



中國石油大學 (华东)
CHINA UNIVERSITY OF PETROLEUM

信息系统设计实训报告

2016-2017-3 学期

2017. 7. 16

小组成员组成及成绩评定

小组总评成绩			
姓名	学号	负责内容概况	成绩
万益嘉	1608020213	后端 sqlite 数据库的建表、连接；报告、PPT 第五部分撰写。	
王丽丽	1608020214	前端票务相关界面的代码的书写、调试；与数据库连接；报告、PPT 第三四部分梳理。	
郑畅	1608020223	前段登录注册界面的代码书写、调试；与数据库连接；报告、PPT 的第一二部分梳理。	

目录

第 1 章 项目概况及要求	4
1.1 项目背景	4
1.3 参考资料	5
第 2 章 系统需求分析	6
2.1 用户功能要求	6
2.2 系统性能要求	7
2.3 输入输出要求	8
2.4 用户其他要求	8
2.5 可行性论证	9
第 3 章 系统分析	10
3.1 组织结构及业务流程分析	10
3.2 用例图分析(含用例描述和活动图)	11
3.3 实体类图分析	16
第 4 章 系统设计	17
4.1 总体设计（功能树设计）	17
4.2 顺序图设计	17
4.3 类图设计	19
第 5 章 系统测试与部署	22
5.1 系统架构选择（或应用程序结构设计）	22
5.2 系统部分代码示例	23
5.3 系统界面实现（贴界面实际图）	27
5.4 系统测试	30
5.5 系统实施应用	31
第 6 章 其他说明	32
6.1 拟实现功能说明	32

第 1 章 项目概况及要求

1.1 项目背景

1.1.1 编写目的

本信息系统设计实训报告的编写目的在于分析校车订退票系统的系统需求、结构设计、业务流程，以及形成新系统的逻辑方案等内容，以成为下一阶段开发实施等工作的依据和今后验收目标系统的检验标准。报告使用者主要是与校车订退票系统开发有联系的决策者、开发人员、软件测试人员以及负责验收本项目的相关领导。

1.1.2 系统名称

校车订退票系统

1.1.3 系统来源

传统的线下校车服务效率低，大部分学校使用的仍然是纸质车票，给教职工和学生带来许多不便。相反，如果采取线上购票模式，将大大提高校车服务的效率，便于师生购买到适时的车票。该校车订票系统的分析与设计正是基于这一点，致力于实现订票、退票、查询等功能，为师生筛选最适宜的车次信息，并提供更好的用户体验。

1.1.4 系统背景

随着大学校区的不断增加及教职工和学生人数的不断增长，很多大学根据实际情况给予师生一定的福利，例如为在校师生提供了往返不同校区的车次，方便师生新老校区之间的来往，解决他们在学习及工作之外的其他难题。

在此之前，车票代理窗口购买车票是最传统的方式，但随着师生工作生活节奏的加快和互联网的普及以及信息化时代的到来，计算机科学日渐成熟，其强大的功能已为人们深刻认识，它已进入人类社会的各个领域并发挥着越来越重要的作用。提前到代理点购买车票就显得耗费时间，而且整个班车数据的统计及地点的设计等都存在一定的问题，无

法达到大规模合理的使用，造成资源的浪费。于是，越来越多的人开始采用在线上系统上购票的方式，订票系统的需求越来越明显，性能良好的订票系统应运而生。

1.2 定义

本系统是一个主要服务于在校大学生及教职工人员，为方便不同校区间交通出行而设计，致力于实现订票、退票、查询等功能的校车订退票系统。

1. 校车：用于运送教职工及学生往返学校的交通工具，本系统特指往返大学各个校区的班车。

2. 退票：退票是一种我们在生活中经常碰到的情况，由于各种原因我们无法使用票据，可以将其返还给原始售票者，这其中可能要收取一定比例的手续费。退票必须是在票有效时间内完成。

3. 网上订票：网上订票省时省力，已成为不少人的首选。用户根据出行需要自行上网设定条件查询机票信息，起飞时间、票价、附加费用等，选定需要的航班，并通过安全、便捷、高效的认证支付完成机票购买。

1.3 参考资料

1、卫红春，《信息系统分析与设计》，西安电子科技大学出版社；

2、吴忠，朱君璇，《信息系统分析与设计》，清华大学出版社；

3、复旦大学班车时刻表最新版，

<http://www.xyfw.fudan.edu.cn/db/20/c1954a121632/page.htm>；

4、复旦大学校园生活服务平台车辆服务，

<http://www.xyfw.fudan.edu.cn/p2049c1954/list.htm>；

5、一种高校校车预约和调度管理系统及方法与流程，

http://www.xjishu.com/zhuanli/57/201610209096_2.html

第 2 章 系统需求分析

2.1 用户功能要求

对于一个校车订退票管理系统，准确快捷的车次信息查询功能和方便简单的订票操作是十分必要的，因此系统必须为用户（教职工和学生）提供准确的车次信息和当前的车票状况，这也是订票管理系统的最基本要求。该软件系统应给乘客提供订票，退票和查询的功能。在该系统内订票，退票等需要首先认证登录人的身份（是否为在校学生及教职工）。用户填写自己的真实基本信息进行注册登陆后，输入想要乘坐的车次或要求（如始发站、目的地、出发时间等），系统为乘客查询对应的车次票务数据，并完成订票工作。另外客户端后台管理员要能够及时添加、更改车次等。

本校车订退票系统应该具备如下功能：

2.1.1 注册

用户可以通过该系统使用自己的学号或教工号进行注册，注册后可以进行相关的操作。

2.1.2 订票

师生填写自己的真实基本信息进行注册登陆后，在查询界面根据自己的需求输入想要乘坐的车次要求（如始发地、目的地、日期等），系统为师生查询对应的车次票务数据，并完成订票工作。

2.1.3 退票

用户通过登录该系统，查询到自己已经购买的校车票，通过退票模块退去已购车票。

2.1.4 查询

分为用户对个人登录信息的查询，校车信息的查询和已购车票信息的查询。要求：

- (1) 用户登录该系统后，可以查询到自己的个人信息和已经预定的车票信息。
- (2) 对车次的查询，可以按照出发地和目的地对列车信息进行查询；
- (3) 车次信息只允许用户查询，不能修改。

2.2 系统性能要求

系统的整体性能需求应该满足以下四点：

第一，承受能力。由于师生出行需求的增加，该订票系统在系统设计和开发过程中要充分考虑系统当前和将来能够承受的工作量,使系统的业务处理能力和响应时间能够满足基本要求，做到无 BUG，无闪退，用户量大时不能出现崩溃现象。

第二，可扩充性。面对假期回家等高峰时期师生出行需求的大幅增加以及校车车次的相应调整，系统应用规模将不断扩大，用户的使用需求也会渐趋个性化和多样化。这就要求该系统功能实现不断更新和完善等。

第三，易用性和易维护性。系统应该尽量使用用户熟悉的术语，避免生僻的专业术语。针对用户可能出现的问题，如退票手续费等规则问题，可以直接查询帮助功能了解规则，加深用户对系统功能熟悉的程度，尽量减少因系统规则或操作带来的不便。

第四，容错性和健壮性。在使用过程中，系统对于意外死机、停电、网络故障等突发事件应有足够的应对能力。系统应该能够保证在出现故障时能确保有效的现场数据以备故障排除后恢复数据。另外,还要充分考虑到用户在使用系统的过程中可能进行不当操作，能及时给予用户提示或警告,而不能影响系统的正常运行。

