**架构设计文档**

软工1803-U201817044-王粟鹏

软工1803-U201817033-李佳欣

软工1803-U201817030-刘文文

软工1801-U201816968-刘欣媛

软工1803-U201817039-周澍

**工作分配**

王粟鹏：

系统需求分析，分层架构设计，逻辑视图，接口设计，安全设计，运维设计，部分模型设计

李佳欣：

质量属性分析，过程视图，物理视图，主要模型设计

刘文文：

限制条件，数据库设计，数据结构设计，部分模型设计

刘欣媛：

质量属性分析，过程视图，物理视图，部分模型设计

周澍：

大部分系统需求分析，用例视图，部署视图，部分模型设计

|  |  |  |
| --- | --- | --- |
| 项目名称  Project Name | | 密级  Confidentiality Level |
| 60321火车购票系统 | | 仅供收件方查阅 |
| 项目编号  Project ID | 版本  Version | 文档编号  Document Code |
| 1 | 1.0 | 1\_SD\_001 |

60321火车购票系统

架构设计说明文档

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制： | 全组人员 | 日期： |  |
| 评审： |  | 日期： |  |
| 批准： |  | 日期： |  |

Revision Record

修订记录

| 日期 | 修订版本 | CR/ Defect号 | 修改章节 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- | --- |
| 2020-6-19 | 初稿 |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[一、简介 6](#_Toc54990896)

[1.1、目的 6](#_Toc54990897)

[1.2、范围 6](#_Toc54990898)

[1.2.1、名称 6](#_Toc54990899)

[1.2.2、应用 6](#_Toc54990900)

[1.2.1、背景 7](#_Toc54990901)

[二、需求分析 7](#_Toc54990902)

[2.1、功能性需求分析 7](#_Toc54990903)

[**2.1.1、用户账户** 7](#_Toc54990904)

[**2.1.1.1、用户注册** 7](#_Toc54990905)

[**2.1.1.2、用户登录** 8](#_Toc54990906)

[**2.1.1.3、用户注销** 9](#_Toc54990907)

[**2.1.2、票务查询及购买** 9](#_Toc54990908)

[**2.1.2.1、票务查询** 9](#_Toc54990909)

[**2.1.2.2、购票** 10](#_Toc54990910)

[**2.1.3、用户票务操作** 11](#_Toc54990911)

[**2.1.3.1、查询未完成订单** 11](#_Toc54990912)

[**2.1.3.2、查询已支付订单** 12](#_Toc54990913)

[**2.1.3.3、查询候补订单** 12](#_Toc54990914)

[**2.1.3.4、查询本人车票** 13](#_Toc54990915)

[**2.1.3.5、改签** 13](#_Toc54990916)

[**2.1.3.6、变更到站** 14](#_Toc54990917)

[**2.1.3.7、退票** 14](#_Toc54990918)

[**2.1.3.8、查看购票帮助** 14](#_Toc54990919)

[**2.1.4、后台管理** 15](#_Toc54990920)

[**2.1.4.1、管理员登录** 15](#_Toc54990921)

[**2.1.4.2、管理员设置** 15](#_Toc54990922)

[**2.1.4.3、管理员修改密码** 16](#_Toc54990923)

[**2.1.4.4、增加/修改车次信息** 16](#_Toc54990924)

[**2.1.4.5、修改车次信息** 16](#_Toc54990925)

[**2.1.4.6、查询用户信息** 17](#_Toc54990926)

[2.2、质量属性 17](#_Toc54990927)

[2.3、限制条件 20](#_Toc54990928)

[三、系统场景 22](#_Toc54990929)

[四、软件架构设计 23](#_Toc54990930)

[3.1、系统分层架构设计 23](#_Toc54990931)

[3.1.1、技术结构 23](#_Toc54990932)

[3.1.2、功能结构 24](#_Toc54990933)

[3.2、用例视图 24](#_Toc54990934)

[3.3、逻辑视图 26](#_Toc54990935)

[3.4、过程视图 27](#_Toc54990936)

[3.5、部署视图 30](#_Toc54990937)

[3.6、物理视图 31](#_Toc54990938)

[五、关键技术设计 31](#_Toc54990939)

[5.1、接口设计 31](#_Toc54990940)

[5.1.1、支付请求接口 31](#_Toc54990941)

[5.1.2、用户信息接口 32](#_Toc54990942)

[5.1.3、票务信息查询接口 32](#_Toc54990943)

[5.2、数据结构 33](#_Toc54990944)

[5.3、数据库设计 36](#_Toc54990945)

[5.4、安全架构设计 41](#_Toc54990946)

[5.5、运维架构设计 42](#_Toc54990947)

[六、其他 42](#_Toc54990948)

[6.1、优先级 42](#_Toc54990949)

# 一、简介

## 1.1、目的

该文档的读者主要为项目的开发人员、项目经理以及文档审核人员。

设计该文档的目的在于以下的几个方面：

**1、 有利于信息持有者之间的沟通：**体系结构是一个系统的个高层表示，可以作为不同信息持有者之间信息讨论的焦点，有利于开发人员对整个系统有更好的把握，更好的进行开发和交流。

**2、 便于系统分析：**在项目开发的早期阶段给出系统的确切结构，实际上就是对系统的分析过程。体系设计决策对系统额能否满足关键性需求（如系统的功能、可靠性、可维持行等）有着重要的影响。

**3、 有利于后续项目开发：**使用add的方式对系统进行设计，以质量属性为指导，为后续的项目开发提供了模板，既有利于开发人员对系统整体的把握，又为开发人员确定规范了开发结构，开发人员在这个体系上开发项目，既有利于后续的管理，又有利于最后的整合。

## 1.2、范围

### 1.2.1、名称

60321购票系统。

### 1.2.2、应用

主要应用领域为火车购票领域，采用线上平台的方式。

该系统旨在为用户提供一个便捷的网络购票平台。在该平台上，使用者可以查询票务信息，以及具体的车次信息。在登陆并实名认证之后，才能具有购买车票资格。

### 1.2.1、背景

步入21世纪之后，我国的经济迅速发展，旅行出游也成为了当代人的常态，随着我国铁路系统的迅速发展，高铁也成为了游客出行的重要方式。

在网络不发达的年代，人们常常要到火车站购票，这一购票方式的弊端有很多，很明显的就是不利于提前了解车次时间安排计划，而且还会发生到了车站却没有票的尴尬情况。由于票务种类的繁多，客户信息的量大复杂，所以在其管理也存在较大困难，特别是早期单用人力和纸张进行管理，导致信息的不全面和错误率高，加之存储介质的约束，难以长期有效的管理。

而网络购票平台的出现就很好的解决了这些问题。用户可以提前在网络购票系统上了解车务信息，提前购票，方便了出行。在当前市场上已经有了很多成熟的网络购票软件，比如12306和携程，但他们也存在着很多缺点，影响着用户的体验。我们设计此火车购票系统，旨在集成市场上其他火车购票系统的优点，并对他们的缺点进行改进，最终为用户提供更加便捷的购票以及出行体验。

# 二、需求分析

## 2.1、功能性需求分析

**2.1.1、用户账户**

**2.1.1.1、用户注册**

实现用户在该网站账号的注册。

**用户需求：**

用户要输入用户名、密码、验证码、手机号、实名认证信息来创建账户。

**系统需求：**

如果用户名或密码没有填写，系统会以将输入框变红的方式提示用户没有填写这一栏。

在用户输入用户名的过程中，系统会实时检测当前用户名有没有被注册，如果被注册了就会在用户名右侧以红色字体提示用户名已被注册。与用户点击注册，系统再判断用户使用的用户名有没有被注册，并将结果返回给用户的方式相比，这种方式可以提高用户的注册效率，提高易用性。

在输入用户密码的一栏，系统会实时检测用户输入的密码格式。密码需要满足长度大于八位少于16位，且同时包含阿拉伯数字、大小写英文字母的要求。如果系统检测到当前的密码格式不符合要求，则会在输入密码这一栏的右侧以红色字体显示“您的密码位数少于8位” “您的密码中没有同时包含阿拉伯数字、英文字母大小写”，让用户能及时修正密码。对密码设置的高要求可以提高账户的安全性。

用户在手机号一栏输入其手机号后，系统会检测手机号位数是否符合规范。如果不符合规范则会用醒目的红色字体提示用户检查输入的手机号并修改；如果符合规范，系统会进一步检测这一手机号有无注册过账号，如果已经注册过账号，则用红字提示用户该手机号已经注册过账户，如需要继续注册则更换手机号；如果系统检测该手机号没有注册过账户，则将输入手机号一栏右侧的获得验证码的按钮变为可用。用户点击“获得验证码”按钮，系统会将随机生成的6位验证码以短信的形式发到用户输入的手机上，并等待用户输入收到的验证码。验证码应在用户点击按钮后的一分钟内发送到用户的手机上，且验证码的有效时长为10分钟。

系统在用户进行实名认证时提供多个选项。包括居民身份证、港澳居民来内地通行证、台湾居民来大陆通行证、外国公民在中国的护照。让所有有买票需求且能购票的人都能注册账号。同时系统会检测输入的证件号是否与所选的证件类型、用户姓名相匹配。也会检测该证件是否注册过账号，一个证件只能注册一个账号。

实名认证，一个手机号、一个证件号只能注册一个账户可以有效减少恶意抢票、倒卖车票的情况，让用户能以相对合适的价格在所需的时刻买到想要买的票。

在注册完成后，系统要求用户将账号与微信或者支付宝绑定。用户在之后的购票中可以用绑定的微信或者支付宝账号进行付款，也能用绑定的微信或者支付宝账号进行登录。

**2.1.1.2、用户登录**

实现用户在该系统的个人登录。

**用户需求：**

要提供多种不同的登录方式，不能仅限于用户名和密码的方式，还需要提供忘记密码的功能，在用户忘记密码时，帮助用户找回密码或者修改密码。

**系统需求：**

提供多种登录方式：用户名+密码、微信登陆、支付宝登录、手机登录，采用后三者方式进行登录时需要先对账号进行绑定。

如果用户没有输入用户名或密码就点击登录，则会出现提示信息，将没有写的输入框变红，提醒用户填写对应信息。

如果用户名错误，会返回给用户“用户名不存在”的信息，并建议用户检查用户名是否输入正确。

如果密码错误，会返回给用户“密码错误”的信息，并建议用户再次输入。

**2.1.1.3、用户注销**

用户可以注销掉自己的账户。

**用户需求：**

清除当前的账户信息，删除该用户的数据，在删除用户时，要进行相应的验证，确保是用户本人的操作。

**系统需求：**

在用户发起“删除用户”操作时，要向用户绑定的第二方应用发送验证信息，确保是用户本人发起的删除操作，在第二方应用没有应答之前，不能继续进行删除操作。给第二方应用发送的信息应该包括是否同意删除用户，如果用户不同意删除操作，要进一步询问是否强制下线发起删除操作的设备、是否修改密码，来防止其他人恶意操作用户账号。

