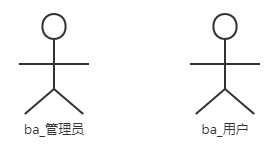
**新冠疫情信息管理系统**

一、概要说明

本信息管理系统是为了便捷疫情常态化下统计学校学生、教职工等的基础信息而建立的，用户也可以通过系统提交出校申请，并且在系统中可以查看全国及世界范围内的感染情况，管理员可以查看用户行动轨迹及所提交的基础信息。

二、需求分析

1.系统角色识别

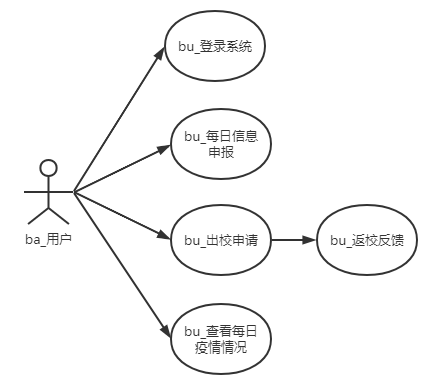


角色说明

|  |  |
| --- | --- |
| 角色名称 | 说明 |
| ba\_用户 | 用户即广大学生与教职工，具有登录系统、提交每日信息申报、提交出校申请、提交返校反馈、查看实时感染情况等功能。 |
| ba\_管理员 | 辅导员及学院领导，具有登录系统、查看用户行动轨迹、查看用户健康信息、辅助用户修改信息、回复用户信息等功能。 |

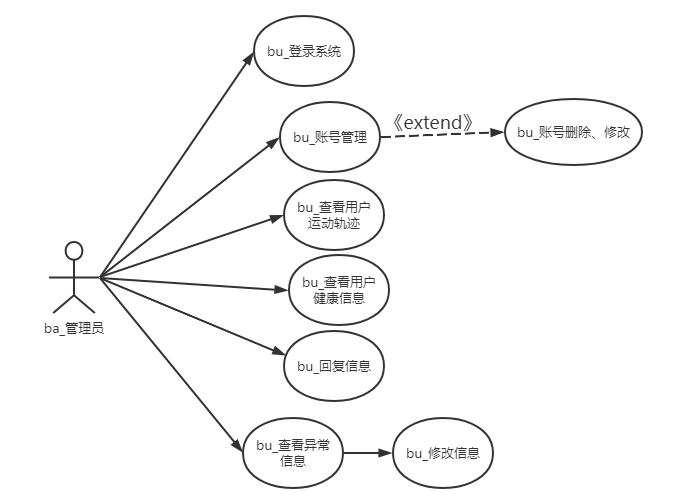
2.系统功能性需求识别

（1）用户



用户端可以通过账号和密码登录系统，填写每日基础信息申报，提交出校申请，返校时需要填写返校反馈，并且可以查看实时感染情况。

（2）管理员



管理员端具有登录系统、查看用户行动轨迹、查看用户健康信息、辅助用户修改信息、回复用户信息等功能。在查看用户健康信息，如果用户填报有误，将联系用户辅助修改健康信息，如学生体温过高，在医院确认并无感染的情况下，辅导员可以将其从疑似感染名单中排除。

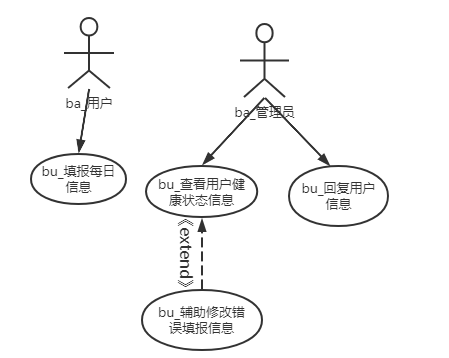
3.系统功能性需求概览

（1）登录业务：用户和管理员的登录操作



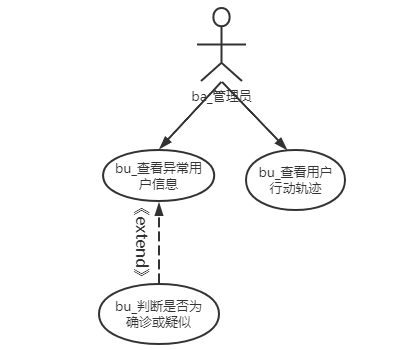
用户与管理员输入账号、密码及验证码登录系统，系统后台数据库根据账号比对密码并返回登录结果（成功或失败）。