3.2.1 数据精确度

- (1) 各种数据参数保持一致性，符合规范，易于修改与完善；
- (2) 数据的数值类型要尽量符合输入输出的要求，易于存储与调用。

3.2.2 时间特性

- (1) 信息查询要求 5 秒以内；
- (2) 登录要求响应时间在 5 秒以内；
- (3) 每笔订退票业务的响应时间在 52 秒；
- (4) Web 服务持续稳定工作时间 ≥ 72 小时。

3.2.3 适应性

- (1) 根据使用者的不同，系统提供三种入口，一种是师生入口，一种是管理员入口，一种是司机入口，以适应不同使用者的需求。
- (2) 模块化的开发使得该系统可以在开发计划发生改变时及时作出相应改变。
- (3) 数据库可选择 MySQL、SQL Server 等，符合数据库设计的一致性、完整性、安全性原则；移动端可采用安卓或 iOS、微信公众号或小程序，提高用户粘性；服务器端可在 java、c#、python、nodejs、ruby、php 选择其中一个或多。
- (4) 客户端能够方便的查询系统及简单明了的订票、退票操作，以防止用户重复订票、退票操作，影响系统数据的正确性。

2.3 输入输出要求

- (1) 页面设计要求清晰明了，功能划分明确，使用户可以一目了然的知道自己需要使用的功能。

- (3) 用户登录使用查询与订票功能时，要求稳定性高、流畅性高,余票情况的更新处理要及时。

2.4 用户其他要求

(1) 要求系统所占内存尽量小，不妨碍现有程序的运行速度。

(2) 加强对管理员的培训，提升管理员对操作系统的熟练程度，目标是做到新增车票和车次更改信息的时差最小化。

2.5 可行性论证

2.5.1 技术可行性

使用 visual studio 开发窗口应用程序，相关知识在以前课程学过，技术难度小

2.5.2 经济可行性

本项目不需要财务花费

第 3 章 系统分析

3.1 组织结构及业务流程分析

3.1.1 组织结构分析

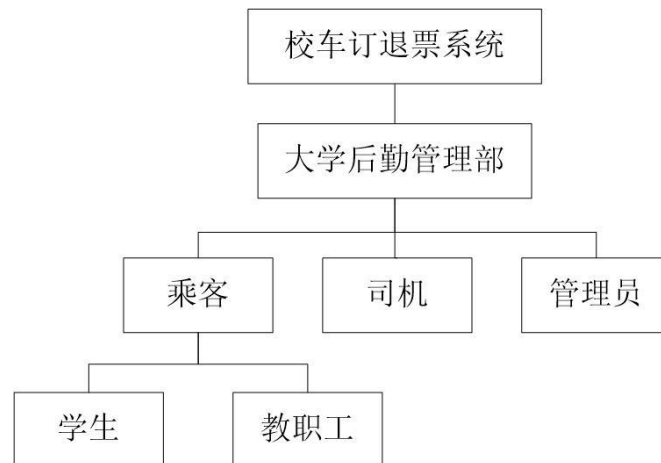


图 3-1 组织结构图

3.1.2 业务流程分析

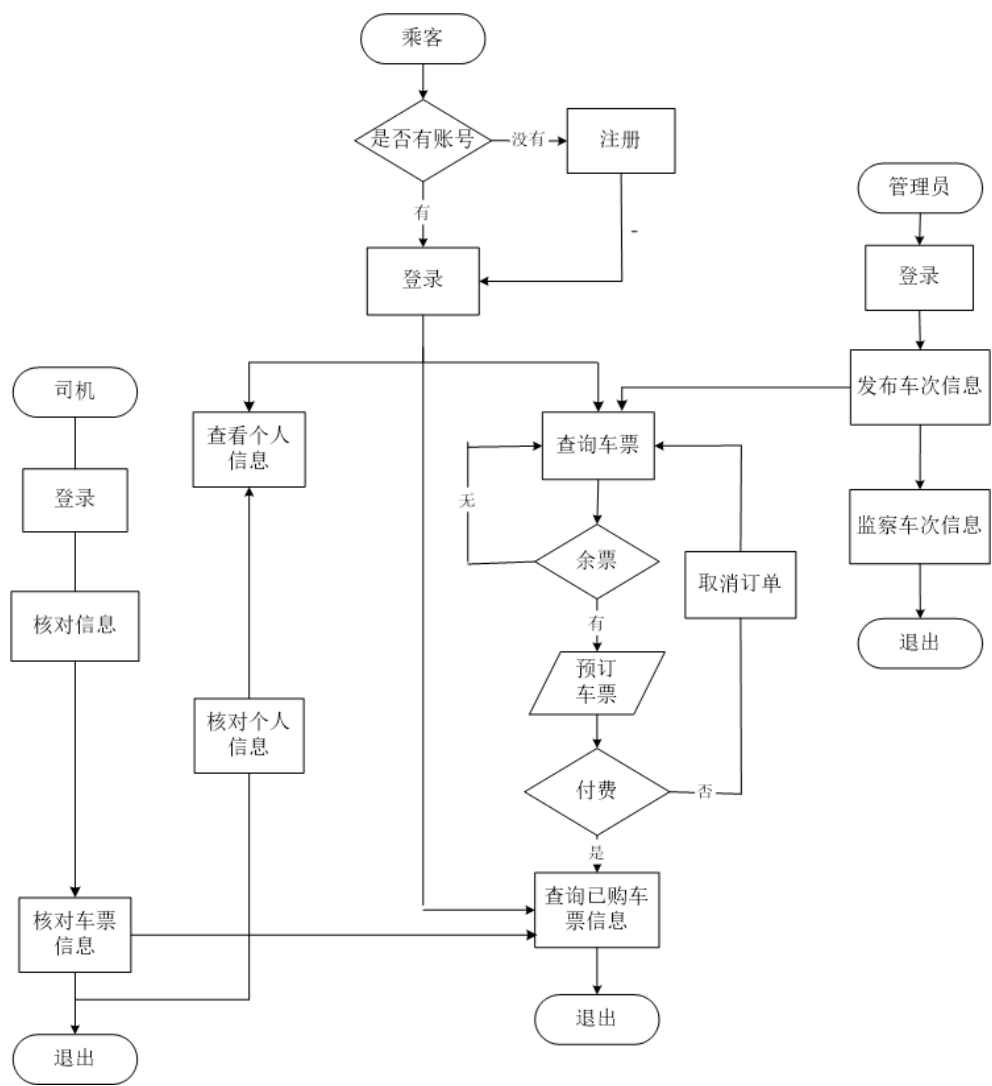


图 3-2 业务流程图

3.2 用例图分析(含用例描述和活动图)

