

铁路管理系统报告

0. 分工安排

学号	姓名	工作内容
2018114228	殷子浩	参与数据库设计，完成管理员端鉴权、认证、模糊搜索、车次、票务管理模块的前端、后端开发，区分用户身份与坐等，github工作流管理
2018115563	周依萍	参与数据库设计，用户端对页面的前端框架进行二次调整；管理端负责整体页面框架决定与调整，车站、线路模块的前、后端开发
2018110753	金嘉淇	参与数据库设计与实现；ER图；部署图；用户端鉴权、认证；人脸识别扩展；数据分析预测前后端开发；列车3D展示模块前后端开发；GitHub commit最多
2018111062	单招文	参与数据库设计，用户端：前端html框架，css初级美化 模糊搜索，js读取后端数据，类图、PPT制作
2018112268	刘立敏	参与数据库设计，用户端后端逻辑功能实现，模糊搜索，后端实现 根据用户类型显示票价，顺序图绘制
2018113562	吴国桐	参与数据库设计；用例图绘制；云服务器搭建； 管理员端：注册登陆模块、用户模块、站点模块、管理模块前后端实现。 用户端：头像上传功能

1. 需求分析

1.1 登录注册模块

1.1.1 目标

实现功能完备，带较强有健壮性的登录注册页面

1.1.2 用户特点

分为两类人群：

- 用户：从电脑小白到电脑高手都有可能是我们的潜在用户
- 管理员：等级森严，不同等级的管理员权限不同

1.1.3 功能说明

- 默认页面是登录界面
- 登录界面下侧分别提供注册账户按钮和忘记密码按钮
- 登录页面右上角提供管理员登陆入口

1.2 用户信息模块

1.2.1 目标

实现功能完备，带较强有健壮性的用户信息模块

1.2.2 用户特点

- 具有定期更换头像的需求
- 具有定期更换密码的需求
- 具有修改电话号码的需求

1.2.3 功能说明

默认界面是一个表格，分别显示用户的用户名，真实姓名，身份证件，电话号码，用户类型，头像等，其中电话号码和头像可以进行修改。

1.3 管理员信息模块

1.3.1 目标

实现功能完备，带较强有健壮性的管理员信息模块

1.3.2 用户特点

具有以下两种类型的管理员

- 超级管理员，可以对其他所有的管理员的权限进行管理，并且拥有所有模块的权限
- 普通管理员，权限由超级管理员分配，并且可以对具有权限的模板进行管理

1.3.3 功能说明

- 对于超级管理员，页面左侧除了对线路车站，票务，车次，用户进行管理之外，还有一个管理员模块对管理员进行管理
- 对于普通管理员，页面左侧只能进入自己拥有权限的模块，无权限的模块将无法进入。

1.4 线路管理模块

1.4.1 目标

实现功能完备，带较强有健壮性的线路信息模块

1.4.2 用户特点

- 具有查看线路功能
- 具有添加线路功能

1.4.3 功能说明

- 对于查看线路，默认界面是一个表格，表格每一行显示一条线路，并且具有翻页，跳转，以及模糊搜索功能。
- 对于添加线路，默认界面是一个表格，需要输入线路名称，出发站，到达站，实际里程等信息才可以实现添加线路的功能。

1.5 车站管理模块

1.5.1 目标

实现功能完备，带较强有健壮性的车站管理信息模块

1.5.2 用户特点

- 具有查看车站功能
- 具有添加车站功能

1.5.3 功能说明

- 对于查看车站，默认界面是一个表格，表格每一行显示一个车站的信息，并且具有翻页，跳转，以及模糊搜索功能。
- 对于添加车站，默认界面是一个表格，需要输入车站名，车站等级等信息才可以实现添加车站的功能。

1.6 票务管理模块

1.6.1 目标

实现功能完备，带较强有健壮性的票务管理信息模块

1.6.2 用户特点

- 具有查看已购票功能
- 具有进行车票购买功能（实现售票员功能）

1.6.3 功能说明

- 对于查看已售出票，默认界面是一个表格，表格每一行显示一个车站的信息，并且具有翻页，跳转，以及模糊搜索功能。
- 对于车票购买功能，默认界面是一个表格，需要输入用户凭证类型，用户凭证，车次号，出发日期，座位等级等才能够进行购票。

1.7 车次管理模块

1.7.1 目标

实现功能完备，带较强有健壮性的车次管理信息模块

1.7.2 用户特点

- 具有查看车次功能
- 具有添加车次功能

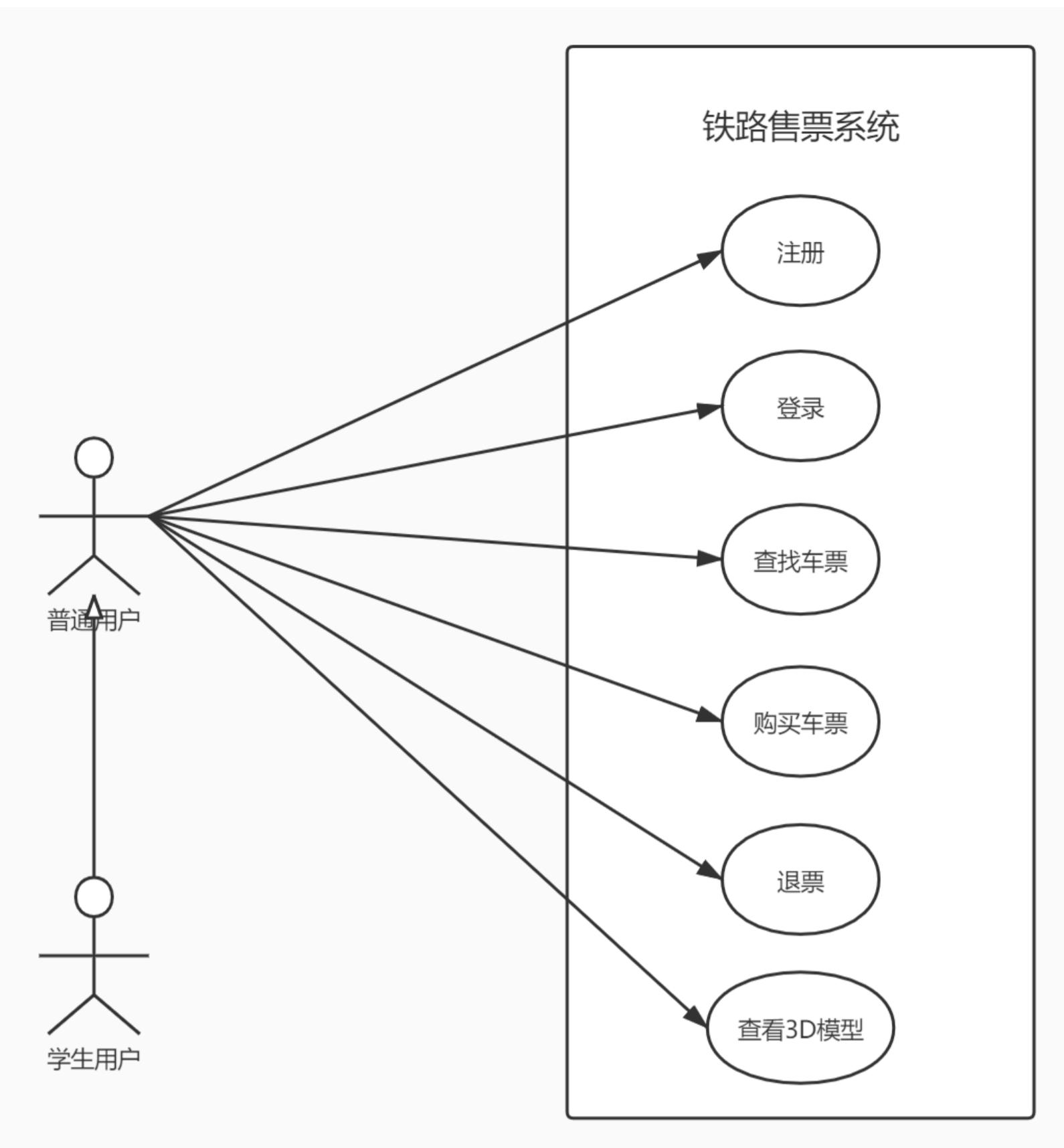
1.7.3 功能说明

- 对于查看车次，默认界面是一个表格，表格每一行显示一个车次的信息，并且具有翻页，跳转，以及模糊搜索功能。
- 对于添加车次，默认界面是一个表格，需要输入，车次号，车号，线路号，出发时间和到达时间等信息才可以实现添加车次的功能。

