四 川 轻 化 工 大 学

课 程 设 计 书

科目 数据库系统课程设计

学院 计算机科学与工程学院

专业 软件工程

班级 2020 级 4 班

题目 物业管理系统设计

教师 梁 兴 建

学生 李万余、胡鹏、罗荣

新闻信息系统设计

摘要

目前，我国房地产业相关产业的发展迅速，但是物业管理水平仍然相对滞后，为此我们团队设计了该物业管理系统来取代传统的管理模式，来改变这一落后的现状。

该物业管理系统采用 C++编写，Qt 框架进行界面设计，MySQL 数据库进行数据存储，程序中采用 MySQL C++ API 对数据库进行操作。使用 VSCode 的 MySQL 插件进行数据库的构建，Qt Creator 4.11.1 (Community)进行软件开发。

该物业管理系统分为用户和管理员两个部分。

用户有账户的登录/注册、信息、投诉/建议、服务、缴费五部分。

管理员有账户登录、产权更新、维修管理、缴费管理、发信息五部分。

关键字：物业管理　MySQL　Qt　C++

应用程序分工情况：

完成人	完成内容	工作量比例
李万余	数据库构建、登录部分、软件设计	40%
胡鹏	用户部分	30%
罗荣	管理员部分	30%

目　录

物业管理系统设计

1、项目开发目的与目标

1.1 设计目的

一方面是方便物业管理人员的管理和服务工作，比如资源的统计和管理，业主档案管理等，另一方面是为了给业主提供方便，比如在线查费，在线缴费，在线报修等。

1.2 设计目标

1) 合理的设计数据库

尽量合理地减少数据库的冗余，使重复的数据保持在最小的限度，这样将会少占用存储空间，减少产生混乱影响的危险，还能提高计算机的运行速度。

2) 设计出友好的界面

界面的友好与否是软件优劣的重要方面之一。窗口界面的各个控件布局哟合理、美观。

3) 强大的信息管理力，分析能力

可以对物业管理工作相关数据进行精确的增、删、改、查操作，提高各个模块的数据交换。

2、需求分析

2.1 主要功能简介

目前，我国房地产业欣欣向荣，相关产业的发展同样迅速，但是物业管理水平仍然相对滞后，相当部分的物业管理仍处于单项数据处理或纯手工管理阶段。随着计算机技术的快速发展，电子设备的迅速普及，我们可以设计一种新型的管理系统，取代传统的管理模式，来改变这一落后的现状。

相关资料表明，国外都推出了一系列信息管理系统软件，来加快小区物业管理水平发展，提高工作效率。我国的小区物业管理也在积极的向信息化的方向努力。基于以上现状的考虑，开发一款物业管理系统可以很好的解决小区物业管理的实际需求。现在可以结合计算机技术和数据库技术，完成对小区的主要日常业务的管理，将传统的手工操作转变为信息操作，提高工作效率，相比传统操作，新的管理系统具有以下特点：信息存储大量、检索迅速、准确性高、效率高、信息传递快速、共享方便等。

本系统由业主模块、管理员模块两部分组成。

业主模块包括由账户登录\注册、查看消息、投诉/建议、服务申请、缴费、查看房屋信息六部分组成。

管理员模块包括由查看账户、发布消息、评价管理、维修管理、服务管理五部分组成。

本系统的功能模块如图 2-1 所示

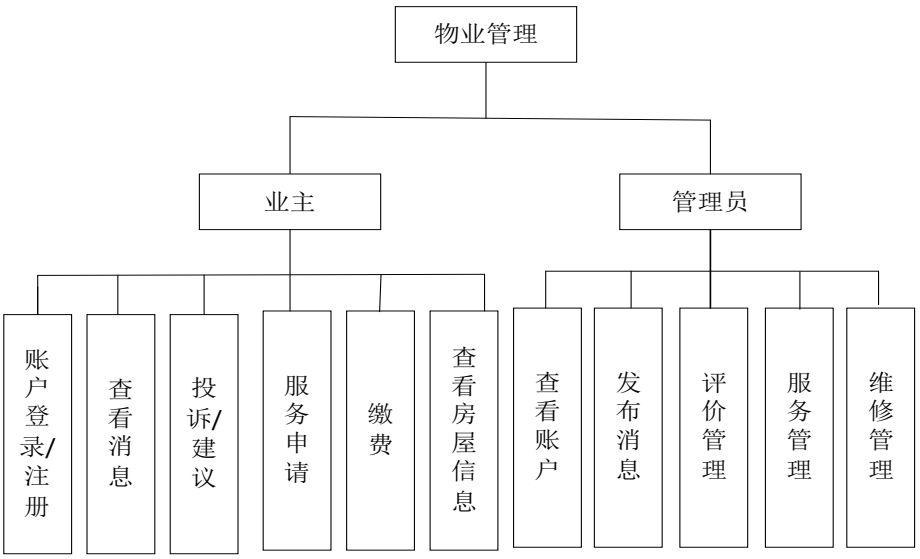


图 2-1 系统功能模块

2.2 功能模块分析

1. 业主功能分

此模块提供给业主使用。

1) 账户登录/注册

点击用户登录界面

用户可注册唯一账号和密码用于登录系统；查看和修改基本信息

2) 查看消息

在此可收到和查看公告（社区信息通知），缴费欠费等提醒，服务处理进度通知；用户收到的信息显示发送时间、对于服务管理还应该显示发布信息的物业管理人员。用户未读的消息会被标记，用户查看后标记消失。

3) 投诉/建议

用户对物业管理人员及其相关事项插入投诉和建议信息

4) 服务申请

服务包含两个子功能：维修和停车信息查询

维修：用户提交维修请求，在管理人员受理后得到反馈，得到受理后任务状态变化由“未受理”→“处理中”→“完成”每次状态更新用户都会收到相关信息。

停车信息查询：用户可以查询自己的停车相关信息，显示车主、停车地点、停车费用、购买时间，备注；

5) 缴费

用户点击查询进入界面可查询相关缴费信息，显示流水号，住户姓名，起始时间，结束时间，金额。点击可以查看各费用详情，显示水费，电费，燃气费，物管费；

点击当月账单显示代缴账单，点击“缴费“可线上缴费

6) 查看房屋信息

先列表显示业主及业主房屋的基本信息。可通过“查看详情”查看具体信息

2. 管理员功能分析

1) 查看账户

账号和密码用于登录系统，具有唯一编号，对应相应物业管理人员

点击管理员登陆登入系统

2) 发布消息

一般发布：选择指定用户可以多选，编辑信息点击发送；

全体发布：向全体住户发送相关信息；

3) 评价管理

可以查看来自住户的投诉和建议，点击回复可以编辑回复内容可以和住户交流

4) 服务管理

分类显示“新任务”,“处理中” ,“完成” 。点击可查看各状态的任务。来自用户的报修点击“已接受” 系统改变用户端状态“处理中” ，管理人员寻找维修人员处理相关问题，并且验收后点击“完成维修”，系统改变用户状态“完成”。

5) 住户管理

住户查询→可以根据户名查询住户信息，点击编辑可以修改住户信息；

添加住户→填入住户姓名，楼号，门牌号，建筑面积，迁入时间，身份证号，联系电话，点击“添加”完成添加住户。

不能添加不存在的楼号门牌号。

6) 缴费管理

分类显示“未交“,“查询”。

点击未显示未交住户显示欠费天数；

查询：输入户号可以查询住户缴费信息，显示每个段时间住户缴费信息，住户名，户号，流水单号，缴费金额，开始时间，结束时间，备注；点击可以查看费用详情；

3、概念结构设计

3.1 数据实体及相关的联系分析

根据需求分析，本系统主要包含实体如下：

1. 用户信息：用户 ID，密码，姓名，手机号，邮箱，激活状态；
2. 投诉/建议：投诉/建议 ID，投诉/建议时间，投诉/建议内容，被投诉/建议的对象；
3. 消息：消息 ID，发布时间，消息内容，消息有效期,消息类型；
4. 费用：费用 ID，开始时间，缴费时间，金额；
5. 费用类型：费用类型，费用类型 ID；
6. 维修任务：维修 ID，维修状态，维修内容，发布时间；
7. 停车位：停车位 ID，停车位位置，购买时间，停车位状态，停车位费用；
8. 管理员信息：管理员 ID，密码，姓名，手机号，邮箱，激活状态；
9. 房屋：房屋 ID，房屋位置，房屋状态；

本系统实体之间的联系如下：

1. 一个用户可以发送多条投诉/建议，一条投诉/建议只能由一个用户发出，即 1：n 关系；
2. 一个管理员可以响应多条投诉/建议，一条投诉/建议只能被一个管理员响应，即 1：n 关系；
3. 一个管理员可以发布多条消息，一条消息只能由一个管理员发布，即 1：n 关系；
4. 一个房屋可以产生多项费用，一项费用只能由一个房屋用户产生，即 1：n 关系；
5. 一个用户可以缴纳多项费用，一项费用只能由一个用户缴纳，即 1：n 关系；
6. 一个费用类型有多条费用，一条费用只属于一个费用类型，即 1：n 关系；
7. 一个用户可以发布多条维修任务，一个维修任务只能被一个用户发布，即 1：n 关系；
8. 一个管理员可以处理多条维修任务，一个维修任务可以被管理员多次处理，即 n：m 关系；
9. 一间房屋可以购买多个停车位，一个停车位只能被一个用户拥有，即 1：n 关系；
10. 一个用户可以拥有多个房屋，一个房屋可以属于多个用户，即 n:m 关系

