

学号	201825040306
成绩	

移动终端开发技术

课程设计报告

题目	火车票务系统
班级	软件技术三班
学号	201825040306
姓名	董沐函
小组成员	张泽阳、张潇文、董沐涵
指导教师	陈媛媛

2019 年 7 月 5 日

目 录

第 1 章绪论.....	1
1.1 汽车票网上订票系统的开发背景.....	2
1.2 系统设计目标.....	3
1.3 开发工具简介.....	4
1.3.1 Java.....	5
1.3.2 Anroid studio.....	6
1.4 系统软硬件要求.....	7
1.5 可行性研究.....	8
1.5.1 技术可行性.....	9
1.5.2 经济可行性.....	10
1.5.3 操作可行性.....	11
第 2 章需求分析.....	12
2.1 功能需求.....	13
2.2 性能需求.....	14
2.3 业务流程图.....	15
2.4 数据字典.....	16
第 3 章 总体设计.....	17
3.1 总体设计描述.....	18
3.3 系统功能设计.....	19
3.4.1 子模块描述.....	20
第 4 章数据库介绍.....	21
4.1 数据库介绍.....	22
第 5 章界面以及实现代码.....	23
5.1 登陆页面.....	24
5.2 注册页面.....	25
5.3 主页面.....	26
5.4 车次添加页面.....	27
第 6 章系统调试与测试.....	28
6.1 调试与测试概述.....	29
第 7 章性能分析.....	30
7.1 完整性分析.....	31
7.2 安全性分析.....	32
第 8 章结论.....	33

第 1 章 绪论

1.1 汽车票网上订票系统的开发背景

汽车站市场的管理和规范问题，是困扰我们多年的一个老问题，订票是客运业务中的一个最基本的业务，表面上看，它只是汽车站业务的一个简单的部分，但是它涉及到管理与客户服务等多方面，因此，过去传统的售票方式已经不能满足现代客运业务流量剧增的客观要求，这就要求一种全新的订票方式——网上订票，来缓解订票高峰时期的客运压力，并为用户提供方便快捷的订票服务。本次设计便是利用开发工具 MyEclipses 和 MySQL5.7 数据库共同开发的一个汽车站网上订票系统，它能方便快捷地运用在汽车站订票业务的营运之中。

1.2 系统设计目标

本系统是利用网络的方便、快捷和计算机的准确、高效实现汽车票网上订票系统。汽车票网上订票系统是通过主页访问、身份识别实现区别不同身份用户完成车票、车次的管理操作。系统有三个大的功能模块：登录功能模块，验证登录信息；管理员功能模块，实现车票和车次管理；普通用户功能模块，实现车票和个人信息管理。

1.3 开发工具简介

1.3.1 Java

Java 是一个纯的面向对象的程序设计语言，它继承了 C++ 语言面向对象技术的核心，舍弃了 C++ 语言中容易引起错误的指针（以引用取代）、运算符重载（operatoroverloading）、多重继承（以接口取代）等特性，增加了垃圾回收器功能用于回收不再被引用的对象所占据的内存空间，使得程序员不用再为内存管理而担忧。Java 不同于一般的编译执行计算机语言和解释执行计算机语言。它首先将源代码编译成二进制字节码（bytecode），然后依赖各种不同平台上的虚拟机来解释执行字节码，从而实现了“一次编译、到处执行”的跨平台特性。Sun 公司对 Java 编程语言的解释是：Java 编程语言是个简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言。

总的来说，Java 语言具有以下特点：

Java 能运行于不同的平台。Java 引进虚拟机原理，并运行于虚拟机，实现不同平台的 Java 接口之间。使用 Java 编写的程序能在世界范围内共享。Java 的数据类型与机器无关，Java 虚拟机（Java Virtual Machine）是建立在硬件和操作系统之上，实现 Java 二进制代码的解释执行功能，提供于不同平台的接口的。

Java 的编程类似 C++，学习过 C++ 的读者将很快掌握 Java 的精髓。Java 舍弃了 C++ 的指针对存储器地址的直接操作，程序运行时，内存由操作系统分配，这样可以避免病毒通过指针侵入系统。Java 对程序提供了安全管理器，防止程序的非法访问。

Java 吸取了 C++面向对象的概念，将数据封装于类中，利用类的优点，实现了程序的简洁性和便于维护性。

Java 建立在扩展 TCP/IP 网络平台上。库函数提供了用 HTTP 和 FTP 协议 传送和接受信息的方法。这使得程序员使用网络上的文件和使用本机文件一样容易。

Java 程序的基本组成单元就是类，有些类是自己编写的，有些是从类库中引入的，而类又是运行时动态装载的，这就使得 Java 可以在分部环境中动态的维护程序及分类。