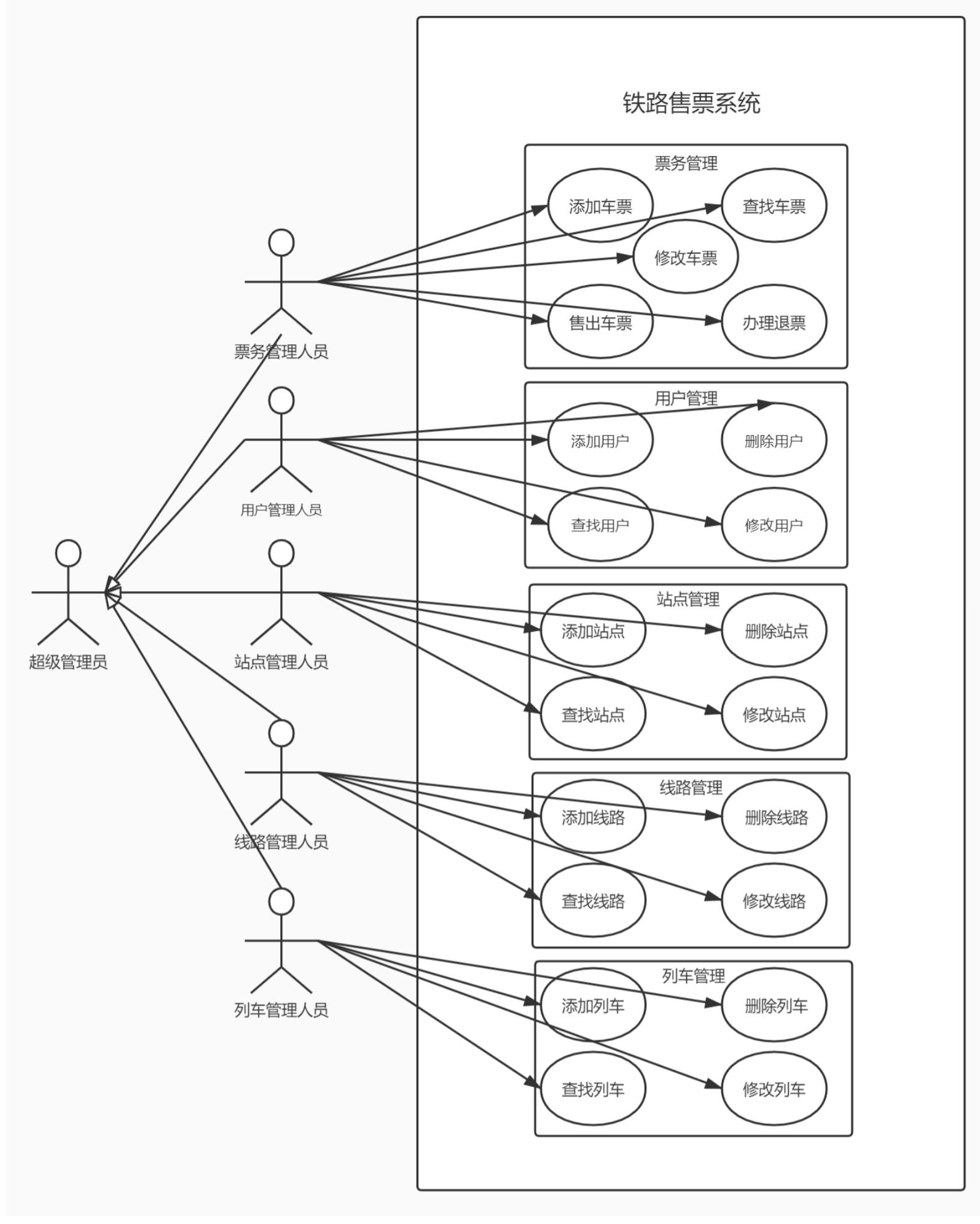
2. UML

2.1 用例图（多补几张）

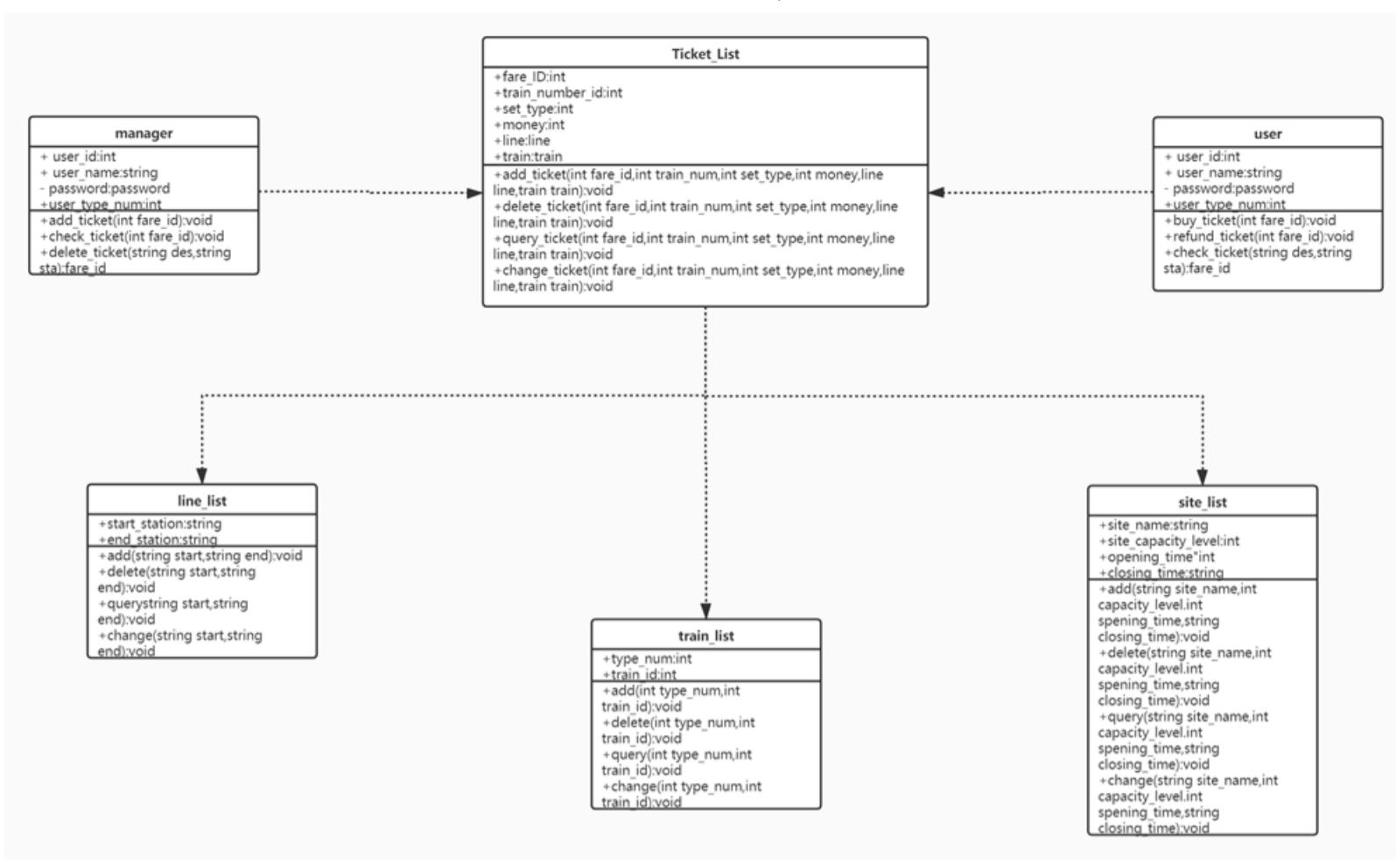
1. 普通用户 用例图



2. 管理员用户 用例图



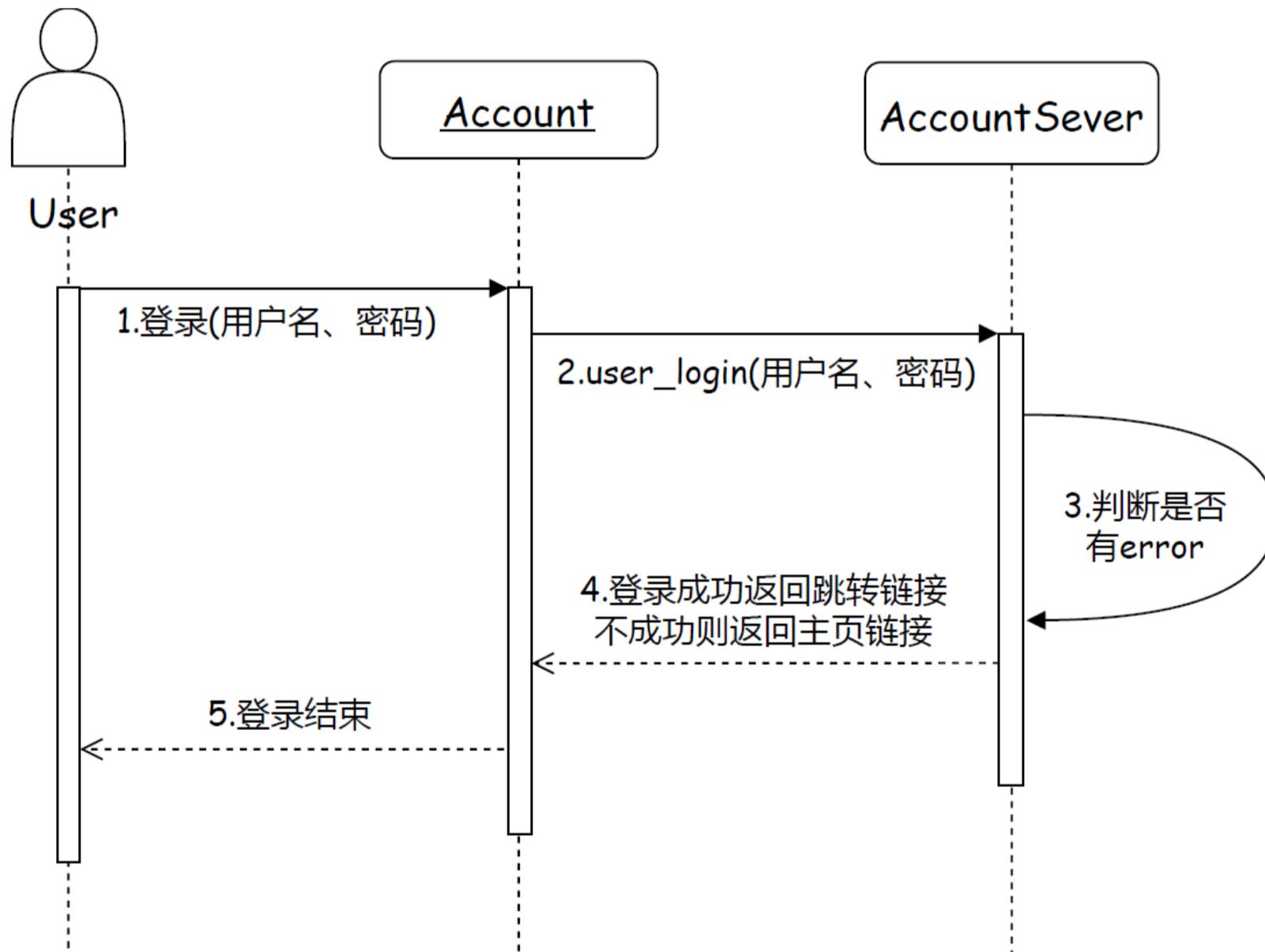
2.2 类图



2.3 顺序图

1. 登录 顺序图

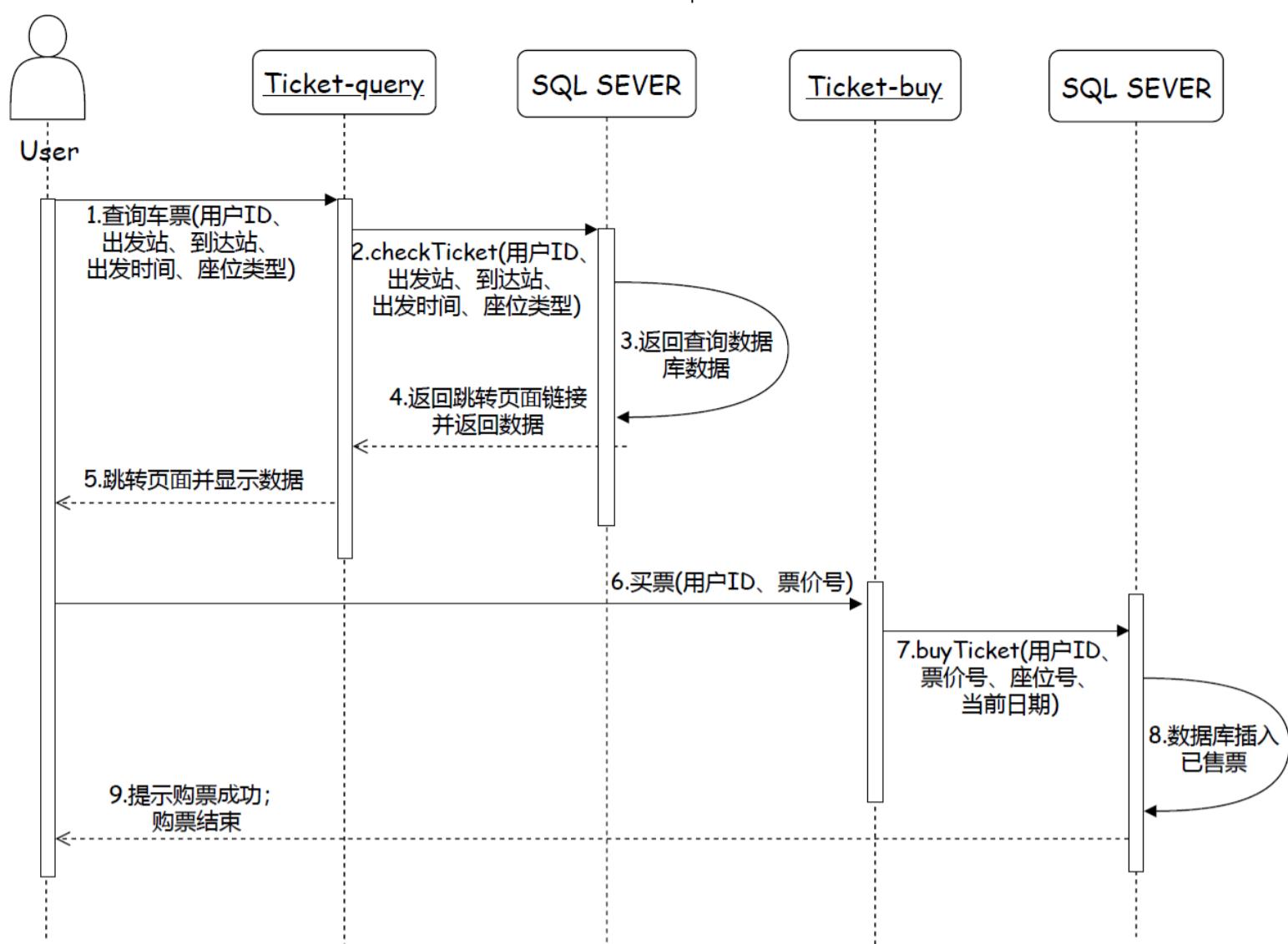
传入参数: 用户名(User_id)、密码(password)



2. 普通用户 购票、退票 顺序图

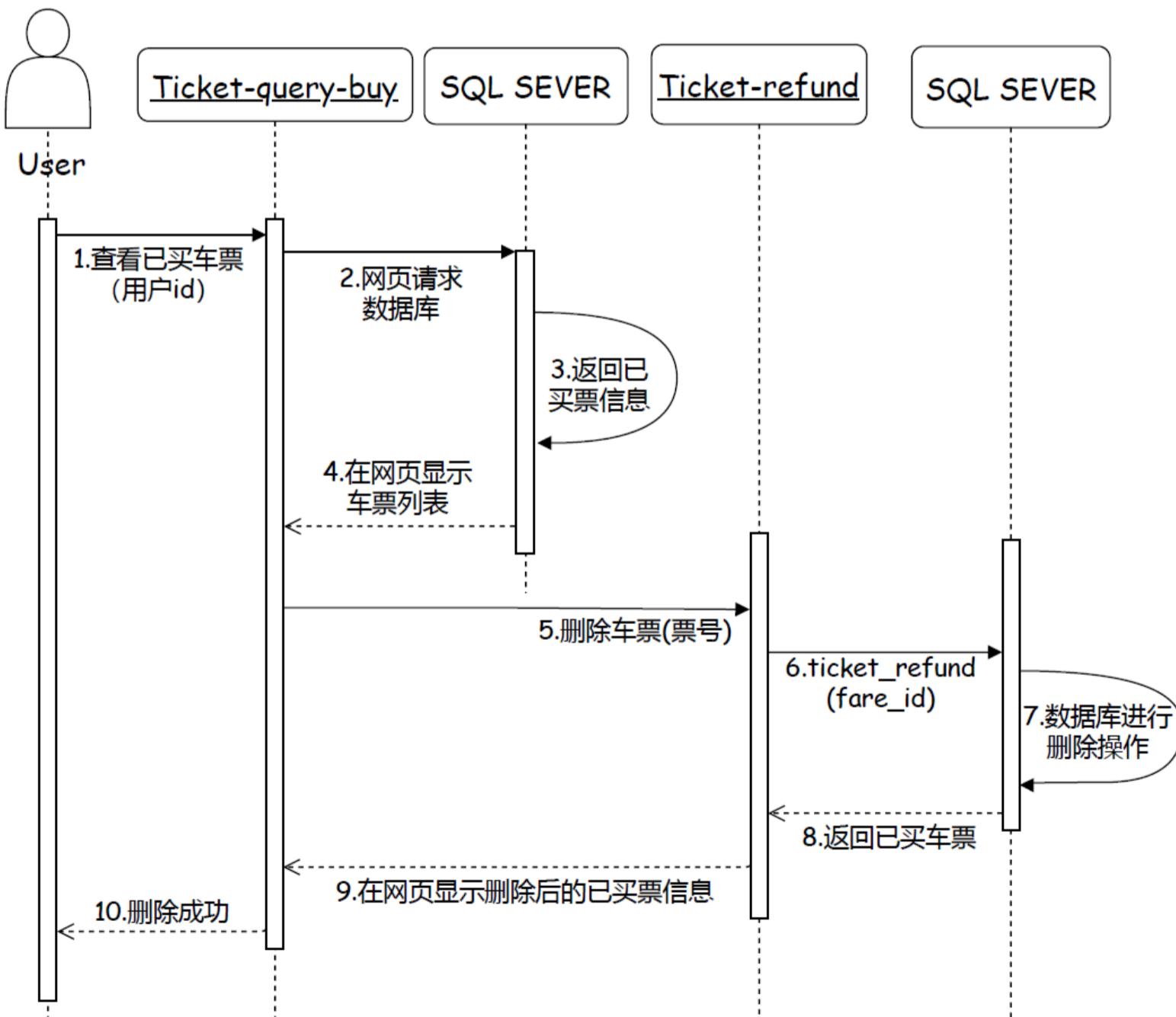
- 购票

传入参数: 用户ID(User_id)、出发站(departure_station)、到达站(arrival_time)、出发时间(departure_time)、到达时间(arrival_time)、座位类型(seat_type)