11. 一个管理员可以更新多个房屋信息，一个房屋信息可以被一个管理员多次更新，即 n：m 关系；

12. 一个上级管理员可以有多个子管理员，一个下级管理员只能有一个上级管理员，即 1：n 关系。

3.2 E-R 模型设计

本系统采用 E-R 模型进行概念结构设计，用户实体图如图 3-1 所示，管理员实体图如图 3-2 所示，房屋实体图如图 3-3 所示，投诉/建议实体图如图 3-4 所示，消息实体图如图 3-5 所示，费用实体图如图 3-6 所示，停车位实体图如图 3-7 所示，维修任务实体图如图 3-8 所示合并后的 E-R 图如图 3-9 所示。

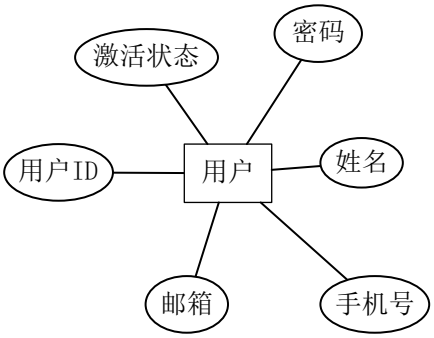


图 3-1　用户实体图

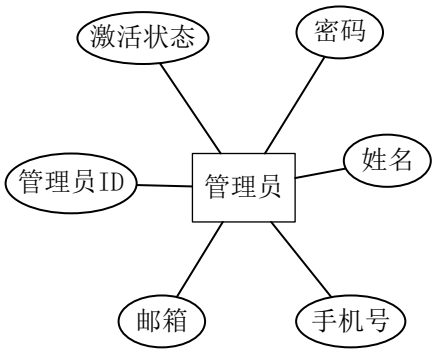


图 3-2　管理员实体图

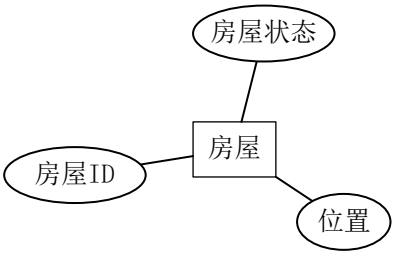


图 3-3　房屋实体图

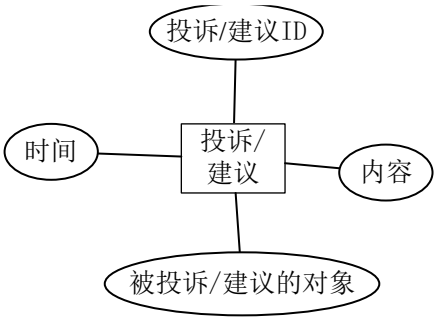


图 3-4　投诉/建议实体图

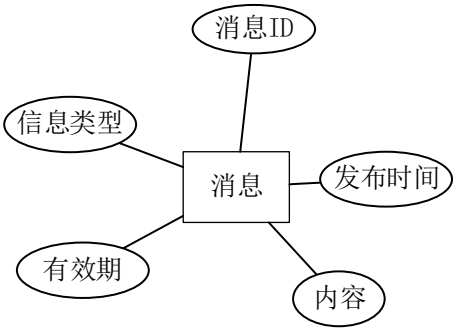


图 3-5　消息实体图

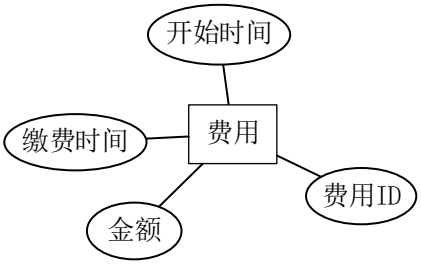


图 3-6　费用实体图

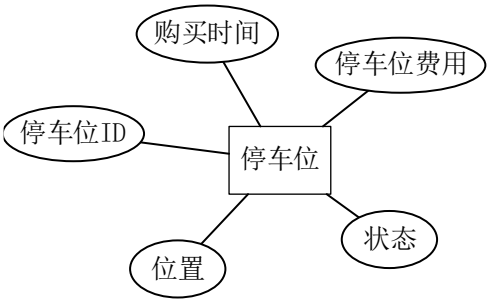


图 3-7　停车位实体图

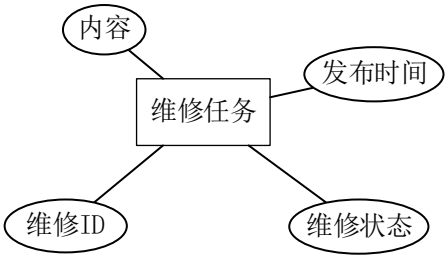


图 3-8　维修任务实体图

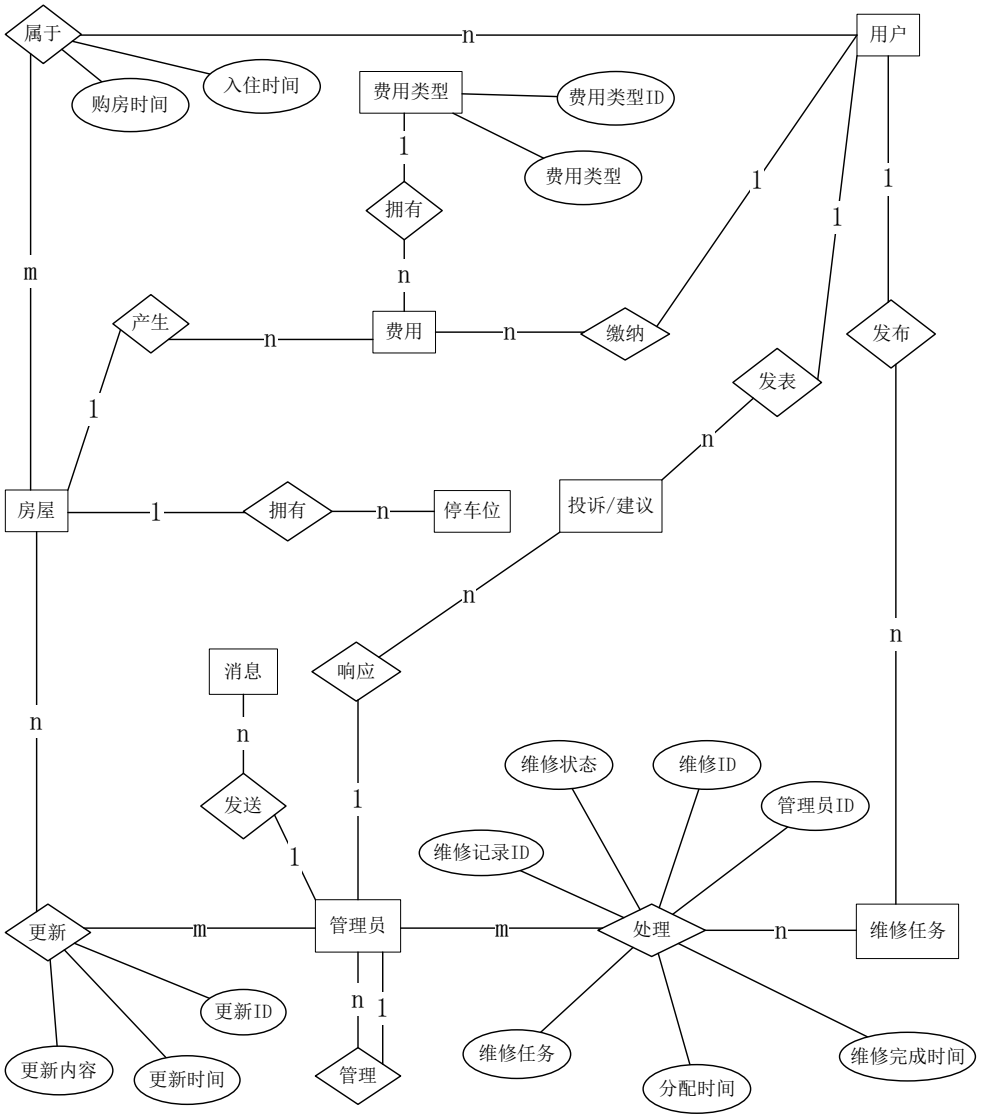


图 3-9　系统总体 E-R 图

4、逻辑结构设计

4.1 关系模式建立

根据需求分析，本系统主要包含实体如下：

概念模型已经确定了实体以及属性联系等，逻辑结构设计主要是针对 E-R 模型转化为关系模式，根据转化规则，逻辑模型设计结果如下，其中划线的属性为主码。

1. 用户（用户 ID，姓名，密码，激活状态，手机号，邮箱）
2. 管理员（管理员 ID，姓名，密码，激活状态，手机号，邮箱，父管理员 ID）
3. 房屋（房屋 ID、位置、房屋状态、管理员 ID）
4. 投诉/建议（投诉/建议 ID、被投诉/建议的对象、时间、内容、管理员 ID, 用户 ID）
5. 消息（消息 ID、内容、时间、有效期、管理员 ID、消息类型，消息标题）
6. 费用（费用 ID、开始时间、缴费时间、金额、房屋 ID、用户 ID、费用类型 ID、费用是否为当前用户）
7. 费用类型（费用类型 ID、费用类型）
8. 维修任务（维修 ID、维修状态、内容、用户 ID、发布时间）
9. 停车位（停车位 ID、停车位位置、购买时间、停车位状态、房屋 ID、停车位费用）
10. 用户_房屋关联（用户 ID、房屋 ID、购房时间、入住时间）
11. 房屋_管理员关联（更新 ID、更新时间、更新内容、房屋 ID、管理员 ID）
12. 管理员_维修任务关联(维修记录 ID、维修状态、维修任务分配时间、维修完成时间、维修 ID、管理员 ID)

4.2 数据关系表结构

通过关系的完整性分析，本系统可以分为：用户信息表、管理员信息表、房屋信息表、投诉/建议信息表、消息信息表、费用信息表、维修任务信息表、停车位信息表、用户_房屋关联表、用户_停车位关联表、用户_费用关联表。