1.3.2 Android Studio

Android Studio 是谷歌推出了新的 Android 开发环境，开发者可以在编写程序的同时看到自己的应用在不同尺寸屏幕中的样子。

谷歌对开发者控制台进行了改进，增加了五个新的功能，包括优化小贴士、应用翻译服务、推荐跟踪、营收曲线图、用版测试和阶段性展示。

1、优化小贴士：在主体中打开你的应用，点击小贴士，会得到这样的建议：为你的应用开发平板电脑版本。

2、应用翻译服务：允许开发者直接在开发主体中获得专业的翻译。上传你的需求，选择翻译，其会显示翻译方和价格，并在一周内发回译本。

3、推荐跟踪：允许开发者找出最有效的广告

4、营收曲线图：向开发者展示其应用营收，以国家进行划分

5、试用版测试和阶段性展示：开发者可以对应用进行测试，然后向测试用户推出，测试结果不会对外公布。当一个版本的测试结束，开发者可以向特定比例用户推出 [4] 。

Android Studio 这款开发工具被首次公布，这也是为了方便开发者基于 Android 开发。

首先解决的一个问题是多分辨率。Android 设备拥有大量不同尺寸的屏幕和分辨率，根据新的 Studio，开发者可以很方便的调整在各个分辨率设备上的应用。

同时 Studio 还解决语言问题，多语言版本（**但是没有中文版本**）、支持翻译都让开发者更适应全球开发环境。Studio 还提供收入记录功能。 [1]

最大的改变在于 Beta 测试的功能。Studio 提供了 Beta Testing，可以让开发者很方便试运行 [2] 。

1.4 系统软硬件要求

系统要求：Windows XP/vista 或者 Window 7 操作系统。

软件要求：本系统采用 Android Studio 为开发工具；Java 为开发语言；数据库采用 MySQL5.7 数据库系统。

硬件要求：硬盘 160G 以上、CPU2.4Ghz 以上、内存 2G 以上。

1.5 可行性研究

可行性分析是在用户的要求和系统调研的基础上进行的，对新系统的开发从社会、技术、经济、管理等方面进行分析，并得出新系统的开发工作可行、不可行、需要修改、追加投资、暂缓开发、分步实施等方案和结论，最后完成可行性分析

可行性分析一般可定义为：可行性分析是在建设的前期对工程项目的一种考察和

鉴定，对拟议中的项目进行全面与综合的技术、经济能力的调查，判断它是否可行。可行性研究的目的是不是解决问题，而是确定问题是否值得去解决。其最根本任务就是对以后的行动方针提出建议。一般来说，至少应该从下述三方面研究每种解法的可行性：

1.5.1 技术可行性

本系统开发的软硬件资源都已具备，前台开发工具使用 MyEclipse, 开发语言为 Java, 后台数据库使用 MySQL5.7, 其基本操作实质还是对数据库进行添加、删除、查找等操作开发环境安全，技术可靠，两者使用都相对比较熟练，因此具备技术可行性。

1.5.2 经济可行性

开发该系统所需的相关资料可以通过已存在的网上订票系统进行调查采集，所需的其他应用软件、硬件系统也易于获得。因此，开发成本较低。而引进使用本系统后，与传统方式相比，具有高效率、低成本、高质量的特点，可以节省不少人力、物力及财力。所以，从经济的角度来看，该系统可行。

1.5.3 操作可行性

本系统界面设计十分亲切，功能标题醒目，各种功能的实现简单容易，易于理解，用户只需对订票流程和业务调查了解即可，不需掌握数据库等相关知识，上手较快，系统管理人员可以简单的实现车票的销售，车次的管理，统计，退票等功能。

通过以上可行性分析，认为该系统的开发方案切实可行，可进行开发。

第 2 章需求分析

需求分析是软件设计的一个重要的环节。本阶段对订票系统的应用情况作全面调查，以确定系统目标，并对系统所需要的基础数据以及数据处理要求进行分析，从而确定用户的需求。

2.1 功能需求

本网上售票系统应该具备如下功能：

查询

分为对车次信息的查询和客户对已订车票信息的查询。要求：

- 1) 对车次的查询，可以按照发车车次进行查询。
- 2) 车次信息包括：车号、出发地、目的地、发车日期、开出时刻、票价。
- 3) 座位类型设定。
- 4) 车次信息只允许用户查询，不能修改。

添加

管理员用户可实现车次的添加、途经站点的添加和车票信息的管理。

订票、退票

普通用户可实现订票和退票，通过查询系统，客户可以根据自己的账户找到车票信息，通过订票模块实现订票，也可通过退票模块退去已购车票。

2.2 性能需求

为了保证系统能够长期、安全、稳定、可靠、高效的运行，本系统应该满足以下的性能需求。

1. 准确性和及时性

系统处理的准确性和及时性是系统的必要性能。系统应能及时而且准确的根据用户权限及所输入的信息做出响应。由于本系统的查询功能对于整个系统的功能和性能完成举足轻重。作为系统的很多数据来源，而车票的数量和时间又影响用户的决策活动，其准确性和及时性很大程度上决定了系统的成败。在系统开发过程中，必须采用一定的方法保证系统的准确性和及时性。

易用性

本系统是直接面对用户的，而用户往往对计算机并不是非常熟悉。这就要求系统能够提供良好的用户接口，易用的人机交互界面。要实现这一点，就要求系统应该尽量使用用户熟悉的术语和中文信息的界面，从而保证系统的易用性。