- 退票

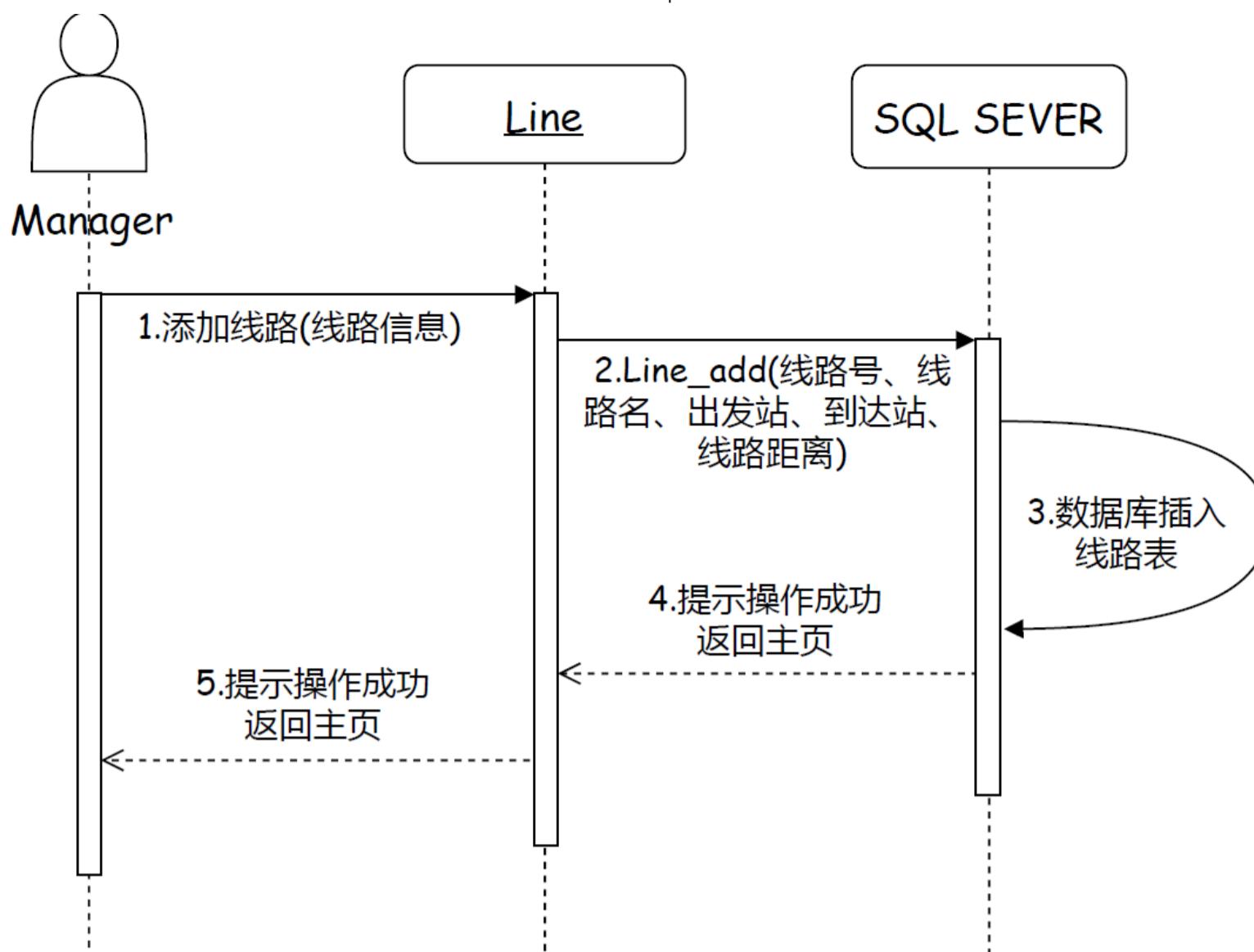
传入参数: 用户ID(*User_id*)



3. 管理员用户 添加线路、修改站点、删除车票 顺序图

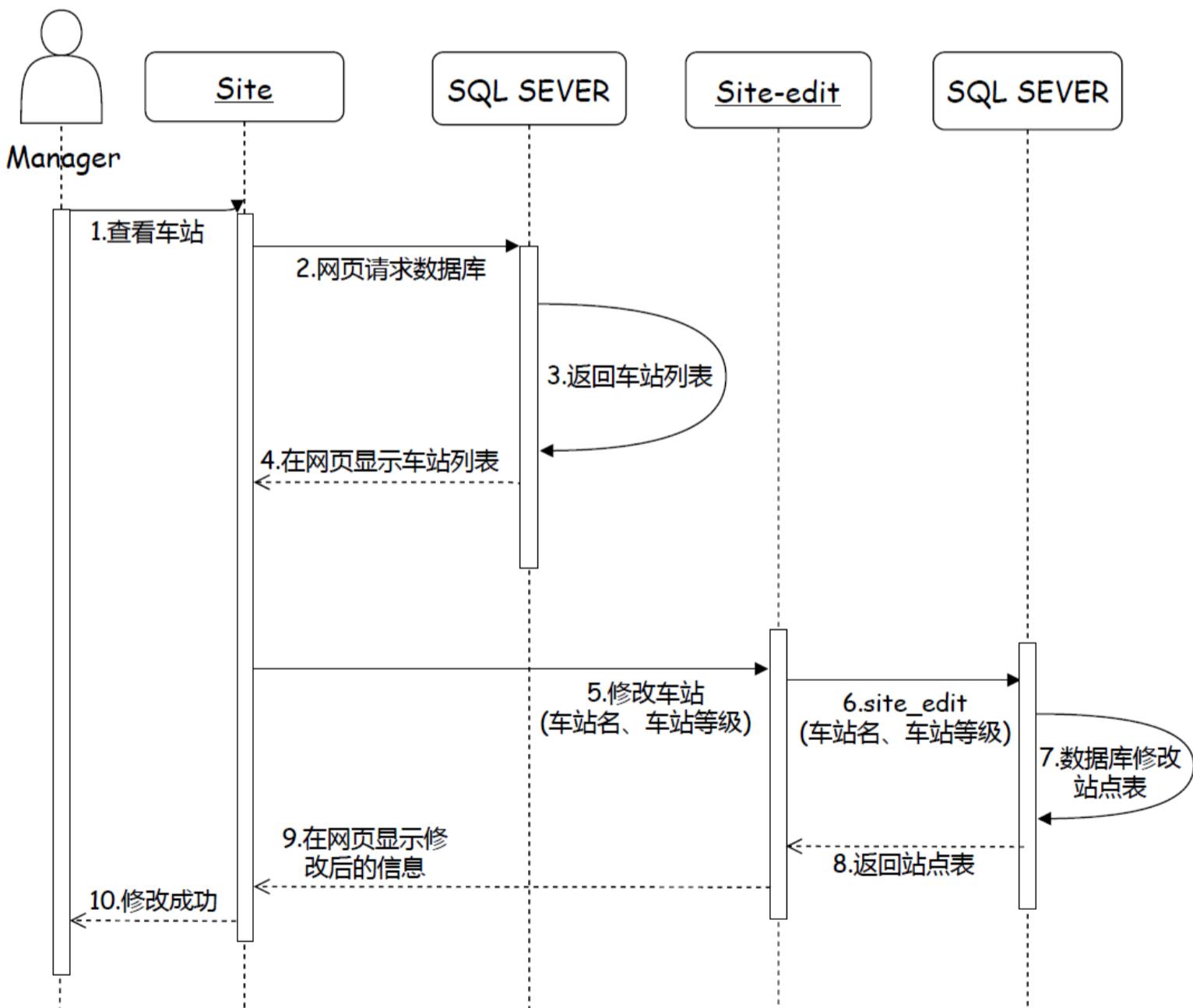
- 添加线路

传入参数: 线路号(*Line_id*)、线路名(*Line_name*)、出发站(*departure_station*)、到达站(*arrival_time*)、线路距离(*Line_distance*)



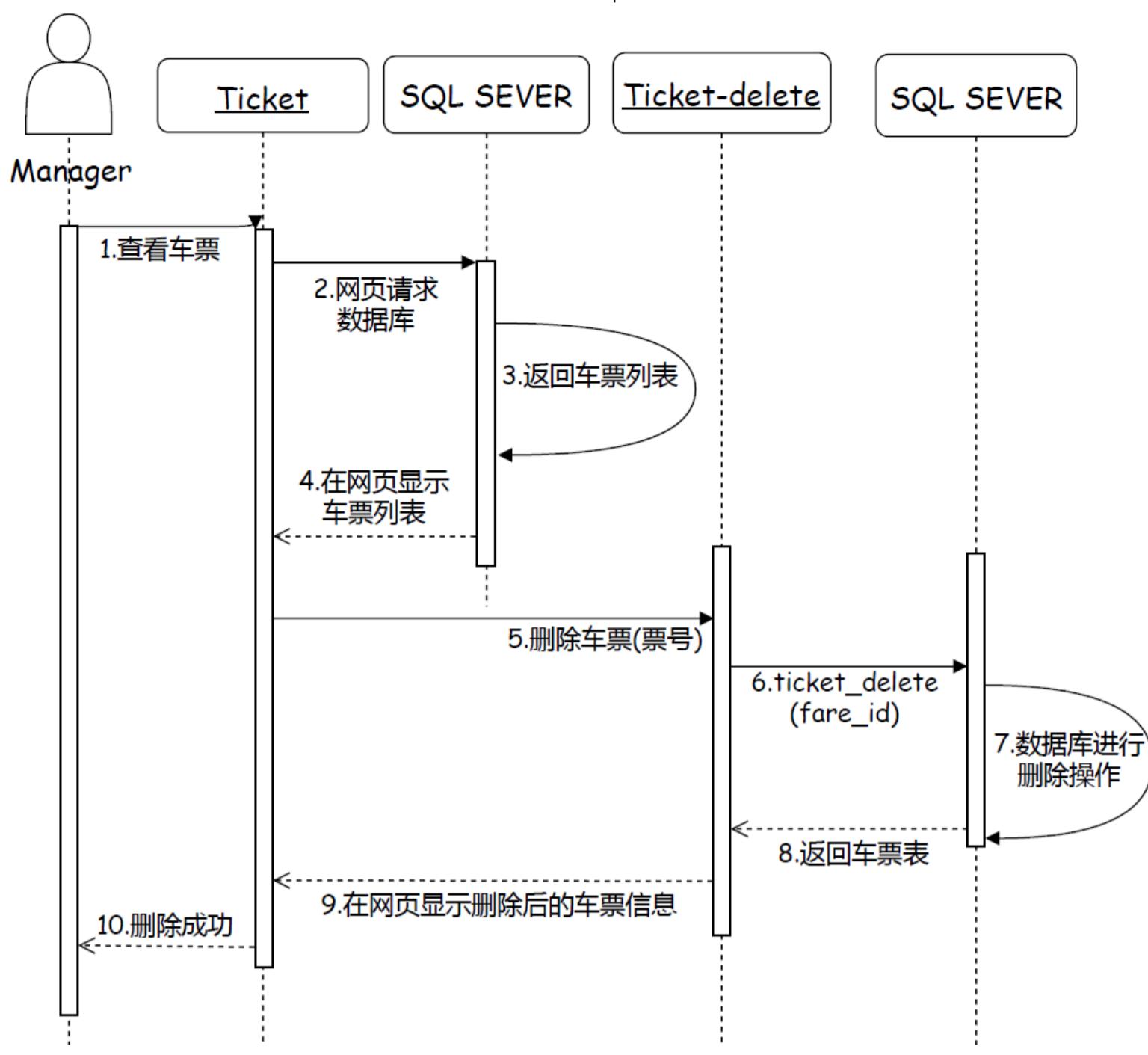
- 修改站点

传入参数: 车站名(site_name)、车站等级(site_type)

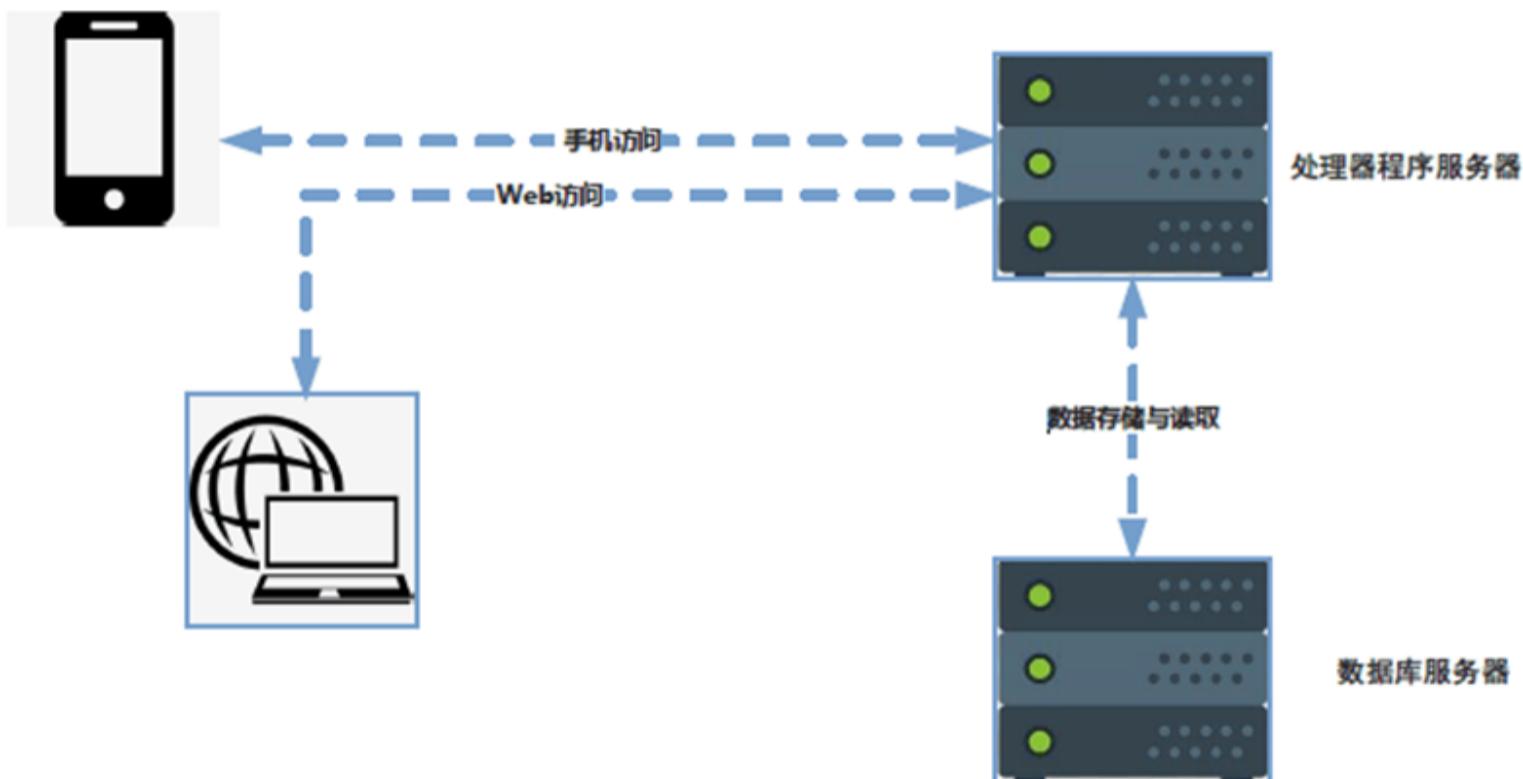


- 删除车票

传入参数: 票号(fare_id)