用户信息表(UserInfo)：用于记录用户的信息，手机号和邮箱必须一项非空且唯一，激活状态默认为未激活，主要数据结构如表 4-1 所示：

表 3-1 用户信息表(UserInfo)表结构

字段名	类型	长度	约束条件	备　注
UserID	int	11	自增，主键	用户 ID
UserName	varchar	12	非空	姓名
UserPassWord	varchar	128	非空	密码
IsActive	tinyint	1	非空，默认 0	是否激活（0：未激活，1：已激活）

UserPhone	varchar	11	唯一	手机号
UserEmail	varchar	50	唯一	邮箱

管理员信息表(AdminInfo)：用于记录管理员的信息，手机号和邮箱必须一项非空且唯一，激活状态默认为未激活，主要数据结构如表 3-2 所示：

表 3-2 管理员信息表(AdminInfo)表结构

字段名	类型	长度	约束条件	备　注
AdminID	int	11	自增，主键	管理员 ID
AdminName	varchar	12	非空	姓名
AdminPassWord	varchar	128	非空	密码
IsActive	tinyint	1	非空，默认 0	是否激活（0：未激活，1：已激活）
AdminPhone	varchar	11	唯一	手机号
AdminEmail	varchar	50	唯一	邮箱
PAdminID	int	11	默认 NULL	父管理员 ID

房屋信息表(HouseInfo)：用于记录房屋的基本信息，房屋状态有无用户和有用户两种状态，默认为无人状态。主要数据结构如表 3-3 所示：

表 3-3 房屋信息表(HouseInfo)表结构

字段名	类型	长度	约束条件	备　注
HouseID	int	11	自增，主键	房屋 ID
Location	varchar	20	非空	位置
HouseState	tinyint	1	非空，默认 0	房屋状态（0：无用户，1：有用户）
House_AdminID	int	11	非空，外键	管理员 ID

投诉/建议信息表(Complaint_and_AdviceInfo)：用于记录用户对小区各类服务的投诉/建议信息。主要数据结构如表 3-4 所示：

表 3-4 投诉/建议信息表(Complaint_and_AdviceInfo)表结构

字段名	类型	长度	约束条件	备　注
C_AID	int	11	自增，主键	投诉/建议 ID
C_ATime	Timestamp		非空，默认 CURRENT_TIMESTAMP	时间

C_AObject	varchar	20		对被投诉/建议的对象的描述
C_AContent	Text		非空	内容
C_A_UserID	int	11	非空，外键	用户 ID
C_A_AdminID	int	11	外键	管理员 ID

消息信息表(NewsInfo)：用于记录管理员发送的公告和信息，类型分为三类：单用户，多用户，全体用户。主要数据结构如表 3-5 所示：

表 3-5 消息信息表(NewsInfo)表结构

字段名	类型	长度	约束条件	备　注
NewsID	int	11	自增，主键	消息 ID
NewsTime	Timestamp		非空，默认 CURRENT_TIMESTAMP	发送时间
NewsContent	Text		非空	内容
News_AdminID	int	11	非空，外键	管理员 ID
NewsType	tinyint	1	非空，默认 0	消息类型（0：全体用户，1：多用户，2：单用户）
NewsEndTime	Timestamp			到期时间

费用类型信息表(CosttypeInfo)：用于记录各种费用类型信息。主要数据结构如表 3-6 所示：

表 3-6 费用类型信息表(CosttypeInfo)表结构

字段名	类型	长度	约束条件	备　注
CosttypeID	int	11	自增，主键	费用类型 ID
CostType	varchar	20	非空	费用类型

费用信息表(CostInfo)：用于记录各种费用信息。主要数据结构如表 3-7 所示：

表 3-7 费用信息表(CostInfo)表结构

字段名	类型	长度	约束条件	备　注
CostID	int	11	自增，主键	费用 ID
CostStartTime	Timestamp		默认 CURRENT_TIMESTAMP	开始时间
PayTime	Timestamp			缴费时间
Money	double	8	非空	金额

HouseID	int	11	非空，外键	房屋 ID
UserID	int	11	非空，外键	用户 ID
CosttypeID	int	11	非空，外键	费用类型 ID

维修任务信息表(TaskInfo)：用于记录维修信息，维修状态分为（未开始，已开始，已完成）三种，默认为未开始。主要数据结构如表 3-8 所示：

表 3-8 维修任务信息表(TaskInfo)表结构

字段名	类型	长度	约束条件	备　注
TaskID	int	11	自增，主键	维修 ID
TaskState	tinyint	1	非空，默认 0	维修状态（0：未开始，1：已开始，2：已完成）
TaskContent	Text		非空	维修内容
Task_UserID	int	11	非空，外键	用户 ID

停车位信息表(PCarInfo) ：用于记录停车位信息，状态分为三种（已购买 ，未购买）默认为空闲，只有当状态为已预定时，用户 ID 为非空。主要数据结构如表 3-9 所示：

表 3-9 停车位信息表(PCarInfo)表结构

字段名	类型	长度	约束条件	备　注
PCarID	int	11	自增，主键	停车 ID
PCarState	tinyint	1	非空，默认 0	停车位状态（1：已购买 0：未购买）
PCarLocation	varchar	20	非空	位置
BuyTime	Timestamp			购买时间
HouseID	int	11	外键	房屋 ID

用户_房屋关联表(User_HouseRelation)：用于记录房屋的归属用户，购买信息。主要数据结构如表 3-10 所示：

表 3-10 用户_房屋关联表(User_HouseRelation)表结构

字段名	类型	长度	约束条件	备　注
UH_Relation_UserID	int	11	外键，主键	用户 ID
UH_Relation_HouseID	int	11	外键，主键	房屋 ID

GetHouseTime	Timestamp		非空	购房时间
MoveTime	Timestamp		非空	入住时间

房屋_管理员关联(UpdataRelation)：用于记录管理员更新房屋信息（一个管理员可能多次更新同一房屋信息）。主要数据结构如表 3-11 所示：

表 3-11 房屋_管理员关联(UpdataRelation)表结构

字段名	类型	长度	约束条件	备　注
UpdataID	int	11	自增，主键	更新 ID
UpdataContent	Text		非空	更新内容
UpdataTime	Timestamp		非空，默认 CURRENT_TIMESTAMP	更新时间
Updata_HouseID	int	11	非空，外键	房屋 ID
Updata_AdminID	int	11	非空，外键	管理员 ID

管理员_维修任务关联(Admin_TaskRelation)：用于记录管理员分配维修任务信息，维修状态分为正在维修，和维修完成，维修失败三种，默认正在维修，当维修完成时，维修完成时间为非空。主要数据结构如表 3-12 所示：

表 3-12 管理员_维修任务关联(Admin_TaskRelation)表结构

字段名	类型	长度	约束条件	备　注
Admin_TaskID	int	11	自增，主键	维修信息 ID
ATaskState	tinyint	1	非空，默认 0	维修状态（0：正在维修，1：维修完成，2：维修失败）
TaskStartTime	Timestamp		非空，默认 CURRENT_TIMESTAMP	维修任务分配时间
TaskEndTime	Timestamp			维修完成时间
Admin_Task_TaskID	int	11	非空，外键	维修 ID
Admin_Task_AdminID	int	11	非空，外键	管理员 ID

5、应用系统设计与实现

本程序只有在接入数据库时可用，否则拒绝进入。如图 5-1 所示

连接数据库操作

```
void MainWindow::InitMySQL(MYSQL* MySQL_Data)
{
    //初始化 mysql
    mysql_init(MySQL_Data);
    //设置字符集
    mysql_options(MySQL_Data, MYSQL_SET_CHARSET_NAME, "utf8");
    //连接数据库
    if (!mysql_real_connect(MySQL_Data, "192.168.43.108", "luorong", "123456",
"property_management", 3306, NULL, 0))
    {//失败将弹窗,关闭后退出
        QString text = "数据库连接失败:";
        text += + mysql_error(MySQL_Data);
        QMessageBox::critical(this, "Error", text);
        QTimer *CloseTimer = new QTimer();
        CloseTimer->start(0);
        connect(CloseTimer, &QTimer::timeout, this, [=](){this->close(); qApp->quit();});
    }
}
```

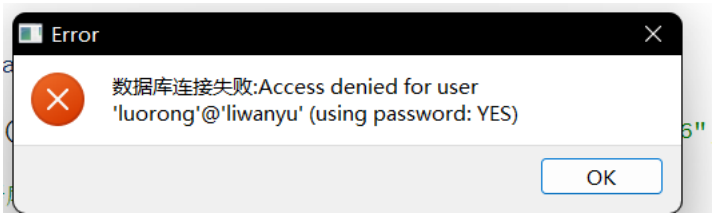


图 5-1 数据库连接失败

执行 SQL 语句：

```
    if (mysql_query(&this->MySQL_Data, MySQLSentence.c_str()))//执行 SQL 语句
    {
        QString Error = "error:";
        Error += mysql_error(&this->MySQL_Data);
        this->ui->ShowLabel->setText(Error);
        return false;
    }
```

获取结果集：

```
    if (!(res = mysql_store_result(&this->MySQL_Data)))                    //获取结果集
    {
        QString Error = "error:";
```

```
        Error += mysql_error(&this->MySQL_Data);
        this->ui->ShowLabel->setText(Error);
        return false;
    }
```

获取结果集行数：

```
    mysql_affected_rows(&this->MySQL_Data)
```

获取一行数据：

```
    MYSQL_ROW column;                    //一行数据
    column = mysql_fetch_row(res);
```