安全性

网上订票系统中涉及到的数据是客运公司相当重要的信息，系统要保证用户的权限，对于车次等信息用户只享有查询服务，不得更改；系统还要提供方便的手段供系统维护人员进行数据备份、日常安全管理、以及系统意外崩溃时数据的恢复等工作。同时系统还要保证对数据库进行及时更新，保证数据一致性。

2.3 业务流程图

该图为汽车票网上订票系统业务流程图：

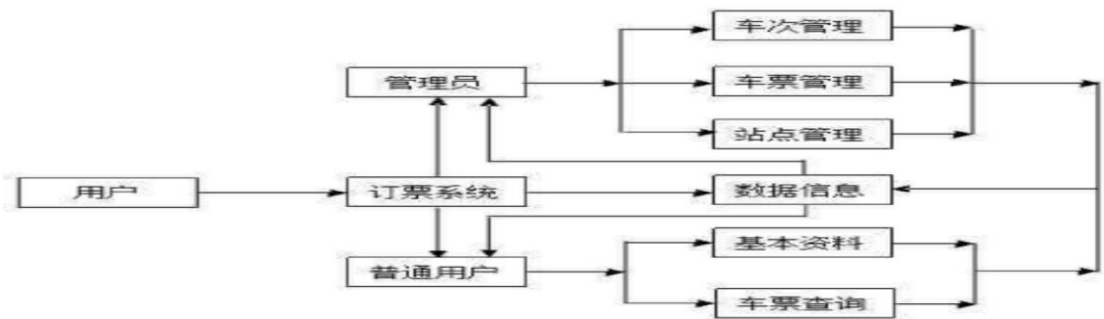


图 2.1 汽车票网上订票系统业务流程图

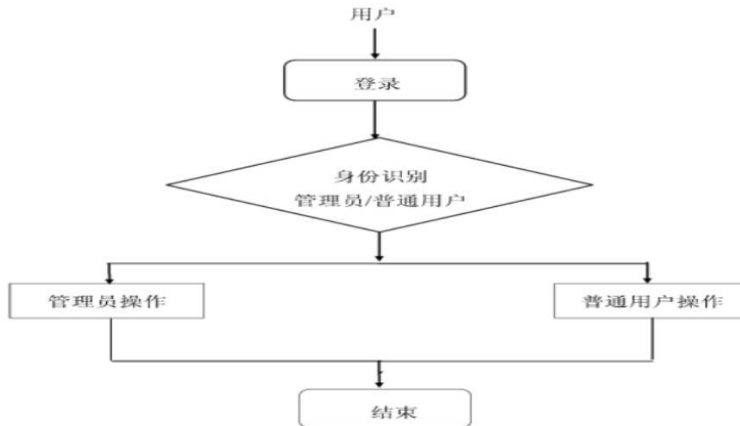


图 2.2 汽车票网上订票系统登录管理数据流图

2.32.3 数据流程图

数据流程图是以图形的方式表达在问题中信息的变换和传递过程。它把系统看成是由数据流联系的各种概念的组合，用分解及抽象手段来控制需求分析的复杂性，采用分层的数据流程图来表示一个复杂的系统。