2.4 部署图



3. 数据库设计

3.1 表的设计

3.1.1 工作人员信息表

1 | 工作人员信息表主要囊括了工作人员的基本信息，包括ID号、工作人员用户名、密码和部门类型号，其中ID号为主键。

3.1.2 用户信息表

1 | 用户信息表主要囊括了用户的基本信息，包括ID号，用户名、密码和用户类型号，其中ID号为主键。

3.1.3 站点信息表

1 | 站点信息表主要囊括各个站点的基本信息，包括站点名、站点容量级别、开站时间和闭站时间，其中站点名为主键。

3.1.4 线路表

1 | 线路表主要是各条线路的基本信息，包括线路号、名称、到达站、出发站和线路距离，其中线路号是主键，到达站和出发站的外键是站点信息表中的站点名。

3.1.5 车类表

1 | 车类表描述了系统中包括的列车类型，包括该车类的类型号、一等座数和二等座数，其中类型号是主键。

3.1.6 火车信息表

1 | 火车信息表主要囊括了所有火车的信息，包括车编号、类型号，其中车编号是主键，类型的外键是车类表的类型号。

3.1.7 车次表

1 | 车次表主要描述了各个车次的信息，包括车次、车编号、线路号、出发时间和到达时间，其中车次是主键，车编号的外键是火车信息表的车编号，线路号的外键是线路表的编号。

3.1.8 票价信息表

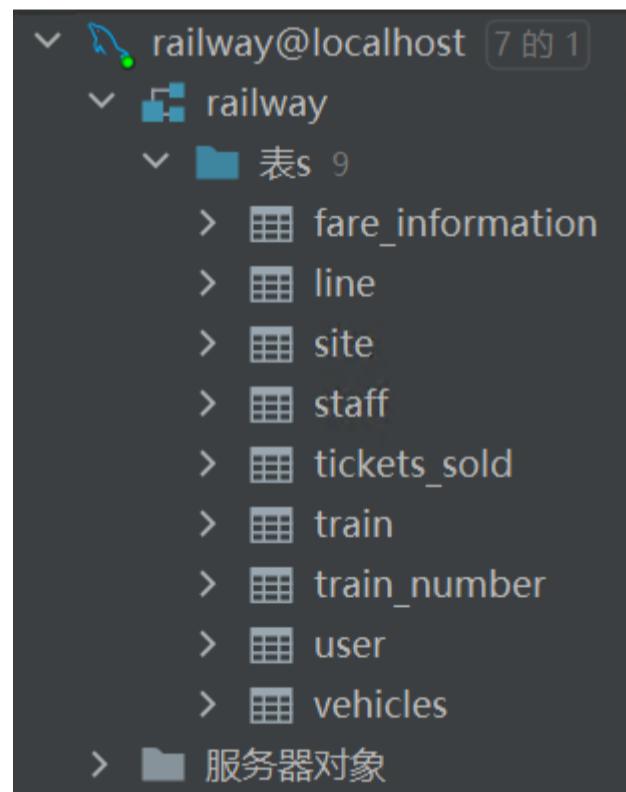
1 | 票价信息主要描述了系统内各类票的情况，包括票价号、车次、座位类型和票价，其中票价号是主键，车次的外键是车次表的车次。

3.1.9 已售票信息表

1 | 已售票信息表记录所有的售票记录，包括票号、票价号、用户ID和座位号，其中票价号的外键是票价信息表的票价号，用户ID的外键是用户信息表的ID号。

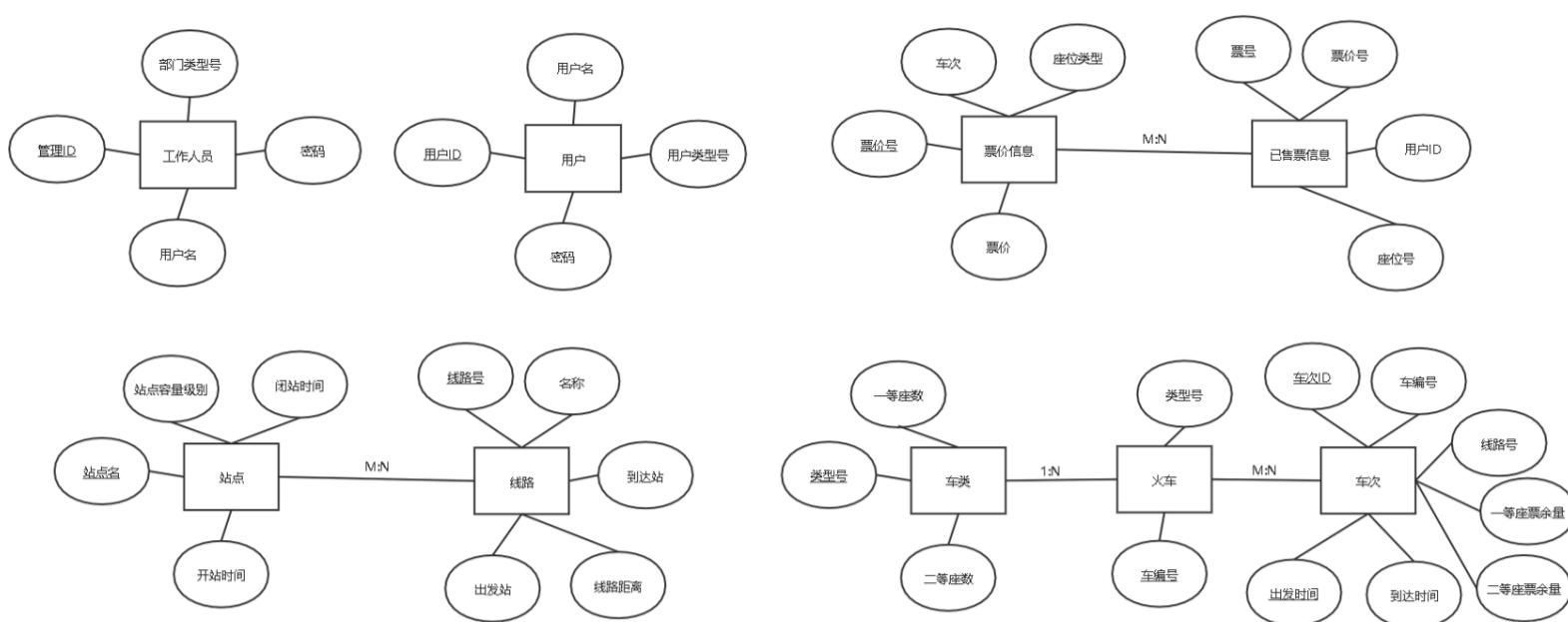
3.2 数据库构建

1 | 数据库采用MySQL，数据库管理IDE选用DataGrip，具体编程采用Python3.8+flask_sqlalchemy+Pycharm实现。最后生成的数据库如下图所示。



3.3 ER图的绘制

1 | 基于各个表的信息，参照主键、外键约束，绘制出ER图如下所示。



4. 系统功能实现

4.1 用户端

4.1.1 首页

本模块为用户系统首页，同时也是整个系统的主页，为默认页

1. 默认页代码，默认用户名为游客（html中设置）

```
1 | @app.route('/')
2 | def hello():
3 |     return redirect(url_for('user_bp.user_index'))
```

2. 界面展示



Our Services

我们提供的服务



4.1.2 登录注册

此模块是为普通用户登录注册以及忘记密码等相关功能

4.1.2.1 登陆

首先介绍登录功能主要的逻辑实现

1. 获取前端输入的用户名与密码，并在数据库数据查询

```
1 | login_username = request.form['user']
2 | login_password = request.form['password']
3 | login_user = User.query.filter(User.user_name == login_username).first()
```

2. 处理错误抛出异常信息，采用flash在前端渲染

```
1 | if login_user is None:
2 |     login_error = 'User {} does not exist!'.format(login_username)
3 | elif login_user.password != login_password:
4 |     login_error = 'Incorrect password!'
```

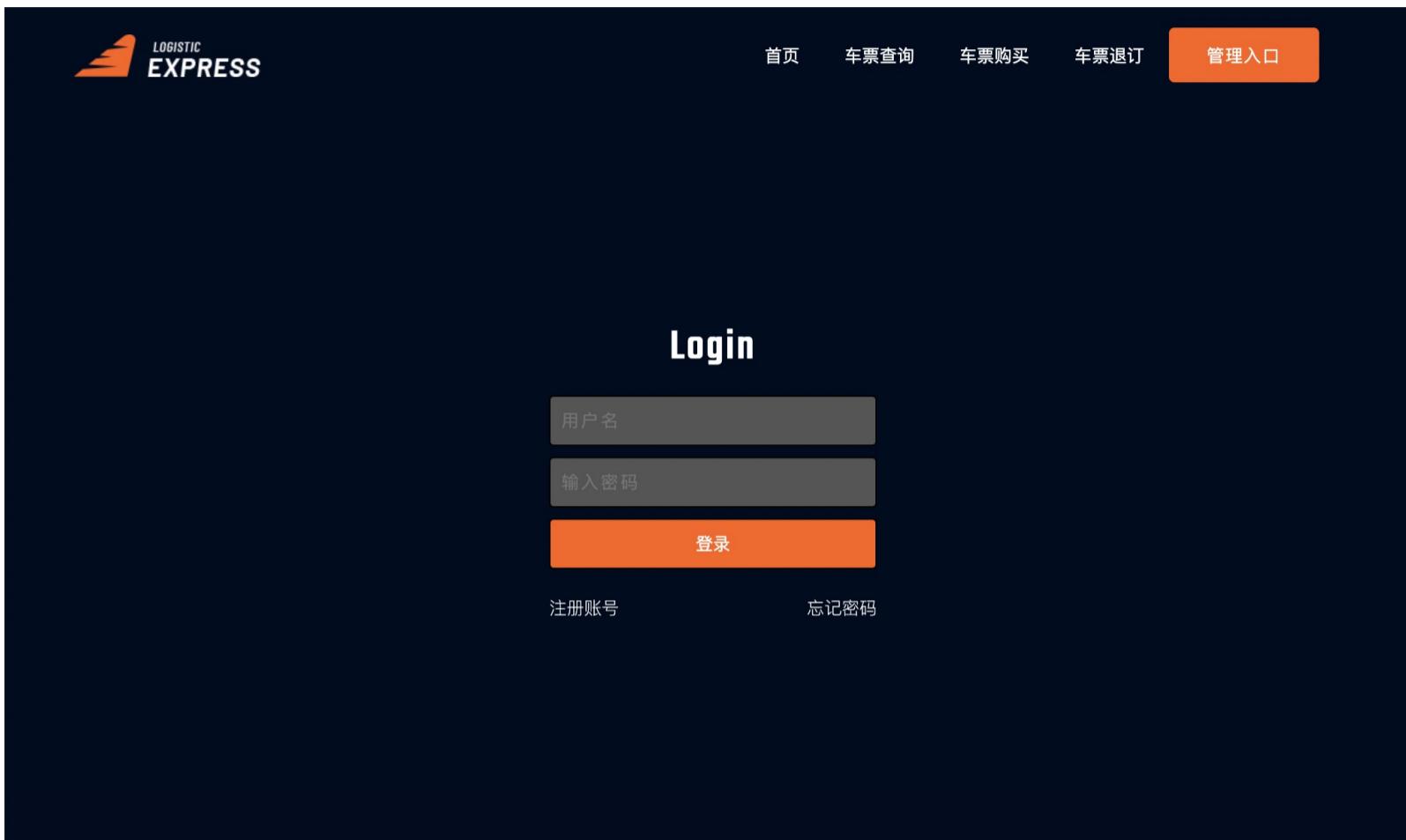
3. 若无错误，再次进入主页，但会显示采用设置好的cookie来获取对应用用户名

```

1 if login_error is None:
2     session.clear()
3     session['user_ID'] = login_user.user_ID
4     respond = redirect(url_for('user_bp.user_index'))
5     respond.set_cookie('customer_id', str(login_userid))
6     respond.set_cookie('customer_name', login_username)

```

4. 页面展示



4.1.2.2 注册

1. 获取前端输入的用户名、密码、重复密码、身份与头像信息

```

1 reg_username = request.form.get('user')
2 reg_password1 = request.form.get('password1')
3 reg_password2 = request.form.get('password2')
4 if request.form.get('browser') == '成人':
5     user_type = 0
6 else:
7     user_type = 1

```

2. 进行输入信息的简单验证，例如二次密码是否一致等

```

1 if reg_password1 != reg_password2:
2     reg_error = 'Inconsistent password.'
3 elif User.query.filter(User.user_name == reg_username).first() is not None:
4     reg_error = 'User {} is already registered.'.format(reg_username)