（2）每日申报：每日信息申报、查看用户信息、辅助用户修改健康信息与回复用户信息



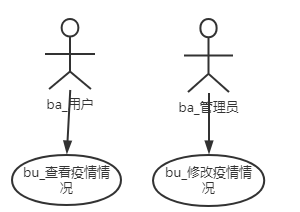
用户填报每日信息并提交，管理员端可以可视化查看用户健康状态等信息，并在用户信息填报有误时辅助修改。如体温超过正常标准，但可以在医院确认安全后将用户从疑似名单中删除。

（3）进出校：出校申请、返校反馈、查看用户行动轨迹与查看出校异常用户信息



非确诊或疑似用户可以提交出校申请，无需经过管理员审批即可出校，只需确定出校日期即可，在返校后必须填写返校反馈，即何时何地停留多久，这些信息将会反馈到管理员端，管理员可以查看用户行动轨迹图与异常用户名单（即提交出校申请时的预定返校时间与实际返校时间不一致的情况）。

（4）实时信息：查看实时感染情况与修改实时感染情况



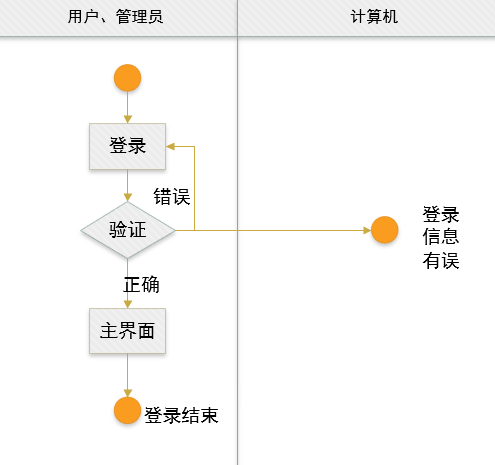
用户和管理员可以查看全国范围内的新冠感染情况（通过后台抓包实现），当抓包数据出现异常时，管理员可以辅助修改感染数据。

4.功能性需求规定

（1）登录系统

|  |  |
| --- | --- |
| 用例名称 | 登录系统 |
| 实现名称 | login.jsp |
| 用例描述 | 用户通过此用例进入系统 |
| 参与者 | 1.用户  2.管理员 |
| 前置条件 | 打开登录界面 |
| 后置条件 | 进入系统平台 |
| 主事件流 | 1.用户填写账号、密码和验证码，计算机显示平台界面  2.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a账号、密码或验证码错误  1.拒绝登录  2.提示信息错误，用例结束  2.a账号与身份不匹配  1.拒绝登录  2.提示身份不正确，用例结束 |
| 业务规则 | 填写个人信息 |
| 设计的业务实体 | 信息验证 |
| 非功能性需求 | 支持中英文 |

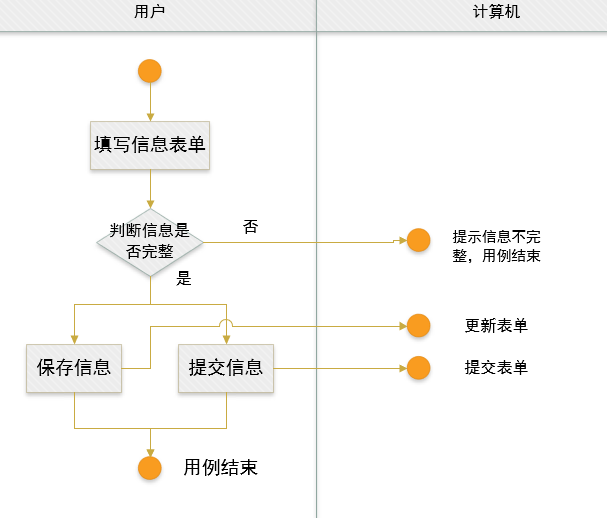
登录系统业务的场景分析：



（2）每日信息申报

|  |  |
| --- | --- |
| 用例名称 | 每日信息申报 |
| 实现名称 | everyReport.jsp |
| 用例描述 | 用户通过此用例提交每日信息 |
| 参与者 | 用户 |
| 前置条件 | 1.登录系统 |
| 后置条件 | 1.填写信息表单  2.保存信息  3.提交信息 |
| 主事件流 | 1.计算机根据用户上次保存的信息显示界面  2.用户选择保存信息，计算机更新修改后的表单  3.用户选择填写备注，计算机将其发送到管理员  4.用户选择提交信息，计算机显示提交成功发送界面  5.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a用户未完成所有必填项  1.拒绝保存或提交  2.提示信息不完整，用例结束 |
| 业务规则 | 用户根据实际情况填写信息 |
| 设计的业务实体 | 信息表单 |
| 非功能性需求 | 支持多用户同时提交信息 |

