



---

# [数据库系统原理]

---

[课程设计总结]



2024-6-28

21281159  
琚王琳博

# 目录

- 1. 系统规划与可行性分析报告 .....2
  - 1.1 项目简介 .....2
  - 1.2 系统需求 .....2
  - 1.3 业务场景 .....2
  - 1.4 性能指标 .....3
  - 1.5 市场分析 .....3
  - 1.6 可行性分析 .....3
- 2. 系统需求规格说明书 .....4
  - 2.1 业务需求分析 .....4
  - 2.2 业务流程分析 .....4
  - 2.3 数据需求 .....5
  - 2.4 非功能性需求 .....13
- 3. 系统详细设计说明书 .....13
  - 3.1 系统功能概述与系统功能模块结构 .....13
  - 3.2 系统界面设计 .....13
  - 3.3 系统物理模型 .....15
  - 3.4 系统安全体系设计 .....16
  - 3.5 系统运行环境设计与部署结构 .....17
  - 3.6 源代码列表及说明 .....17
- 4. 用户安装与使用手册 .....18
- 5. 所有源代码与脚本 .....18
- 6. 系统设计总结与课程总结 .....18

# 1. 系统规划与可行性分析报告

## 1.1 项目简介

本项目的名称为铁路 12306 自助查票系统。铁路 12306 自助查票系统是中国铁路总公司官方推出的一种网上查票系统，乘客可以通过该系统查询火车票的余票信息，方便用户。

## 1.2 系统需求

前台页面设计：系统需要设计一个直观、简洁、易于操作的前台页面，以提供给用户使用，用户可以方便地进行查询、选座、购票等操作。

数据采集：系统需要从国家铁路局的后台服务器上获取实时的列车运行信息和票务信息，为用户提供最新的信息。

查票功能：系统需要提供查询票务信息的功能，以方便用户在不同时间、不同地点完成火车票相关业务。

数据存储与管理：系统需要建立一个统一的数据库来存储所有的系统数据，并提供相应的管理功能，以保证数据的安全性和稳定性。

系统安全：系统需要采取一系列措施，如数据加密、用户身份验证等，来确保系统的安全性和稳定性。

## 1.3 业务场景

### 1. 登录系统并查询余票

- a. 用户打开对应的网页或者手机 APP，输入账号密码登录
- b. 如果输入正确，进入主界面，输入错误则返回登陆页面并提供找回密码服务
- c. 在搜索页面查询特定日期、始发、终点站的车票余量。
- d. 网页或手机 app 返回结果。

### 2. 更新车票信息

- a. 管理员打开管理端网页，输入账号密码登录
- b. 如果输入正确，进入主界面，输入错误则返回登陆页面
- c. 在车票信息管理界面对车票信息进行更改或删除
- d. 网页返回更改结果。

### 3. 更新车次信息

- a. 管理员打开管理端网页，输入账号密码登录
- b. 如果输入正确，进入主界面，输入错误则返回登陆页面
- c. 在车次信息管理界面对车次信息进行更改或删除
- d. 网页返回更改结果。

### 4. 更新用户信息

- a. 管理员打开管理端网页，输入账号密码登录
- b. 如果输入正确，进入主界面，输入错误则返回登陆页面
- c. 在用户信息管理界面对用户信息进行更改或删除

d.网页返回更改结果。

## 1.4 性能指标

对于核心业务响应时间，即对数据库信息进行检索或者增删减改的时间消耗。如查询于4月5日由北京开往上海的列车的车票余量时，快速查询并展示相应的查询结果。若不受网络影响，整个查询过程不能超过1min。

由于除管理员外的用户对整个系统只有查询权限，因而对普通用户的并发操作限制应当宽松，而对管理员更新数据的并发操作进行较为严格的限制，防止数据出错。并发用户数量的设置应以不影响数据完整性为前提，分用户群体设计。

由于数据库系统面向整个社会，因此数据库用户数极大，需要着重维护用户信息。系统每秒处理的事务数较高。整个系统需要有处理亿级吞吐量的能力。

数据库需要存储的用户信息、车票信息条目极多，系统需要长期存储。需要保证以上数据的安全性。

## 1.5 市场分析

12306 是中国铁路客户服务中心推出的官方购票应用软件，服务于全体社会公众。作为一项基于铁路运营的公共服务，系统依赖于铁路运营赢得客户，获得收益。由于我国铁路为国家所有，此系统在市场上没有竞品，市场地位稳固。

投资和运营成本：

对于运营成本，最主要两点就是服务器的购买与维护以及在进行内容审查时的人工成本。因为12306查票系统自身的数据量相对较大，同时又需要维持快速的查询回复，所以对于服务器以及系统的需求往往比较高。同时对于一个查票系统，需要保证信息的时效性和有效性，维护车票数据十分重要，增加了维护成本。网站，APP的创建成本也需要考虑。

本系统借助铁路售票获得收益。

## 1.6 可行性分析

技术可行性：12306 查票系统使用现代化的互联网技术，如 PHP、MySQL、Javascript 等，技术成熟，可实现。

经济可行性：投入总成本相对较低，相较于传统方式，12306 查票系统能够节省了人力、物力、财力资源，有较高的经济效益。

管理可行性：12306 查票系统需要建立一个统一的数据库来存储所有的系统数据，并提供相应的管理功能。这些功能包括数据库备份、故障恢复、权限控制等，以保证数据的安全性和稳定性。因此，需要具备数据库管理与运维经验的人员对系统进行后台管理和维护。