```

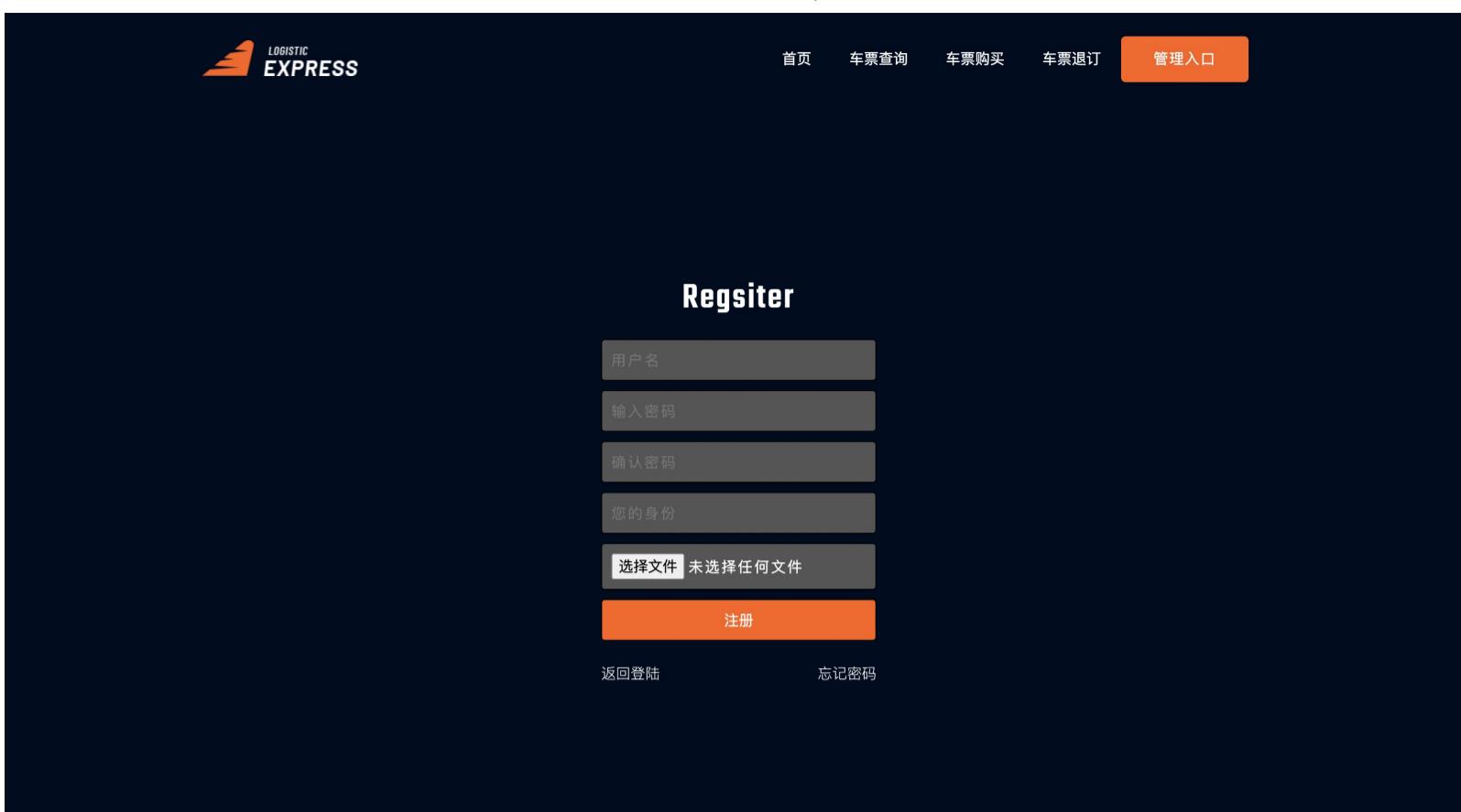
3. 若无错误，则向数据库提交，向用户表添加一条新用户信息

```

1 reg_user = User(user_name=reg_username, password=reg_password1, user_type_number=user_type)
2 db.session.add(reg_user)
3 db.session.commit()

```

4. 页面展示



4.1.2.3 忘记密码

若进入忘记密码逻辑，可直接输入用户名与新密码，

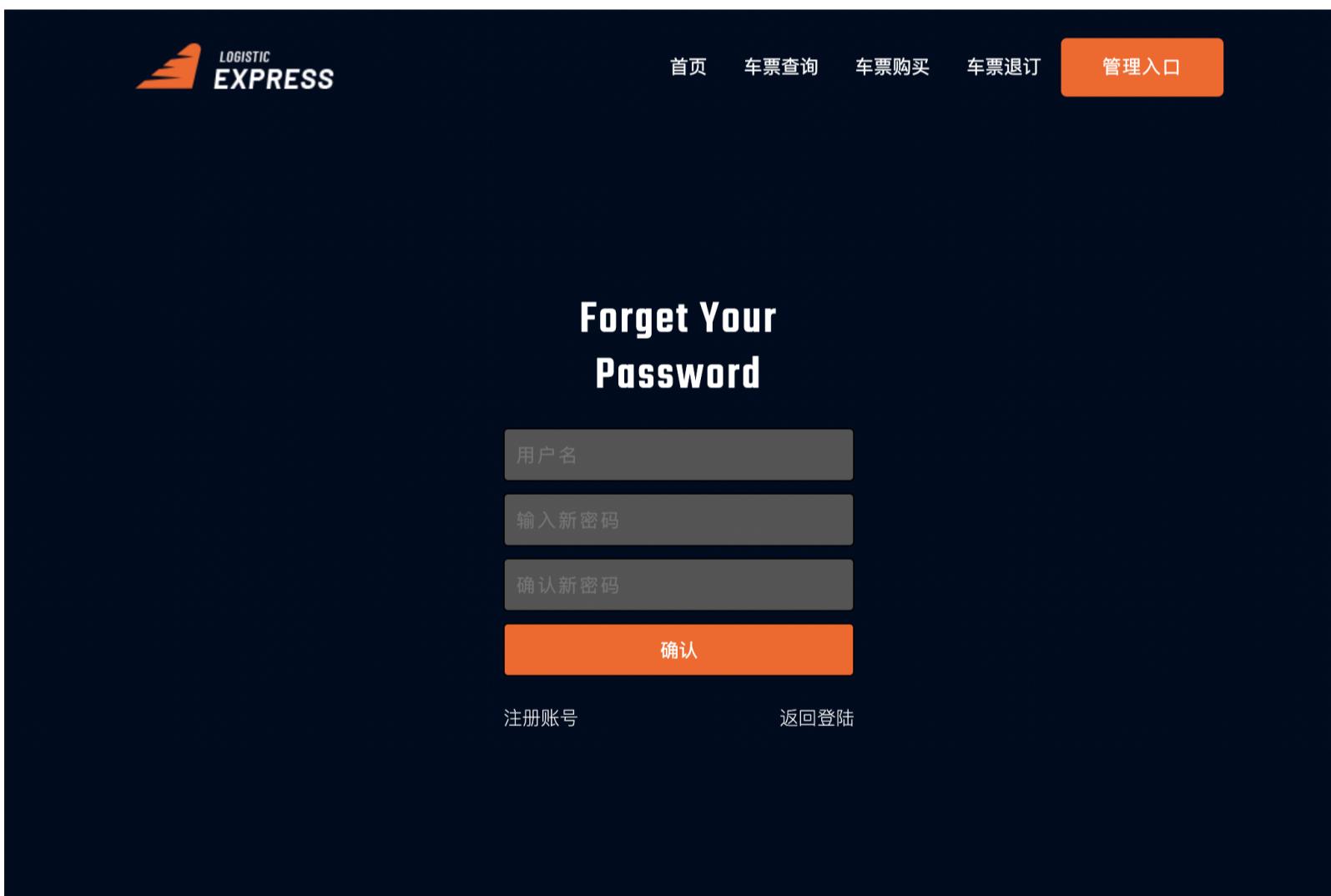
1. 获取前端输入的用户名、新密码、重复密码

```

1 | reg_username = request.form.get('user')
2 | reg_password1 = request.form.get('password1')
3 | reg_password2 = request.form.get('password2')

```

2. 页面展示



4.1.2 用户信息

本模块为登陆后显示个人信息页面，且可直接修改

1. 采用cookie获取用户名，根据此用户名在数据库查询获取相关信息进行显示

```

1 | reg_username = request.cookies.get('customer_name')
2 | user=User.query.filter(User.user_name == reg_username).first()

```

2. 页面展示



修改个人信息

个人信息

未选择任何文件

4.1.3 购买车票

- 购票逻辑第一步为输入起始站、终点站、座等与出发时间

```

1 | buy_arrival_station = request.form['arrival_station']
2 | buy_departure_station = request.form['departure_station']
3 | temp = request.form['departure_time']
4 | buy_departure_time = datetime.datetime.strptime(temp, "%H:%M:%S")
5 | seat_type = request.form['seat_type']

```

- 购买前置查询页面



请输入购票信息

上午 07:00:01

确认信息

3. 后端通过cookie获取的用户身份以及输入的数据查询对应车次

```

1 checkbuytickets = db.session.query(FareInformation.fare_ID, TrainNumber.train_number_ID, Line.line_ID,
2     FareInformation.seat_type,
3     Line.departure_station, Line.arrival_station, TrainNumber.departure_time, TrainNumber.arrival_time). \
4     filter(TrainNumber.train_number_ID == FareInformation.train_number_id,
5            TrainNumber.line_ID == Line.line_ID,
6            Line.departure_station == buy_departure_station,
7            Line.arrival_station == buy_arrival_station,
8            FareInformation.seat_type == buy_seat_type,
9            TrainNumber.departure_time >= buy_departure_time).all()
10 site_list = db.session.query(site).filter().all()

```

4. 购买页面



Result

火车号	线路号	座位类型	起点站	终点站	出发时间	到达时间	操作
D1626	18	1	Guangzhou	Nanjing	18:22:48	23:53:44	购买



4.1.4 查询车票

用户端提供车票的检索功能

车次	始发站-终点站	出发站	到达站	出发时间	到达时间
D1304	Shanghai-Wuhan	Wuhan	Shanghai	00:40:33	08:40:09
D1304	Shanghai-Wuhan	Wuhan	Shanghai	00:40:33	08:40:09
D1626	Nanjing-Guangzhou	Guangzhou	Nanjing	18:22:48	23:53:44
D1626	Nanjing-Guangzhou	Guangzhou	Nanjing	18:22:48	23:53:44
D2588	Guangzhou-Chengdu	Chengdu	Guangzhou	05:24:16	08:20:32
D2588	Guangzhou-Chengdu	Chengdu	Guangzhou	05:24:16	08:20:32
D2700	Chengdu-Beijing	Beijing	Chengdu	12:09:56	14:05:57
D2700	Chengdu-Beijing	Beijing	Chengdu	12:09:56	14:05:57
D2744	Wuhan-Beijing	Beijing	Wuhan	03:08:50	13:51:56
D2744	Wuhan-Beijing	Beijing	Wuhan	03:08:50	13:51:56
D3276	Guangzhou-Shanghai	Shanghai	Guangzhou	17:43:27	22:10:41
D3276	Guangzhou-Shanghai	Shanghai	Guangzhou	17:43:27	22:10:41

4.1.5 退订车票

用户端提供车票退订的功能，进入模块显示当前用户的所有购票记录，在各行末尾设置一个退订按钮，点击实现退票



My Tickets

票号	车次	线路名	出发时间	座位类型	票价	座位号	操作
5	G2547	Beijing-Shanghai	13:59:33	2	1029	1669	<button>退订</button>
6	D6628	Guangzhou-Wuhan	14:25:32	2	4869	8532	<button>退订</button>
20	G6621	Beijing-Shanghai	14:01:46	1	4596	7791	<button>退订</button>
23	D2700	Chengdu-Beijing	12:09:56	2	6346	9429	<button>退订</button>

4.2 管理端

4.2.1 登录

此模块是为管理员用户登录使用，现以登录为例，介绍管理员登录模块主要的逻辑实现

- 首先从前端的HTML读入用户名和密码，进入数据库查询结果

```

1 login_error = None
2 login_username = request.form['login_username']
3 login_password = request.form['login_password']
4 login_user = UserStaff.query.filter(UserStaff.user_name == login_username).first()

```

- 然后处理错误异常信息，如果产生错误，将错误信息存储在flash中，稍后会以消息闪现的方式在HTML上进行渲染，用户将继续留在登录界面

```

1 if login_user is None: # 用户名错误
2     login_error = 'Incorrect username'
3 elif not check_password_hash(login_user.password, login_password): # 密码以哈希形式存储在数据库，保证安全
4     login_error = 'Incorrect password'
5 flash(login_error)

```

- 如果没有发生错误，跳转到管理员主页，并为浏览器设置cookie，免去下次用户打开浏览器重复验证身份的麻烦

```

1 if login_error is None:
2     session.clear()
3     session['user_ID'] = login_user.staff_ID
4
5     respond = redirect(url_for('manager_bp.manager_index'))
6     respond.set_cookie('user_name', login_username)
7     respond.set_cookie('user_type', str(login_user.department_type_number))

```

- 界面效果展示

The image contains two screenshots of a web-based Railway Manager system. Both screenshots show a 'Login' page with fields for 'Username' and 'Password'. Below the password field is a 'Sign In' button. The bottom screenshot includes a red error message 'Incorrect username' next to the 'Sign In' button.

4.2.2 用户信息

此模块位于管理员部分，用于管理普通用户，只有超级管理员和用户管理管理员的模块权限，主要实现以下功能：

1. 普通用户的增加，相当于实际生活中，铁路工作人员帮助普通用户注册账号。点击最左侧菜单栏中“添加用户”，进入相关界面，输入用户名和用户身份，确认密码后点击提交即可。

The image shows a detailed view of the 'Add New User' form. The form has fields for '用户名 *', '新密码 *', and '确认密码 *'. Below these fields is a '用户身份' dropdown with options '学生用户' (highlighted in blue) and '非学生用户'. At the bottom of the form are three buttons: 'Cancel', 'Reset', and 'Submit' (highlighted in green).