**2.1.2、票务查询及购买**

**2.1.2.1、票务查询**

用户选择相应的信息查询车票，如果有符合用户要求的票务信息则显示，没有符合要求的信息则给用户反馈。

**用户需求：**

用户首先选择起点站、终点站、出行日期，非必须的选项有出行方式类型（高铁/动车/火车）、是否购买学生票。点击“查询车票”按钮之后，会显示符合要求的所有直达车次，用户可以选择车次的排列方式，可以按照耗时的长短、发车时间的早晚来排序，可以在车次显示界面修改乘车时间。对于某一个确定的车次而言，要显示该车次的编号、发车和到站时间、全程时长、每一种座位的票价、每一种座位是否有票。点击车票之后会显示该车次的具体车务信息，具体的经停站和每一站的时间。

如果没有对应的直达车票，应该提供可以选择的换乘方案。

**系统需求：**

在查询车票的时候，系统会保存用户的查询历史。默认的起点站和终点站分别为上一次购票的终点站和起点站，默认的出行时间为第二天。

如果没有用户搜索的站台，则应该给用户提供反馈信息，告诉用户查询不到车站。

如果用户查询的车票已经没有席位了，则会提示用户其所要购买的车票已经售完。并询问用户是否要委托系统抢票。抢票功能会帮用户实时检测是否有其他用户退订了该用户所要购买的车票，若有，则自动帮用户购买这一张车票，并将车票信息以短信的方式发送至与用户账号绑定的手机上。如果该班次发车前两个小时仍然没有合适的票，系统则会发送短信到与用户账号绑定的手机上，告知用户仍未抢到票，用户可以在售票系统上继续抢票，则系统帮助用户继续抢票，如果在之后的时间里抢到票，则发短信通知用户并告知车票信息，如果在发车前半个小时依旧没有抢到票，则以短信的形式告知用户这一结果，并将用户预留的购票费用退还到用户账号上。用户也可以在购票系统上取消抢票。取消抢票后，系统将用户预留的购票费用退回到用户账号上。

**2.1.2.2、购票**

用户查询到车务信息之后可以购买相应车次的车票。

**用户需求：**

查询到车票之后，点击相应的车次进入购票界面。可以选择座位的类型、添加乘客和受让人、添加随行儿童，如果具有学生认证可以选择是否购买学生票，还需要提供选座服务。点击提交订单之后，会再次显示待购买车票的具体信息，以及乘客信息、金额明细，提供购买返程车票的便捷通道，用户可以选择支付或者取消订单。支付时可以选择多种不同的支付方式。用户可以在账户设置里更改默认支付方式与各种支付方式的使用顺序。

**系统需求：**

在购票时为乘客提供座位类型。如果是高铁和动车，则有“二等”、“一等”、“商务”三种座位类型；如果是火车，则有“软卧”、“硬卧”、“硬座”和“无座”四种类型。

在购票的时候必须选择至少选择一名乘客，选中之后系统会检测该乘客的信息是否验证，如果没有验证或者没有选择乘客，则会弹出相应提醒。

如果至少有一排座位全为空时，会为用户提供选座服务，从左到右依次为“窗”“A”“B”“C”“过道”“D”“F”，当用户选择一个座位时，系统会优先给该乘客分配相应位置的座位，如果无法满足选座需求，系统将会自动分配座位。

用户在提交订单之后，系统应该进一步显示购置车票的具体信息以及乘客的基本信息。

用户提交订单之后系统会进行倒计时，用户有三十分钟的时间支付订单，如果超过时间却未支付订单，系统会自动给用户取消订单。用户也可以自己取消订单，点击取消订单之后，系统要弹出提醒，确定用户是否确认取消订单，如果一天之内取消三次订单，则不能购置其他的车票。在支付时，要能够为用户提供多种不同的支付方式。

支付成功会弹出提醒，并显示具体的车票信息。

**2.1.3、用户票务操作**

**2.1.3.1、查询未完成订单**

用户可以查询下单但没有支付的车票。

**用户需求：**

用户可以查询未完成订单，要能够看到该票务的具体信息，并可以执行“取消订单”、“支付订单”功能。

**系统需求：**

用户点击查询未完成订单选项，系统判断用户有无登录账号，若无，则弹出登录界面；若用户已登录，则将界面跳转到未完成订单显示界面。

如果有未完成订单，则在界面上显示该订单的基本信息如出发地与目的地、出发时间与到达时间、乘客信息、座位信息等，并为用户提供取消订单、支付订单、购买铁路乘意险、购买返程、查看退改说明等选项，并显示提醒信息，用户需要在30分钟内处理未完成订单，超过30分钟，订单自动取消，同时界面为用户提供剩余处理时间的提示信息；如果没有未完成订单，界面显示订单未完成。

两种界面都提供返回上一级界面的选项。在返回界面时，系统弹出提示框，提示用户是否要放弃支付，并且提示用户同一用户一天只能取消三次订单。

**2.1.3.2、查询已支付订单**

**用户需求：**

用户点击“查询已支付订单”选项可以查询已经支付的订单。

**系统需求：**

用户点击查询已支付订单选项后，系统判断用户有无登录账号，若无，则弹出登录界面；若用户已登录，则将界面跳转到已支付订单显示界面。

在已支付订单界面，有两个可选显示的子界面，分别显示未出行订单与历史订单。

未出行订单显示用该账号支付的车票的信息，车票信息包括出发地与目的地、出发时间与到达时间、乘客信息、座位信息等。车票信息按距离出行时刻的远近排序，距离当前时间近的排在前面。

历史订单显示用户已经支付并完成出行的车票，并显示车票的基本信息，包括出发地与目的地、出发时间与到达时间、乘客信息、·座位信息等。车票按完成订单的时刻距离当前时间的远近排序，距离当前时刻近的车票排在前面。

显示已支付订单界面提供退出选项。车票以无限下拉的形式显示，不采用分页的方式。

**2.1.3.3、查询候补订单**

**用户需求：**

用户可以查询正处于“抢票”状态的车票订单。

**系统需求：**

用户点击查询候补订单选项后，系统判断用户有无登录账号，若无，则弹出登录界面；若用户已登录，则将界面跳转到候补订单显示界面。

候补订单界面显示用户之前想要购买但当时已经售罄的车票。显示车票的信息包括出发地与目的地、出发时间与到站时间。同时在每张车票上显示当前该车票是否可以购买（当有其他用户退订或改签会有车票多余）。车票显示顺序按照出行时间距离当前时间的远近排序，距离当前时间近的车票显示在前面。

若没有候补订单，则显示“当前没有候补订单”。

界面左上方提供返回上一级菜单的选项。

**2.1.3.4、查询本人车票**

**用户需求：**

用户可以查询与用户本人相关的车票。

**系统需求：**

用户点击查询本人车票选项，系统判断用户有无登录账号，若无，则弹出登录界面；若用户已登录，则将界面跳转到本人车票显示界面。

在本人车票界面系统显示本人未出行的车票及其基本信息包括出发地与目的地、出发时间与到达时间、乘客信息、座位信息等，并按照发车时间距离当前时间的远近显示，距离当前时间近的先显示。

若没有本人的车票，则显示“没有本人车票”。

在界面的左上方提供返回上一级界面的选项。

**2.1.3.5、改签**

**用户需求：**

用户可以对已经购置的车票进行改签。

不更改终点站、始发站和乘客信息的情况下，可以修改乘车的日期和车次，依然要提供选座服务，可以改变座位的类型，而且在车票改签的时候，不需要二次付款。

**系统需求：**

在改签界面为用户提供改签说明。

固定乘客的始发站、终点站以及乘客信息，允许乘客修改乘车日期和车次，在用户选择好改签之后的车票之后，允许用户更改座位类型，允许用户重新选择座位，如果席位不足以满足选座需求，则会自动为用户分配座位。用户提交订单之后，会计算两次车票的差额，采取多退少补的原则。

已经变更到站的车票无法进行改签。

**2.1.3.6、变更到站**

**用户需求：**

用户可以变更已经购买车票的到站。

不更改始发站和乘客信息的情况下，可以修改乘车的终点站、日期和车次，依然要提供选座服务，可以改变座位的类型。

**系统需求：**

在更改到站界面为用户提供更改到站说明。

固定乘客的始发站以及乘客信息，允许乘客修改终点站、乘车日期和车次，在用户选择好变更到站的车票之后，允许用户更改座位类型，允许用户重新选择座位，如果席位不足以满足选座需求，则会自动为用户分配座位。用户提交订单之后，会计算两次车票的差额，采取多退少补的原则。

**2.1.3.7、退票**

用户可以变更已经购买车票的到站。

**用户需求：**

对于已经购买的车票，可以取消订单，并按照退订时间与发车时间的间隔长短支付一定的违约金，违约金最多不超过总金额的30%，而且退款要按照原路径及时退到用户的账户。发车前24小时退票不退还车票费用。

**系统需求：**

用户在点击退票之后，弹出提醒，是否确定退票，如果用户确定退票则进入退票界面，显示要退票的具体信息、乘车人信息以及温馨提示，点击确定退票之后，系统会立即执行退票操作，并将退款退还到用户账户。

**2.1.3.8、查看购票帮助**

**用户需求：**

用户点击“购票帮助”选项可以查看详细的用户购票指南。

**系统需求：**

系统在用户的购票系统的主界面、购票界面都为用户提供“购票帮助”的选项。当用户点击这个选项后，系统跳转至展示用户购票帮助的界面。

界面显示有关用户购票、改签、退订的流程提示与注意事项。并将重要的信息标红，让用户能够快速了解到这些重要的信息，避免不必要的损失。

界面左上方提供返回前一个界面的选项。

**2.1.4、后台管理**

**2.1.4.1、管理员登录**

售票系统的管理员登录该售票系统的后台。

**用户需求：**

管理员在登录的时候要输入账号、密码以及验证码。

管理员在忘记密码后可以在登录界面找回密码。

**系统需求：**

系统对管理员输入的账号、密码、验证码进行判断。若管理员输入了不存在的账号，则会在输入账号一栏的右侧用红色字体提示“您输入的账号不存在”。如果输入的密码与账号不匹配，则会在输入账号一栏的右侧用红色字体提示“用户名账号不匹配”。如果验证码输入错误，则会在输入验证码一栏的右侧用红色字体提示“验证码输入错误”，并重置验证码。

处于安全性考虑，当一个账号段时间内连续5次输入账号密码错误，系统会暂时锁定这个账号，使其在一个小时内无法登录，并将这个异常登录的信息发送给售票系统最高权限的管理员。

系统在帮管理员找回密码时，需向管理员绑定的手机号发送短信验证码，管理员在修改密码前需要先输入验证码。验证码的有效期为30分钟。该验证码每个小时内向一个手机号只能发送两次。