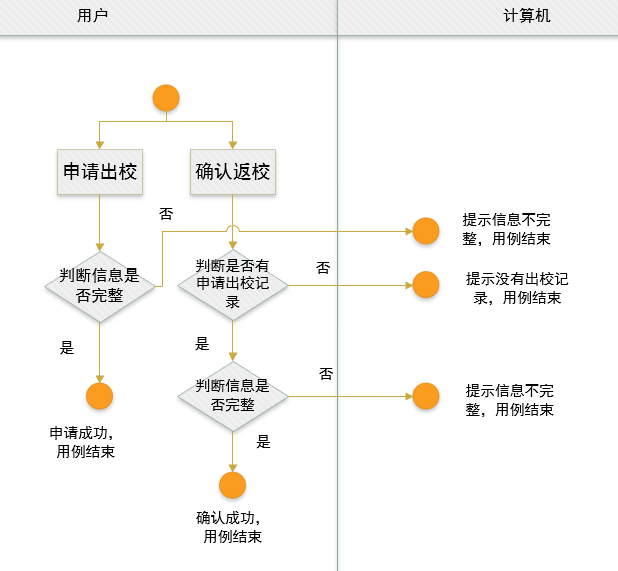
每日信息申报业务场景分析：



（3）进出校业务

|  |  |
| --- | --- |
| 用例名称 | 进出校 |
| 实现名称 | askOut.jsp |
| 用例描述 | 用户通过 |
| 参与者 | 用户通过此用例申请并确认出入校 |
| 前置条件 | 用户登录界面，通过验证 |
| 后置条件 | 1.申请出校  2.确认返校 |
| 主事件流 | 1.用户选择申请出校，计算机显示申请界面  2.用户选择确认返校，计算机显示确认返校信息界面  3.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a用户申请出校信息未完成所有必填项  1.拒绝申请出校  2.提示申请信息不完整，用例结束  2.a用户未成功申请出校，选择核实返校信息  1.拒绝访问  2.提示未申请出校，用例结束  3.a用户返校信息填写不完整  1.拒绝确认返校  2.提示返校信息不完整，用例结束 |
| 业务规则 | 根据实际情况填写出入校信息 |
| 设计的业务实体 | 申请出校表单、确认返校表单 |
| 非功能性需求 | 支持多用户同时填写 |

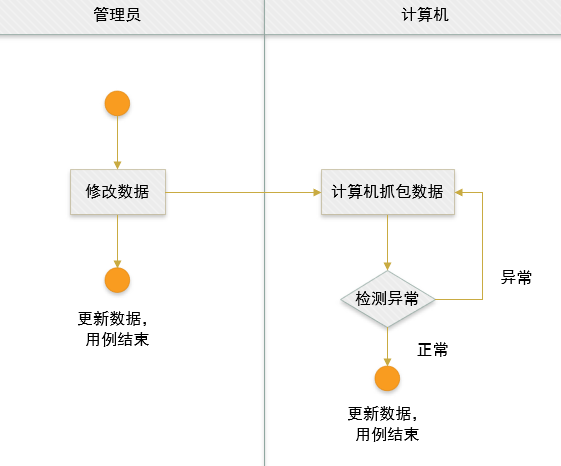
出入校业务的场景分析：



（4）每日感染情况

|  |  |
| --- | --- |
| 用例名称 | 每日感染情况 |
| 实现名称 | currentSituation.jsp |
| 用例描述 | 用户通过此用例查看全国和学校新冠感染情况  管理员通过此用例修改感染数据 |
| 参与者 | 用户、管理员 |
| 前置条件 | 用户登录界面，通过验证 |
| 后置条件 | 查看每日感染情况 |
| 主事件流 | 1.用户进入系统，计算机自动显示感染情况  2.管理选择更改数据，计算机显示更改界面  3.管理员确认更改数据，计算机更新数据 |
| 备选事件流 | 1.a抓包数据出现异常  1.拒绝更新  2.提示异常信息，用例结束 |
| 业务规则 | 管理员根据官方发布信息更新数据 |
| 设计的业务实体 | 感染数据 |
| 非功能性需求 |  |

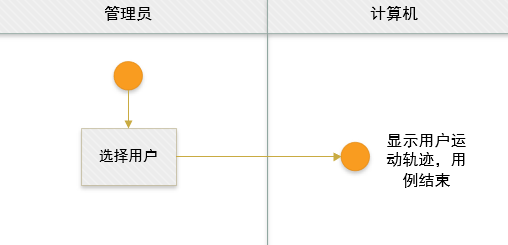
查看每日感染情况业务的场景分析：



（5）查看用户运动轨迹

|  |  |
| --- | --- |
| 用例名称 | 查看用户运动轨迹 |
| 实现名称 | track.jsp |
| 用例描述 | 管理员通过此用例查看用户运动轨迹 |
| 参与者 | 管理员 |
| 前置条件 | 用户登录界面，通过验证 |
| 后置条件 | 查看用户运动轨迹 |
| 主事件流 | 1.管理员选择特定用户，计算机显示该用户的运动轨迹  2.管理员选择轨迹显示范围，计算机显示不同范围的轨迹  3.计算机执行后置条件，用例结束 |
| 备选事件流 |  |
| 业务规则 | 管理员可以根据需求选择特定用户的运动轨迹 |
| 设计的业务实体 | 运动轨迹 |
| 非功能性需求 |  |