用户体验可行性：在现代互联网时代，用户体验是非常重要的。12306 查票系统需要具备良好的用户界面设计、快速的查询响应时间、完善的购票流程及售后服务等方面，以提高用户满意度和忠诚度。

技术更新可行性：随着科技的不断进步和更新，12306 查票系统需要不断升级和更新技术，以满足用户需求和提高系统效率。系统技术更新包括硬件服务器升级、软件架构升级、

数据库优化等方面。

系统性能可行性：12306 查票系统需要保证具有高可用性、高性能等方面的性能，以满足高并发的需求。同时，也需要考虑系统的伸缩性，即在用户量、访问量增大的情况下，能够自动扩展资源，保证系统的稳定性。

## 2 系统需求规格说明书

### 2.1 业务需求分析

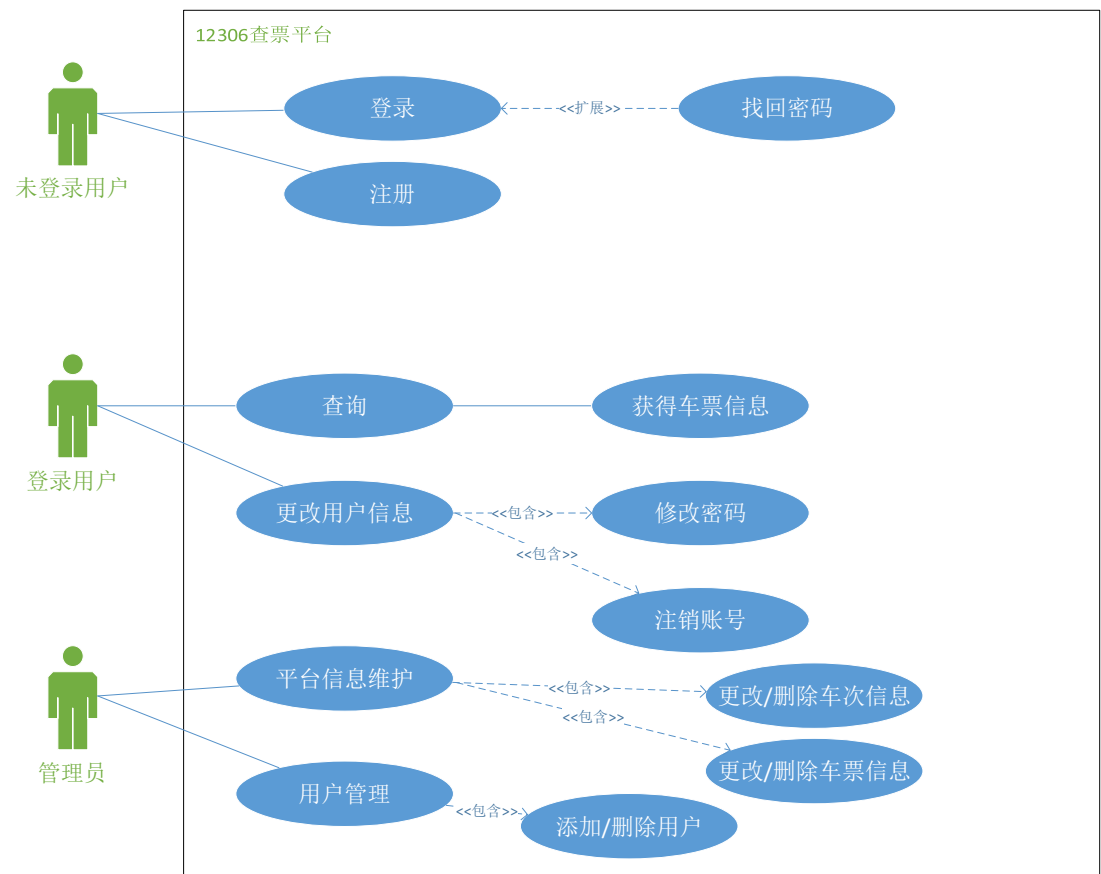
12306 查票系统的主要业务功能包括以下几个方面：

票务查询：提供高效便捷的票务查询功能，方便用户了解车票类型、车次信息、日期和站名等相关信息。

更改用户信息：通过账户维护功能，为用户提供修改账户信息的接口。

信息维护：管理员通过平台信息维护功能修改车次、车票信息并管理用户。

用例图如下：



### 2.2 业务流程分析

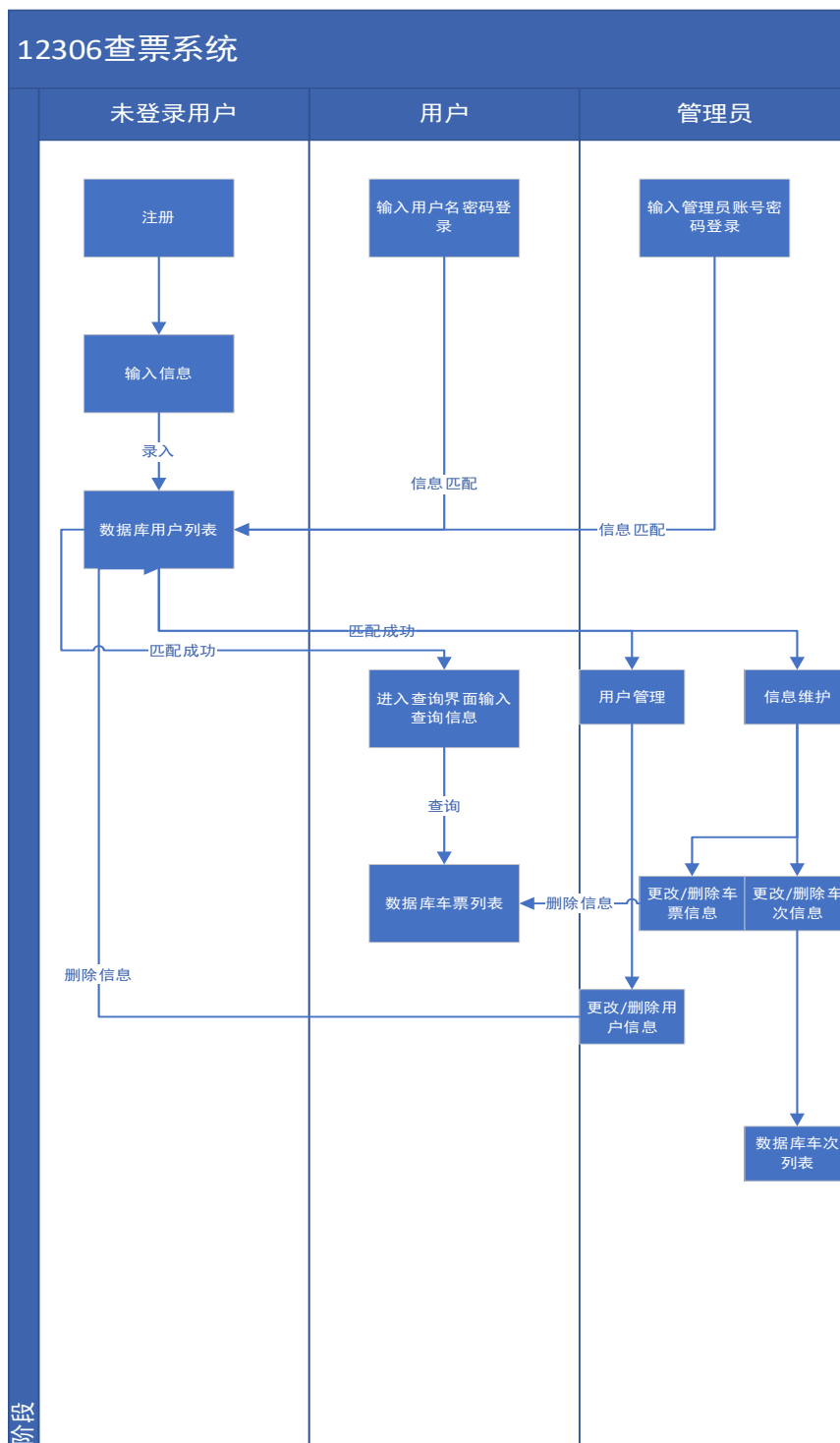
用户注册：用户在访问 12306 查票系统之前，需要先进行账号注册。用户需提供手机号等信息，完成账号注册。