管理员的密码每发生一次改变，系统都应在管理员相关的数据库里更新相关信息。

**2.1.4.2、管理员设置**

**用户需求：**

在管理员设置界面，有最高权限的管理员可以管理其他管理员的信息。包括新增管理员，删除管理员，修改管理员信息如管理员的密码、管理员的权限等。其中，管理员数量不应超过10个。

**系统需求：**

在对其他管理员对信息进行操作前，系统会对当前管理员对权限进行判断。只有拥有最高权限的管理员才能在管理员设置界面对其他管理员对信息进行修改。

**2.1.4.3、管理员修改密码**

**用户需求：**

管理员可以修改自己的密码。

**系统需求：**

在修改密码时需要管理员先输入原先的密码，再输入新的密码，并重复输入一次新密码。密码需要包括英文的大小写和阿拉伯数字，且长度为8至16个字符。当点击确定按钮后，系统会对输入的密码进行判断，如果原密码正确，新密码符合密码设定的规范且两次新密码的输入相符，则弹出提示框，显示“密码修改正确”并返回上一级菜单；如果有任意一项条件不满足，则弹出提示框并显示该次输入不符合的条件，并刷新修改密码的界面。

**2.1.4.4、增加/修改车次信息**

**用户需求：**

管理员可以新增车次信息。车次信息包括列车号；各种类型座位（二等座、一等座、商务座）的数量；起点站站名和出发时间、终点站站名和到站时间以及每个经停站站名、到站时间、车站停留时间；每两个站之间的不同种类座位的票价。其中，到站时间和站台停留时间精确到分钟。

**系统需求：**

管理员每新增一项车次信息，系统就会更新相应的数据库信息。

**2.1.4.5、修改车次信息**

**用户需求：**

管理员可以修改车次信息。管理员可以选中某一列还未出发的列车并修改其信息。

可修改的信息包括列车号；各种类型座位（二等座、一等座、商务座）的数量；起点站站名和出发时间、终点站站名和到站时间以及每个经停站站名、到站时间、车站停留时间；每两个站之间的不同种类座位的票价。其中，到站时间和站台停留时间精确到分钟。

**系统需求：**

管理员每新增一项车次信息，系统就会更新相应的数据库信息。

**2.1.4.6、查询用户信息**

**用户需求：**

管理员可以查询用户信息。管理员可以根据用户的订单号、用户名称查询某一用户的信息。管理员可以查到的用户信息包括用户名、用户的注册时间、用户登录售票系统的时间与总次数、用户的订单信息以及用户身份。其中，用户的订单信息包括订单号，订单对应的车次、座位种类（二等座、一等座、商务座）、购票数量、购票的乘客姓名及其证件号；用户身份包括成人、儿童、军人、学生。如果用户不止一份订单，则根据订单创建的时间距离当前时刻的远近显示订单，距离当前时刻近的订单在头部显示。

**系统需求：**

系统应记录管理员查询用户信息的每条操作记录。这在一定程度上可以保护用户的隐私。

## 2.2、质量属性

**可用性：**

1. 系统达到较高可用性标准，年度停机时间小于等于8.8小时，达到可用性级别99.9%。
2. 在正常操作下，系统数据库出现错误时，在一小时内完成自动更换备份数据库。
3. 在正常操作下，系统出现错误或异常但不影响操作结果时，系统给出警告信息并保持运行。
4. 在用户购票时进行一致性检查，也就是在并发时需要对数据加锁，保证数据一致性，从而避免出现用户下单后却没票了的情况。

**性能：**

1. 在一千万用户试图同时登陆购票系统即本系统的可承受最高峰时期内，系统保证正常运行，购票请求被处理的平均响应时间低于7秒，票务查询请求被处理的平均响应时间低于2秒，超出15秒未做出响应时系统弹出警告。
2. 在用户同时访问数低于五百万时，系统保证正常运行，购票请求被处理的平均响应时间低于3秒，票务查询请求被处理的平均响应时间低于1秒，超出10秒未做出响应时系统弹出警告。
3. 用户完成购票操作后，余票量显示在0.1秒内完成更新。
4. 车次信息在数据库中更新后，在0.1秒内完成界面显示更新。
5. 用户申请退款后，全部的退款操作在5秒内完成执行。

**安全性：**

1. 用户个人信息数据，票卡数据和用户使用数据全部加密存放于数据库中，杜绝非法用户试图绕过服务器直接连接到数据库服务器端口，防止100%的信息盗取。
2. 提供冗余的系统数据存储和数据备份，对数据库进行在线数据备份，当数据库遭到破坏用户数据丢失时，可在3分钟内恢复全部数据和系统的正常使用。
3. 当非法票卡进入本系统票务流程时，实现100%的非法票务拦截。
4. 用户进行在线支付时，系统可以阻止100%的病毒攻击，防止用户交易信息泄露和资金被非法转移。
5. 用户角色识别，当管理者登录系统时使用多重验证，100%保证角色识别正确防止系统信息泄露，当用户进行不可逆操作时，进行多方验证防止他人恶意操作，多方验证不通过时提供修改密码、强制下线和冻结账户等操作保证用户账号安全性。
6. 系统使用限流阀机制和过期机制，以及辅助检测手段防止Dos攻击，当抢票插件小程序或票务黄牛进行抢票攻击时系统可检测出95%的抢票行为并拦截。
7. 屏蔽某IP短时间内大量无意义访问，以防正常用户无法使用。
8. 限制单个用户单日订单取消次数防止恶意占票行为，保证正常用户余票量。
9. 用户进行退票操作时，退款金额返还给原支付账户且不可更改退款路径，在退款过程中拦截100%病毒攻击防止资金非法转移。

**易用性：**

1. 当用户进行错误操作后，可在运行时1秒内取消错误操作使影响最低。
2. 所有具有基本电脑操作常识的人，可在3分钟内根据界面设计掌握本系统的基本使用方法，熟练用户可以使用快捷键进行操作。
3. 用户操作易于记忆，用户在使用本系统2周后，可以熟悉记起其基本操作步骤，不需要重新学习。
4. 系统设计保证交互效率，使新用户平均可以在10分钟内完成选票和购票流程，熟练用户平均可以在3分钟内完成选票和购票流程。
5. 系统错误率为，新用户正常进行系统操作时，平均出错率低于20%，熟练用户正常进行系统操作时，平均出错率低于2%。
6. 用户进行账号注册和账号登录等输入操作时，若出现输入错误或输入内容不当情况，系统利用界面文字提醒用户及时更正，使用户每次出错平均更正时间低于2秒。
7. 当用户需要查看帮助或进行交互时，可读的排版和易用的导航可以为用户提供其所需要的信息。例如在帮助界面等将重要信息标红。
8. 用在查询车票的时候，系统会有默认的起点站和终点站，其分别为上一次购票的终点站和起点站，默认的出行时间为第二天。
9. 对于全部用户操作系统在2秒内提供文字或声音反馈，并实现合适的界面跳转。

**可修改性：**

1. 由于人们对售票系统的使用激增，数据爆炸产生，开发人员需要对数据库方面进行更新迭代，以满足需求，同时避免影响到未来的收益。设计时，降低模块间的依赖性并使之满足“接口隔离”原则；维护时，查找架构中所需要修改的位置，只对数据库相关的实现类进行修改，不会影响其他功能，在维护期间启用备用数据库，不影响用户使用。
2. 用户希望调整系统界面，在维护时，开发人员只修改界面及相关配置文件，不影响其他部分，能在5小时内完成修改并发布更新。
3. 用户产生在高铁上点外卖的需求，开发人员需要增加订餐功能，只需实现相关功能并将新功能与相关功能关联，对无关功能没有副作用。
4. 随着移动支付的普及，在窗口现金支付的选项不再被需要，开发人员需要删除该功能，只需在维护时删除相关代码，修改不影响到其他文件，可在两小时内完成。
5. 保证系统的开放性，只要符合系统规范，可以简单的加入或减少系统模块，可以通过软件的修补完成系统升级和更新。

**可移植性：**

1. 在新的平台上运行的需求出现，开发人员需要修改代码，使之与新环境适配，只需要修改与目标环境相关的适配层代码，不用修改与目标环境无关的主体层代码，最小程度上降低了修改量，可以在一周内完成。

**可测试性：**

1. 开发人员在设计时统一做一个操作面板，其为一个可操作性整个系统的独立模块，并提供两种形式，一种为命令形式执行操作并获得基本输出，另一种为GUI形式，在界面上展现输入输出。
2. 单元测试人员在组件完成时执行单元测试，系统内的各组件有控制行为的接口并且其输出可观察，例如测试买票是否成功时可以在操作面板的数据输出观察到余票数量的变化等。可以在3小时内测试85%的路径。
3. 系统用户在使用所交付的系统时，对完整的应用进行测试。系统可以对出现的错误或故障进行详细提示，使测试结果易于判断，有可分析性、可获得性。且出现故障的概率小于0.1%。
4. 提供事务日志使过去的系统状态和变量可见，在运行中可查询。

**可维护性：**

1. 设计人员在设计时尽可能地将不同模块分离，降低后合度，从而使后面维护人员更容易地对单独的模块进行修改，替换和升级。
2. 设计人员在设计时撰写详细的设计和帮助文档，以便后续的维护人员更好的理解系统并进行维护。

## 2.3、限制条件

**技术限制：**

1. 售票系统的实现必须运行在Windows系统上；
2. 需要实现系统框架，对使用语言形式不限，可以是C、java等；
3. 小组成员对java更为熟悉，所以使用java进行开发；
4. 使用Eclipse进行系统程序编写；
5. 使用MySQL数据库进行数据的存储和读取；

**法律限制：**

1. 在实现实名制的同时，不能泄露用户的个人信息，不能非法使用用户信息；
2. 所有软件都选用正版；
3. 所有技术资料由开发一方保管；
4. 合同制定确定违约责任；
5. 所销售的火车票必须是全国联网售票系统数据库里可查询到班次的火车票，不能非法收取票价以外的手续费，不能私自圈票抬价出售等；