为展示校车订退票系统的用户与系统之间的交互，本系统的系统用例图及业务用例图（包括乘客事务管理、票务管理和司机事务管理）如下：

3.2.1 总体用例建模

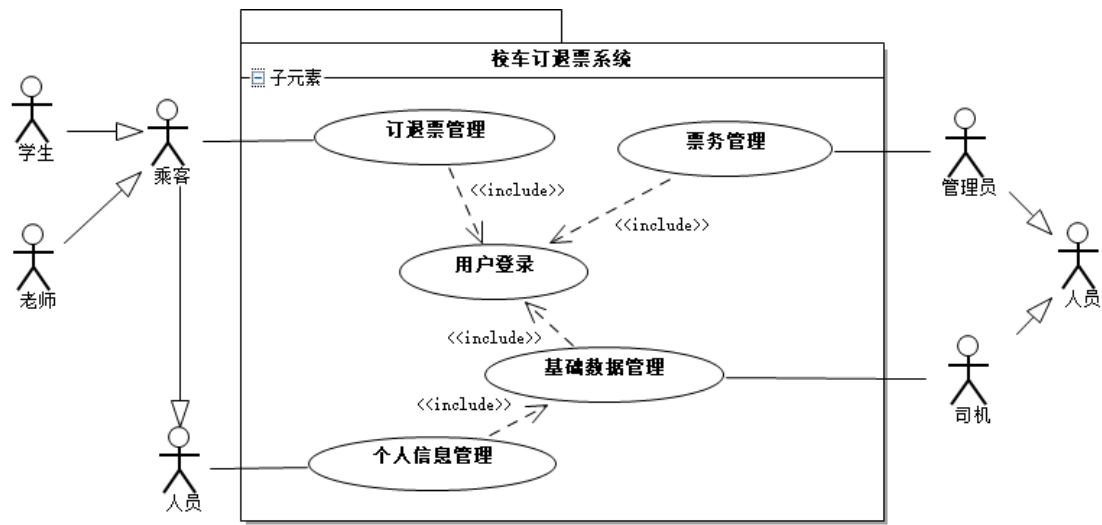


图 3-3 系统用例图

3.2.2 业务用例建模

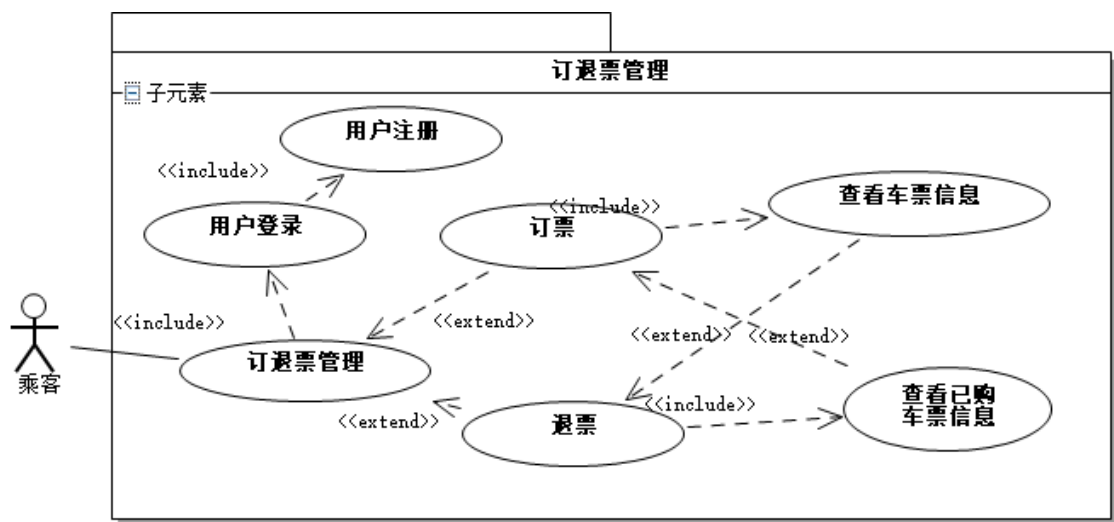


图 3-4 乘客事务管理

用例名称：乘客事务管理

参与者：乘客

前置条件：乘客有条件通过网络访问系统并已成功登录系统

后置条件：系统增加一条订退票记录

基本事件流：

- 1、乘客通过网络登录成功访问系统
- 2、乘客查看车票信息并订购车票
- 3、乘客查看已购车票信息并选择是否退票
- 4、操作后提交系统校验

异常事件流：

- 1、车票信息无法查看
- 2、订退票操作失败

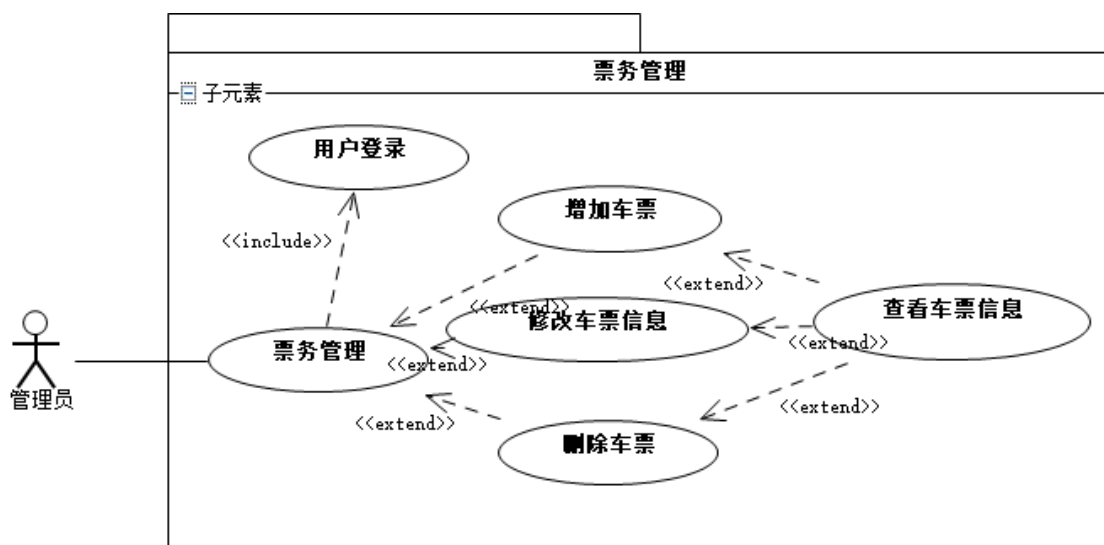


图 3-5 票务管理

用例名称：票务管理

参与者：管理员

前置条件：管理员有条件通过网络访问系统并已成功登录系统

后置条件：系统增加、删除或修改一条车票信息

基本事件流：

- 1、管理员通过网络登录成功访问系统
- 2、管理员查看车票信息并更改车票