登录系统：用户在注册成功后，通过输入正确的账号和密码，登录到系统中。

查询票务信息：登录到系统后，用户可以选择出发站和目的站，选择乘车日期及车次，点击查询按钮，获取相应的票务信息。

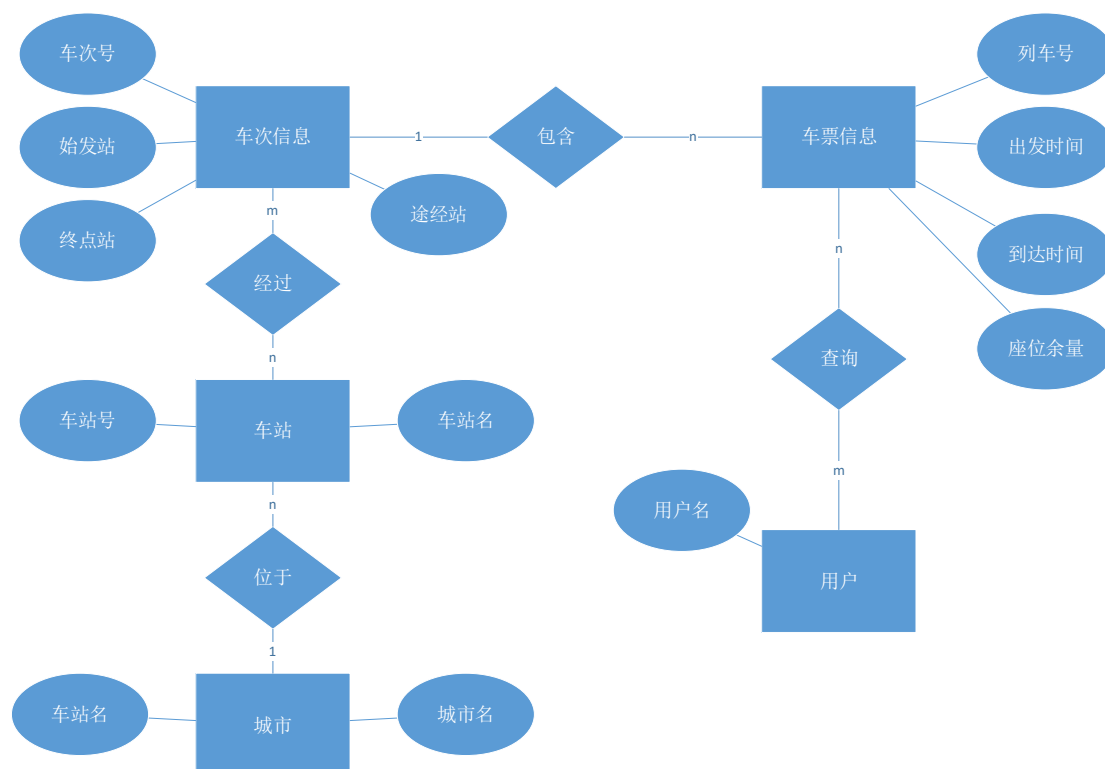
更改用户、列车信息：管理员登录系统后可以选择系统中的各种信息进行修改。

泳道图如下：



## 2.3 数据需求

通过对系统逻辑结构的分析得出如下的 E-R 图



通过 E-R 图表现的关系，编写系统数据表如下：  
Order 表用于存储订单信息与用户信息的映射关系。

表名	Order					
数据库用户	root					
主键	No					
外键	User_phone_num					
排序字段	Order_id					
索引字段	Order_id					
字段名称	数据类型	允许为空	唯一	默认值	约束条件	
no	int	N	Y			
User_phone_num	varchar	N	N		外键	
Passenger_phone_num	varchar	N	N			
Passenger_name	varchar	N	N			

Train_no	varchar	N	N		
Origin_name	varchar	N	N		
Destination_name	varchar	N	N		
Carriage_no	varchar	N	N		
Seat_no	varchar	N	N		
Order_money	varchar	Y	N		
Order_status	varchar	Y	N		
Train_start_time	datetime	Y	N		
Order_create_time	datetime	Y	N		

Passenger 表用于存储用户 user 信息和乘车人 passenger 信息的映射关系。

表名	Passenger				
数据库用户	root				
主键	No				
外键	user_id				
排序字段	No				
索引字段	user_id				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
No	varchar	N	N		外键
phone	varchar	N	Y		主键
realname	varchar	N	N		
user_id	varchar	N	N		
type	varchar	N	N		



address	varchar	Y	N		
---------	---------	---	---	--	--

Seat 表用于存储列车与列车中座位信息的映射关系。

表名	seat				
数据库用户	root				
主键	No				
外键	train_no				
排序字段	train_no				
索引字段	train_no				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
no	int	N	Y		主键
train_no	int	N	Y		
carriage_no	int	N	Y		
seat_type	int	N	N		
seat_count	int	N	N		

station 表用于存储列车与停靠站信息之间的映射关系。

表名	stations				
数据库用户	root				
主键	station_id				
外键	station_name				
排序字段	station_id				
索引字段	station_id				
字段名称	数据类型	允许为空	唯一	默认值	约束条件

station_id	varchar	N	Y		主键
station_name	varchar	N	N		
train_id	varchar	N	Y		外键
train_number	varchar	N	N		
arrive_day	datetime	N	N		
start_time	datetime	N	N		
arrive_time	datetime	Y	N		
running_time	datetime	Y	N		

Train 表用于存储列车的基本信息

表名	train				
数据库用户	root				
主键	no				
外键	station_name				
排序字段	no				
索引字段	no				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
id	varchar	N	Y		主键
number	varchar	N	N		
type	varchar	N	N		
carriage_num	varchar	N	N		
origin_station	int	N	N		
destination_station	varchar	N	N		

start_time	varchar	N	N		
arrive_time	varchar	Y	N		
arrive_day	varchar	Y	N		
running_time	varchar	Y	N		