2. 普通用户的删除，相当于实际生活中，铁路工作人员删除某购票账号。在用户信息表中点击Delete，即可删除该用户

普通用户号	普通用户名	身份	操作
1	lilm	非学生用户	Delete Edit
2	zyp	非学生用户	Delete Edit
3	jjq	学生用户	Delete Edit
4	szw	学生用户	Delete Edit
5	yzh	学生用户	Delete Edit
6	aaa	非学生用户	Delete Edit
7	lllm	学生用户	Delete Edit
8	hhh123	非学生用户	Delete Edit
9	wgt	非学生用户	Delete Edit
10	test	学生用户	Delete Edit

3. 普通用户的修改，相当于实际生活中，管理员帮助某用户修改其基本账号信息。在上面的页面中点击Edit，进入对单个用户的修改页面，注意此时不能修改用户名，仅仅能修改密码和用户身份。

4. 普通用户的查看，可以查找某一特定用户的购票信息。在用户信息表，点击想要查询的用户名，可以看到该用户的详细信息，包括车号、线路（起始地、目的地）、座位号、座位类型等。

序号	列车	线路	座位号	座位类型	时间	出发时间	到达时间	车费
1	G2547	Beijing-Shanghai	1669	二等座	2020-01-26	13:59:33	15:17:09	1029 RMB
2	D6628	Guangzhou-Wuhan	8532	二等座	2020-02-01	14:25:32	15:34:49	4869 RMB
3	G6621	Beijing-Shanghai	7791	一等座	2020-06-07	14:01:46	23:56:06	4596 RMB
4	D2700	Chengdu-Beijing	9429	二等座	2021-06-11	12:09:56	14:05:57	6346 RMB
5	G4015	Chengdu-Shanghai	2795	二等座	2021-12-20	18:28:48	21:23:23	4934 RMB
6	G3987	Wuhan-Nanjing	8452	一等座	2021-12-20	02:26:12	10:45:01	7543 RMB
7	D6070	Chengdu-Guangzhou	8658	二等座	2021-10-13	18:02:42	22:39:47	9578 RMB
8	D2700	Chengdu-Beijing	3247	一等座	2021-10-25	12:09:56	14:05:57	9423 RMB
9	D8792	Guangzhou-Chengdu	3992	二等座	2021-01-03	03:49:44	19:08:33	4405 RMB
10	G9809	Shanghai-Wuhan	5930	二等座	2020-08-07	05:09:06	08:38:14	3527 RMB

4.2.3 管理员信息

此模块用于对管理员的管理，只有超级管理员有该模块的权限，注意超级管理员只有一个，可以注册、修改、删除、查找上述五种权限的管理员，主要功能如下：

1. 管理员注册，在菜单栏点击“添加管理员”即可，注册完成后的管理员用户，可以直接在登录界面登录，并拥有对某一模块的管理权限。

The screenshot shows the 'Add New Admin' form. It includes fields for 'New Username*', 'New Password*', 'Confirm Password*', and 'User Identity*' (with 'Line Management' selected). Below the form are 'Cancel', 'Reset', and 'Submit' buttons.

2. 管理员修改，点击管理员信息表中的Edit按钮，可以给对应的管理员修改权限和密码。

The screenshot shows the 'Modify Admin' form for user 'zlb'. It includes fields for 'New Username*', 'New Password*', 'Confirm Password*', and 'User Identity*' (set to 'Line Management'). Below the form are 'Cancel', 'Reset', and 'Submit' buttons.

3. 管理员查找，可以直接在管理员信息表输入查找的信息，此功能基于JavaScript实现

The screenshot shows the 'View Admin' page. It displays a table titled 'Administrator Information Table' with one entry:序号 1, 管理员名 llb, 权限 超级管理员, 操作 无权限操作. The search input field at the top right contains 'llb'.

4. 管理员删除，直接点击管理员信息表中Delete按钮，可以删除该用户

序号	管理员名	权限	操作
1	llb	超级管理员	无权限操作
2	zlb	线路管理	Delete Edit
3	wlb	车站管理	Delete Edit
4	jlb	票务管理	Delete Edit
5	sib	列车管理	Delete Edit
6	ylb	普通用户管理	Delete Edit
9	swjtu	列车管理	Delete Edit

4.2.4 线路管理

- 该模块可供对应权限管理员浏览并编辑，当前铁路数据库所有的线路信息

线路号	线路名称	出发站	到达站	实际里程	操作
1	Beijing-Chengdu	Chengdu	Beijing	999	Delete Edit
2	Beijing-Guangzhou	Guangzhou	Beijing	648	Delete Edit
3	Beijing-Nanjing	Nanjing	Beijing	904	Delete Edit
4	Beijing-Shanghai	Shanghai	Beijing	1426	Delete Edit
5	Chengdu-Beijing	Beijing	Chengdu	4168	Delete Edit
6	Chengdu-Guangzhou	Guangzhou	Chengdu	3901	Delete Edit
7	Chengdu-Nanjing	Nanjing	Chengdu	4030	Delete Edit
8	Chengdu-Shanghai	Shanghai	Chengdu	4859	Delete Edit
9	Chengdu-Wuhan	Wuhan	Chengdu	2545	Delete Edit
10	Guangzhou-Beijing	Beijing	Guangzhou	1577	Delete Edit

- 可对应进行模糊搜索，例如查看所有含成都站的线路信息

线路号	线路名称	出发站	到达站	实际里程	操作
1	Beijing-Chengdu	Chengdu	Beijing	999	Delete Edit
5	Chengdu-Beijing	Beijing	Chengdu	4168	Delete Edit
6	Chengdu-Guangzhou	Guangzhou	Chengdu	3901	Delete Edit
7	Chengdu-Nanjing	Nanjing	Chengdu	4030	Delete Edit
8	Chengdu-Shanghai	Shanghai	Chengdu	4859	Delete Edit
9	Chengdu-Wuhan	Wuhan	Chengdu	2545	Delete Edit
11	Guangzhou-Chengdu	Chengdu	Guangzhou	1495	Delete Edit
16	Nanjing-Chengdu	Chengdu	Nanjing	1878	Delete Edit
21	Shanghai-Chengdu	Chengdu	Shanghai	2880	Delete Edit
25	Wuhan-Chengdu	Chengdu	Wuhan	4851	Delete Edit

Showing 1 to 10 of 10 entries (filtered from 29 total entries)

Previous 1 Next

3. 对应插件可直接对数据进行复制、打印等操作，例如打印

The screenshot shows a modal window titled "打印" (Print) over a table of train routes. The modal has a dropdown for "目标打印机" (Target Printer) set to "另存为 PDF" (Save as PDF). It also has dropdowns for "网页" (Webpage) set to "全部" (All) and "布局" (Layout) set to "纵向" (Portrait). There is a "更多设置" (More Settings) button at the bottom.

4. 可通过操作栏进行删除，删除则直接从数据库级联删除

1	Beijing-Chengdu	Chengdu	Beijing	999	Delete Edit
5	Chengdu-Beijing	Beijing	Chengdu	4168	Delete Edit
6	Chengdu-Guangzhou	Guangzhou	Chengdu	3901	Delete Edit

5. 点击Edit则可进入修改页面，以上图1号线为例，将里程修改为2345

正在修改 Beijing-Chengdu

线路

线路名称 *	Beijing-Chengdu
出发站 *	Chengdu
到达站 *	Beijing
实际里程	2345

Cancel Reset Submit

线路号	线路名称	出发站	到达站	实际里程	操作
1	Beijing-Chengdu	Chengdu	Beijing	2345	Delete Edit

6. 同时也可增添一个新线路，但线路的站点必须存在于车站表，例如接下来演示将Beijing- Chengdu删除后再重新添加

线路号	线路名称	出发站	到达站	实际里程	操作
1	Beijing-Chengdu	Chengdu	Beijing	2345	Delete Edit

Copy	CSV	Print						Search: <input type="text" value="beijing-chengdu"/>			
线路号	↓	线路名称	↑	出发站	↓	到达站	↑	实际里程	↑	操作	↑
No matching records found											

添加新线路

线路

线路名称 *	<input type="text" value="Beijing-Chengdu"/>
出发站 *	<input type="text" value="Beijing"/>
到达站 *	<input type="text" value="Chengdu"/>
实际里程	<input type="text" value="3601"/>

[Cancel](#) [Reset](#) [Submit](#)

线路信息表					
Copy	CSV	Print	Search: <input type="text" value="Beijing-chen"/>		
线路号	↓	线路名称	↑	出发站	↓
29		Beijing-Chengdu		Beijing	Chengdu
Showing 1 to 1 of 1 entries (filtered from 29 total entries)					
Previous 1 Next					

4.2.5 车站管理

车站模块逻辑功能与线路模块相似，同样可实现增删改查，车站表是更为基础的表项，不依赖于任何外键

1. 查询功能 (同样具有模糊查找和插件功能)

站点信息表

站点信息总览					
Copy	CSV	Print	Search: <input type="text"/>		
车站号	↓	车站名称	↑	车站等级	↑
1		Beijing		一等站	Delete Edit
2		Chengdu		二等站	Delete Edit
3		Guangzhou		一等站	Delete Edit
4		Nanjing		一等站	Delete Edit
5		Shanghai		特等站	Delete Edit
6		Wuhan		二等站	Delete Edit

Showing 1 to 6 of 6 entries

[Previous](#) [1](#) [Next](#)

2. 修改功能 (修改站点等级)

正在修改 Beijing

车站

车站名 *	Beijing
车站等级	特等站 <input checked="" type="checkbox"/> 一等站 二等站 三等站
<input type="button" value="Cancel"/> <input type="button" value="Reset"/> <input type="button" value="Submit"/>	

3.删除功能（同样位于操作栏），删除后级联删除相关线路

车站号	车站名称	车站等级	操作
1	Beijing	一等站	Delete Edit
2	Chengdu	二等站	Delete Edit
3	Guangzhou	一等站	Delete Edit
4	Nanjing	一等站	Delete Edit
5	Shanghai	特等站	Delete Edit
6	Wuhan	二等站	Delete Edit

Copy CSV Print Search: Beijing

线路号	线路名称	出发站	到达站	实际里程	操作
No matching records found					

4.增加站点

添加新车站

车站	<input type="button" value="^"/> <input type="button" value=""/> <input type="button" value="X"/>		
车站名 *	<input type="text" value="Beijing"/>		
车站等级	<input type="button" value="▼"/>		
<input type="button" value="Cancel"/> <input type="button" value="Reset"/> <input type="button" value="Submit"/>			
1	Beijing	特等站	Delete Edit

4.2.6 票务管理

1. 管理员可以通过本系统的票务模块实现对已购票的查询和查看

||b 您正在查看票务

票务信息表 b														
Show	10	▼ entries	Search: <input type="text"/>											
票号	车号	线路	用户号	用户名	座等	座位	出发日期	出发时间	到达时间	票价	Action	↑		
2	G697	Wuhan-Beijing	2	zyp	1	6485	2021-03-19	08:53:02	17:05:19	8066	Delete	↑		
4	G4015	Chengdu-Shanghai	4	jjq	2	2972	2021-04-27	18:28:48	21:23:23	4934	Delete	↑		
5	G2547	Beijing-Shanghai	1	llm	2	1669	2020-01-26	13:59:33	15:17:09	1029	Delete	↑		
6	D6628	Guangzhou-Wuhan	1	llm	2	8532	2020-02-01	14:25:32	15:34:49	4869	Delete	↑		
7	G3257	Nanjing-Guangzhou	4	jjq	1	6726	2020-11-23	02:01:35	15:11:08	6075	Delete	↑		
8	D2588	Guangzhou-Chengdu	2	zyp	1	6546	2020-01-16	05:24:16	08:20:32	8890	Delete	↑		
10	D8792	Guangzhou-Chengdu	2	zyp	1	4695	2020-06-15	03:49:44	19:08:33	7394	Delete	↑		
11	D3666	Nanjing-Shanghai	4	jjq	1	5594	2021-01-08	03:42:40	06:18:02	2941	Delete	↑		
12	G6601	Nanjing-Guangzhou	4	jjq	2	1744	2020-07-12	07:46:16	14:51:48	5330	Delete	↑		
14	G1837	Wuhan-Shanghai	4	jjq	1	3905	2021-09-05	00:04:46	11:02:50	3022	Delete	↑		

Showing 1 to 10 of 71 entries

Previous 1 2 3 4 5 ... 8 Next

管理员可以通过右上角的检索框，根据票号、车次号、线路名称、用户号、用户名等关键词进行模糊搜索

下面以检索用户 llm 的购买情况为例

票务信息表 b														
Show	10	▼ entries	Search: <input type="text"/> llm											
票号	车号	线路	用户号	用户名	座等	座位	出发日期	出发时间	到达时间	票价	Action	↑		
5	G2547	Beijing-Shanghai	1	llm	2	1669	2020-01-26	13:59:33	15:17:09	1029	Delete	↑		
6	D6628	Guangzhou-Wuhan	1	llm	2	8532	2020-02-01	14:25:32	15:34:49	4869	Delete	↑		
20	G6621	Beijing-Shanghai	1	llm	1	7791	2020-06-07	14:01:46	23:56:06	4596	Delete	↑		
23	D2700	Chengdu-Beijing	1	llm	2	9429	2021-06-11	12:09:56	14:05:57	6346	Delete	↑		
31	G4015	Chengdu-Shanghai	1	llm	2	2795	2021-12-20	18:28:48	21:23:23	4934	Delete	↑		
39	G3987	Wuhan-Nanjing	1	llm	1	8452	2021-12-20	02:26:12	10:45:01	7543	Delete	↑		
45	D6070	Chengdu-Guangzhou	1	llm	2	8658	2021-10-13	18:02:42	22:39:47	9578	Delete	↑		
48	D2700	Chengdu-Beijing	1	llm	1	3247	2021-10-25	12:09:56	14:05:57	9423	Delete	↑		
66	D8792	Guangzhou-Chengdu	1	llm	2	3992	2021-01-03	03:49:44	19:08:33	4405	Delete	↑		
67	G9809	Shanghai-Wuhan	1	llm	2	5930	2020-08-07	05:09:06	08:38:14	3527	Delete	↑		