- 3、操作后提交系统校验

异常事件流：

- 1、车票信息无法查看
- 2、车票更改操作失败

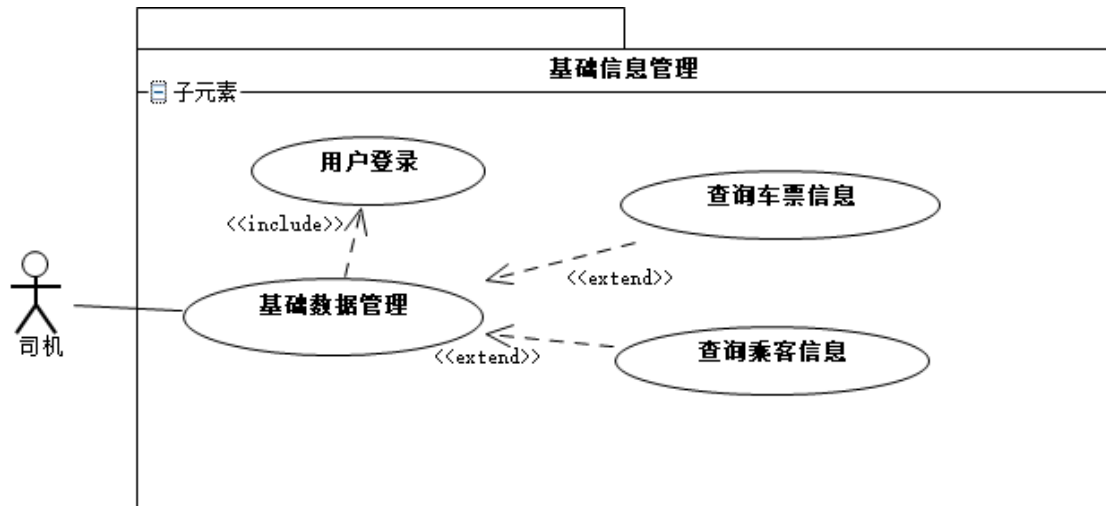


图 3-6 司机事务管理

用例名称：司机事务管理

参与者：司机

前置条件：司机有条件通过网络访问系统并已成功登录系统

后置条件：无

基本事件流：

- 1、司机通过网络登录成功访问系统
- 2、司机查询车票信息
- 3、司机查询乘客信息
- 4、操作后提交系统校验

异常事件流：

- 1、车票信息无法查看
- 2、车票更改操作失败

3.2.4 活动图

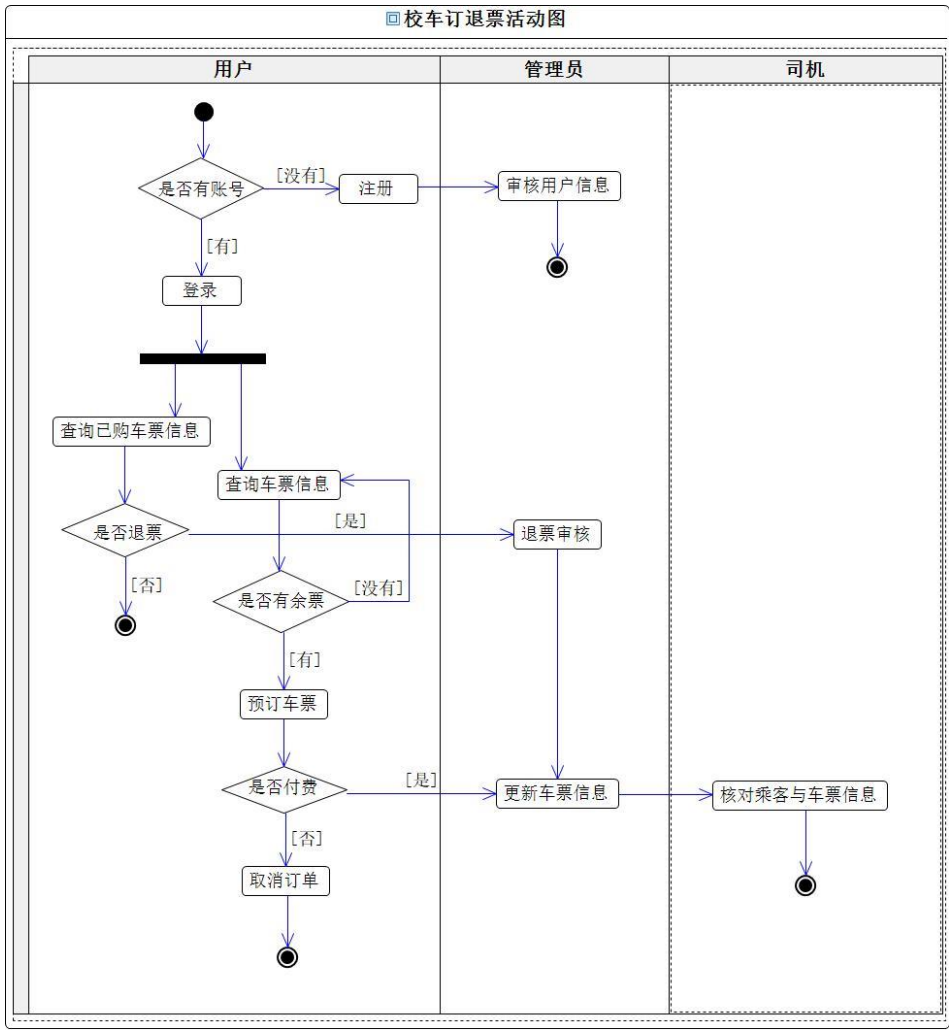


图 3-7 活动图

3.3 实体类图分析

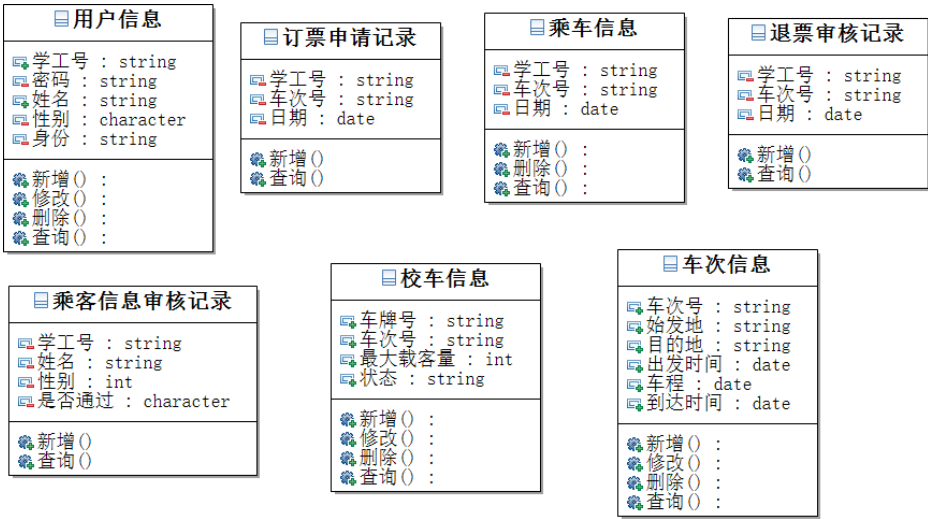


图 3-8 实体类图

第 4 章 系统设计

4.1 总体设计（功能树设计）

本校车订退票系统分为乘客信息系统和车票信息系统两个子系统。其中，乘客信息系统负责乘客信息的管理，存储已在系统注册的、有乘坐校车需求的师生信息，由管理员进行后台操作。车票信息系统负责车票信息的管理，存储可供购买、已购买等不同状态的车票信息，由管理员进行后台增加、删除、修改车票等操作。