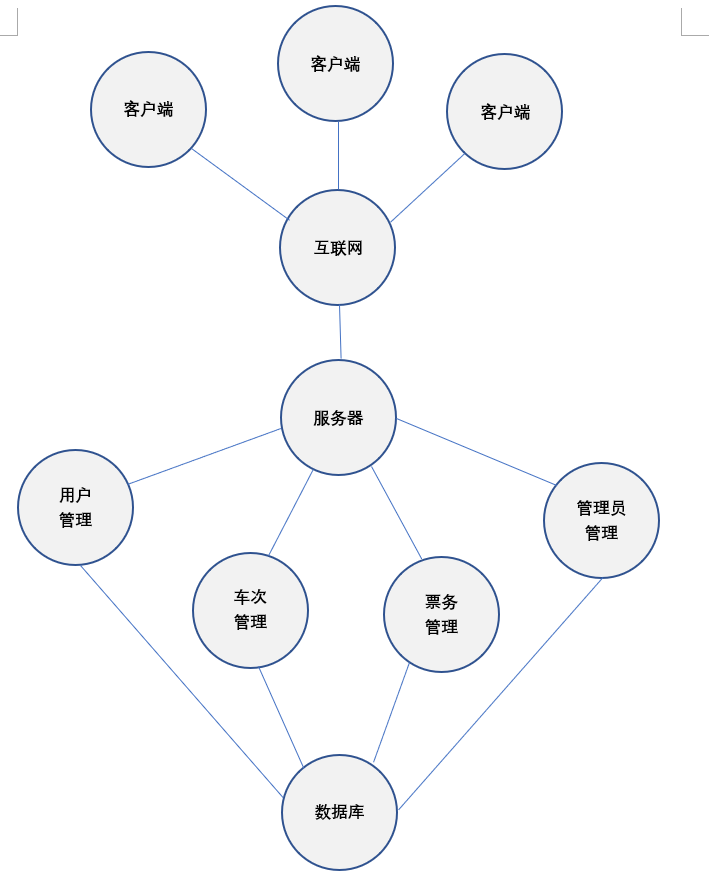
**社会限制：**

1. 必须实现实名制；
2. 实现实名制同时要保证用户的信息安全；
3. 售票系统出售火车票的票价，应当按照全国联网售票系统的中心数据库提供的实时价格执行；

**商业限制：**

1. 完成时间要求：需要在本课程结束的一周完成售票系统的实现和文档的完成并且提交；
2. 成本、预算：需要花费小组成员的大量时间精力，在合理调配小组成员的精力和时间的基础上实现系统的完成和完善；
3. 预计的系统生命周期的长短：预计系统周期为4年，最长可达5年；
4. 目标市场：面向所有人，所有需要购买火车票的人都可以在该售票系统上购买；
5. 推出计划：首先实现一个基础的系统，在此基础上不断完善优化，或是增加新的功能；
6. 集成：不与老系统集成，实现一个全新的系统。