Showing 1 to 10 of 12 entries (filtered from 71 total entries)

Previous 1 2 Next

也可以通过点按表头，实现根据某一个特殊关键字的排序，例如根据线路关键字排序，并设置页面上能够显示的条目数量。

票务信息表 b														
Show	25	▼ entries	Search: <input type="text"/>											
票号	车号	线路	用户号	用户名	座等	座位	出发日期	出发时间	到达时间	票价	Action	↑		
41	G3269	Beijing-Chengdu	6	yzh	1	1390	2021-03-02	10:35:45	18:54:55	4466	Delete	↑		
103	G3269	Beijing-Chengdu	2	zyp	2	961	2021-05-23	10:35:45	18:54:55	4740	Delete	↑		
5	G2547	Beijing-Shanghai	1	llm	2	1669	2020-01-26	13:59:33	15:17:09	1029	Delete	↑		
71	G6621	Beijing-Shanghai	2	zyp	1	4556	2020-06-03	14:01:46	23:56:06	4596	Delete	↑		
20	G6621	Beijing-Shanghai	1	llm	1	7791	2020-06-07	14:01:46	23:56:06	4596	Delete	↑		
76	G2547	Beijing-Shanghai	2	zyp	1	4469	2020-07-07	13:59:33	15:17:09	6618	Delete	↑		
37	G2547	Beijing-Shanghai	4	jjq	2	3149	2020-08-24	13:59:33	15:17:09	1029	Delete	↑		
50	G6621	Beijing-Shanghai	2	zyp	1	7151	2020-10-28	14:01:46	23:56:06	4596	Delete	↑		
33	G2547	Beijing-Shanghai	5	szw	1	7354	2021-04-12	13:59:33	15:17:09	6618	Delete	↑		
65	D2700	Chengdu-Beijing	5	szw	1	1347	2021-05-05	12:09:56	14:05:57	9423	Delete	↑		
23	D2700	Chengdu-Beijing	1	llm	2	9429	2021-06-11	12:09:56	14:05:57	6346	Delete	↑		
30	D2700	Chengdu-Beijing	2	zyp	1	7064	2021-06-16	12:09:56	14:05:57	9423	Delete	↑		
48	D2700	Chengdu-Beijing	1	llm	1	3247	2021-10-25	12:09:56	14:05:57	9423	Delete	↑		

2. 管理员可以通过点击Action-Delete 按钮实现对一张票进行退票的操作

3. 管理员能够使用本系统中的票务子系统，实现为用户购票

我们提供根据用户名，用户号两种方式进行购票

IIb 正在购票

购票

The screenshot shows a 'Buy Ticket' form with fields for user credentials, train number, departure date, and seat class. A dropdown menu is open under the 'Train Number' field, displaying several train route and time options.

用户凭证类型	user_name
用户凭证 *	user_name user_id
车次号 *	
出发日期 *	yyyy/mm/日
座等	Class 1

Cancel Reset Check Submit

车次号提供模糊搜索匹配的下拉框功能，提示管理员用户方便输入

IIb 正在购票

购票

The screenshot shows the same 'Buy Ticket' form. The 'Train Number' field contains the placeholder 'Re-enter!' instead of a valid train number. A dropdown menu is open above it, showing the correct train route and time.

用户凭证类型	user_name
用户凭证 *	
车次号 *	Re-enter!
出发日期 *	年 /月/日
座等	Class 1

输入错误时，强制要求重新输入

The screenshot shows the 'Buy Ticket' form with a summary at the top stating '应收 CNY: 9385'. The 'Check' button is highlighted in green, indicating it has been clicked.

用户凭证类型	user_name
用户凭证 *	
车次号 *	Re-enter!
出发日期 *	年 /月/日
座等	Class 1

Cancel Reset Check Submit

点击check查询价格，价格按账户类型（学生、成人）、座等、车次号变化

The screenshot shows the 'Buy Ticket' form with a summary at the top stating '应收 CNY: 9385'. All fields are filled with sample data. The 'Check' button is highlighted in green.

用户凭证类型	user_name
用户凭证 *	llm
车次号 *	D1626-Nanjing-Guangzhou-18:22:48
出发日期 *	2021/06/22
座等	Class 1

Cancel Reset Check Submit

4.2.7 车次管理

车次管理模块提供车次的查看、查询功能，可以在网页右上角进行条目内容的模糊匹配查询，在左上角进行页面条目量设置

llb 您正在查看车次

车次信息表 llb

Show 10 entries

Search:

车次号	车号	线路号	线路	出发时间	到达时间
D1304	36	25	Shanghai-Wuhan	00:40:33	08:40:09
D1626	31	18	Nanjing-Guangzhou	18:22:48	23:53:44
D2588	30	12	Guangzhou-Chengdu	05:24:16	08:20:32
D2700	12	6	Chengdu-Beijing	12:09:56	14:05:57
D2744	5	26	Wuhan-Beijing	03:08:50	13:51:56
D3276	22	14	Guangzhou-Shanghai	17:43:27	22:10:41
D3304	27	26	Wuhan-Beijing	07:49:13	18:37:23
D3354	3	7	Chengdu-Guangzhou	04:11:00	21:49:14
D3666	19	19	Nanjing-Shanghai	03:42:40	06:18:02
D3988	7	20	Nanjing-Wuhan	03:17:34	15:10:26

Showing 1 to 10 of 40 entries

Previous 1 2 3 4 Next

类似的，车次添加界面要求指定一个车次号作为唯一凭证，并根据提示选择车号、线路号，当输入错误时强制要求重新输入，在时间选择控件中设置时间，点按添加即可完成添加

添加新车次

车次

车次号*

车号*

线路/线路号

出发时间

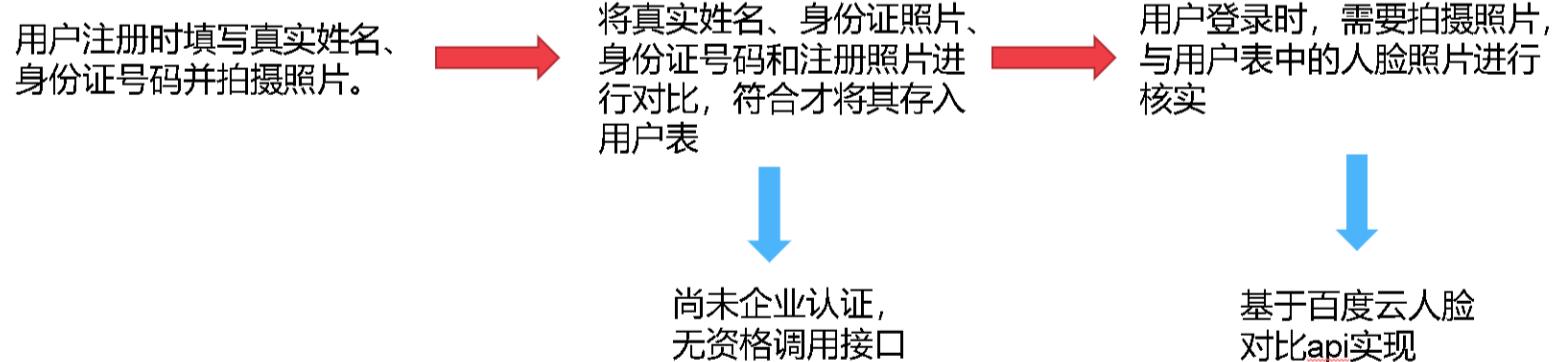
到达时间

- 1-Beijing-Chengdu
- 2-Beijing-Guangzhou
- 3-Beijing-Nanjing
- 4-Beijing-Shanghai
- 6-Chengdu-Beijing

5. 扩展功能实现

5.1 基于阿里云人脸对比SDK实现的登录人脸核验

1 为进一步落实铁路购票实名制需求，特设计人脸核验功能，在用户登录时，需要核验当前登录用户人脸与数据库中存储的人脸信息是否相符，并将错误信息反馈到登录界面。具体实现思路如下图所示。



5.2 基于sketchfab实现的列车模型3D展示

1 为进一步宣扬铁路文化，普及列车发展知识，特在用户端设计了列车模型的3D展示模块，展示我国最新一代的CRH5高速列车。考虑到利用本地资源渲染模型需要存储、加载大量的静态资源，不利于多并发访问且存在诸多无意义的编程，特采用网络资源，即基于sketchfab的api实现。设置调用超链接，用户在打开网页时将自动加载网络资源，并且用户还可以利用鼠标拖动模型和设置一些参数来实现简单的交互。

1 页面效果图如下所示。



5.3 基于Prophet模型实现的数据趋势分析和预测

5.3.1 数据源

1 | 分析数据采用特定时间范围内的某车次总购票人数，预测数据采用过去时间内的某车次总购票人数。

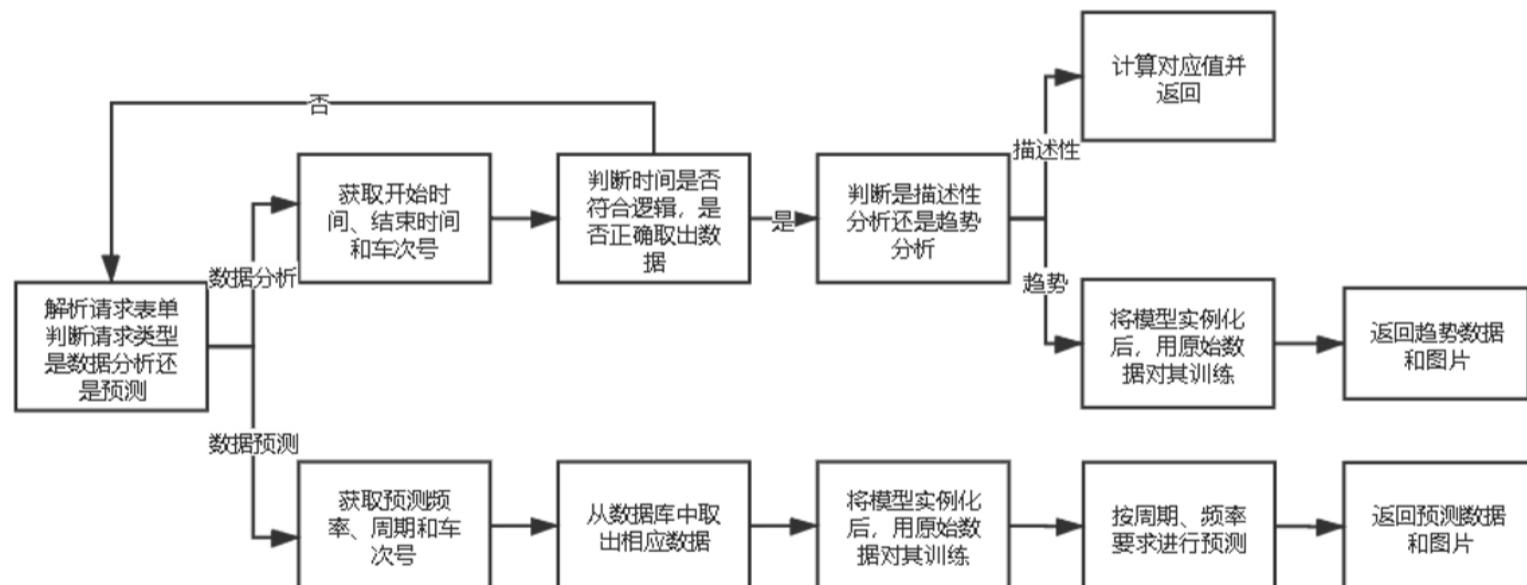
5.3.2 Prophet模型简介

1 | Prophet模型由Facebook提出并开源，其是基于非线性关系拟合的时间序列模型，并非机器学习方法。因此他无需麻烦的特征工程，用户操作友好，适用于铁路信息中的这种自动化处理。并且其将快速的完成模型训练更新，符合铁路系统数据更新较为频繁的现状。同时，其将对假期和特征事件建模，如额外关注国庆节、劳动节等假期；其支持自动分析和手动添加周期性变化规律，如一周内的趋势。这些特征符合铁路旅客数据的周期性和特殊性。