根据本系统的系统功能分析，可画出系统的功能结构如图所示。

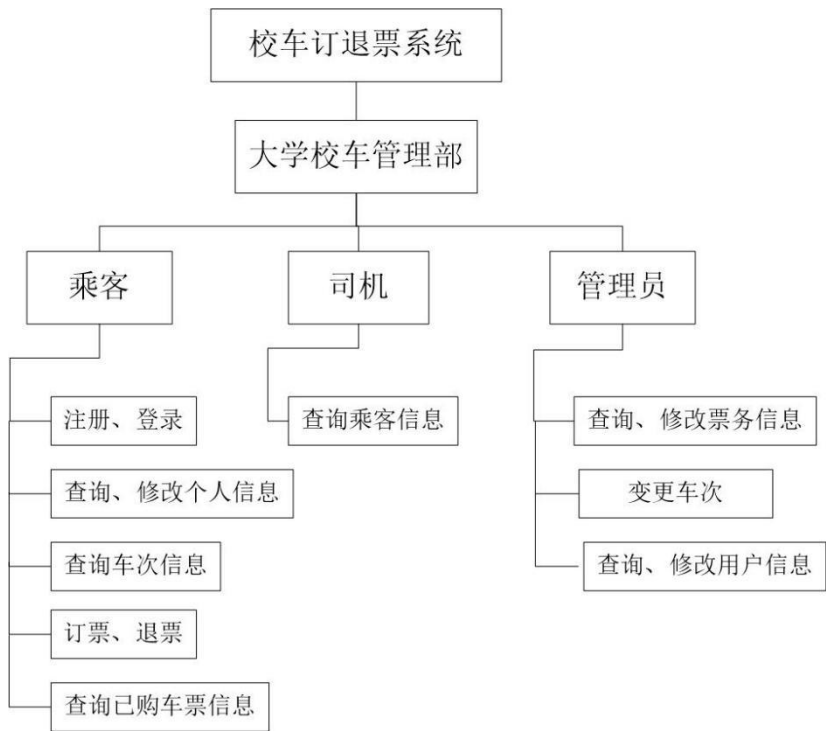


图 4-1 功能结构图

4.2 顺序图设计

为展示校车订退票系统设计模式的应用以及系统运行过程，顺序图如下所示：

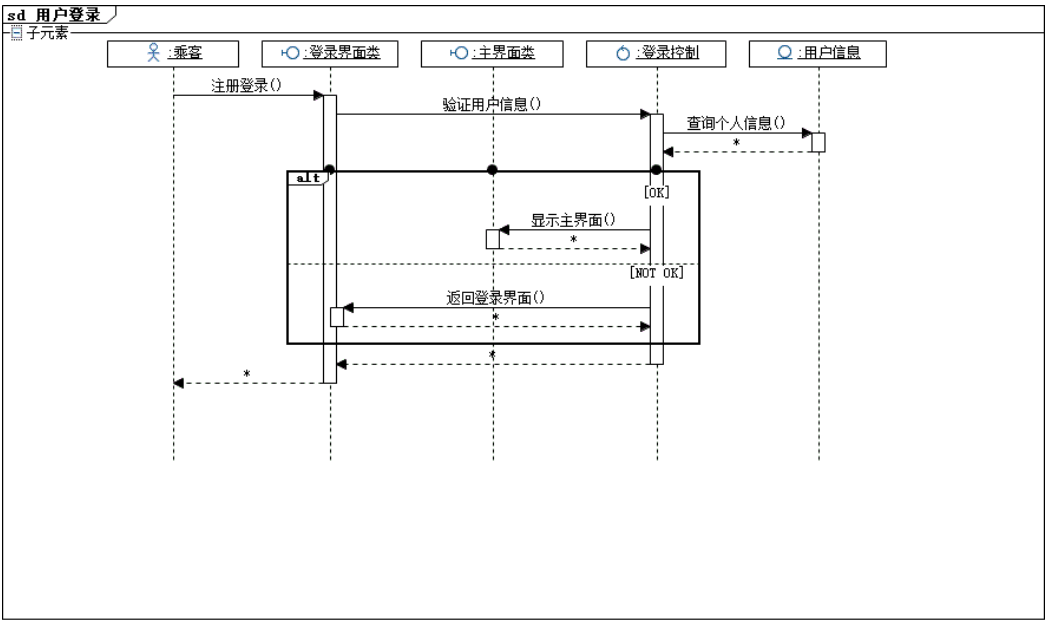


图 4-2 用户登录

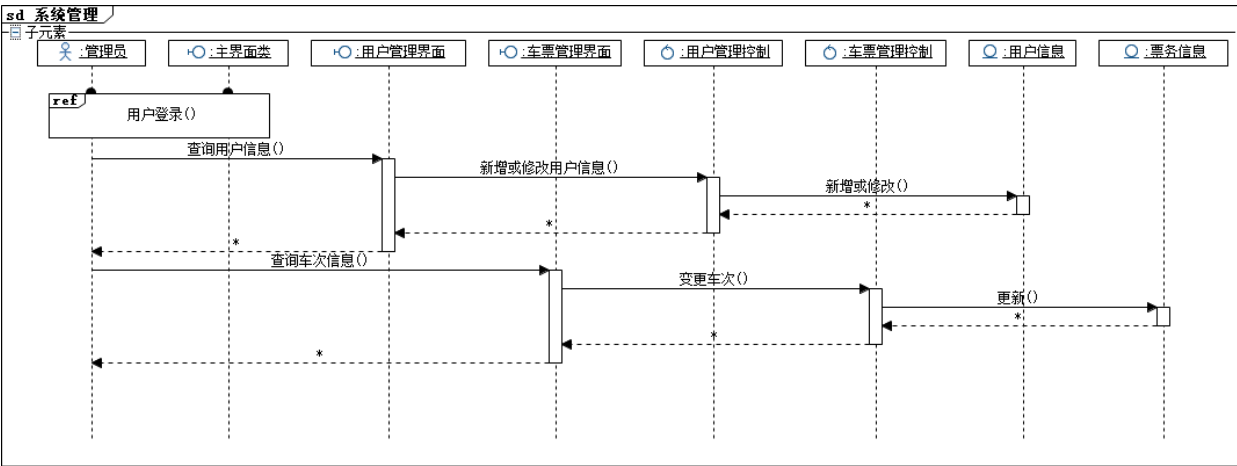


图 4-3 系统管理

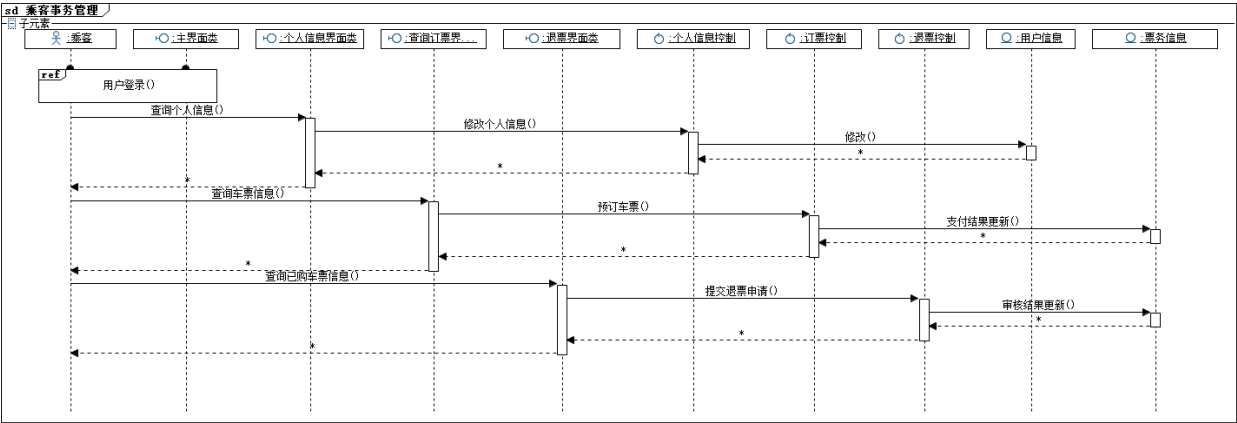


图 4-4 乘客事务管理

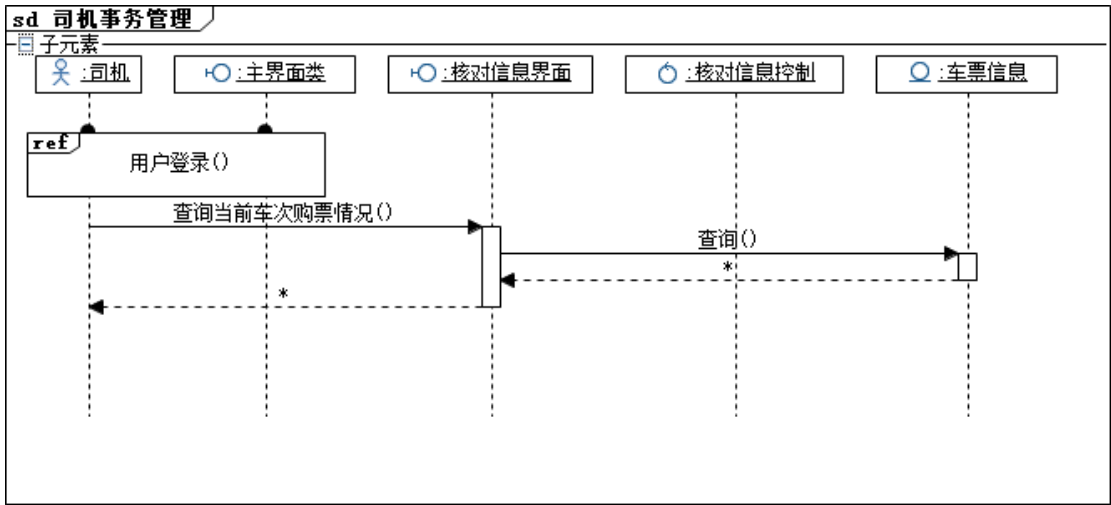


图 4-5 司机事务管理

4.3 类图设计

4.3.1 类关系图建模

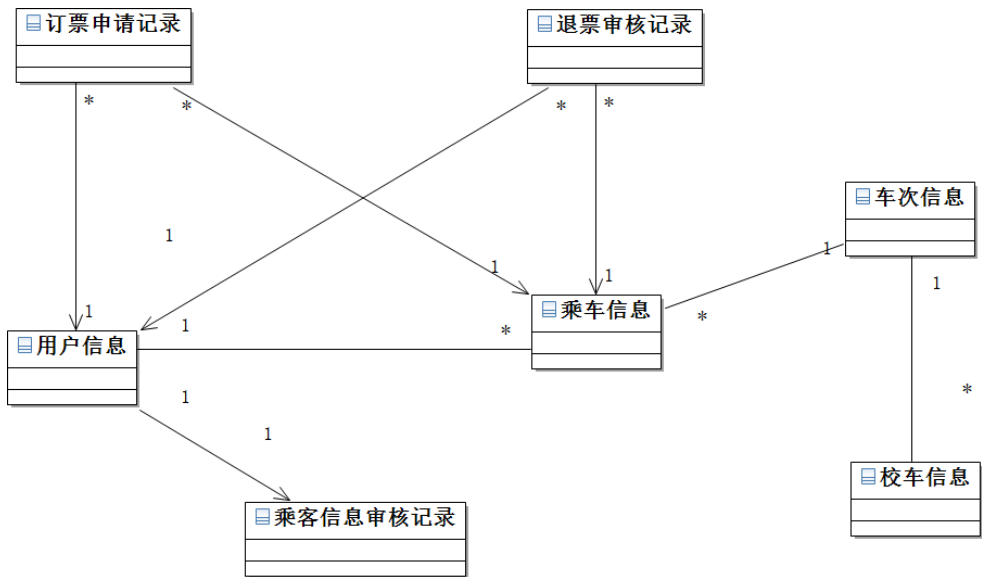


图 4-6 类关系图