汽车票网上订票系统系统登录数据流图：

汽车票网上订票系统车票查询数据流图：



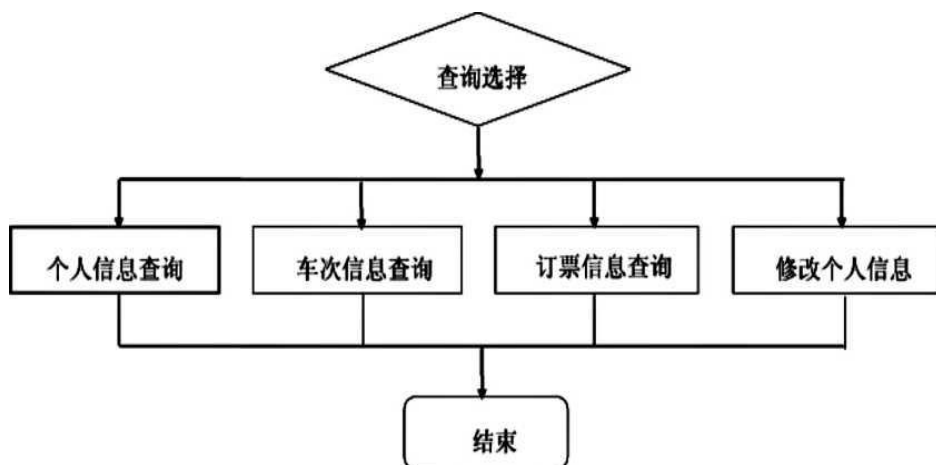


图 2.3 汽车票网上订票系统查询管理数据流图

汽车票网上订票系统车次、车票、站点管理数据流图：

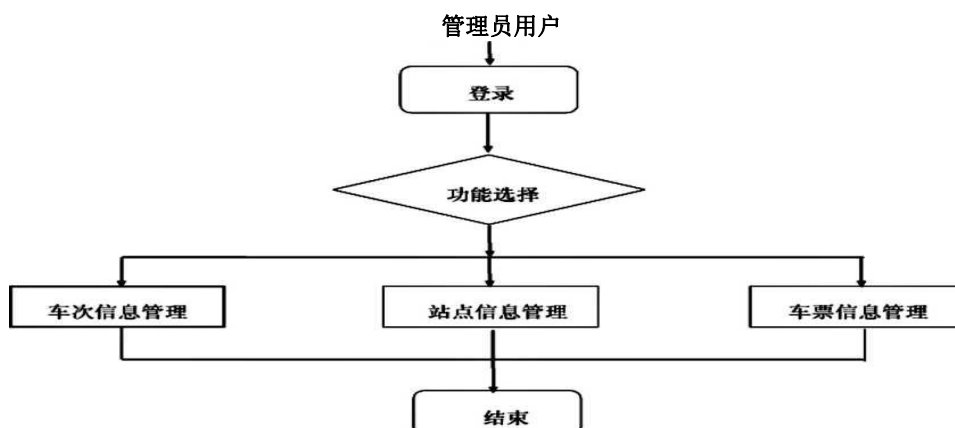


图 2.4 汽车票网上订票系统车次、车票、站点管理数据流图

2.4 数据字典

数据字典 (Datadictionary) 是一种用户可以访问的记录数据库和应用程序元数据的目录。主动数据字典是指在对数据库或应用程序结构进行修改时，其内容可以由 DBMS 自动更新的数据字典。被动数据字典是指修改时必须手工更新其内容的数据字典。

数据字典则是系统中各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要成果。数据字典通常包括数据项、数据结构、数据流、数据存储和处理过程五个部分。数据字典是关于数据的信息的集合，也就是对数据流图中包含的所有元素的定义的集合。

以下是数据流图中主要元素的数据字典卡片：

表 2.1 用户信息的定义

名字：用户信息
别名：无
描述：数据库中用来存储用户信息的关系表
定义：用户名+密码+姓名+性别+出生日期+电话+地址+类型+等级+身份证号码
位置：SQL 数据库中用户信息表

表 2.2 车票信息的定义

名字：车票信息
别名：无
描述：数据库中用来存储车票信息的关系表
定义：车次号+起始站+终点站+起始时间+终止时间+里程+余票+日期+票价
位置：SQL 数据库中车票信息表

表 2.3 订票信息的定义

名字：订票票信息
别名：无
描述：数据库中用来存储订票信息的关系表
定义：票号+用户帐号+车次+起始站+终点站+票价+开车时间+日期
位置：SQL 数据库中订票信息表

表 2.4 行车路线信息的定义

名字：行车路线信息
别名：无
描述：数据库中用来存储行车路线信息的关系表
定义：车次+站点序号+站点名称+到站时间+出站时间+里程+票价
位置：SQL 数据库中行车路线信息表

表 2.5 车次信息的定义

名字：车次信息
别名：无
描述：数据库中用来存储车次信息的关系表
定义：车次号+始发站+终点站+始发时间+到达终点站时间+历时+总里程数
位置：SQL 数据库中车次信息表

第3章 总体设计

3.1 总体设计描述

总体设计是软件开发人员根据软件需求说明书的要求，运用结构化程序设计思想，将软件自上而下逐层分解成多个软件模块，直到分解成每一个模块只具有单一的功能，能用一个或几个程序实现的树形结构为止。总体设计阶段还要定义各模块的数据传递关系，设计软件的编码方案、文件存储策略、输入输出格式，以及硬件和系统软件配置，最后编制概要设计说明书。

总体设计的内容主要包括：(1)结构化模块设计；(2)代码设计；(3)文件设计；(4)输入设计；(5)输出设计；(6)系统软硬件配置设计；(7)编制设计说明书。

总体设计过程可分为两个主要阶段：功能设计，确定车站售票系统的实现方案；结构设计，确定该软件的结构。功能设计是在需求分析的基础上进行的，这里所说的“功能”是泛指，不仅指问题定义中列出的功能，还包括软件定义时确定的任何一个独立的数据加工或处理步骤，例如添加、删除、查询和各个特定功能的算法实现等。结构设计，是将整个系统按照不同的功能和层次划分为一个个功能简单明确且相对独立的部分（模块），每个模块实现系统的一项具体功能，自顶向下，逐步细化。结构设计是确定程序由哪些模块组成，以及这些模块之间的关系。

3.3 系统功能设计

本系统分三个模块，登录功能模块，验证登录信息；管理员功能模块，实现车票和车次管理；普通用户功能模块，实现车票和个人信息管理。汽车票网上订票系统简单、操作灵活、方便、反应快速、计算准确，系统运行稳定、安全可靠，而且有良好的交互界面，让使用者可以方便、快速地掌握。

3.4.1 子模块描述

各个子模块描述如下：

登录功能模块，进入登录功能模块，选择登录身份区别是普通用户或是系统管理员。

管理员功能模块：

车次管理功能模块主要实现车次的添加、车次更新、删除等功能。

普通用户功能模块：

查询功能模块：主要完成站点信息查询、车票信息查询、订票信息查询等功能。站点信息查询是通过输入始发站和终点站显示对应的车次、车票价格、到站时间等信息；车票信息查询是通过输入车次显示对应的票价及是否还有剩余车票。白亲寺伯局。