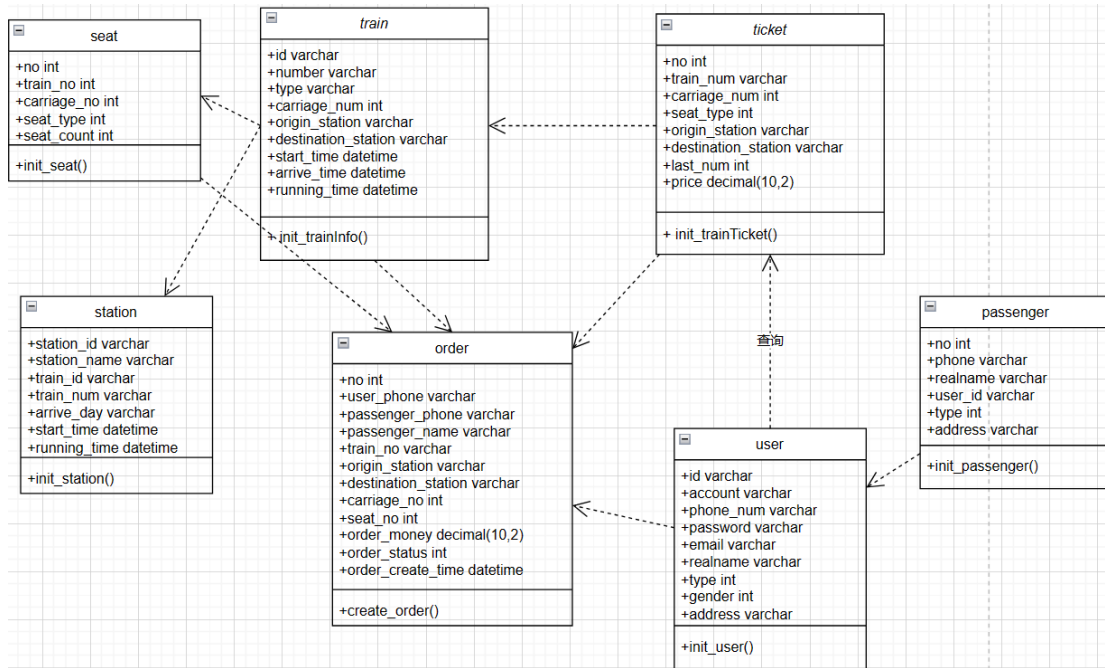
User 表用于存储用户基本信息。

<b>表名</b>	<b>user</b>				
数据库用户	root				
主键	id				
外键					
排序字段	id				
索引字段	id				
<b>字段名称</b>	<b>数据类型</b>	<b>允许为空</b>	<b>唯一</b>	<b>默认值</b>	<b>约束条件</b>
id	varchar	N	Y		主键
account	varchar	N	N		
phone_num	varchar	N	Y		
password	varchar	N	N		
email	varchar	N	N		
realname	varchar	N	N		
type	int	N	N		
gender	int	N	N		0/1
address	varchar	Y	N		

Ticket 表用于存储车票信息的基础信息。

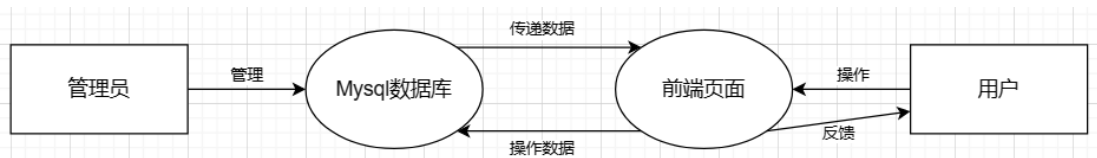
表名	ticket				
数据库用户	root				
主键	no				
外键	Train_num				
排序字段	no				
索引字段	no				
字段名称	数据类型	允许为空	唯一	默认值	约束条件
no	varchar	N	Y		主键
train_num	varchar	N	N		
carriage_num	varchar	N	Y		
seat_type	int	N	N		
origin_station	varchar	N	N		
destination_station	varchar	N	N		
last_num	int	N	N		
price	decimal(10,2)	N	N		

对于系统的数据结构，这里按照数据项的种类进行类的划分。由于用户查询车票信息是一对多的关系，即一个用户可以查询到多个车票信息。因而 User 与 Trainticket 为直接关联关系。同时，由于 Ticket 表中有来自 Train 的外键约束；Train 中含有来自 Station 的外键约束。因而为依赖关系。对于每个类里都会存在一个 Init 的函数，用来对类内变量进行初始化，方便其它类内函数的调用。表示的类图如下。

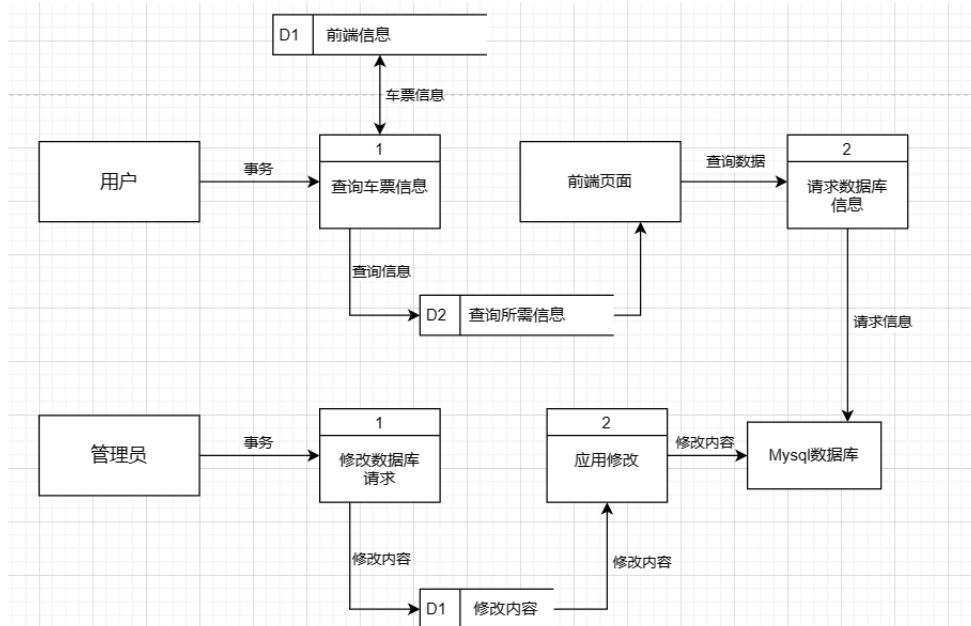


对于数据的存储需求，首先对于数据量大小，为了让数据能够正常存储，磁盘容量需要足够大。而对于数据的物理存储方式，由于 **Trainticket** 是有频繁查找需求的表，并且涉及到相对较多的数据插入操作以及较大量的数据，所以不适合添加聚簇索引，而 **Traininfo**, **Station** 作为有频繁查找需求的表，自身插入操作较少，可以添加聚簇索引。**User** 因为自身属性原因，采用随机存取方式即可。