4.3.2 健壮性分析

为实现本系统静态建模与动态建模之间的转换，反映界面类、中间件、数据库的结构关系及调用关系，健壮性分析图如 4-7、4-8、4-9 所示。

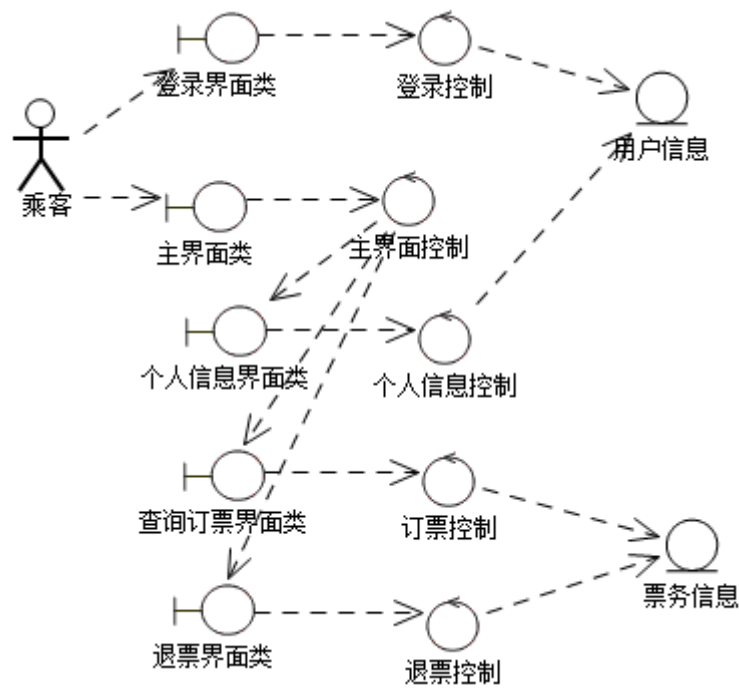


图 4-7 乘客事务管理

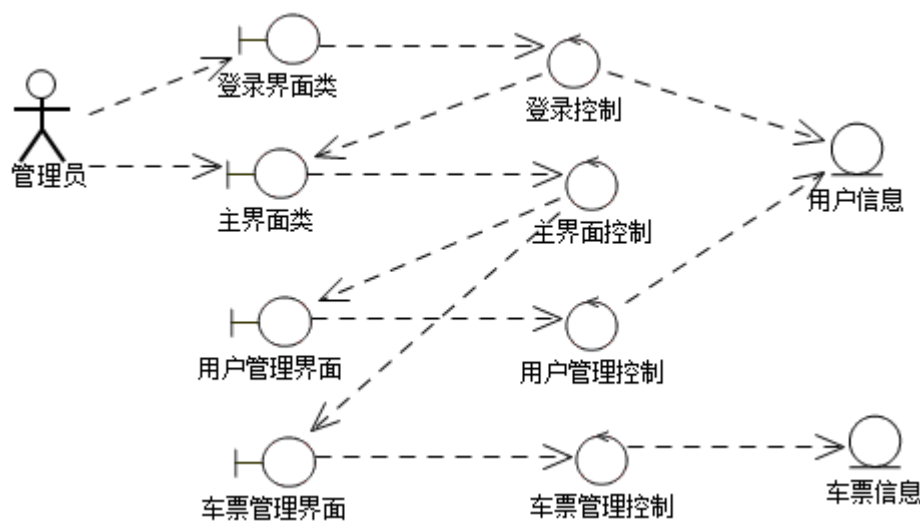


图 4-8 系统管理

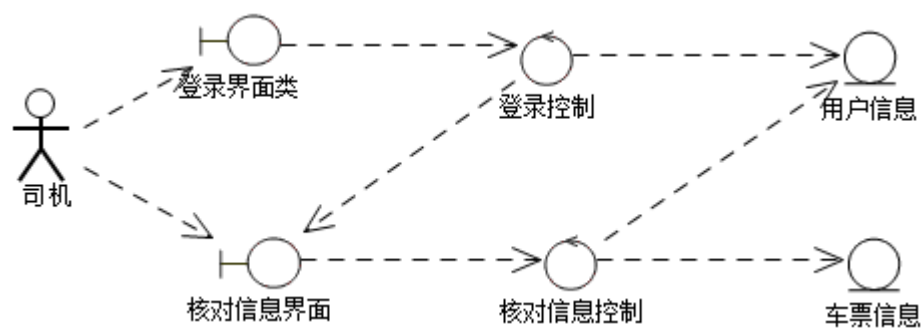


图 4-9 司机事务管理

第 5 章 系统测试与部署

5.1 系统架构选择（或应用程序结构设计）

本系统采取 B/S 网络架构设计，即浏览器请求，服务器响应的工作模式。教职工及学生可以直接通过浏览器访问 Internet 上由 Web 服务器产生的文本、数据、图片等信息而无需安装应用程序；而每一个 Web 服务器又可以通过各种方式与数据库服务器连接，大量的数据实际存放在数据库服务器中。