5.1 账户登录与注册

图 5-2 用户登录界面

图 5-3 管理员登录界面

（1） 用户登录界面，该页面为整个软件的入口，提供登录和注册以及忘记密码功能。如图 5-2 所示

（2） 管理员登录界面，该页面为整个软件的入口，只提供登录。如图 5-3 所示登录后进入主页显示管理员/用户基本信息（姓名，手机号，邮箱）。

1、 登录功能实现

首先通过正则表达式判断输入的手机号或邮箱是否合法，再查询数据库验证密码是否正确，将用户/管理员信息存入类‘MainWindow’的属性‘User’或‘Admin’中。

验证正则表达式：

电话号码正则表达式

```
const static QRegExp PhoneReg("^[1][3,4,5,7,8,9][0-9]{9}$");
```

邮箱正则表达式

```
const static QRegExp EmailReg("[a-zA-Z0-9-_]+@[a-zA-Z0-9-_]+\\.[a-zA-Z]+");
```

用户账户查询

```
if (IsPhone)
    MySQLSentence = "SELECT * FROM userinfo LEFT OUTER JOIN user_houserelation ON userid = uh_relation_userid WHERE userphone = '"+ UserName.toStdString() + "';";
```

else

MySQLSentence = "SELECT * FROM userinfo LEFT OUTER JOIN user_houserelation ON userid = uh_relation_userid WHERE useremail = '" + UserName.toStdString() + "';";

管理员账户查询

if (IsPhone)

MySQLSentence = "SELECT * FROM admininfo WHERE adminphone = '" + UserName.toStdString() + "';";

else

MySQLSentence = "SELECT * FROM admininfo WHERE adminemail = '" + UserName.toStdString() + "';";

当查询结果行数为 0 时，说明用户不存在数据库中

通过 strcmp(column[3], "0") 判断账户是否激活

通过图 5-10 验证密码是否正确

QString PassWord = this->ui->PassWord_LineEdit->text();

if (PassWord != QString(column[2])) //判断密码是否正确

{

this->ui->SignInShow_Label->setText("密码错误");

mysql_free_result(res);

return false;

}注册功能实现

该功能只有用户拥有，满足用户注册账户需求。如图 5-11 所示

图 5-4 用户注册

密码需包含小写字符、大写字符、数字，不小于 8 位。通过正则表达式验证

密码正则表达式

const static QRegExp PassWordReg("^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[$@$!%*?&])[A-Za-z\\d$@$!%*?&]{8,}");

通过下面查询语句查询结果的行数判断输入的手机号/邮箱是否被注册过

MySQLSentence = "SELECT * FROM userinfo WHERE useremail = '" + Email.toStdString() + "' OR userphone = '" + PhoneNumber.toStdString() + "';";

最后将注册信息存入数据库

```
bool MainWindow::Register(std::string Name, std::string Phone, std::string Email, std::string PassWord)
{
    std::string MySQLSentence;          //MySQL 语句
    MySQLSentence = "INSERT INTO userinfo(username, userpassword,userphone, useremail, isactive) VALUES ('" + Name + "', '" + PassWord + "', '" + Phone + "', '" + Email + "', 1);";
    if (mysql_query(&this->MySQL_Data, MySQLSentence.c_str()))
    {
        QString Error = "error:";
        Error += mysql_error(&this->MySQL_Data);
        this->ui->ShowLabel->setText(Error);
        return false;
    }
    return true;
}
```

5.2 用户消息

用户登录后在左侧菜单栏选择消息页面，本页面为用户查看消息，为用户以列表的方式展示所有消息，也可以查看单条消息详情。如图 5-15 和 5-16 所示。

图 5-5 消息列表

图 5-6 单条消息详情

1、 以列表的方式展示所有消息

查询用户的所有消息

MySQLSentence = "SELECT newsid,newstitle,newstime,adminname FROM newsinfo,admininfo WHERE news_adminid = adminid and newstype = 0 UNION ALL SELECT newsid,newstitle,newstime,adminname FROM admininfo,newsinfo,user_newsrelation WHERE news_adminid = adminid and un_newsid = newsid and un_userid = " + this->User.UserID + " ORDER BY newstime DESC;";

将查询结果显示在 QTableWidget 中

row = mysql_affected_rows(&this->MySQL_Data);

```
this->ui->NewsList_TableWidget->setRowCount(row);
int j=0;
while((column = mysql_fetch_row(res)))
{
    this->ui->NewsList_TableWidget->setItem(j,0,new
QTableWidgetItem(QString("%1").arg(QString::fromStdString(std::string(column[0
])).toInt(), 10, 10, QLatin1Char('0'))));
    for(int i=1;i<4;i++)
    {
        QString tmp(column[i]);
        tmp += "    ";
        this->ui->NewsList_TableWidget->setItem(j,i,new
QTableWidgetItem(tmp));
    }
    j++;
}
```

2、 单条消息详情

查询单条消息

```
MySQLSentence ="SELECT newstitle,newscontent,newstime,adminname FROM
newsinfo,admininfo WHERE news_adminid = adminid and newsid = " + tmp + ";";
```

显示单条消息

```
column = mysql_fetch_row(res);
this->ui->Newstitle->setText(column[0]);
this->ui->Newscontent->setText(column[1]);
this->ui->Newstime->setText(column[2]);
this->ui->Adminname->setText(column[3]);
```

5.3 用户投诉/建议

用户登录后在左侧菜单栏选择投诉/建议页面，本页面为用户提供了发送投诉/建议的功能。如图 5-7 所示。

5-7 发送投诉/建议

获取用户输入

std::string text = this->ui->ComplaintandAcvice_TextEdit->toPlainText().toStdString();

std::string text1 = this->ui->lineEdit_4->text().toStdString();

将其添加到数据库

MySQLSentence = "INSERT INTO

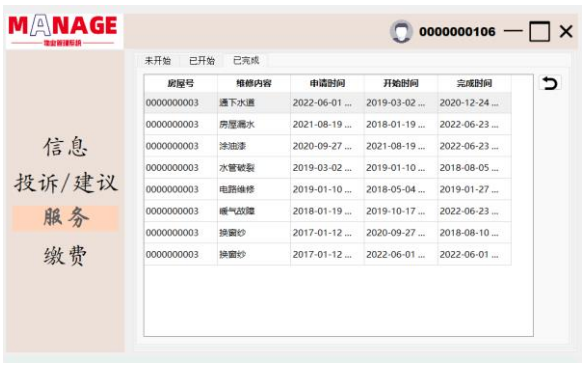
complaint_and_acviceinfo(c_acontent,c_atime,c_aobject,c_a_userid) VALUES('" + text + "', NOW(), '" +text1+ "', " + this->User.UserID + "); ";

5.4 用户服务

用户登录后在左侧菜单栏选择服务页面，本页面为用户提供了维修任务申请、维修任务查看、购买停车位、已购停车位查看功能。如图 5-8，5-9，5-10，5-11….所示。

5-8 未开始维修任务查看

5-9 已开始维修任务查看



5-10 已完成维修任务查看

5-11 发布维修任务

5-12 拥有的停车位查询

5-13 购买停车位

1、 未开始维修任务查询

SQL 查询

```
MySQLSentence = "SELECT task_houseid,taskcontent,applicationtime FROM
taskinfo where taskstate = 0 and (task_houseid =    ";
for (auto house : this->User.HouseId)
{
    MySQLSentence += house;
    MySQLSentence += " or task_houseid = ";
}
MySQLSentence += "0)";
MySQLSentence += " ORDER BY applicationtime DESC;";
```

将查询结果显示在 QTableWidget 中

```
row = mysql_affected_rows(&this->MySQL_Data);
this->ui->Nottable->setRowCount(row);
int j=0;
while((column = mysql_fetch_row(res)))
{
    this->ui->Nottable->setItem(j,0,new
QTableWidgetItem(QString("%1").arg(QString::fromStdString(std::string(column[0])).toInt(),10,
10, QLatin1Char('0'))));
    for(int i=1;i<3;i++)
    {
        QString tmp(column[i]);
        tmp += "    ";
        this->ui->Nottable->setItem(j,i,new QTableWidgetItem(tmp));
    }
    j++;
}
```

2、 已开始维修任务查询

SQL 查询

```
MySQLSentence = "SELECT task_houseid,taskcontent,applicationtime ,taskstarttime FROM
taskinfo,admin_taskrelation where taskid=admin_task_taskid AND taskstate = 1 and
(task_houseid =    ";
for (auto house : this->User.HouseId)
{
    MySQLSentence += house;
    MySQLSentence += " or task_houseid = ";
}
MySQLSentence += "0)";
MySQLSentence += " ORDER BY applicationtime DESC;";
```

将查询结果显示在 QTableWidget 中

```
row = mysql_affected_rows(&this->MySQL_Data);
this->ui->Hastable->setRowCount(row);
int j=0;
```

```
while((column = mysql_fetch_row(res)))
{
        this->ui->Hastable->setItem(j,0,
        new QTableWidgetItem(QString("%1").arg(QString::
        fromStdString(std::string(column[0])).toInt(),
        10, 10, QLatin1Char('0'))));
    for(int i=1;i<4;i++)
    {
        QString tmp(column[i]);
        tmp += "    ";
        this->ui->Hastable->setItem(j,i,new QTableWidgetItem(tmp));
    }
    j++;
}
```

3、 已完成维修任务查询

SQL 查询

```
MySQLSentence = "SELECT
task_houseid,taskcontent,applicationtime,taskstarttime,taskendtime FROM
taskinfo,admin_taskrelation where taskid=admin_task_taskid AND taskstate = 2 and
(task_houseid =    ";
for (auto house : this->User.HouseId)
{
    MySQLSentence += house;
    MySQLSentence += " or task_houseid = ";
}
MySQLSentence += "0)";
MySQLSentence += " ORDER BY applicationtime DESC;";
```