数据处理需求如 DFD 所示：



第 0 层 DFD 如下：



## 2.4 非功能性需求

对于业务处理性能，系统要采取集群方式提供服务，有异常检查系统，及时识别有问题的节点和重启，保证服务的高可用性和数据的完整性。

对于业务处理的安全性，作为一个铁路查票平台，隐私信息存储极大，要保障用户的隐私信息不会泄漏。另外，平台的操作权限只能由登陆用户拥有，网站要对用户的密码进行保护。

对于业务处理的完整性需求，用户在平台对数据信息的修改要符合数据库表的三种完整性要求，否则会给用户无法执行的信息反馈。

## 3. 系统详细设计说明书

### 3.1 系统功能概述与系统功能模块结构

12306 查票系统是一个基于网络的售票平台，主要提供列车时刻表等服务。用户可以在该系统中进行账号注册、登录、查询车票等操作。系统主要功能模块如下：

**前端功能：**前端功能模块是用户直接与之交互的模块，包括网站设计、页面布局、交互功能等。这个模块主要由 HTML、CSS、JavaScript 等语言组成，使用 VUE 框架进行具体实现，它们共同实现了该系统的可视化效果和操作功能。前端模块包括列车时刻表、余票查询、用户信息、订单信息等多个模块。在用户进行相关操作时，前端将请求转发给 WEB 服务端，从而实现后续的业务处理。

**WEB 服务端：**WEB 服务端模块主要负责系统的业务逻辑和数据处理工作。其主要负责接受前端模块的请求并进行相关业务处理，在处理完成后将结果返回给前端模块。本项目的 WEB 服务端模块采用 python 编程语言进行开发。WEB 服务端模块包括列车时刻表、余票查询等多个模块。在这些模块中，WEB 服务端会调用数据库端的相应模块进行数据查询、数据插入和数据更新等操作，并根据具体情况做出相应的响应。

**数据库端：**数据库端模块主要负责系统中的数据存储和查询工作。本项目选用 MySQL 数据库。该模块主要包括数据管理、数据备份与恢复、性能优化等多个方面，以保证数据的合理组织、高效查询和稳定存储。数据库端模块包括了列车时刻表、余票信息、个人用户数据等多个数据表，这些数据表包含了系统中各个模块所需要的数据。在 WEB 服务端和前端模块的请求下，数据库端会根据要求提供相关数据，并进行数据更新、维护和备份工作，确保系统的可靠性和稳定性。