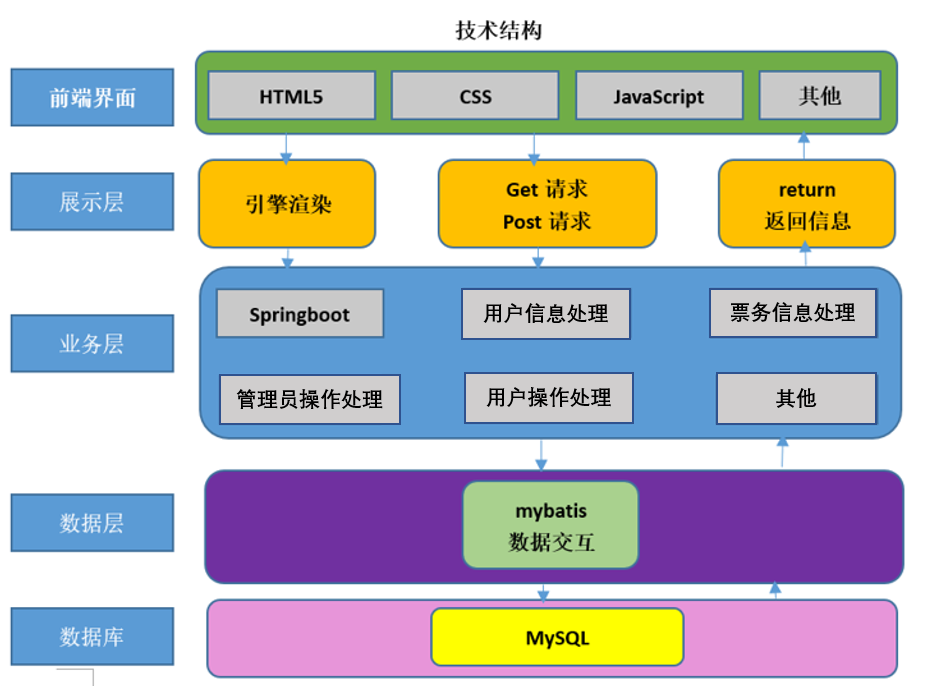
# 三、系统场景



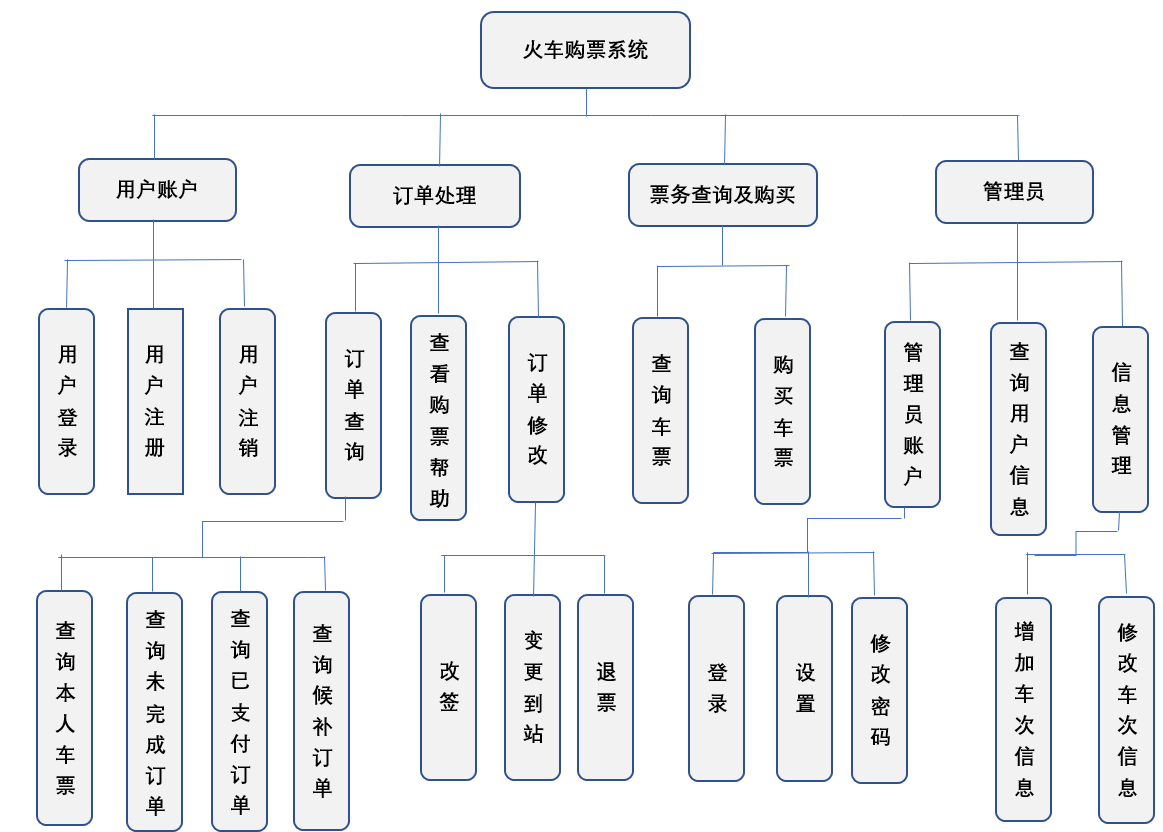
# 四、软件架构设计

## 3.1、系统分层架构设计

### 3.1.1、技术结构

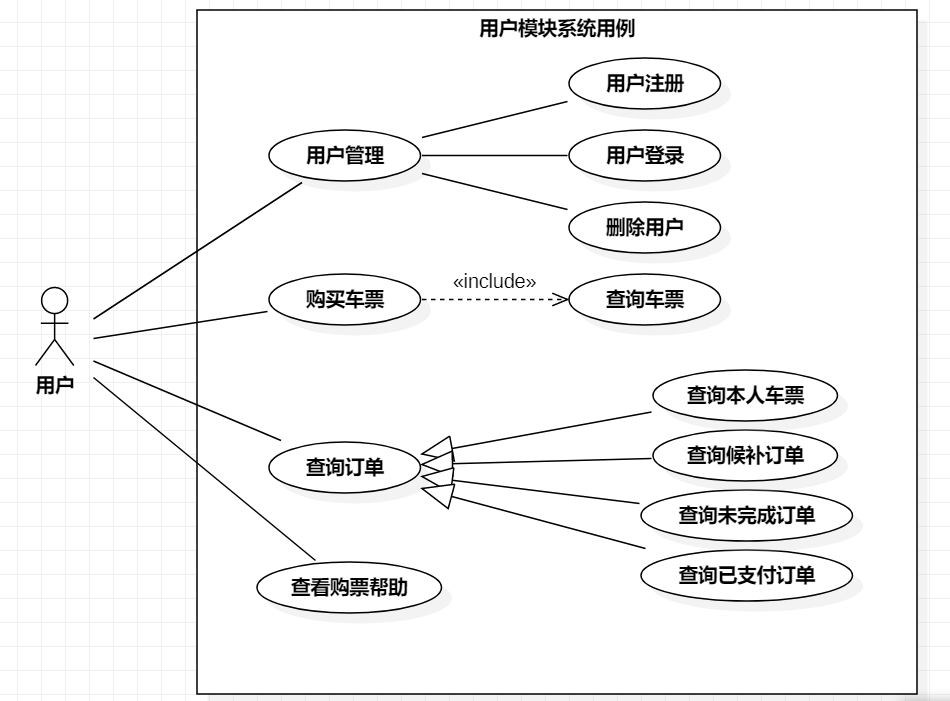


### 3.1.2、功能结构

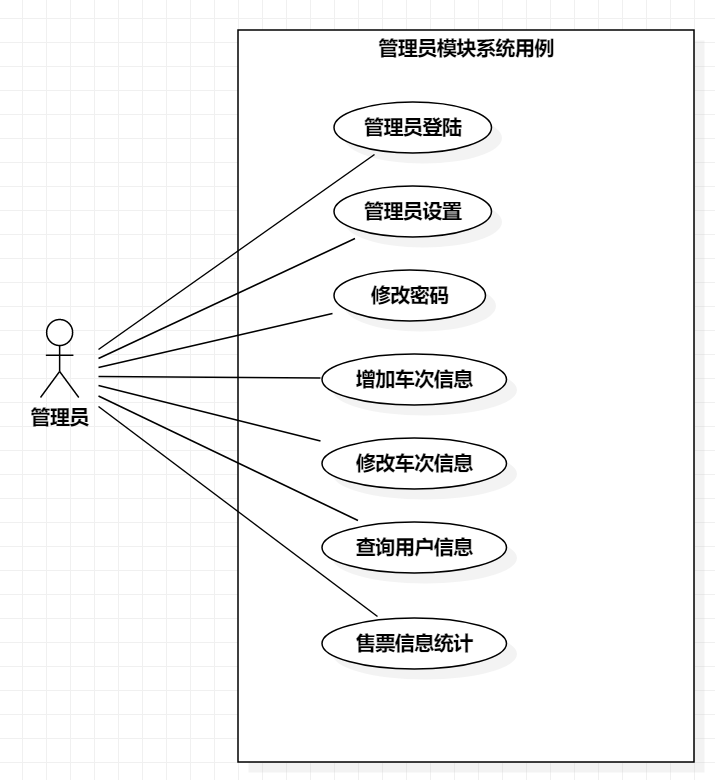


## 3.2、用例视图

**用户模块用例图：**

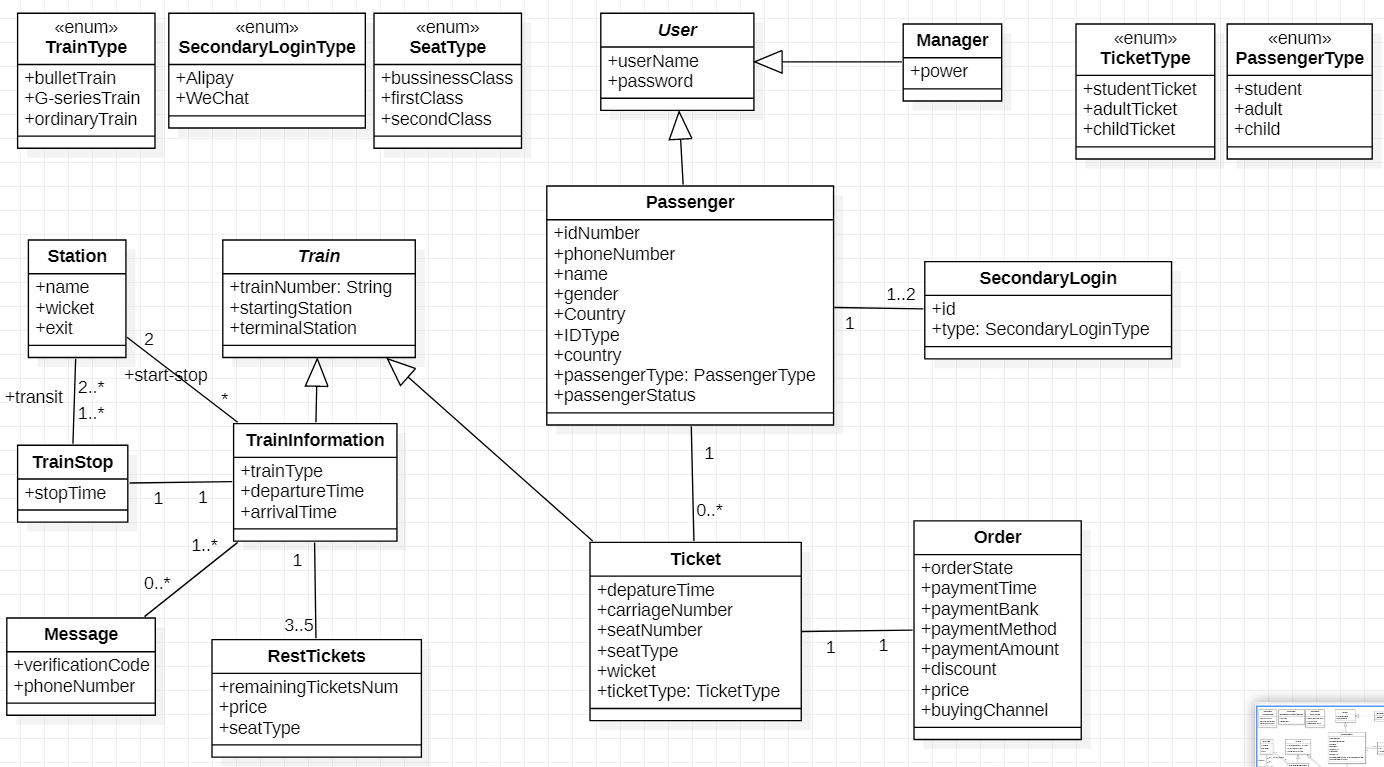