将查询结果显示在 QTableWidget 中

```
row = mysql_affected_rows(&this->MySQL_Data);
this->ui->Fitable->setRowCount(row);
int j=0;
while((column = mysql_fetch_row(res)))
{
    this->ui->Fitable->setItem(j,0,
            new QTableWidgetItem(QString("%1").arg(QString::
            fromStdString(std::string(column[0])).toInt(),
            10, 10, QLatin1Char('0'))));
        for(int i=1;i<5;i++)
        {
            QString tmp(column[i]);
            tmp += "    ";
            this->ui->Fitable->setItem(j,i,new QTableWidgetItem(tmp));
        }
        j++;
```

```
}
    //this->ui->Nottable->setEditTriggers(QAbstractItemView::NoEditTriggers);
    mysql_free_result(res);
}
```

4、 发布维修任务

获取输入内容，SQL 语句

```
std::string text = this->ui->TasktextEdit->toPlainText().toStdString();
std::string text1 = this->ui->Task_comboBox->currentText().toStdString();
if(text.empty() || text1.empty())
    return;
MySQLSentence = "INSERT INTO taskinfo(taskstate, taskcontent,task_houseid,
applicationtime) VALUES (0, '"+ text +"', "+ text1 +", now());";
```

5、 拥有的停车位查询

SQL 查询

```
MySQLSentence = "SELECT pcarid, pcarloction, buytime, location FROM pcarinfo,
houseinfo, user_houserelation WHERE pcar_houseid = houseid and pcar_houseid =
uh_relation_houseid and uh_relation_userid = " + this->User.UserID + " ORDER
BY buytime DESC;";
```

将查询结果显示在 QTableWidget 中

```
row = mysql_affected_rows(&this->MySQL_Data);
this->ui->Mypcar_Table->setRowCount(row);
int j=0;
while((column = mysql_fetch_row(res)))
{
    this->ui->Mypcar_Table->setItem(j,0,
            new QTableWidgetItem(QString("%1").arg(QString::
            fromStdString(std::string(column[0])).toInt(),
            10, 10, QLatin1Char('0'))));
        for(int i=1;i<3;i++)
        {
            QString tmp(column[i]);
            tmp += "    ";
            this->ui->Mypcar_Table->setItem(j,i,new QTableWidgetItem(tmp));
        }
        j++;
}
```

6、 购买停车位

SQL 查询

```
MySQLSentence ="SELECT pcarid, pcarloction, pcarcost FROM pcarinfo WHERE pcarstate <> 2;";
```

将查询结果显示在 QTableWidget 中

```
row = mysql_affected_rows(&this->MySQL_Data);
this->ui->Buypcar_Table->setRowCount(row);
int j=0;
while((column = mysql_fetch_row(res)))
```

```
{
    this->ui->Buypcar_Table->setItem(j,0,
            new QTableWidgetItem(QString("%1").arg(QString::
            fromStdString(std::string(column[0])).toInt(),
            10, 10, QLatin1Char('0'))));
        for(int i=1;i<3;i++)
        {
            QString tmp(column[i]);
            tmp += "    ";
            this->ui->Buypcar_Table->setItem(j,i,new QTableWidgetItem(tmp));
        }
        this->ui->Buypcar_Table->setItem(j,3,new QTableWidgetItem("购买"));
        j++;
}
```

未出售停车位详情

```
void MainWindow::on_Buypcar_Table_cellDoubleClicked(int row, int col)
{
    if(col==3)
    {
        std::string MySQLSentence;
        this->ui->Pcar_file->setCurrentIndex(3);
        this->ui->Pcarid_lable->setText(this->ui->Buypcar_Table->item(row,
0)->text());
        this->ui->Pcarseat_lable->setText(this->ui->Buypcar_Table->item(row,
1)->text());
        this->ui->Pcarcost_lable->setText(this->ui->Buypcar_Table->item(row,
2)->text());

    }
}
```

车位出售

```
void MainWindow::on_BuypushButton_clicked()
{
    std::string MySQLSentence;
    std::string tmp = this->ui->Pcarid_lable->text().toStdString();
    std::string text1 = this->ui->Pcar_comboBox->currentText().toStdString();
    MySQLSentence = "UPDATE pcarinfo SET pcarstate = 2, pcar_houseid = "+ text1 +","
                    " buytime = localtime() WHERE pcarid = "+ tmp +";";
    if (mysql_query(&this->MySQL_Data, MySQLSentence.c_str()))          //执行查询语句
    {
        QString Error = "error:";
        Error += mysql_error(&this->MySQL_Data);
        this->ui->ShowLabel->setText(Error);
        return;
```

```
    }
    MySQLSentence = "INSERT INTO costinfo(cost_costtypeid, cost_houseid,"
                    "cost_userid, costmoney, coststarttime, paytime, istrue) "
                    "VALUES (7, " + text1 + ", "
                    + this->User.UserID + ", "
                    + this->ui->Pcarcost_lable->text().toStdString() +","
                    "NOW(), NOW(), 1);";
    if (mysql_query(&this->MySQL_Data, MySQLSentence.c_str()))          //执行查询语句
    {
        QString Error = "error:";
        Error += mysql_error(&this->MySQL_Data);
        this->ui->ShowLabel->setText(Error);
        return;
    }
    Pcar_mypcar();
    Pcar_buycar();
    this->ui->Pcar_file->setCurrentIndex(2);
}
```

5.5 用户缴费

用户登录后在左侧菜单栏选择缴费页面，本页面为用户提供了缴费记录查看和缴费功能。如图 5-14，5-15，5-16. ……. 所示

5-14 已缴费列表

5-15 缴费记录详情

5-16 未缴费列表

5-17 缴费详情

1、 已缴费

SQL 查询

```
MySQLSentence = "SELECT costid, paytime, cost_houseid, costtype FROM costinfo,
costtypeinfo WHERE cost_costtypeid = costtypeid and cost_userid = " + this->User.UserID + "
ORDER BY paytime DESC;";
```

将查询结果显示在 QTableWidget 中

```
row = mysql_affected_rows(&this->MySQL_Data);
this->ui->bill_Table->setRowCount(row);
int j=0;
while((column = mysql_fetch_row(res)))
{
        this->ui->bill_Table->setItem(j,0,
            new QTableWidgetItem(QString("%1").arg(QString::
            fromStdString(std::string(column[0])).toInt(),
            10, 10, QLatin1Char('0'))));
        for(int i=1;i<4;i++)
        {
            QString tmp(column[i]);
            tmp += "    ";
            this->ui->bill_Table->setItem(j,i,new QTableWidgetItem(tmp));
        }
        j++;
}
```

查询缴费记录详情

```
std::string tmp = this->ui->bill_Table->item(row, 0)->text().toStdString();
MySQLSentence ="SELECT costid, coststarttime, costmoney, paytime, cost_houseid, costtype
FROM costinfo, costtypeinfo WHERE cost_costtypeid = costtypeid and costid = " + tmp + ";";
```

显示缴费记录详情

```
column = mysql_fetch_row(res);
this->ui->CostID->setText(QString("%1").arg
        (QString::fromStdString(std::string(column[0])).toInt(),
        10, 10, QLatin1Char('0')));
this->ui->Costcreattime->setText(column[1]);
this->ui->Costmoney->setText(column[2]);
this->ui->Costtime->setText(column[3]);
this->ui->CostHouseId->setText(column[4]);
this->ui->Costtype->setText(column[5]);
this->ui->CostuserId->setText(QString("%1").arg(
        QString::fromStdString(std::string(this->User.UserID)).toInt(),
        10, 10, QLatin1Char('0')));
this->ui->Costusername->setText(QString::fromStdString(std::string(this->User.Use
rName)));
```

2、 未缴费

SQL 查询

```
MySQLSentence = "SELECT costid,coststarttime,costmoney, cost_houseid, costtype FROM
costinfo, costtypeinfo WHERE cost_costtypeid = costtypeid and cost_userid IS NULL and
```

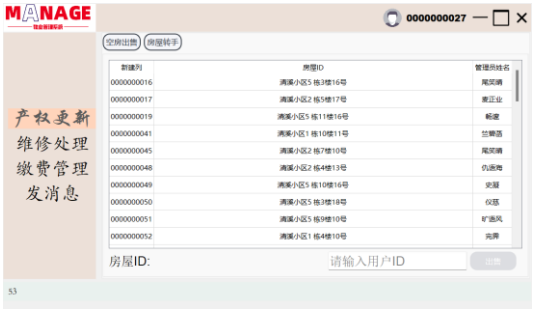
```
(cost_houseid = " ;
for (auto house : this->User.HouseId)
    {
        MySQLSentence += house;
        MySQLSentence += " or cost_houseid = ";
    }
MySQLSentence += "0)";
MySQLSentence += " ORDER BY coststarttime DESC;";
```
将查询结果显示在 QTableWidget 中
```
row = mysql_affected_rows(&this->MySQL_Data);
this->ui->pay_table->setRowCount(row);
int j=0;
while((column = mysql_fetch_row(res)))
    {
        for(int i=0;i<5;i++)
        {
            QString tmp(column[i]);
            tmp += "    ";
            this->ui->pay_table->setItem(j,i,new QTableWidgetItem(tmp));
        }
        this->ui->pay_table->setItem(j,5,new QTableWidgetItem("去缴费"));
        j++;
    }
```
显示待缴纳费用详情
```
void MainWindow::on_pay_table_cellDoubleClicked(int row, int col)
{//信息显示
    if(col==5)
    {
        std::string MySQLSentence;
        this->ui->cost_file->setCurrentIndex(4);
        this->ui->Costidlable->setText(this->ui->pay_table->item(row,
0)->text());
        this->ui->Begin_lable->setText(this->ui->pay_table->item(row,
1)->text());
        this->ui->Money_Lable->setText(this->ui->pay_table->item(row,
2)->text());
        this->ui->HouseID_Lable->setText(this->ui->pay_table->item(row,
3)->text());
        this->ui->Type_Lable->setText(this->ui->pay_table->item(row,
4)->text());

    }
}
```