### 3.2 系统界面设计

系统通过 B/S 模式与用户交互，因此系统前端界面采用 html 格式进行编写。详细代码请见附件，界面详见下图。

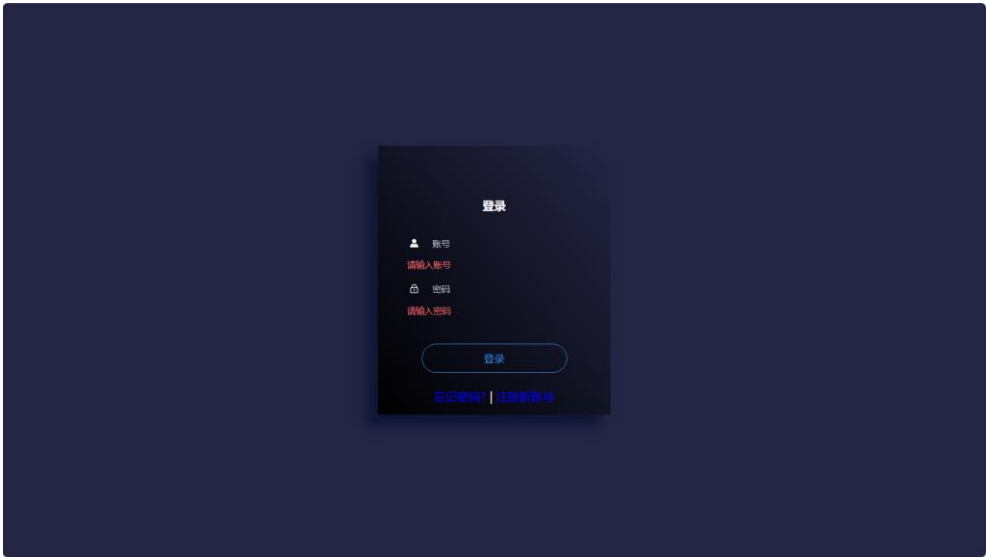


图 用户登录页

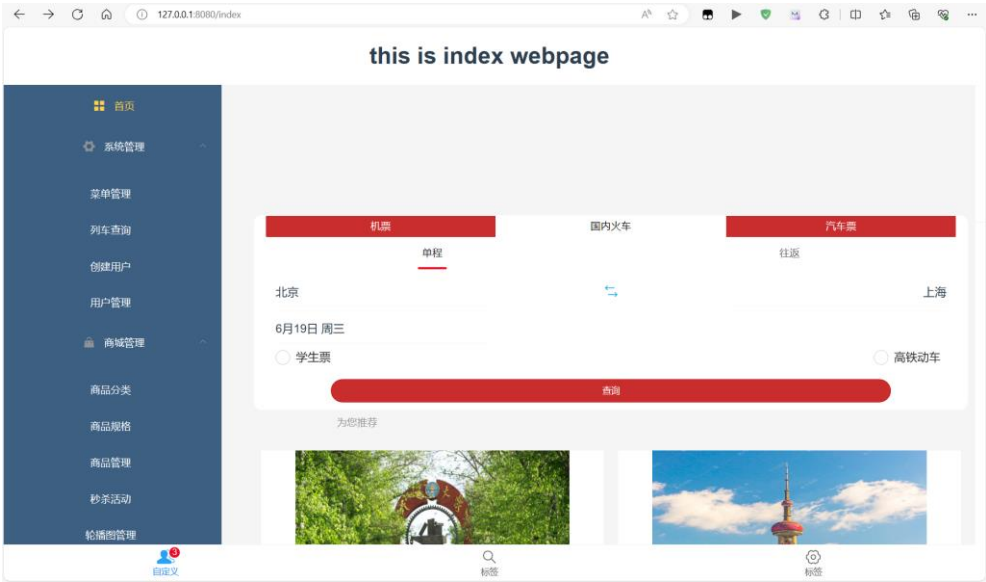


图 用户搜索页

北京 → 上海						抢票
列车号	车厢号	座位类型	发车时间	剩余票数	价格	
G435	7	3	2023-11-23 17:36:07	58	50.50	
G435	6	3	2023-11-23 17:36:07	45	60.00	
G435	8	3	2023-11-23 17:36:07	60	50.50	
G435	5	3	2023-11-23 17:36:07	29	60.00	
4126	5	3	2023-11-23 09:28:28	44	35.50	
4126	6	3	2023-11-23 09:28:28	45	35.50	
4126	7	3	2023-11-23 09:28:28	33	40.00	

图 用户购票页

首页 > 活动管理 > 活动列表

请输入内容

添加用户

#	姓名	邮箱	电话	角色	状态	操作
1	1670864543	1670864543@qq.com	158-1032-7051	管理员	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
2	admin	11111	11111	管理员	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
3	乘车人111111	2221119999@bjtu.edu.cn	13518851243	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
4	zcbcc	cwf1947@mail.com	13311113333	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
5	13251325	12222@qq.com	161235152	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
6	miyamo@1014	miyamotorin10@outlook.com	2	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
7	yidegelami	harold326@yahoo.com	3	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
8	adadad	juwanglinbo@gmail.com	13333333333	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
9	Wangrit	c59066368dupo0899@163.com	13888888888	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
10	dddaaaaa	1555@qq.com	13888888888	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
11	yibaminuosi	twinsted@qq.com	13910779974	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>
12	aminosi	alicewa@gmail.com	5	用户	<input type="checkbox"/>	<a href="#">编辑</a> <a href="#">删除</a> <a href="#">重置</a>

图 用户管理界面

首页 > 订单管理 > 订单列表

请输入内容

#	电话号码	乘客姓名	订单价格	是否付款	下单时间	操作
1	11-288-0711	testing	114514	已付款	Sat, 20 May 2006 18:49:20 GMT	<a href="#">编辑</a> <a href="#">删除</a>