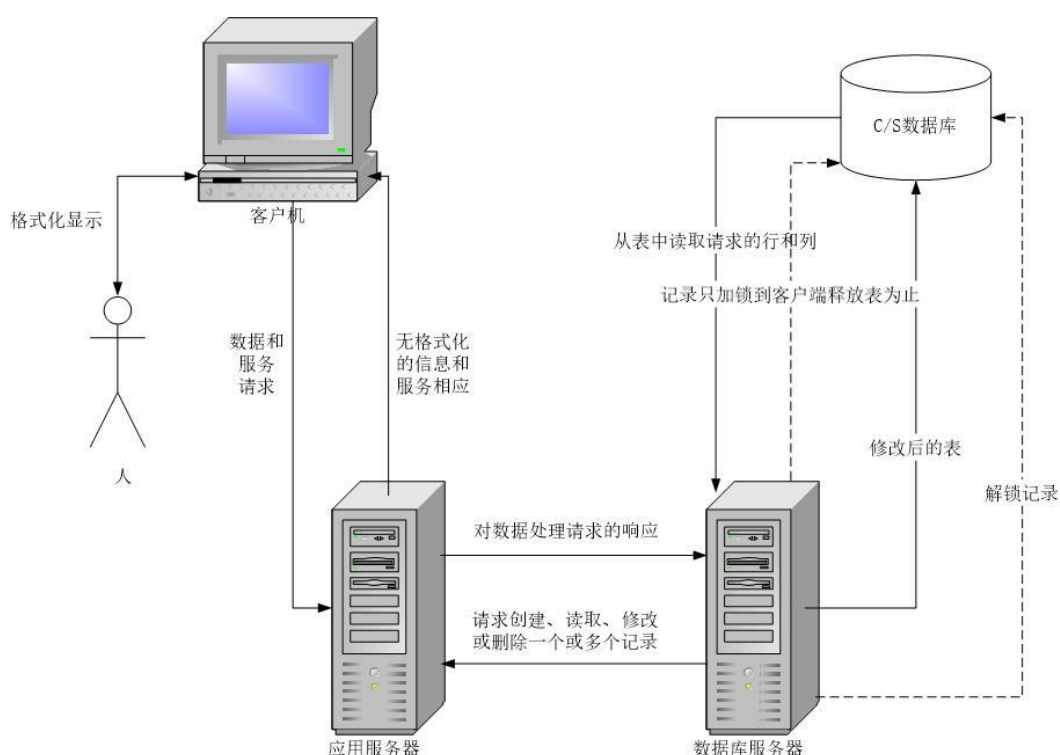


图 5-1 系统 B/S 架构图

工作流程：

1. 客户端发送请求：用户在客户端浏览器页面提交表单操作，向服务器发送请求，等待服务器响应；
2. 服务器端处理请求：服务器端接收并处理请求，应用服务器端通常使用服务器端技术，如 JSP 等，对请求进行数据处理，并产生响应；
3. 服务器端发送响应：服务器端把用户请求的数据（网页文件、图片、声音等）返回给浏览器。

浏览器解释执行 HTML 文件，呈现用户界面。

5.2 系统部分代码示例

用户登录代码示例

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    if (getSupportActionBar() != null) {
        getSupportActionBar().setTitle("用户登陆");
    }

    findViewById(R.id.btn_login).setOnClickListener(this);
    findViewById(R.id.btn_register).setOnClickListener(this);
    edit_pwd = findViewById(R.id.edit_pwd);
    edit_user = findViewById(R.id.edit_user);
}
```

用户注册代码示例

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);

    findViewById(R.id.btn_register).setOnClickListener(this);
    edit_user = findViewById(R.id.edit_user);
    edit_pwd = findViewById(R.id.edit_pwd);
    radio_group = findViewById(R.id.radio_group);
    RadioButton radio_btn_admin = findViewById(R.id.radio_btn_admin);
    RadioButton radio_btn_common = findViewById(R.id.radio_btn_common);
    //radioGroup 清除子 View 选择状态
    radio_group.clearCheck();
    //普通用户 默认选中
    radio_btn_common.setChecked(true);
}
```

主界面代码示例

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);
    //获取 LoginAct 传递过来的 User 对象 对象需要实例化才能够进行传递 UserBean 实现了 Parcelable 接口
    //AndroidStudio 有自动 Parcelable 化的插件
    UserBean user = getIntent().getParcelableExtra("user");
    if (user == null) {
        showToast("用户数据获取失败");
    }
}
```

```

        finish();
        return;
    }

    TextView tv_welcome = findViewById(R.id. tv_welcome);
    TextView btn_query_tickets = findViewById(R.id. btn_query_tickets);
    TextView btn_add_ticket = findViewById(R.id. btn_add_ticket);
    TextView btn_delete_ticket = findViewById(R.id. btn_delete_ticket);
    TextView btn_query_my_tickets = findViewById(R.id. btn_query_my_tickets);
    TextView btn_ticket_purchase = findViewById(R.id. btn_ticket_purchase);
    TextView btn_refund_ticket = findViewById(R.id. btn_refund_ticket);
    findViewById(R.id. btn_logout).setOnClickListener(this);
    btn_refund_ticket.setOnClickListener(this);
    btn_ticket_purchase.setOnClickListener(this);
    btn_query_my_tickets.setOnClickListener(this);
    btn_delete_ticket.setOnClickListener(this);
    btn_add_ticket.setOnClickListener(this);
    btn_query_tickets.setOnClickListener(this);

    boolean isAdmin = user.getUserType() == 0;
    String userName = user.getUserName();
    //使用 SharedPreferences 记录 登陆用户的用户名
    SPUtils.getInstance().put("userName", userName);
    //不同的用户类型显示不同的文字
    tv_welcome.setText("欢迎" + (isAdmin ? "管理员" : "用户") + userName);
    //不同的用户类型 显示不同的按钮
    if (isAdmin) {
        btn_query_my_tickets.setVisibility(View. GONE);
        btn_ticket_purchase.setVisibility(View. GONE);
        btn_refund_ticket.setVisibility(View. GONE);
    } else {
        btn_add_ticket.setVisibility(View. GONE);
        btn_delete_ticket.setVisibility(View. GONE);
    }
}

```

票务处理相关代码示例

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout. activity_tickets);

    list_view = findViewById(R.id. list_view);
    tv_empty = findViewById(R.id. tv_empty);
    //获取传递过来的操作类型
    type = getIntent().getIntExtra("type", 0);
    //使用 SharedPreferences 记录 登陆用户的用户名
}