缴纳费用

```
void MainWindow::on_Pay_PushButton_clicked()
{//费用缴纳
    std::string MySQLSentence;
     std::string tmp = this->ui->Costidlable->text().toStdString();
    MySQLSentence = "UPDATE costinfo set paytime = localtime(), cost_userid
="
                        + this->User.UserID + " WHERE costid =" +tmp+ ";";
    if (mysql_query(&this->MySQL_Data, MySQLSentence.c_str()))          //执
行查询语句
    {
        QString Error = "error:";
        Error += mysql_error(&this->MySQL_Data);
        this->ui->ShowLabel->setText(Error);
        return;
    }
    Cost_bill();
    Cost_pay();
    this->ui->cost_file->setCurrentIndex(3);
}
```

5.6 管理员产权更新

管理员登录后在左侧菜单栏选择产权更新页面，本页面为管理员提供了空房出售和房屋转手功能。如图 5-18、5-19 所示。



5-18 空房出售

5-19 房屋转手

1、 空房出售

列举空房以供选择，输入用户 ID 后，点击出售按钮。调用‘updata_new’存储过程完成数据库的数据更新

2、 房屋转手

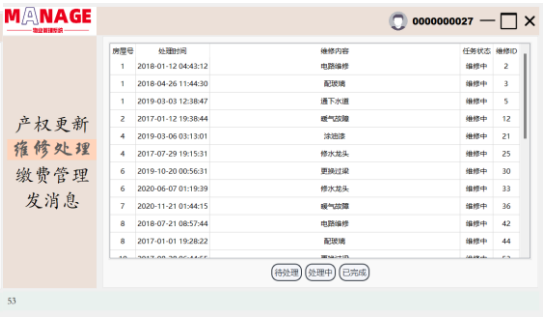
输入用户 ID,房屋 ID 后，点击出售按钮。调用‘updata_old’存储过程完成数据库的数据更新

5.7 管理员维修处理

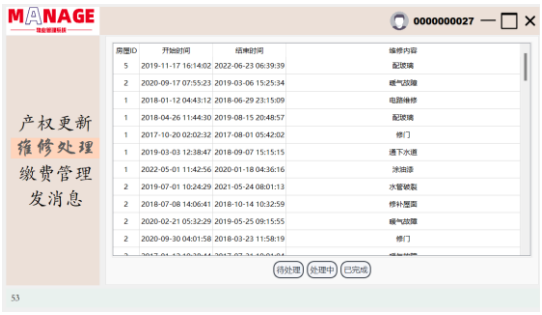
管理员登录后在左侧菜单栏选择维修处理页面，本页面为管理员提供了查看维修任

务和修改维修状态功能。如图 5-20，5-21，5-22 所示。

5-20 待处理维修任务



5-21 处理中维修任务



5-22 处理完成维修任务

1、 查看维修任务

提供 SQL 语句分别查询待处理、处理中、处理完成的维修任务

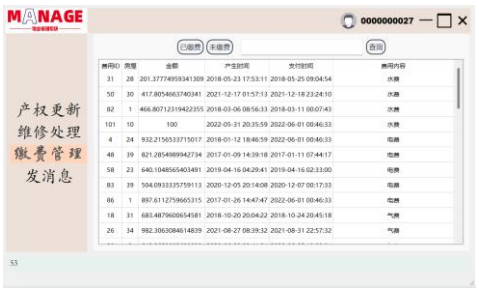
2、 修改维修状态

通过双击进行维修任务的状态修改

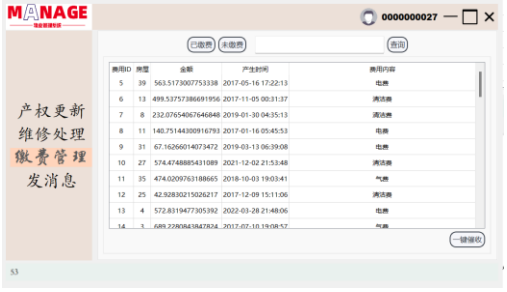
调用‘receive_task’，‘successful_task’存储过程完成数据库的数据更新

5.8 管理员缴费管理

管理员登录后在左侧菜单栏选择缴费管理页面，本页面为管理员提供了查看缴费信息和催收未缴费功能。如图 5-23，5-24 所示。



5-23 已缴费



5-24 未缴费

1、 查看缴费信息

提供 SQL 语句分别查询已缴费，未缴费的缴费数据

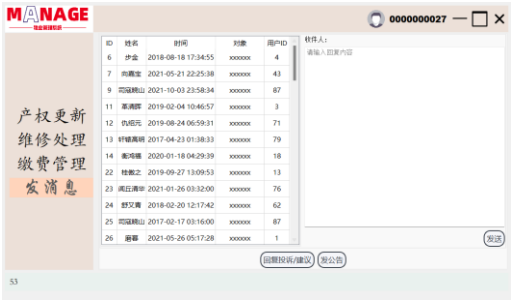
2、 催收未缴费

该功能未实现

5.9 管理员发消息

管理员登录后在左侧菜单栏选择发消息页面，本页面为管理员提供了回复投诉/建

议和发公告功能。如图 5-25，5-26 所示。



5-25 回复投诉/建议

5-26 发公告

1、 回复投诉/建议

首先查询所有未回复的投诉/建议，双击查看详情，编辑消息导入数据库

2、 发公告

编辑消息导入数据库

6、设计总结

本次课程设计中，我们小组设计的是物业管理系统，此系统分为业主端和管理员端两部分，这两个端口我们都实现了登录和注册功能。 在业主端，我们设计了信息、投诉/建议、服务和缴费四个模块。“信息”可以查询业主所有信息；“投诉/建议”提交用户的提交建议内容到数据库；“服务”包含“停车位”和“维修”两个部分，“停车位”实现了查询用户拥有停车位和购买停车位两个功能，“维修”实现了查询维修任务和申请任务两个功能；“缴费”实现了查看已缴费信息和缴费两个功能。在管理员端，我们设计了产权更新、车位管理、维修处理、缴费管理和新消息五个模块。“产权更新”实现了空房出售和房屋转手两个功能；“车位管理”实现了查看车位信息；“维修处理”可以修改维修任务状态；“缴费管理”实现了查看缴费信息；“新消息”实现回复用户投诉/建议和发公告两个功能。

不完善：

1、修改密码功能，用户修改信息功能，催收缴费功能还未实现。

2、根据 ID 查询信息时对用户不太友好。

3、窗口界面不够美观。

7、设计体会

胡鹏：

本次设计，我学会了 Mysql c++API 以及 QT 框架的使用，并更加生客地理解了模块化编程。在设计过程中和组员相互协作也让我体会到团队合作的重要性。

罗荣：

本次设计作业在知识和心得上都获益良多，学习了 c++-mysql Api 等知识踏出了第一步，从无到有。虽然作品还很粗糙，但是积累了经验，也认识到很多不足，要好好总结在以后的设计中注意这些问题。

李万余：

本次设计作业在知识和心得上都获益良多，大体上知道了开发的流程，虽然对于用户的体验未有太多考虑，但是积累了经验，也认识到很多不足，要好好总结在以后的设计中注意这些问题。

附录 A、操作手册

管理员操作手册

打开软件进入登录界面，登录界面左侧有功能模块。本界面包含登录按钮输入正确的密码和电话或邮箱可以登录，否则会在下方提醒错误信息。左侧功能模块只有在登录后才可以使用。如图 A-1 所示

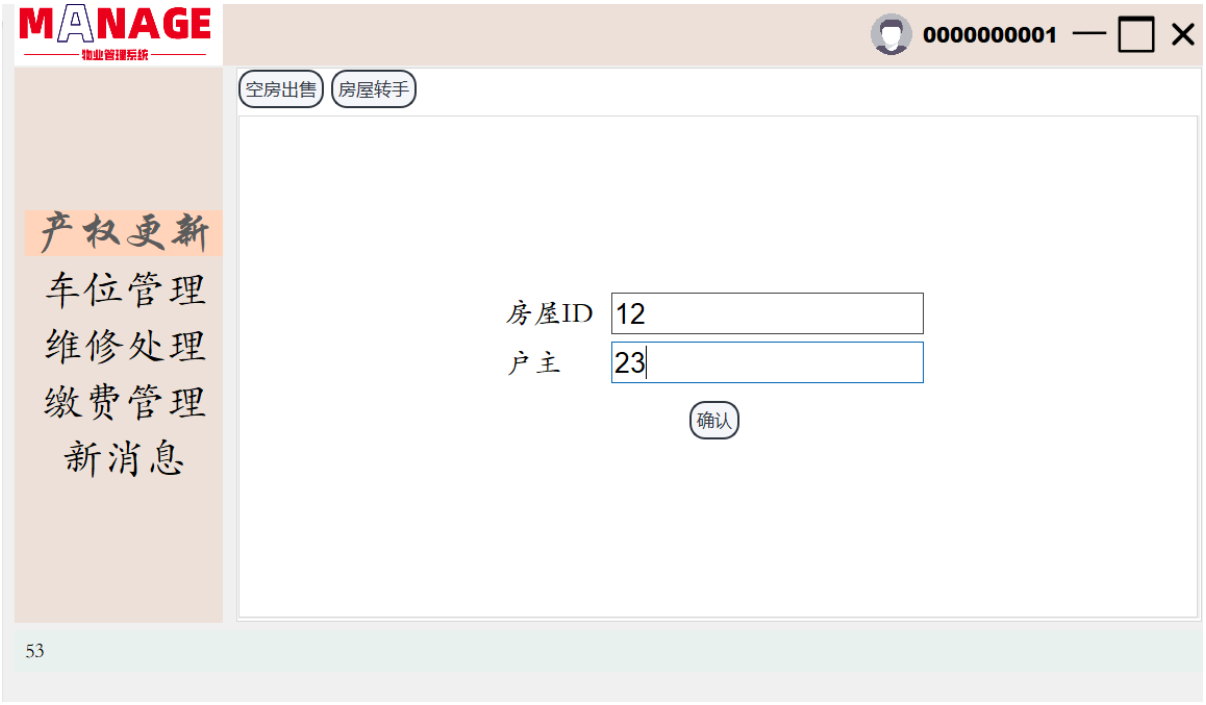
图A-1 登录首页

点击登录后进入主页展示界面，本界面主要展示管理员基本信息（姓名，电话，邮箱），包含退出登录后修改密码两个按钮。点击退出登录按钮可以退到登录页，需要重新登录才能再次进入主页，登陆后左侧按钮可以点击使用。如图 A-2 所示