2	74-243-1663	default	32.1	已付款	Mon, 18 May 2020 17:09:42 GMT	<a href="#">编辑</a> <a href="#">删除</a>
3	21-3008-7579	aaaaa	22.2	已付款	Mon, 16 Nov 2009 16:12:34 GMT	<a href="#">编辑</a> <a href="#">删除</a>
4	614-258-2003	翟王琳博	61	未付款	Mon, 31 Aug 2020 08:14:44 GMT	<a href="#">编辑</a> <a href="#">删除</a>
5	5708 601454	alskfgja	648	已付款	Fri, 16 Nov 2018 15:58:23 GMT	<a href="#">编辑</a> <a href="#">删除</a>
6	112315124	yidegelami	31.5	未付款	Tue, 21 May 2024 20:57:33 GMT	<a href="#">编辑</a> <a href="#">删除</a>
7	112315124	aminuosi	32	未付款	Tue, 21 May 2024 20:57:33 GMT	<a href="#">编辑</a> <a href="#">删除</a>
8	11111	admin1	21.767388956404655	未付款	Tue, 11 Jun 2024 16:48:39 GMT	<a href="#">编辑</a> <a href="#">删除</a>
9	11111	admin1	57.918190823791726	未付款	Tue, 11 Jun 2024 16:48:39 GMT	<a href="#">编辑</a> <a href="#">删除</a>

图 订单管理界面

3.3 系统物理模型

数据表与索引具体内容已在 2.3 中给出，此处不再赘述。系统中存在 v\_ticket 视图，旨在展示完整的车票及车次信息，用于前端展示。内容如下图所示，具体定义详见脚本。



train_num	carriage_num	seat_type	origin_station	destination_station	start_time	last_num	price
Z16	3	2	哈尔滨	北京	2023-11-23 21:21:20	15	50.00
2210	4	5	黑河	大连	2023-11-23 12:52:55	33	186.30
4082	6	5	黑河	齐齐哈尔	2023-11-23 15:33:48	36	6.80
4084	4	1	黑河	齐齐哈尔	2023-11-23 14:26:52	37	272.60
Z16	2	1	哈尔滨	北京	2023-11-23 21:21:20	33	80.00
Z16	4	3	哈尔滨	北京	2023-11-23 21:21:20	50	40.00
Z16	5	3	哈尔滨	北京	2023-11-23 21:21:20	50	40.00
Z16	6	3	哈尔滨	北京	2023-11-23 21:21:20	50	40.00
Z16	7	3	哈尔滨	北京	2023-11-23 21:21:20	50	40.00
2210	5	3	黑河	大连	2023-11-23 12:52:55	50	186.30
2210	6	3	黑河	大连	2023-11-23 12:52:55	50	186.30
2210	7	3	黑河	大连	2023-11-23 12:52:55	48	186.30
2210	8	3	黑河	大连	2023-11-23 12:52:55	48	186.30
G435	7	3	北京	上海	2023-11-23 17:36:07	58	50.50
G435	6	3	北京	上海	2023-11-23 17:36:07	45	60.00
G435	8	3	北京	上海	2023-11-23 17:36:07	60	50.50
G435	5	3	北京	上海	2023-11-23 17:36:07	29	60.00
4126	5	3	北京	上海	2023-11-23 09:28:28	44	35.50
4126	6	3	北京	上海	2023-11-23 09:28:28	45	35.50
4126	7	3	北京	上海	2023-11-23 09:28:28	33	40.00