**管理员模块视图：**

****

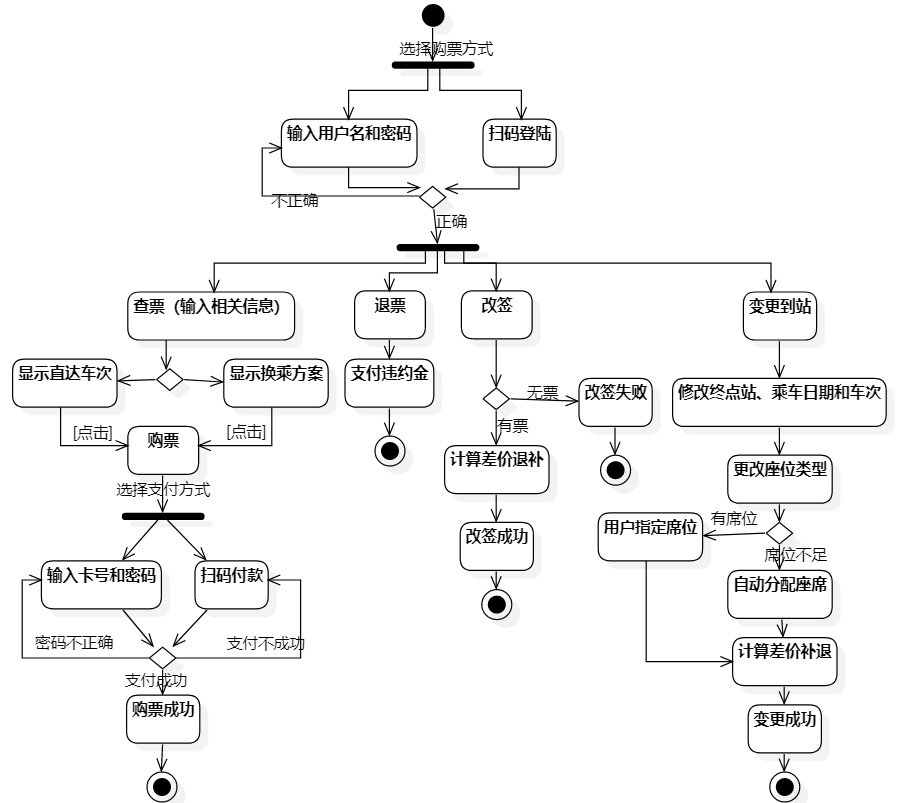
## 3.3、逻辑视图

**类图：**

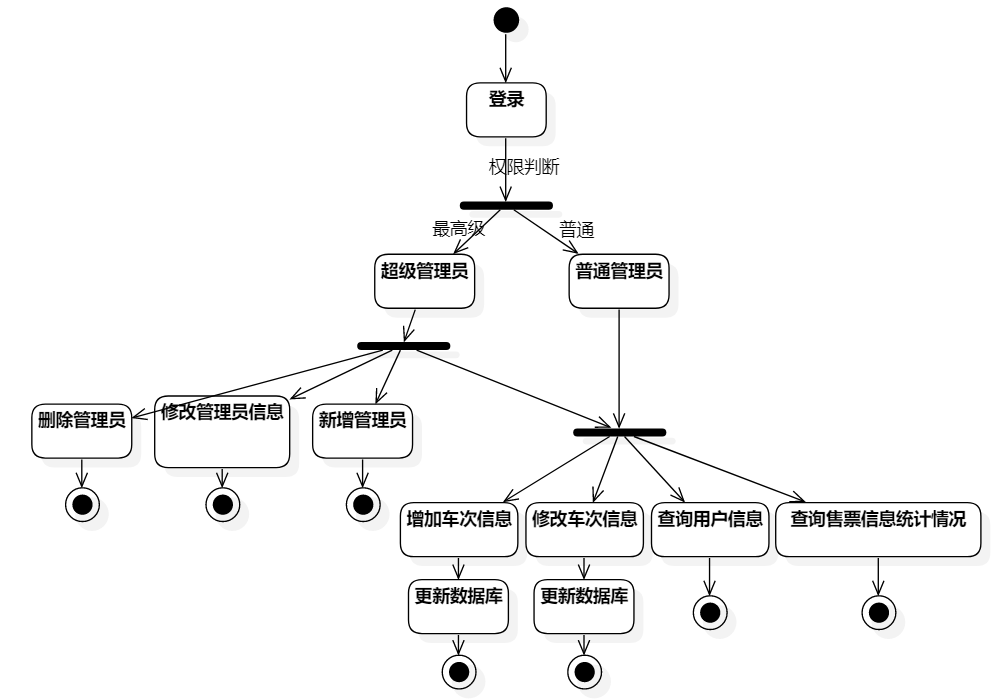
****

## 3.4、过程视图

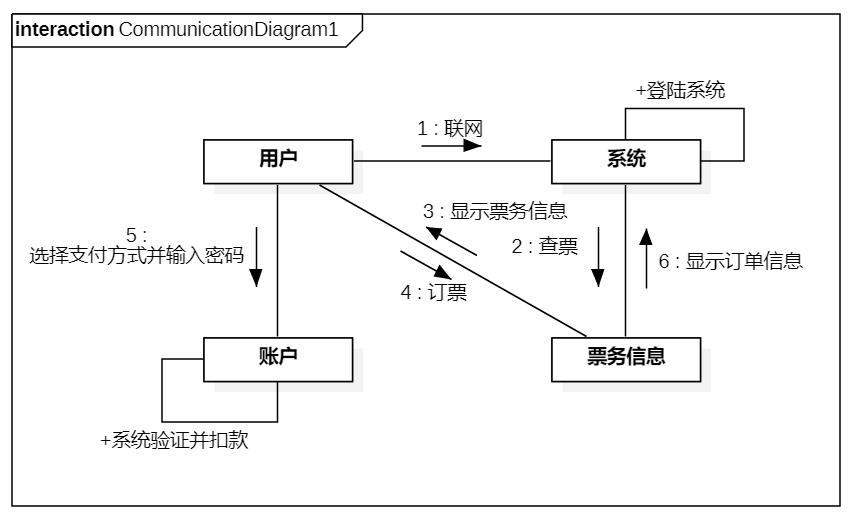
**用户票务操作活动图：**



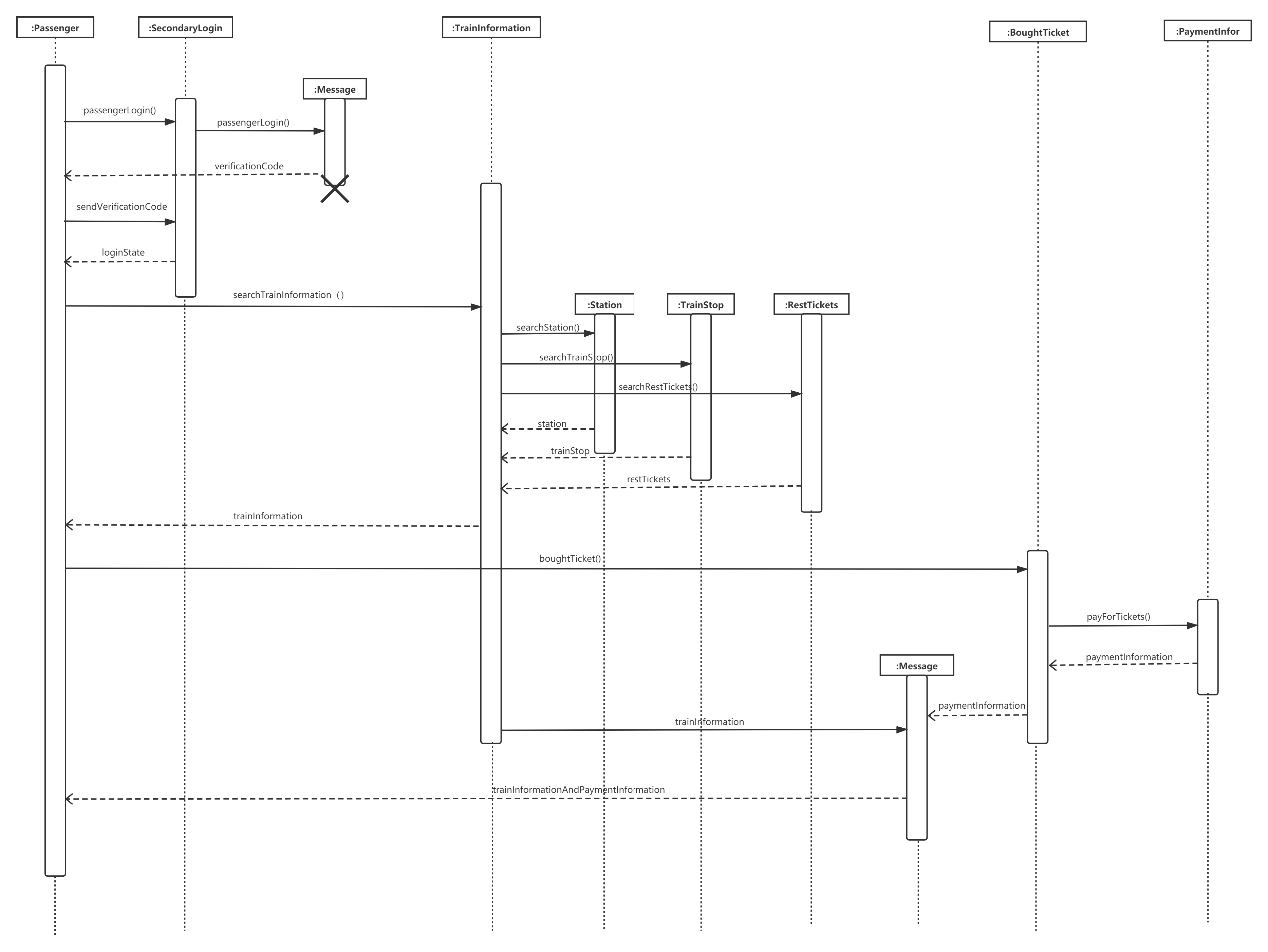
**后台管理活动图：**

****

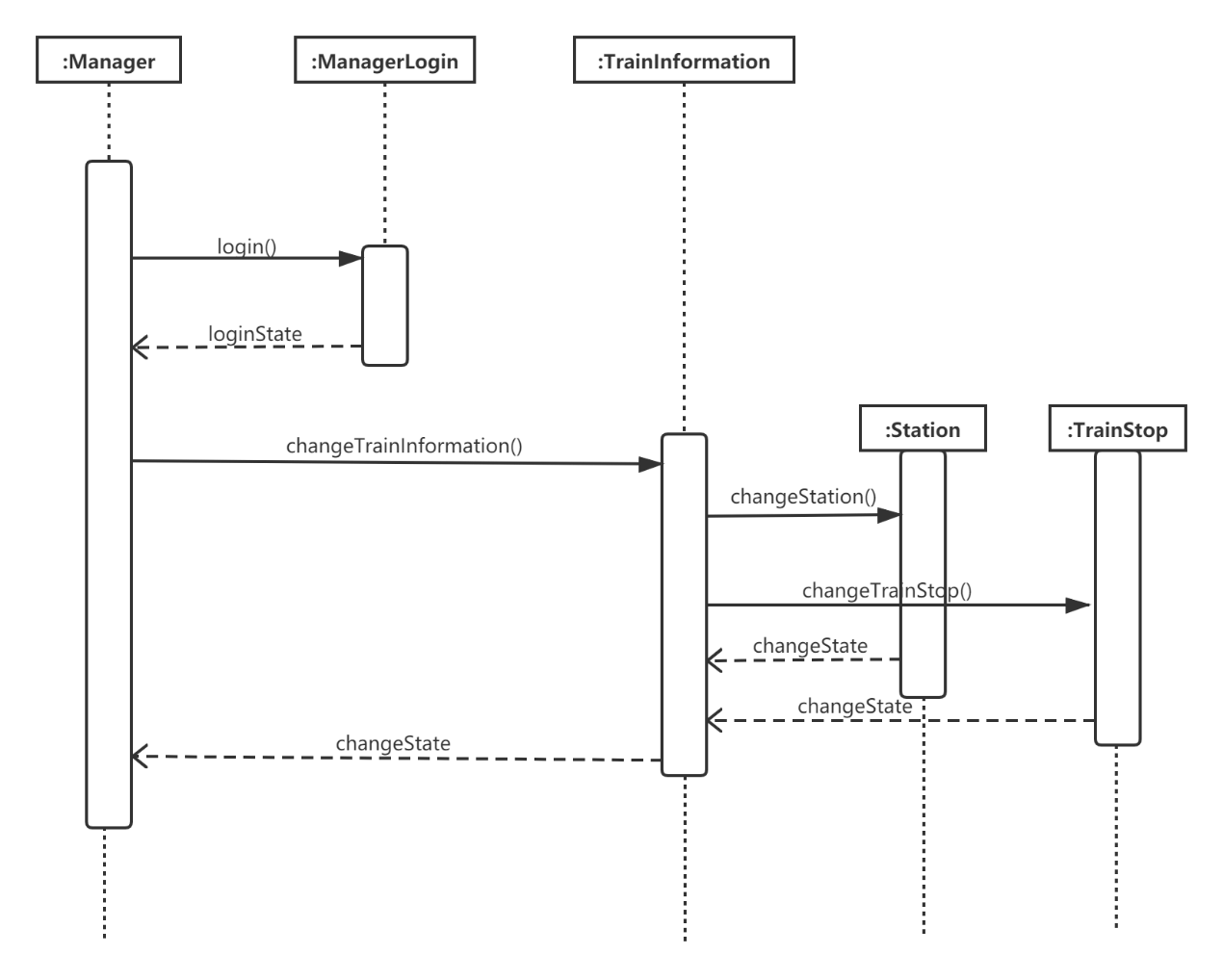
**Communication Diagram：**

****

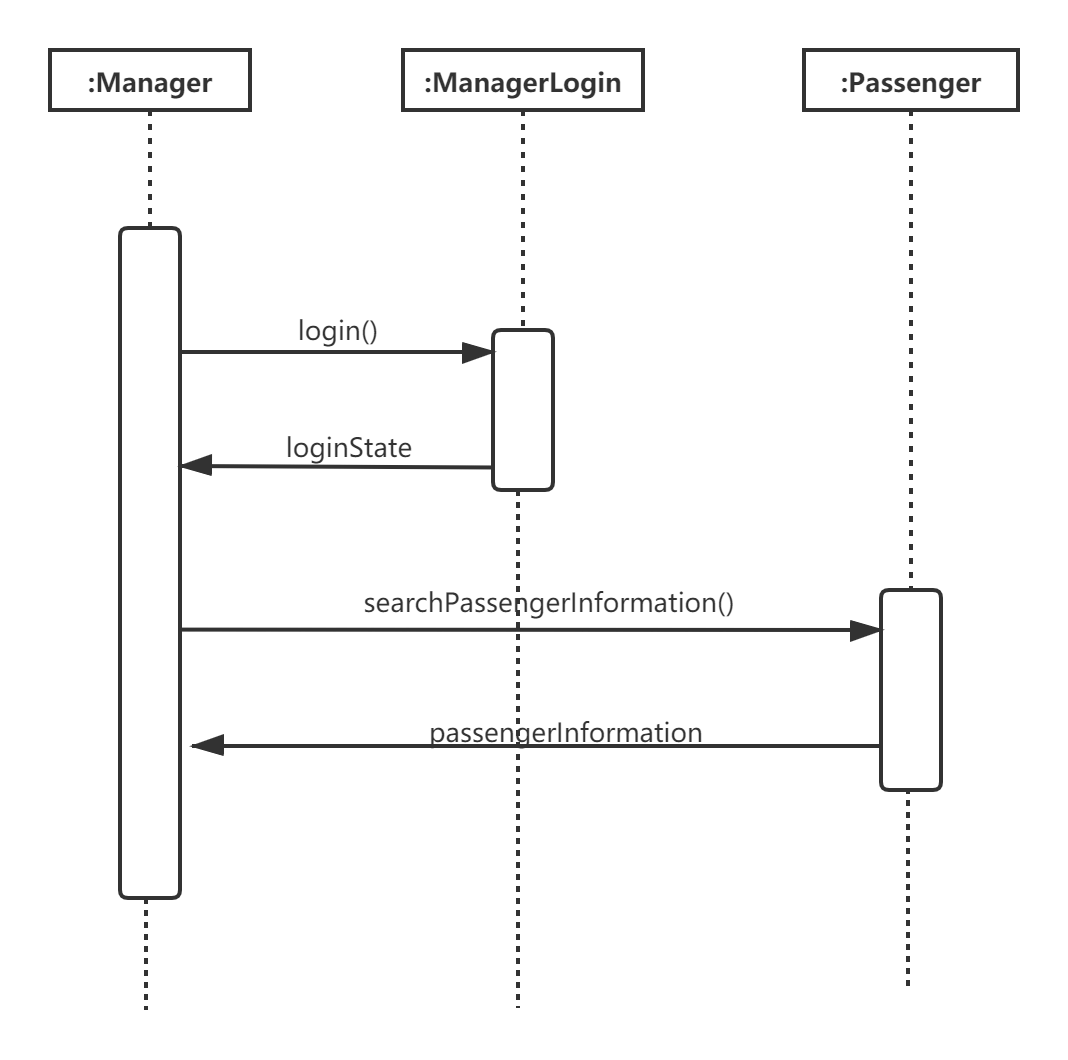