图A-2 主页信息

点击左侧产权更新按钮进入产权更新界面的空房出售页，以表格形式展示空房信息（）双击其中一个空房，用户ID框输入相应ID然后点击出售按钮可以出售空房。点击房屋转售可以进入房屋转售界面输入要转售的ID和出售户主点击确认即可转售房屋。如图5-3和图5-4所示

图A-3 产权更新-房屋出售



图A-4 产权更新-房屋转手

点击车位管理进入车位管理界面，点击已出售将以表格形式展示已出售车位信息（车位ID，业主，购买时间，位置）；点击未出售展示未出售车位信息（车位ID和位置），双击车位，然后在房屋ID处输入信息，点击后出售后即可完成出售。 如图A-5和A-6所示

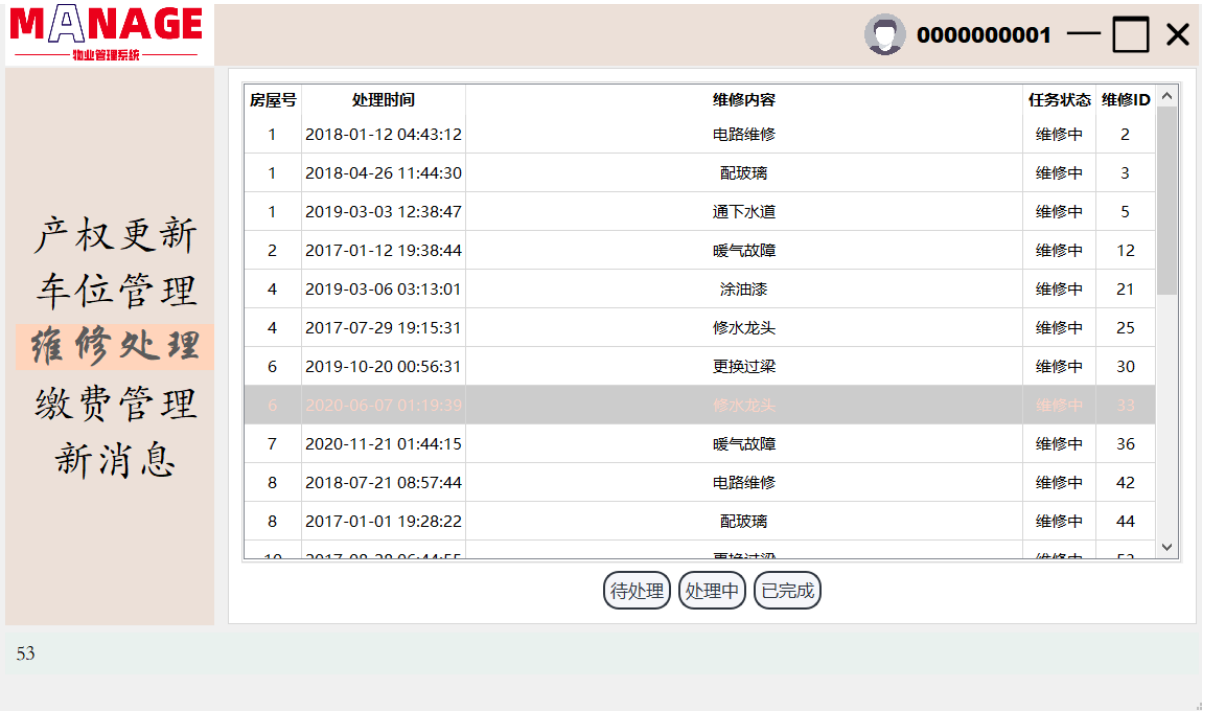
车位ID	业主	购买时间	位置
9	介雅诗	2022-05-31 21:55:55	A区 1-9
11	介雅诗	2022-06-23 10:55:28	A区 2-1
27	竺痴春	2022-06-23 04:37:53	A区 3-7
27	封春冬	2022-06-23 04:37:53	A区 3-7
27	度流婉	2022-06-23 04:37:53	A区 3-7
27	纵侠	2022-06-23 04:37:53	A区 3-7
27	尚尔安	2022-06-23 04:37:53	A区 3-7
27	胡鹏	2022-06-23 04:37:53	A区 3-7
46	贺涵涵	2022-06-23 10:56:06	B区 2-6
46	喻采绿	2022-06-23 10:56:06	B区 2-6

图A-5 车位管理-已出售

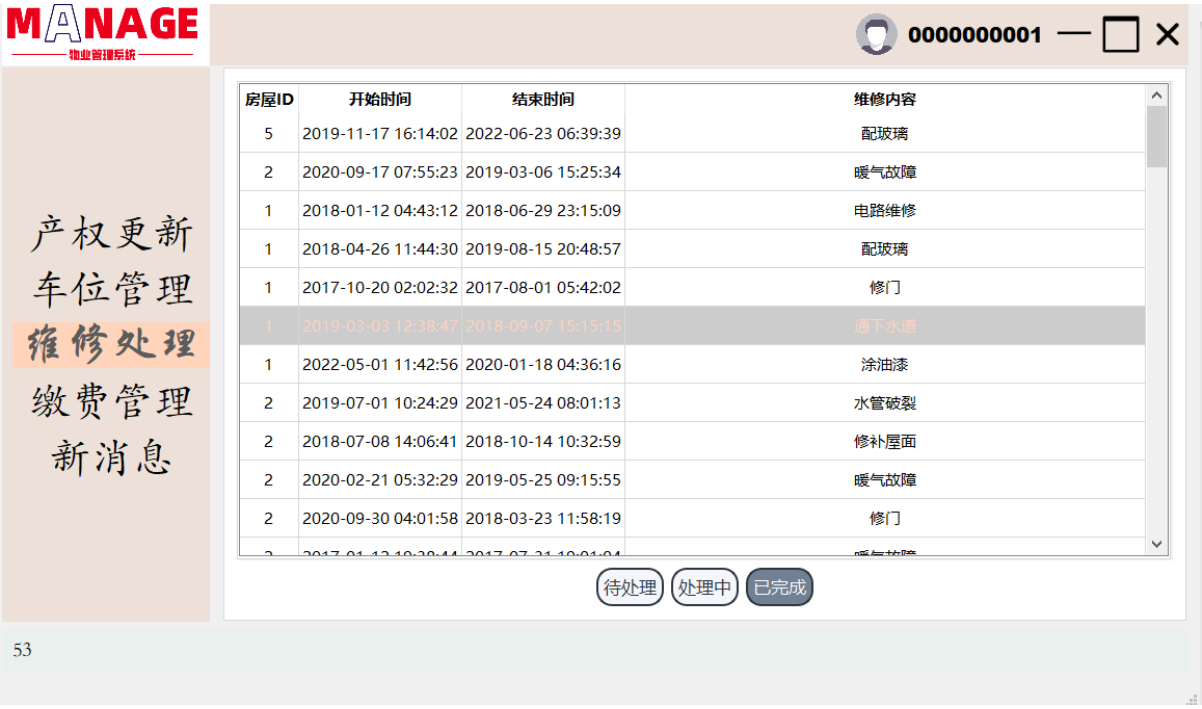
图A-5 车位管理-未出售

点击维修处理进入维修界面，点击待处理展示待处理任务信息（房屋号，维修内容，维修ID），点击处理中可以展示处理中任务信息（房屋号，处理时间，维修内容，任务状态，维修ID），点击已完成按钮展示已完成任务信息（房屋号，开始时间，结束时间，维修内容）。双击信息可以改变任务状态（待处理→处理中→已完成）。如图A-6，A-7，A-8所示