车票管理功能模块：主要完成汽车车票的订票，支付汽车票票额、退票等功能。

第 4 章数据库设计

4. 1 数据库介绍

数据库是指自描述的完整记录的集合。它除了包含用户的源数据外，还包含关于它本身结构的描述。

数据库设计是建立数据库及其应用系统的技术，是信息系统开发和建设中的核心技术，具体说，数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求。合理的数据库结构设计可以提高数据存储的效率，保证数据的完整和一致。另外，合理的数据库结构将非常有利于程序的实现。数据库设计 (Database Design) 是指根据用户的需求，在某一具体的数据库管理系统上，设计数据库的结构和建立数据库的过程，就是规划和结构化数据库中的数据对象以及这些数据对象之间关系的过程。

数据库设计的一般性原则

第一个基本规则：对于每一个要存储其信息的实体（或事物），我们都应该给他一个自己的表；

第二个处理规则：如果你需要在一个列中存储多个值，那证明你的设计可能是有缺陷的。

简单的数据关系：一对一；一对多；多对一；多对多

在数据库的设计中：要做到数据规范化、完整性、数据冗余和规范实例化。

在整个设计过程中，我们必须按步骤认真完成。

数据库设计的过程（六个阶段）

需求分析阶段，准确了解与分析用户需求（包括数据与处理）是整个设计过程的基础，是最困难、最耗费时间的一步

概念结构设计阶段是整个数据库设计的关键，通过对用户需求进行综合、归纳与抽象，形成一个独立于具体 DBMS 的概念模型

逻辑结构设计阶段将概念结构转换为某个 DBMS 所支持的数据模型对其进行优化

数据库物理设计阶段为逻辑数据模型选取一个最适合应用环境的物理结构（包括存储结构和存取方法）

数据库实施阶段运用 DBMS 提供的数据库语言、工具及宿主语言，根据逻辑设计和物理设计的结果建立数据库，编制与调试应用程序，组织数据入库，并进行试运行

数据库运行和维护阶段数据库应用系统经过试运行后即可投入正式运行。在数据库系统运行过程中必须不断地对其进行评价、调整与修改

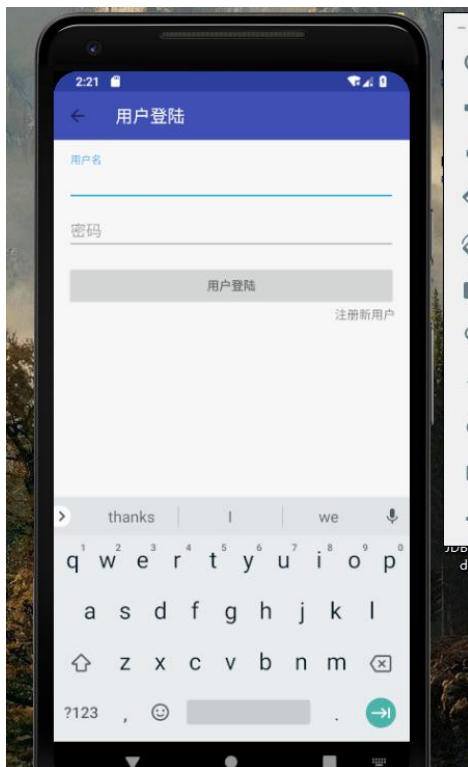
在设计过程中把数据库的设计和对数据库中数据处理的设计紧密结合起来。将这两个方面的需求分析、抽象、设计、实现在各个阶段同时进行，相互参照，相互补充，

以完善两方面的设计

数据库的设计在数据库应用系统的开发中占有很重要的地位。只有设计出合理的数据库，才能为建立在数据库上的应用提供方便。

第5章详细设计

5.1 用户登录界面



```
// Store values at the time of the login attempt.
final String userName = mEtUser.getText().toString();
String password = mEtPwd.getText().toString();

boolean cancel = false;
View focusView = null;

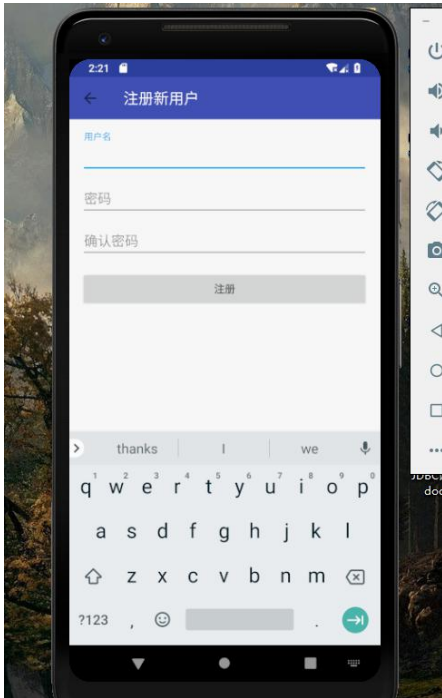
// Check for a valid password, if the user entered one.
if (TextUtils.isEmpty(password) || !isPasswordValid(password)) {
    mEtPwd.setError("密码不合法");
    focusView = mEtPwd;
    cancel = true;
}

// Check for a valid userName address.
if (TextUtils.isEmpty(userName)) {
    mEtUser.setError("用户名不能为空");
    focusView = mEtUser;
    cancel = true;
} else if (!isUserValid(userName)) {
    mEtUser.setError("用户名必须大于5位");
    focusView = mEtUser;
    cancel = true;
}

if (cancel) {
    // There was an error; don't attempt login and focus the first
    // form field with an error.
    focusView.requestFocus();
} else {
    // 登陆操作
    Cursor c = mGuestDAO.queryGuest("guest_name=? and guest_pwd=?",
        new String[]{userName, password});
    // 没有查询结果时,表示登陆失败
    if (!c.moveToNext()) {
        Snackbar.make(mLoginFormView, "用户名或密码错误", Snackbar.LENGTH_SHORT).show()
    } else {
        Snackbar.make(mLoginFormView, "登陆成功", Snackbar.LENGTH_SHORT).show();
        mHandler.postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(LoginActivity.this, GuestActivity.class);
                intent.putExtra(SignupActivity.SEND_USER_NAME, userName);
                startActivity(intent);
                finish();
            }
        }, 1000);
    }
}
```