**时序图1：**

****

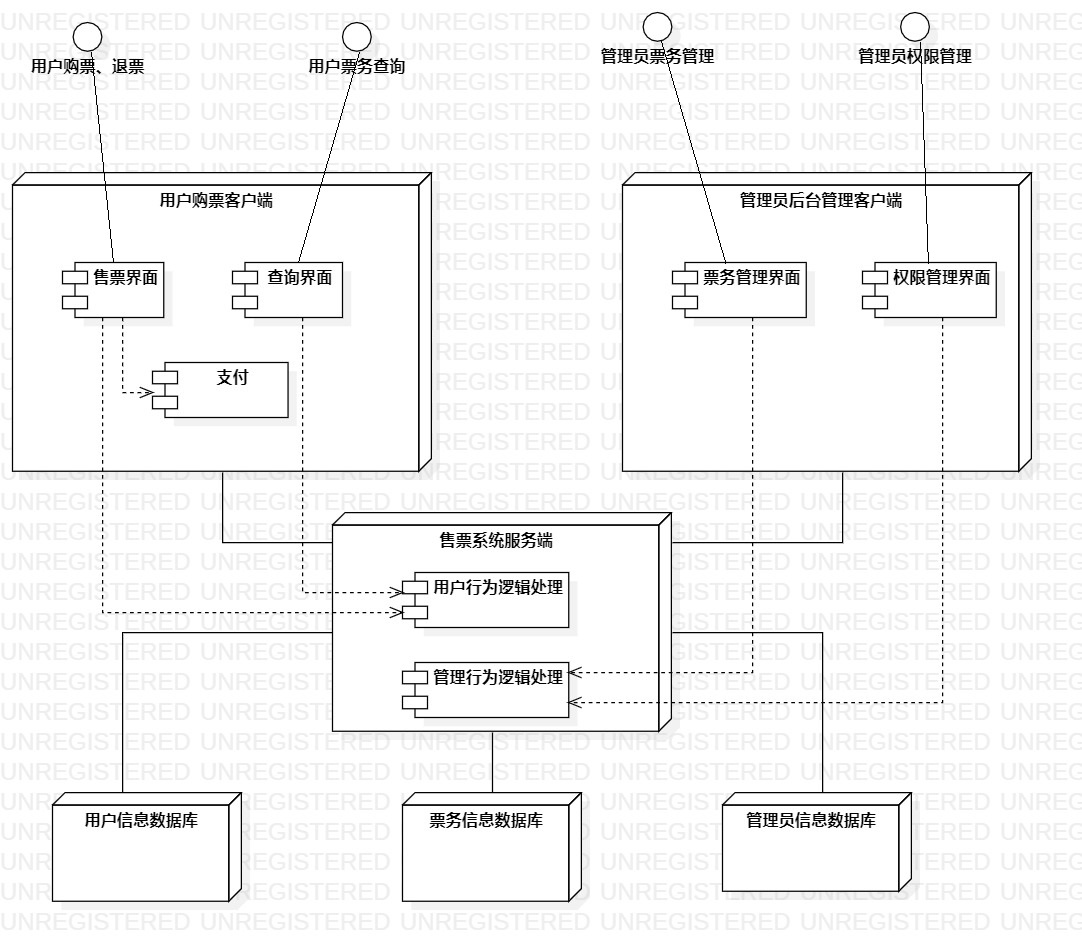
**时序图2：**

****

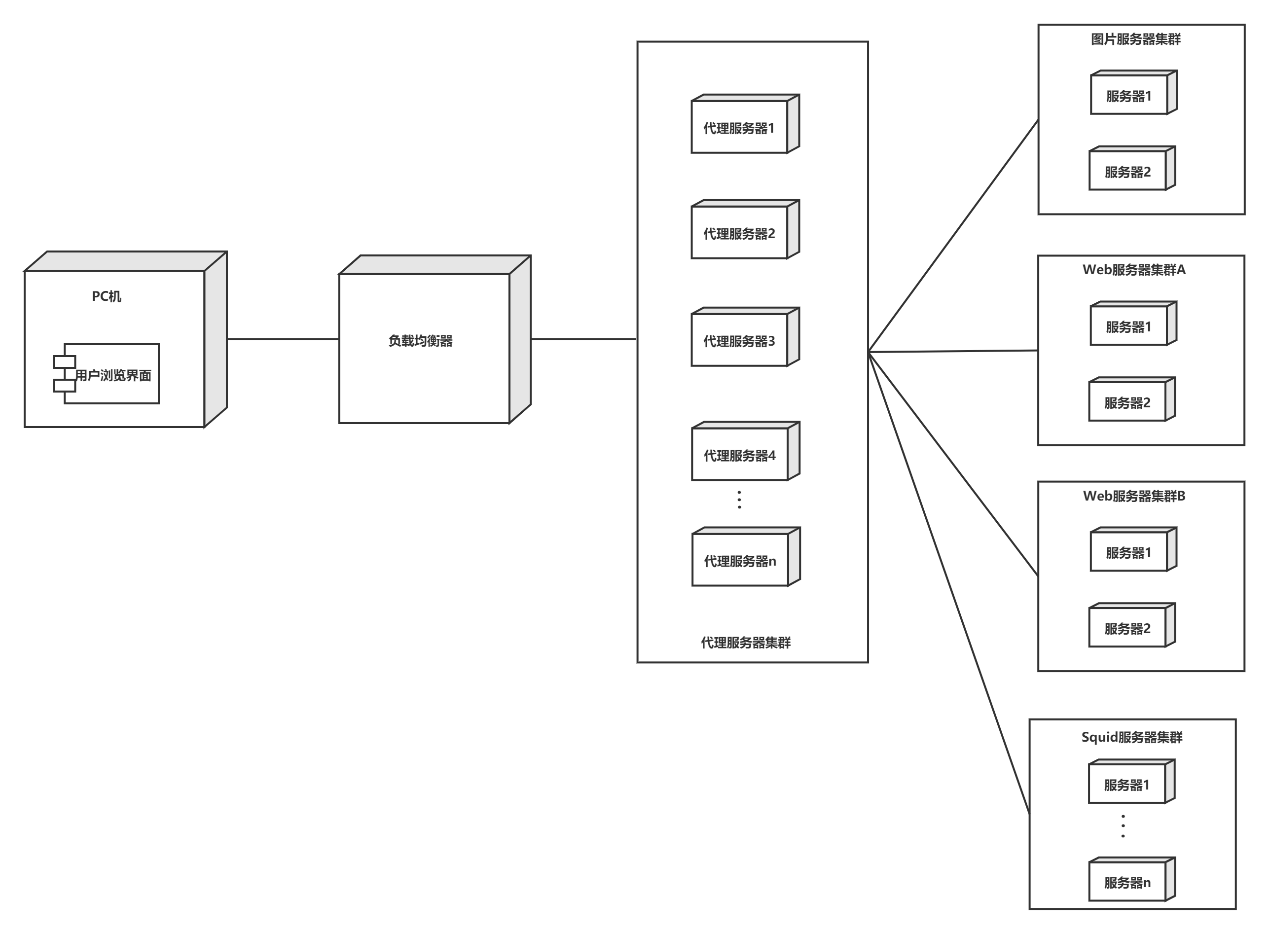
**时序图3：**

****

## 3.5、部署视图



## 3.6、物理视图



# 五、关键技术设计

## 5.1、接口设计

### 5.1.1、支付请求接口

1. 购票系统向对接的系统提出支付请求，让其他系统完成用户的支付操作。
2. 传递的信息包括支付金额、付款人以及收款地址。
3. 传递的数据类型：付款金额为int，付款人为Payer类，收款人地址为Payee类。
4. 异常定义：