图 v\_ticket 视图

### 3.4 系统安全体系设计

用户管理与控制方面：

12306 查票系统采用了完善的用户管理和控制机制，确保只有经过认证的用户才能使用系统，并且系统管理员具有对用户权限进行管理的能力。

具体来说，用户可以通过注册并登录系统获得使用权限，登录时需要验证账号密码等身份信息。系统管理员可以对用户进行管理，包括添加、修改、删除、禁用账号等操作，以确保只有正式用户才能访问数据。

此外，在用户登录时，还会记录登录信息，如 IP 地址、登录时间等，以便于后续的安全审计和跟踪，同时也会设置一些访问限制规则，如同账户在短时间内多次错误登录时会被暂时禁用等。

存储与恢复方面：

12306 查票系统采用了多层次的存储与恢复机制，以确保数据安全性和可靠性。

为防止恶意攻击和数据泄露，系统采用访问控制技术，在前端通过 javascript 脚本限制对关键数据的访问，并定期进行安全测试和漏洞扫描等操作，及时消除系统中存在的安全隐患。

系统采用数据库来存储车票、订单等关键数据，因此对数据库进行定期备份和恢复非常关键。为了防止因各种原因导致的数据丢失、损坏或误操作等情况，系统会对数据库进行定期备份。本系统采用 python 后端定期进行 mysql 数据库备份操作，由于定时进行备份会导致备份文件占用大量空间，故设计为每天备份一次，时间为每晚 23 点 56 分，具体实现详见附录。

同时，在进行数据库备份时，系统还会启用事务机制，以确保备份数据的一致性和可靠性。在发生数据损坏等情况时，系统管理员可以通过备份数据进行恢复，并在恢复之后对数据库进行充分测试，确保系统正常运行。

并发控制方面：

数据库采用 SERIALIZABLE 的事务隔离等级，以避免因为并发操作导致的数据库数据异常。具体如下图所示：

```
mysql> select @@global.transaction_isolation;
+-----+
| @@global.transaction_isolation |
+-----+
| SERIALIZABLE                    |
+-----+
1 row in set (0.00 sec)
```

### 3.5 系统运行环境设计与部署结构

本系统需要运行在两个硬件平台上，分别为数据库服务器和 Web 服务器。这两个服务器都需要具备较高的性能和稳定性，以满足系统的高负载和高可用性要求。一般来说，这些服务器需要配备多核 CPU、大容量存储设备、高速网络接口等硬件资源，并且需要进行负载均衡和集群配置，以提高系统的可靠性和可扩展性。

12306 查票系统的数据库服务器与 Web 服务器需要运行在相同的操作系统上，为 Windows11。数据库服务器需要安装 MySQL 数据库引擎。此外，数据库还可以通过 Navicat 软件 SSH 连接数据库服务器的方式进行管理。

本系统的系统架构采用了分层结构，最上层为用户界面层，进行各种用户交互操作；中间层为应用程序层，负责业务逻辑处理和数据存储管理；最下层为数据存储层，负责对数据进行存储和管理。用户界面层和应用程序层采用 MVC 模型，以实现业务逻辑与用户界面的解耦，提高系统的可维护性和可扩展性。

12306 查票系统的部署结构采用了分布式架构，将数据库服务器和 Web 服务器部署在不同的物理节点上，并通过网络连接进行通信。具体来说，在系统的部署过程中，首先需要安装 Windows11 操作系统，并分别配置数据库服务器和 Web 服务器的硬件和软件环境；然后，将 MySQL 数据库引擎部署在数据库服务器上，并进行数据初始化和备份。同时，在系统部署过程中还需要进行网络设置、安全配置等操作，以确保系统的稳定性和安全性。

### 3.6 源代码列表及说明

#### 1. # 前端文件说明

- |                    |           |
|--------------------|-----------|
| 2. Login.vue       | #用户登录页面   |
| 3. Index.vue       | #首页       |
| 4. Menu.vue        | #左侧菜单组件   |
| 5. Orders.vue      | #订单界面     |
| 6. Register.vue    | #注册界面     |
| 7. Train_list.vue  | #列车购票界面   |
| 8. Usrmng.vue      | #用户信息管理界面 |
| 9. router/index.js | #路由文件     |

10.

#### 11. # 后端文件说明

- |               |          |
|---------------|----------|
| 12. fflask.py | #后端服务器代码 |
| 13. BackUp.py | #备份数据库代码 |

14.

#### 15. # 数据库建库脚本说明

## 4.用户安装与使用手册

见附件。

## 5.所有源代码与脚本

见附件

## 6.系统设计总结与课程总结

通过完整设计整个数据库系统,我对数据库本身以及整个数据库系统的设计有了比较全面的了解。在制作网页前端时也初步学习了 vue 框架下的三种前端编程语言中数据库连接的基本操作。结合课程内容对数据库存储结构进行优化也让我对课本上的知识得到了充分的实践。

在课程的学习中,我掌握了数据库设计、数据结构、查询优化、事务处理以及数据安全等各方面的知识,并且将它们应用于实际的数据库开发和管理工作中,以提高数据库的性能和可靠性。获取新知识的同时,数据库软件开发能力也得到了锻炼。