代码实现：

5.2 注册界面：



```

// Reset errors.
mEtUser.setError(null);
mEtPwd.setError(null);

// Store values at the time of the login attempt.
final String userName = mEtUser.getText().toString();
String password = mEtPwd.getText().toString();
String verify = mEtVerify.getText().toString();

boolean cancel = false;
View focusView = null;

// Check for a valid password, if the user entered one.
if (TextUtils.isEmpty(password) || !isPasswordValid(password)) {
    mEtPwd.setError("密码不合法");
    focusView = mEtPwd;
    cancel = true;
}

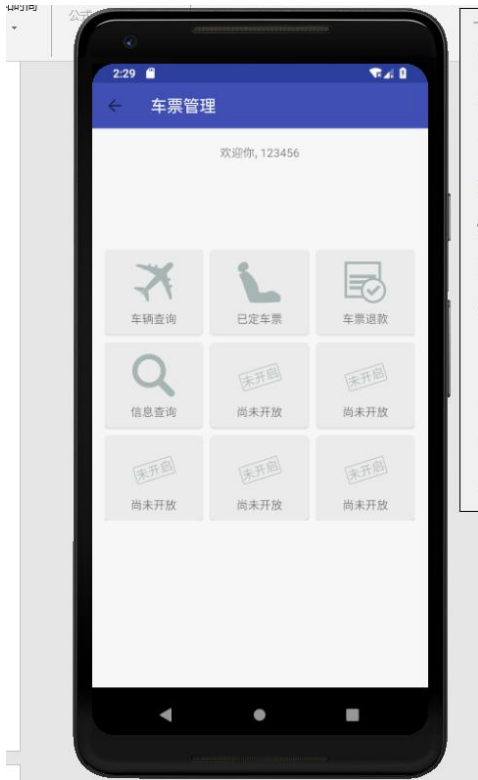
// 验证密码
if (TextUtils.isEmpty(verify) || !verifyPwd(verify, password)) {
    mEtVerify.setError("两次输入不一致");
    focusView = mEtVerify;
    cancel = true;
}

// Check for a valid userName address.
if (TextUtils.isEmpty(userName)) {
    mEtUser.setError("用户名不能为空");
    focusView = mEtUser;
    cancel = true;
} else if (!isUserValid(userName)) {
    mEtUser.setError("用户名必须大于5位");
    focusView = mEtUser;
    cancel = true;
}

if (cancel) {
    // There was an error; don't attempt login and focus the first
    // form field with an error.
    focusView.requestFocus();
} else {
    // Show a progress spinner, and kick off a background task to
    // perform the user login attempt.
    ContentValues values = new ContentValues();

```

代码实现:
5.3 主界面:



代码实现:

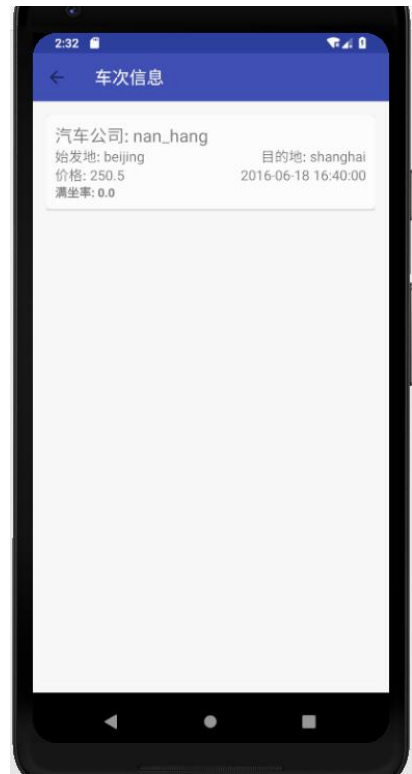
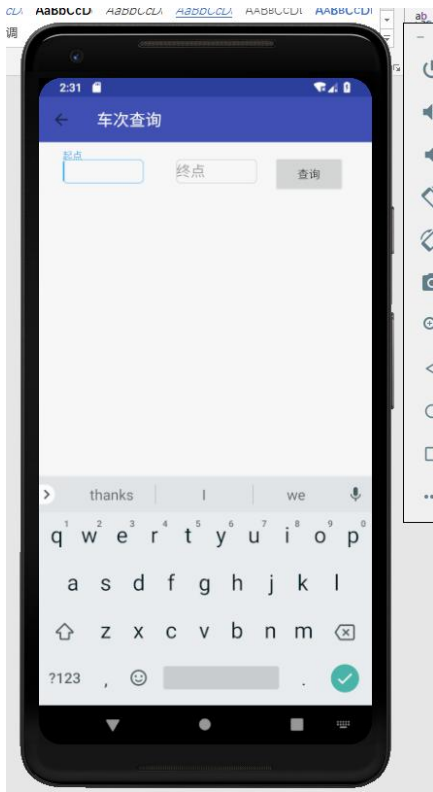
```

public class GuestActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {

    private GridView mGvMenu;
    private Toolbar mToolbar;
    private TextView mTvUser;
    private SimpleAdapter mAdapter;
    private String mUserName;
    private String[] mLabels = new String[] {
        "车辆查询", "已定车票", "车票退款",
        "信息查询", "尚未开放", "尚未开放",
        "尚未开放", "尚未开放", "尚未开放",
    };
    private int[] mImages = new int[] {
        R.drawable.menu_flight_info, R.drawable.menu_ticket, R.drawable.menu_reserve,
        R.drawable.menu_search, R.drawable.menu_dispark, R.drawable.menu_dispark,
        R.drawable.menu_dispark, R.drawable.menu_dispark, R.drawable.menu_dispark,
    };
}

```

部分功能:



代码实现:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_guest);
    mToolbar = (Toolbar) findViewById(R.id.id_toolbar_guest);
    mToolbar.setTitle("车票管理");
    setSupportActionBar(mToolbar);
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    mToolbar.setNavigationOnClickListener((v) -> { onBackPressed(); });
    mGvMenu = (GridView) findViewById(R.id.id_gv_menu);
    mTvUser = (TextView) findViewById(R.id.id_tv_username);
    initData();
    mTvUser.setText("欢迎你, " + mUserName);
    mGvMenu.setAdapter(mAdapter);
    mGvMenu.setOnItemClickListener(this);
}
}
```

5.4: 车次添加:



代码实现:

```
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
```

```
    mRoot = inflater.inflate(R.layout.fragment_add_flight, container, false);
    initView();
    return mRoot;
```

```
}
```

```
private void initView() {
```

```
    mTilCompany = (TextInputLayout) mRoot.findViewById(R.id.id_til_add_company);
    mTilStarting = (TextInputLayout) mRoot.findViewById(R.id.id_til_add_starting);
    mTilEnding = (TextInputLayout) mRoot.findViewById(R.id.id_til_add_ending);
    mTilTime = (TextInputLayout) mRoot.findViewById(R.id.id_til_add_time);
    mTilPrice = (TextInputLayout) mRoot.findViewById(R.id.id_til_add_price);
    mBtnSubmit = (Button) mRoot.findViewById(R.id.id_btn_flight_submit);
    mBtnSubmit.setOnClickListener(this);
    mTilCompany.setHint("汽车公司");
    mTilStarting.setHint("起点");
    mTilEnding.setHint("终点");
```

```

        mTilTime.setHint("时间");
        mTilPrice.setHint("价格");
    }

    @Override
    public void onClick(View v) {
        // 五个字段均非空
        String company = mTilCompany.getEditText().getText().toString();
        String starting = mTilStarting.getEditText().getText().toString();
        String ending = mTilEnding.getEditText().getText().toString();
        String price = mTilPrice.getEditText().getText().toString();
        String flightTime = mTilTime.getEditText().getText().toString();
        if (!TextUtils.isEmpty(company)
            && !TextUtils.isEmpty(starting)
            && !TextUtils.isEmpty(ending)
            && !TextUtils.isEmpty(price)
            && !TextUtils.isEmpty(flightTime)) {
            ContentValues values = new ContentValues();
            values.put("flight_company", company);
            values.put("flight_starting", starting);
            values.put("flight_ending", ending);
            values.put("flight_price", price);
            values.put("flight_time", flightTime);
            int row = mFlightDAO.insertFlight(values);
            if (row > 0) {
                Snackbar.make(mRoot, "添加成功", Snackbar.LENGTH_SHORT).show();
                getLoaderManager().restartLoader(AddTicketFragment.LOADER_ID, null,
                AddTicketFragment.mLoader);
            }
        } else {
            Snackbar.make(mRoot, "输入错误，请重新输入",

```