在支付金额、付款人以及收款地址这三个信息传递过程中，出现信息丢失。

系统在传递支付金额、付款人以及收款地址时，出现信息传输时的错误。

系统在传递支付金额、付款人以及收款地址时，出现信息的遗漏。

1. 质量属性特征：

从系统提交支付请求，到对接系统接收到请求的时间间隔不应超过0.5s。

从系统提交支付请求，到对接系统完成支付的时间间隔不应超过2s。

在1000000000次对接中，最多容忍出错的次数为1次，且该笔的交易金额不能超过100元。

出错之后，能够在10分钟内测试出错误，在30分钟内做出修正。

能容许1000万人同时进行接口使用。

信息的传送方式采用加密通信，提升安全性，

进行测试时，不需要第二方接口即可对该接口进行测试。

### 5.1.2、用户信息接口

1. 在前端获取用户信息之后，通过接口传递给后端。
2. 给后端传递一个用户类User，里面包含了用户的基本信息。
3. 异常定义：

传递的信息为空。

网络故障时，信息传输失败。

前端在获取用户信息时，出现参数的获取错误。

1. 质量属性特征：

在10000000次对接中，最多容忍出错的次数为1次。

出错之后，能够在10分钟内测试出错误，在30分钟内做出修正。

从信息发出，到后端接收到接口的消息，这个时间间隔不应超过1ms。

能容许1000万人同时进行接口使用。

信息的传送方式采用加密通信，提升安全性，

在出现错误时，要能及时给用户发出提醒。

### 5.1.3、票务信息查询接口

1. 在前端获取车次信息之后，通过接口传递给后端。
2. 给后端传递一个车次编码，类型为String
3. 异常定义：

传递的信息为空。

网络故障时，信息传输失败。

前端在获取车次编号时，出现参数的获取错误。

1. 质量属性特征：

在10000000次对接中，最多容忍出错的次数为1次。

出错之后，能够在10分钟内测试出错误，在30分钟内做出修正。

从信息发出，到后端接收到接口的消息，这个时间间隔不应超过1ms。

能容许1000万人同时进行接口使用。

## 5.2、数据结构

1. 旅客信息类Passenger：

功能：包含旅客相关的信息和操作。

具体变量：

用户名userName：存储用户名的变量；

姓名name：必须是实名库里的存在的名字，存储用户现实中的名字；

性别gender：存储用户性别的变量；

国家地区country：存储用户国籍的变量；

证件类型IDType：存储用户证件类型的变量；

证件号码IDNumber：必须对应实名库里的存在的证件号码，并且和名字相对应，存储用户证件号码的变量；

手机号phoneNumber：必须是实际存在的手机号，存储旅客的手机号；

密码password：符合密码规范的密码，存储用户设定的密码；

旅客类型passengerType：存储旅客类型的变量；

旅客状态passengerStatus：存储购买该车票的旅客的状态如登车、未登车等的变量；

订单order：存储旅客的订单的变量；

第二登陆方式secondaryLogin：第二登录方式

1. 车次信息类Train：

功能：包含车次相关信息的类。

具体变量：

车次号trainNumber：必须是实际存在、在库里面的车次，存储车次号的变量；

起始站startingStation：必须是实际存在的车站，存储起始站相关的信息；

终点站terminalStation：实际存在的车站，存储终点站的相关信息；

类型trainType：必须是实际存在的火车类型，存储火车类型信息；

发车时间departureTime：存储相应车次从起始站发车的时间；

到达时间arrivalTime：存储相应车次到达终点站的时间；

车次停靠trainStops：存储该车所有停靠信息的一个集合；

车次状态trainState：存储车次状态的变量。

1. 车次停靠信息类TrainStop：

功能：包含车次停靠的相关信息；

具体变量：

车次号trainNumber：存储余票对应的车次号的变量；

停靠站stopStation：存储相应车次在从初始站到终点站途经的停靠的车站的信息；

到达时间arrivalTime：存储相应车次到达从初始站到终点站途经的停靠车站的时间；

停靠时间stopTime：存储相应车次在从初始站到终点站途经的停靠车站的停靠时间；

1. 余票信息类RestTickets：

功能：存储某一车次对应的余票的信息。

具体变量：

车次号trainNumber：存储余票对应的车次号的变量；

余票类型type：存储余票的类型的变量；

剩余票数remainingTicketNum：存储该票的剩余票数信息；

价格price：存储余票的价格的变量；

1. 车票信息类Ticket：

功能：存储旅客购买的车票的信息。

具体变量：

车次trainNumber：存储该车票对应的车次信息；

起始站startingStation：存储该车票的起始站信息；

终点站terminalStation：存储该车票的终点站信息；

发车时间departureTime：存储该车票对应的发车时间；

车厢号carriageNumber：存储该车票上记录的车厢的变量；

座位号seatNumber：存储该车票上记录的座位号的变量；

座位类型seatType：存储车票上记录的座位类型的变量；

检票口wicket：存储车票上记录的检票口的变量；

车票类型ticketType：存储车票的类型，是否为学生票、儿童票等

1. 订单信息Order：

功能：包含旅客的订单的相关信息；

车票ticket：存储旅客购买的车票的相关信息；

订单状态orderState：存储订单状态如未完成、已支付的变量；

价格price：存储购买的车票的价格；

支付时间paymentTime：存储旅客支付该车票的时间；

支付银行paymentBank：存储旅客支付该车票的所用的银行；

支付金额paymentAmount：存储旅客支付该车票的实际金额；

支付方式paymentMethod：存储旅客支付该车票的方式；

优惠discount：存储旅客支付该车票时获得的优惠；

购票渠道buyingChannel：存储旅客购买该车票渠道的变量；

1. 短信信息Message：

手机号码phoneNumber：存储短信对应的手机号的变量；

验证码[verification](javascript:;)Code：存储发送给旅客的验证码信息；

车票信息ticketInformation：存储短信里包含的车票信息；

1. 管理员Manager：

账户id：管理员的登录账户

密码password：管理员登陆的密码

权限power：管理员执行相应操作的权限

1. 第二登陆方式SecondaryLogin：

账号id：第二方登陆方式的账号

类型type：第二登录方式的类型 （enum SecondaryLoginType）

1. SecondaryLoginType：

Alipay

WeChat

1. 车站信息类Station：

name：车站名

departureTrain：从该站出发的火车

terminalTrain：到达该站的火车

wicket：检票口

exit：出站口

1. 付款人信息Payer：

wechat：绑定的微信支付账号

alipay：绑定的支付宝支付账户

PBC：绑定的人民银行账号

ICBC：绑定的中国工商银行付款账户

CCB：绑定的中国建设银行付款账户

BOC：绑定的中国银行支付账户

ABC：绑定的中国农业银行支付账户

BC：绑定的中国交通银行支付账户

CMB：绑定的中国招商银行支付账户

1. 收款地址Payee：

wechat：绑定的微信收款账户

alipay：绑定的支付宝首款账户

bankAdress：绑定的银行收款账户

## 5.3、数据库设计

使用MySQL数据库

数据库名称TicketingSystem

1. 数据库表passenger：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | user\_name | name | gender | country | id\_type | id\_number | phone\_number | password |
| 1 | … | … |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| type | status | order\_id | secondary\_login\_id |
| … |  |  |  |

id:INTEGER

user\_name:VARCHAR(10)

name: VARCHAR(10)

gender: VARCHAR(2)

country: VARCHAR(20)

id\_type: VARCHAR(20)

id\_number: VARCHAR(20)

phone\_number: VARCHAR(20)

password: VARCHAR(20)

type: VARCHAR(10)

status: VARCHAR(10)

order\_id: INTEGER

secondary\_login\_id: INTEGER

2.数据库表train:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | train\_number | starting\_station\_id | terminal\_station\_id | type | departure\_time | arrival\_time | state |
| 1 | … | … |  |  |  |  |  |

id: INTEGER

train\_number: VARCHAR(10)

starting\_station\_id: INTEGER

terminal\_station\_id: INTEGER

type: VARCHAR(10)

departure\_time:DATETIME

arrival\_time:DATETIME

state: VARCHAR(10)

3.数据库表trainstop：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | train\_id | stop\_station\_id | arrival\_time | stop\_time |
| 1 | … | … |  |  |

id: INTEGER

train\_id: INTEGER

stop\_station\_id: INTEGER

arrival\_time: DATETIME

stop\_time: DATETIME

4.数据库表rest\_ticket：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | train\_id | type | remaining\_ticket\_num | price |
| 1 | … | … |  |  |

id: INTEGER

train\_id: INTEGER

type: VARCHAR(10)

remaining\_ticket\_num: INTEGER

price: INTEGER

5.数据库表ticket：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | train\_id | starting\_station\_id | Terminal\_station\_id | departure\_time | carriage\_num | seat\_num | seat\_type | wicket |
| 1 | … | … | … |  |  |  |  |  |

id: INTEGER

train\_id: INTEGER

starting\_station\_id: INTEGER

terminal\_station\_id: INTEGER

departure\_time:DATETIME

carriage\_num: INTEGER

seat\_num: INTEGER

seat\_type: VARCHAR(10)

wicket: VARCHAR(10)

6.数据库表order：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | ticket\_id | order\_state | price | payment\_time | payment\_bank |
| 1 | … | … |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| payment\_amount | payment\_method | discount | buying\_channel |
| … |  |  |  |

id: INTEGER

ticket\_id: INTEGER

order\_state: VARCHAR(10)

price: INTEGER

payment\_time:DATETIME

payment\_bank: VARCHAR(10)

payment\_amount: INTEGER

payment\_method: VARCHAR(20)

discount: INTEGER

buying\_channel: VARCHAR(20)

7.数据库表message：

|  |  |  |  |
| --- | --- | --- | --- |
| id | phone\_number | verification\_code | ticket\_information |
| 1 | … | … |  |

id: INTEGER

phone\_number: VARCHAR(20)

verification\_code: VARCHAR(20)

ticket\_information: VARCHAR(100)

8.数据库表manager：

|  |  |  |
| --- | --- | --- |
| id | password | power |
| 1 | … | … |

id: INTEGER

password: VARCHAR(20)

power: VARCHAR(10)

9.数据库表secondary\_login:

|  |  |  |
| --- | --- | --- |
| id | passenger\_id | type |
| 1 | … |  |

id: INTEGER

passenger\_id: INTEGER

type: VARCHAR(10)

10.数据库表station:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | name | departure\_train\_id | terminal\_train\_id | wicket | exit |
| 1 | … |  |  |  |  |

id: INTEGER  
name: VARCHAR(10)

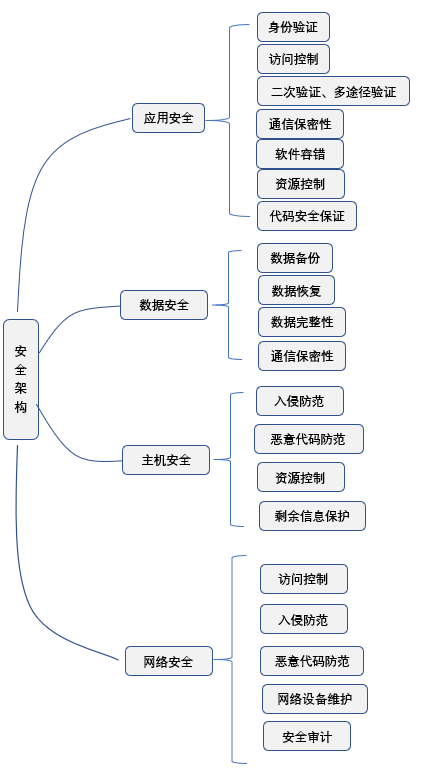
departure\_train\_id: INTEGER

terminal\_train\_id: INTEGER

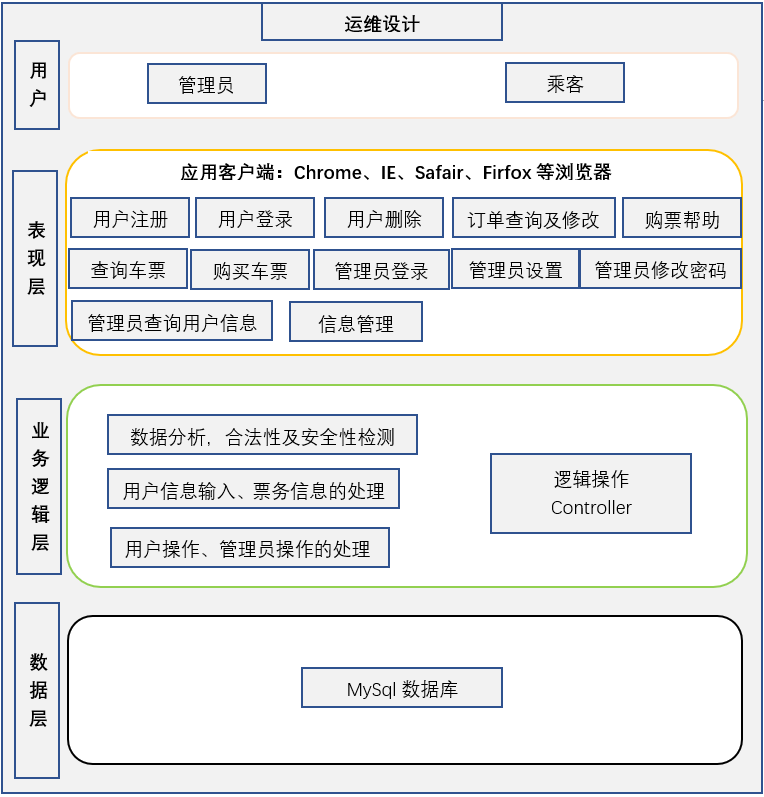
wicket: VARCHAR(100)

exit: VARCHAR(100)

## 5.4、安全架构设计



## 5.5、运维架构设计



# 六、其他

## 6.1、优先级

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 优先级 | 功能名称 | 优先级 |
| 用户注册 | A | 改签 | B |
| 用户登录 | A | 变更到站 | B |
| 票务查询 | A | 查询未完成订单 | B |
| 购票 | A | 查询候补订单 | B |
| 退票 | A | 管理员设置 | C |
| 管理员登录 | A | 查询已支付订单 | C |
| 增加车次信息 | A | 用户注销 | C |
| 修改车次信息 | A | 查询用户信息 | C |
| 查询本人车票 | A | 查询购票帮助 | C |
| 管理员修改密码 | A |  |  |

（注：优先级顺序为A>B>C）