```



```

userName = SPUtils.getInstance().getString("userName");

switch (type) {
    default:
    case 0://查票
    case 1://删票
    case 3://购票
    {
        //查票、删票、购票 这三种 类型都是使用 Tickets 表中的数据
        ActionBar supportActionBar = getSupportActionBar();
        if (supportActionBar != null) {
            if (type == 0) {
                supportActionBar.setTitle("查看车票列表");
            } else if (type == 1) {
                supportActionBar.setTitle("删除车票");
            } else {
                supportActionBar.setTitle("购买车票");
            }
        }

        ticketList = DatabaseHelper.getInstance().getAllTickets();
        break;
    }
    case 2://已购票
    case 4://退票
    {
        //已购票、退票 使用的是 UserTickets 中的数据
        ActionBar supportActionBar = getSupportActionBar();
        if (supportActionBar != null) {
            if (type == 2) {
                supportActionBar.setTitle("已购车票列表");
            } else {
                supportActionBar.setTitle("退票操作");
            }
        }

        ticketList = DatabaseHelper.getInstance().getUserTickets(userName);
        break;
    }
}

if (ticketList == null || ticketList.size() == 0) {
    tv_empty.setVisibility(View.VISIBLE);
    list_view.setVisibility(View.GONE);
    return;
}

tv_empty.setVisibility(View.GONE);
list_view.setVisibility(View.VISIBLE);

```

```

adapter = new BaseAdapter() {
    @Override
    public int getCount() {
        return ticketList == null ? 0 : ticketList.size();
    }

    @Override
    public Object getItem(int position) {
        return ticketList == null ? null : ticketList.get(position);
    }

    @Override
    public long getItemId(int position) {
        return ticketList == null ? -1 : ticketList.get(position).getId();
    }

    @SuppressWarnings("SetTextI18n")
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        ViewHolder holder;
        //ViewHolder convertView ListView 的优化写法
        if (convertView == null) {
            convertView = View.inflate(TicketsActivity.this, R.layout.layout_item, null);
            holder = new ViewHolder(convertView);
            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }
        BaseTicketBean item = ticketList.get(position);
        holder.tvTicketName.setText("票名: " + item.getTicketName());
        holder.tvTicketAmount.setText("票价: " + item.getTicketAmount() + "元");
        holder.tvTicketNumber.setText("数量: " + item.getTicketNumber() + "张");
        holder.tv_start_time.setText("发车时间: " + item.getTicketStartTime());
        holder.tv_end_time.setText("到达时间: " + item.getTicketEndTime());
        holder.tv_origin.setText("始发地: " + item.getTicketOrigin());
        holder.tv_destination.setText("目的地: " + item.getTicketDestination());
        holder.btn.setVisibility(View.VISIBLE);
        //查票和已购票 的操作类型 不显示按钮
        switch (type) {
            default:
            case 0://查票
                holder.btn.setVisibility(View.GONE);
                break;
            case 1://删票

```

```

        holder.btn.setText("删票");
        break;
    case 2://已购票
        holder.btn.setVisibility(View.GONE);
        break;
    case 3://购票
        holder.btn.setText("买票");
        break;
    case 4://退票
        holder.btn.setText("退票");
        break;
    }
    holder.btn.setTag(item);
    holder.btn.setOnClickListener(TicketsActivity.this);
    return convertView;
}
};
list_view.setAdapter(adapter);
}

```

5.3 系统界面实现（贴界面实际图）

1.用户登陆界面与注册界面



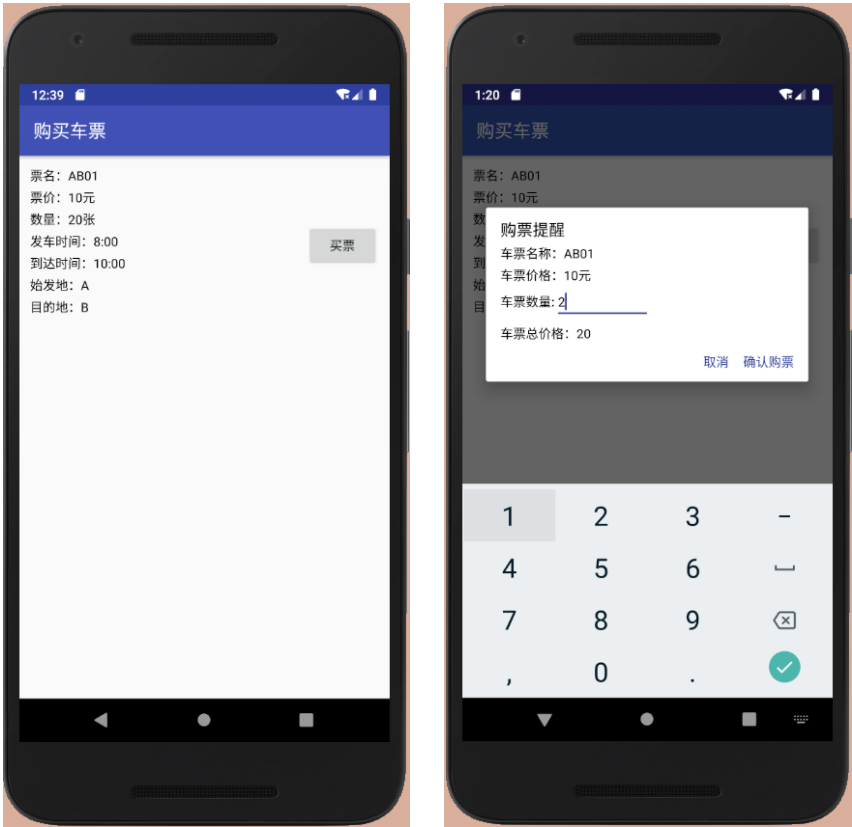
3.用户主界面与管理员主界面



5.添加车票界面



6.购买车票界面



7.退票界面



5.4 系统测试

5.4.1 测试环境

- 1.物理环境： 安卓手机
- 2.软件环境： 安卓系统运行环境，并安装 Android Studio、Sqlite 和 Mysql 等软件

5.4.2 测试项目

本次测试主要针对校车订退票系统进行系统测试，主要包括功能测试、界面测试。
具体的测试内容有：普通用户订退票模块、车票查询模块、管理员增删车票模块等。

5.4.3 登录界面测试

测试内容:背景、颜色、字体、按钮名称、提示信息的一致性等;
友好性、正确性、便捷性等

5.4.4 功能测试

测试 1：普通用户订退票

- (1) 首先用户应先登录该系统，经系统检查合法后才能进入主界面
- (2) 输入：用户名，密码
- (3) 处理过程：用户查询车票或已购车票，确定能否操作。若有余票或已购车票，显示票名、价格、时间等，点击“确定”按钮后购买或退票成功
- (4) 输出：操作是否成功

测试 2：车票查询

输入：点击“查询”按钮

处理过程：根据用户需要，查询相应的车票信息

输出：票名、价格、时间等

测试 3：管理员增删车票

输入：票名、价格、时间等

处理过程：根据校车管理需要，增加或删除车票信息

输出：更新车票信息

5.4.5 测试结论

1.登录界面测试

登录模块可以实现普通用户和管理员的登录

2.功能测试

可以近似模拟实现相应功能

5.5 系统实施应用

第 6 章 其他说明

6.1 拟实现功能说明

管理员：注册登录，查询车票，增加车票，删除车票

乘客：注册登录，查询车票，查询已购车票，买票，退票