1 | 我们利用chinesecalendar库函数寻找数据时间范围内的所有holiday，添加特征事件。并且添加了年、月、周等周期性因素。

5.3.3 编程逻辑

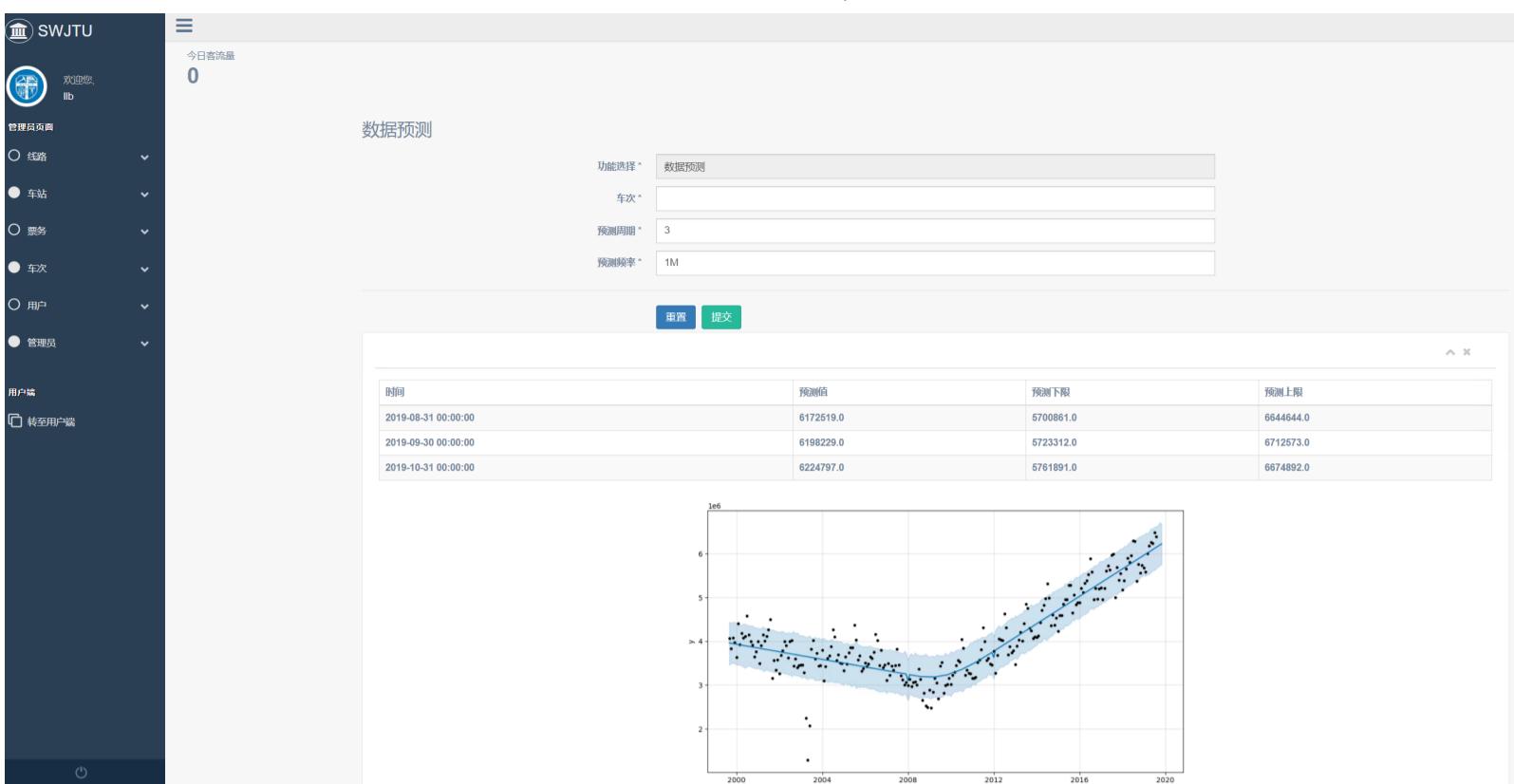
1 | 具体的编程逻辑如下图所示。



5.3.4 效果展示

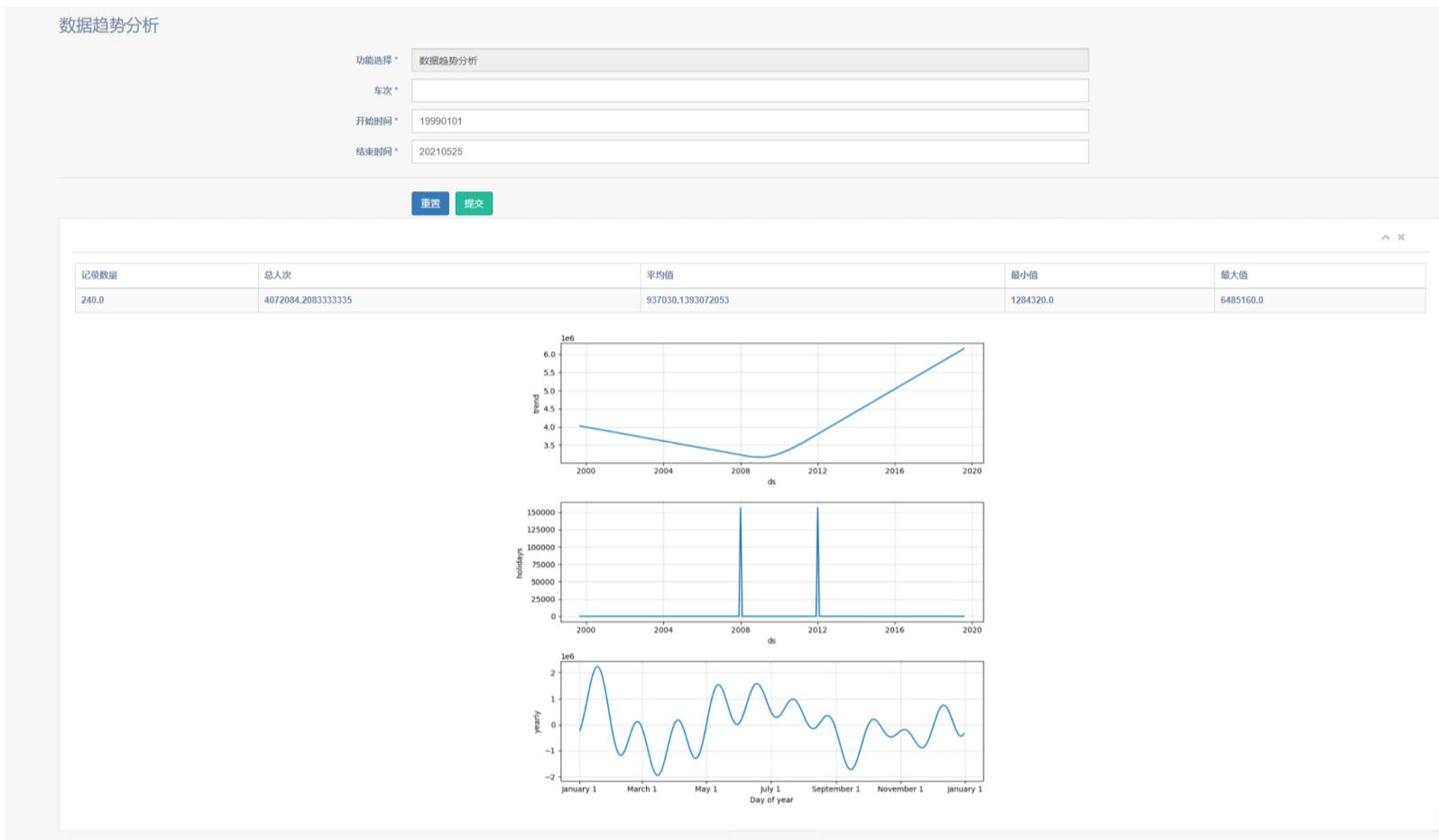
1. 数据预测

1 | 输入为车次、预测周期和预测频率，输出为一个表格和一张图片。其中表格列为时间、预测值、预测上限值和预测下限值。图片中黑点为历史数据，蓝色线为拟合情况，蓝色带为范围值。



2. 数据分析

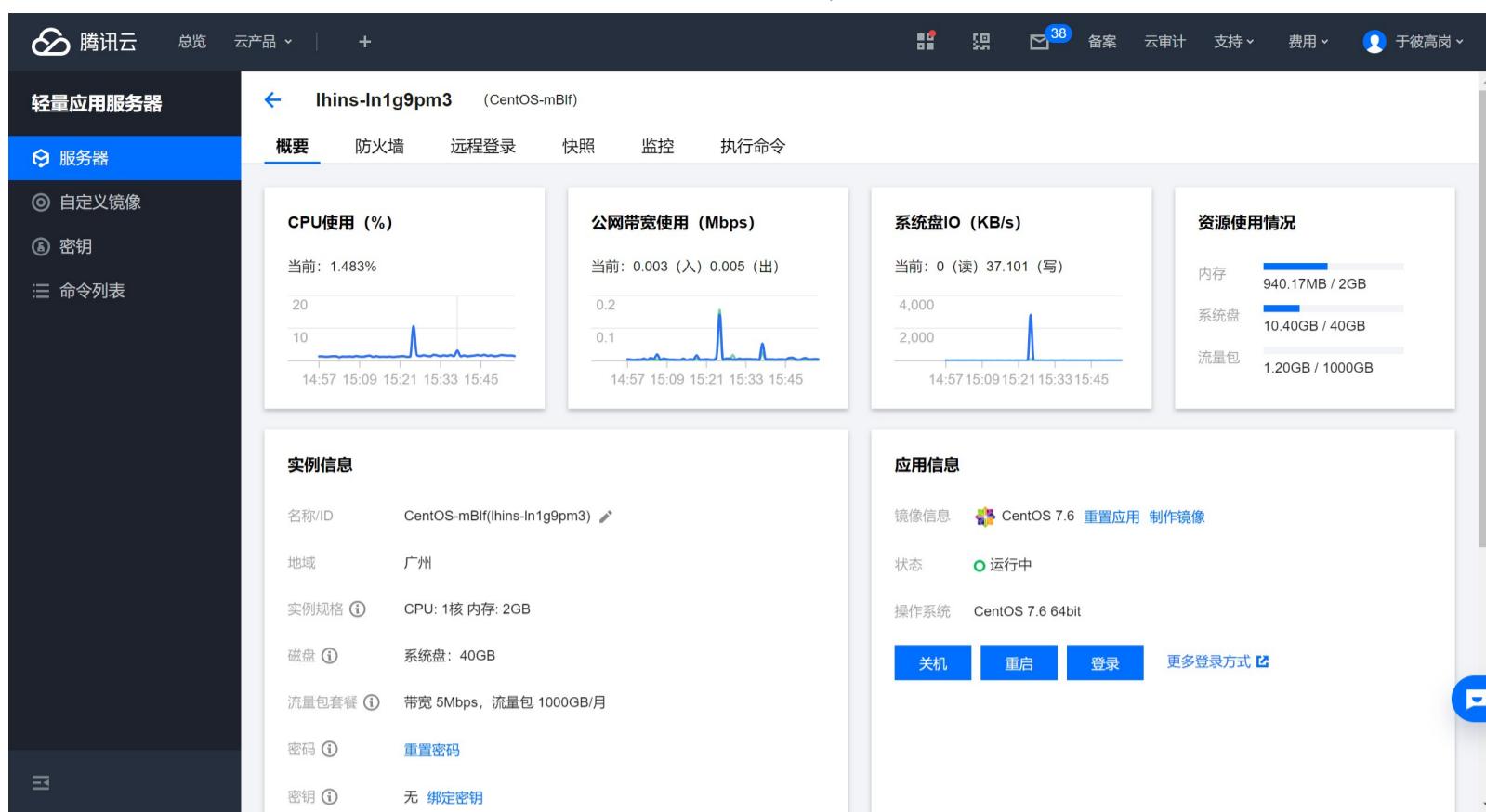
- 1 | 输入为车次、开始时间和结束时间，输出为一个表格和一组图片，其中表格为数据的描述性分析。图片为数据趋势分析，包括数据的总变化趋势，特殊事件带来的影响和周期性影响因素。



6. 服务器

6.1 服务器概述

服务器使用腾讯云的轻量应用服务器，配置为1核CPU，2GB内存，40GB系统盘，5Mbps带宽。



[域名](#)在阿里云注册，已在工信部备案，可以满足日常的简单使用。

6.2 服务器配置

6.2.1 服务器本地运行配置

由于服务器需要和远端仓库的代码实现同步，这里选用git工具进行代码，初次运行需要初始化MySQL数据库：

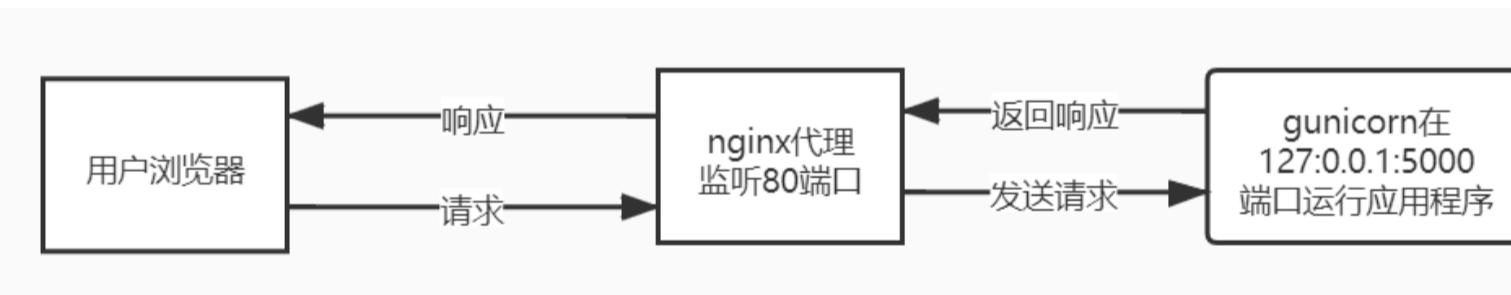
```
1 git clone https://gitee.com/Guotongwu/train-ticket-system
2 python Project/db.py
```

为了在指定的本地端口（127.0.0.1:8000）运行Flask程序，使用gunicorn来运行进程。gunicorn的作用相当于一个容器，可以对进程的资源进行分配：

```
1 cd Project
2 gunicorn -w 4 -b 127.0.0.1:8000 app:app
```

6.2.2 Nginx配置

对服务器本地运行的进程，使用Nginx进行反向代理，Nginx反向代理原理如下：



使用Nginx需要修改其配置，使其同时监听80（HTTP）和443（HTTPS）端口，同时完成反向代理本地端口127.0.0.1:5000。以443端口为例，大概的配置如下：

```

1 server {
2     listen 443 ssl;
3     #配置HTTPS的默认访问端口为443。
4     server_name www.omysscamore.top; #证书绑定的域名。
5
6     client_max_body_size 10M; #最高上传的文件大小
7
8     root html;
9     index index.html index.htm;
10    ssl_certificate cert/5681214-www.omysscamore.top.pem; #SSL证书文件
11    ssl_certificate_key cert/5681214-www.omysscamore.top.key; #SSL密钥
12    ssl_session_timeout 5m;
13    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDSA:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
14    #表示使用的加密套件的类型。
15    ssl_protocols TLSv1 TLSv1.1 TLSv1.2; #表示使用的TLS协议的类型。
16    ssl_prefer_server_ciphers on;
17    location / {
18        proxy_pass http://127.0.0.1:8000; #本地端口
19    }
20
21    #设置错误页面跳转
22    error_page 404 500 502 503 504 /root/github/train-ticket-
system/Project/Manager/templates/manage_404.html;
  
```

```
23     location = /root/github/train-ticket-system/Project/Manager/templates/manage_404.html {  
24         }  
25     }
```

注：由于HTTPS需要数字证书进行安全认证，此处选用了阿里云提供的免费SSL证书。