图A-6 维修处理-待处理

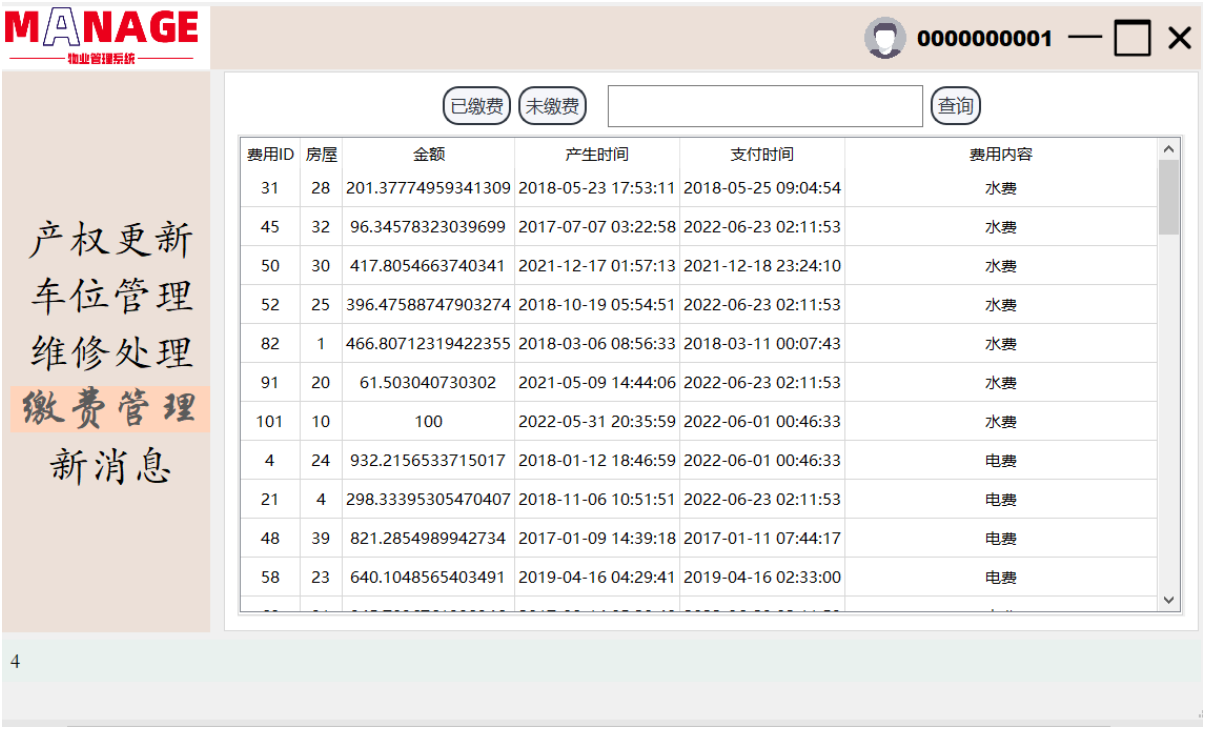


图A-7 维修处理-处理中



图A-8 维修处理-已完成

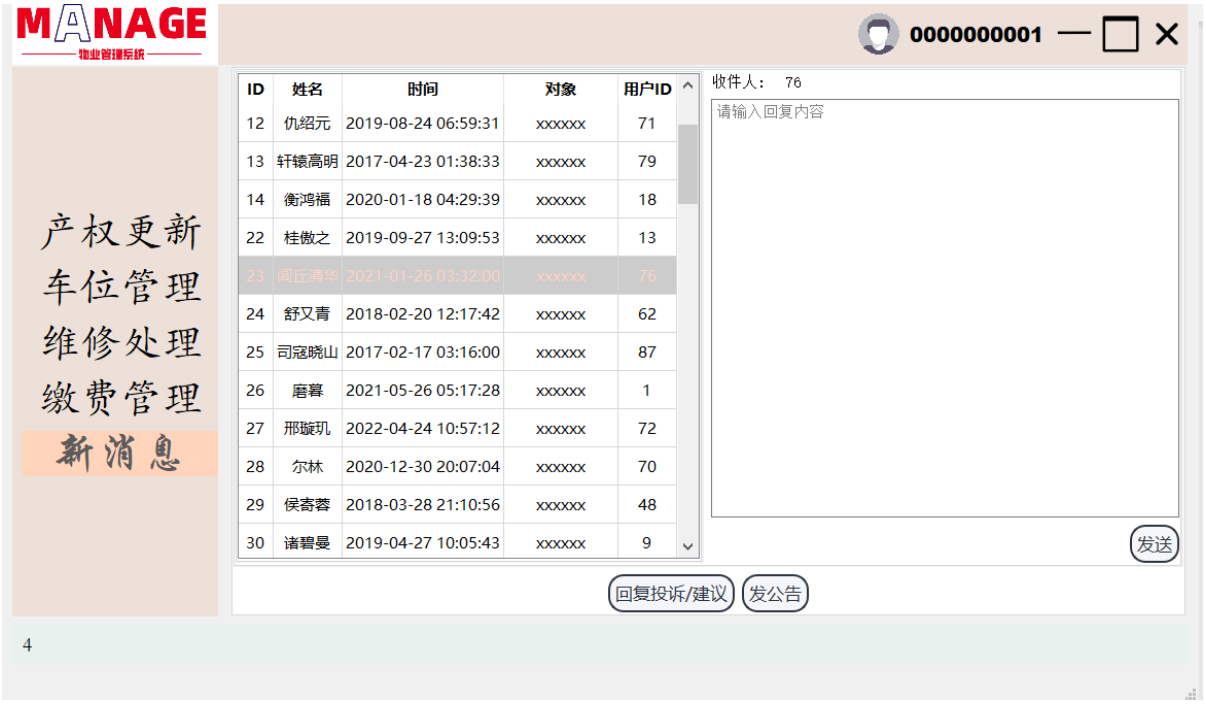
点击缴费管理可以进入缴费界面，点击已缴费展示已缴费信息，点击未缴费也会展示未缴费信息，在未缴费页可以点击一件催收完成催费提醒。在文本框输入房屋ID可以查询该房屋产生的所有费用信息。如图A-9，A-10所示



图A-9 缴费管理-已缴费

图A-10 缴费管理-未缴费

点击新消息进入消息界面，点击回复投诉建议展示待回复的消息记录，双击可以查看消息详情同时自动填充收件人，输入回复内容点击发送完成回复；点击发广告，完成输入后点击发送，完成发送公告。如图A-11，A-12，A-13所示



图A-11新消息-回复投诉/建议

图A-12 新消息-发送

图A-13 新消息-发公告

用户操作手册

用户登录界面包含注册，忘记密码。输入正确的密码账号（电话或邮箱），点击登录进入使用效果如图U-1，U-2，U-3所示

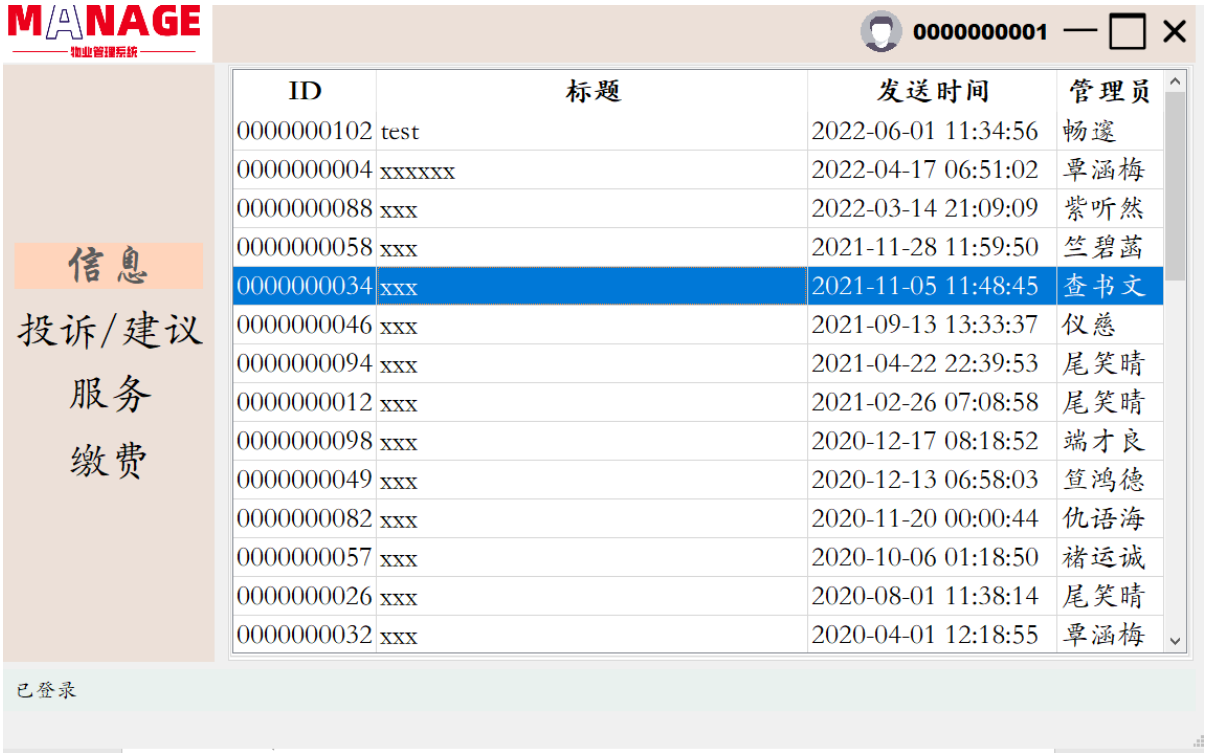
图U-1 登录

图U-2 注册账号

成功登录后在左侧的功能可以点击使用，进入主页将展示用户个人信息。运行效果如图U-3所示

图U-3 主页信息展示

点击消息进入消息界面，将展示管理员发来的信息或通知，双击进入查看消息详情，也可以再点击返回图标回到消息页。如图U-4，U-5所示



图U-4 信息

图U-5 信息-详情

点击投诉/建议，输入投诉建议内容然后输入你所投诉的对象，点击发送完成投诉如图U-6所示

图U-6 投诉/建议

点击服务进入服务页，维修服务可以查看我的任务和申请新任务；。运行效果如图U-7，

图U-7 维修服务

停车服务可以查看我的已经购买车位和购买一个新车位。如图U-8所示

图U-8 停车服务

点击购买后进入结算界面。如图U-9所示

图U-9 支付界面

点击缴费进入缴费界面，用户可以查看缴费记录。如图U-10所示

图U-10 查看缴费记录

点击缴费可以查看用户的未缴费账单，点击可以跳转支付界面。如图U-11

图U-11 支付费用