京邮电大学 2012

[22] 卢大勇,陆琪,姜恺.  [基于模板的高性能计算应用封装方法](https://www.cnki.net/kcms/detail/detail.aspx?filename=JSJC201111014&dbcode=CJFQ&dbname=CJFD2011&v=MTc4ODk5RE5ybzlFWUlSK0MzODR6aDRYbkQwTFRnMlgyaHN4RnJDVVJMT2VadVJ2RkN2bFZyL0FMejdCYmJHNEg=)[J]. 计算机工程. 2011(11)

致谢