```
Snackbar.LENGTH_SHORT).show();  
    }  
}  
}
```

其他功能：
未开放

第 6 章系统调试与测试

6. 1 调试与测试概述

程序测试是指发现程序代码中的错误;程序调试是指从程序中找到每个问题,然后逐一解决。测试和调试是程序开发周期中必不可少的阶段,在程序开发的早期工作中,它们显示尤为重要。当对每个组件全面测试和调试之后,整个应用程序的测试和调试工作就十分简单了。

为了确保本系统能够正常运行,需要在发布之后做一次较全面的测试。现将具体操作及过程举例说明如下:

测试管理员登陆是否成功:

在主页左上角选择“管理员”选项,点击“用户登陆/注册”在“用户名”里输入一个用户名“wangwu”,“密码”框里输入登录密码“0”。填写完成点击“登录”按钮,将会出现管理员功能页面,即管理员已经登陆成功了。

测试新用户注册是否成功:

在主页左上角选择“用户登陆/注册”按钮,系统将跳转到用户注册界面,按照提示输入相关信息,点击“注册”按钮,系统会将用户输入的信息保存到数据库的 User 表中,注册完成系统将会跳转到首页,即新用户注册成功。

测试普通用户登录是否成功:

在主页左上角选择“普通用户登陆/注册”选项，在“用户名”里输入一个用户名“zhangsan”（为刚刚注册的帐号），“密码”（框里输入登录密码“123”为刚刚注册的密码）。填写完成点击“登录”按钮，将会出现普通用户功能页面，即普通用户已经登陆成功了。

测试管理员车次管理功能是否成功：

以管理员身份登录后，选择“车次信息”功能选项，单击“新增车次”按钮，系统会跳转到新增车次页面，根据提示输入新增车次的相关信息，点击“添加”按钮，系统会将管理员输入的信息保存到数据库的 Tickets 表中，添加完成系统将会跳转到管理员管理功能页面，即车次添加成功。单击“修改”按钮，系统会

钮，系统会将管理员输入的信息保存到数据库的 Tickets 表中，修改完成系统将会跳转到管理员管理功能页面，点击“查询”按钮，系统将会显示修改后车次的相关信息，即车次修改成功。点击“删除”按钮，系统会将数据库 Tickets 表中对应车次的信息删除，再次点击“查询”按钮，系统显示的信息将没有以删除车次的信息，即车次删除成功。

(5) 测试管理员站点管理功能是否成功：

以管理员身份登录后，选择“查看站点”功能选项，单击“新增站点”，系统跳转到新增途径站页面，根据提示输入新增站点的相关信息，点击“添加”按钮，系统会将管理员输入的信息保存到数据库的 Ticketst 表中，添加完成系统将会跳转到管理员管理功能页面，即站点添加成功。单击“修改”按钮，系统会跳转到修改站点信息页面，根据提示输入修改后车次的相关信息，点击“修改”按钮，系统会将管理员输入的信息保存到数据库的 Ticketst 表中，修改完成系统将会跳转到管理员管理功能页面，单击“查询”按钮，系统将会显示修改后站点的相关信息，即站点修改成功。点击“删除”按钮，系统会将数据库 Ticketst 表中对应的站点信息删除，再次点击“查询”按钮，系统显示的信息将没有以删除站点的信息，即站点删除成功。

(6) 测试普通用户查询功能是否成功：

以普通用户身份登录后，选择“预订车票”选项，输入车次，点击“查询”按钮系统会将对应车次的信息显示在页面上，即查询功能成功。

(7) 测试普通用户订票功能是否成功：

普通用户登录后，查询相应车次，点击“预订”按钮，系统将会跳转到订票页面，根据提示输入订票的相关信息，点击提交后系统将订票信息保存到数据库中的“order”表中，点击“查看订单”，即用户订票功能成功。

第 7 章性能分析

系统的性能是软件制作的一个比较关键的问题，能否实现相关的功能要求，达到预期的目的是性能分析的一个比较关键的问题。

对本系统的性能分析可以从系统的完整性和安全性方面进行考虑。

此系统在功能上基本达到了开题报告中的要求：能实现对信息的查询与维护功能。在性能上，表与表之间的约束不够好，可能会出现一些不必要信息，或者是冗余信息，但是相对来说也有自己的优点，页面之间的切换比较容易，使用起来方便。

7.1 完整性分析

数据完整性(Data Integrity):数据库中的表都是相互有一定联系的。在这方面，系统的表与表之间的联系良好，没有出现数据的冗余现象。因此系统的完整性良好。

在 Windows 10 操作系统下，采用 Java 语言作为开发工具，MySQL 后台数据库。

7.2 安全性分析

安全性分析可以归类到以下的几个方面：身份识别与验证、保密性、数据完整性与不可篡改性。

身份验证：目的是确认访问者的身份。访问者可能是人或者程序，识别与验证就是验证它们提交的身份识别标志。身份验证是权限控制的基础和必要条件。

保密性：目的是保护敏感信息。当敏感信息被保存在本地时，必须使用权限控制或加密技术，使之得以保护；当敏感信息在网络上传输时，应该被加密。

经过分析，本系统安全性良好，基本上符合要求。

第 8 章结论

经过好长时间的制作和思考我们组三个人尽心尽力的去完成完善我们做的 app 案例，在此间我们遇到的很多的问题，不过我们一步的去改正，去克服这问题，通狗这次的 app 案例开发，我们也学到很多东西，也很感谢老师讲授我们那么多的知识。我们在这次开发中也提升了我们自己独立的能力，同时再合作我们也体验到了乐趣，原本以为开发很难，但是这次的项目加深了我的记忆也加深了我对以前学习知识的巩固，所以说这次期末项目加深我对安卓 app 开发的兴趣，我也很感谢这次的项目让我成长。