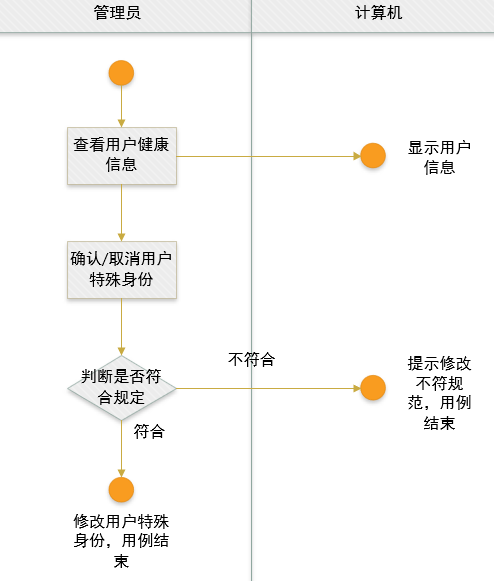
查看用户运动轨迹业务的场景分析：



（6）查看用户健康信息

|  |  |
| --- | --- |
| 用例名称 | 查看用户健康信息 |
| 实现名称 | fitInfo.jsp |
| 用例描述 | 管理员通过此用例查看用户健康信息 |
| 参与者 | 管理员 |
| 前置条件 | 用户登录界面 |
| 后置条件 | 1.查看用户健康信息  2.确认和取消用户身份 |
| 主事件流 | 1.管理员选择查看用户信息，计算机显示该用户信息表  2.管理员选择确认用户身份为确诊或疑似，计算机更新用户身份  3.管理员选择取消用户特殊身份，计算机更新用户身份  4.计算机执行后置条件，用例结束 |
| 备选事件流 | 1.a管理员确认和取消用户特殊身份与实际不符  1.拒绝更改身份信息  2.提示操作有误，用例结束 |
| 业务规则 | 管理员可以根据需求查看特定用户的健康信息 |
| 设计的业务实体 | 健康信息表 |
| 非功能性需求 |  |

查看用户健康信息业务的场景分析：



（7）辅助修改用户信息

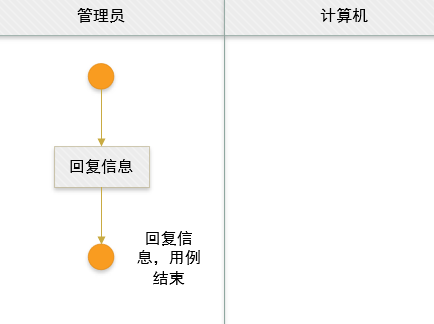
|  |  |
| --- | --- |
| 用例名称 | 辅助修改用户信息 |
| 实现名称 | changeInfo.jsp |
| 用例描述 | 管理员用过此用例修改用户信息 |
| 参与者 | 管理员 |
| 前置条件 | 登录界面 |
| 后置条件 | 1.修改用户账号、密码  2.修改用户填写信息 |
| 主事件流 | 1.管理员选择修改用户账号、密码，计算机显示修改界面  2.管理选择修改用户信息，计算机显示修改用户信息界面  3.计算机执行后置条件，用例结束 |
| 备选事件流 |  |
| 业务规则 | 管理员可以根据需求修改用户信息 |
| 设计的业务实体 |  |
| 非功能性需求 |  |

修改用户信息业务的场景分析

（8）回复用户信息

|  |  |
| --- | --- |
| 用例名称 | 回复用户信息 |
| 实现名称 | replyInfo.jsp |
| 用例描述 | 管理员通过此用例回复用户信息 |
| 参与者 | 管理员 |
| 前置条件 | 登录界面 |
| 后置条件 | 1.回复用户消息  2.发送通知 |
| 主事件流 | 1.用户选择回复用户消息，计算机显示用户消息  2.用户选择发送通知，计算机显示通知发送框 |
| 备选事件流 | 1.a回复消息发送失败  1.拒绝发送  2.提示发送失败，用例结束 |
| 业务规则 | 管理员可以根据需求回复用户消息 |
| 设计的业务实体 |  |
| 非功能性需求 |  |

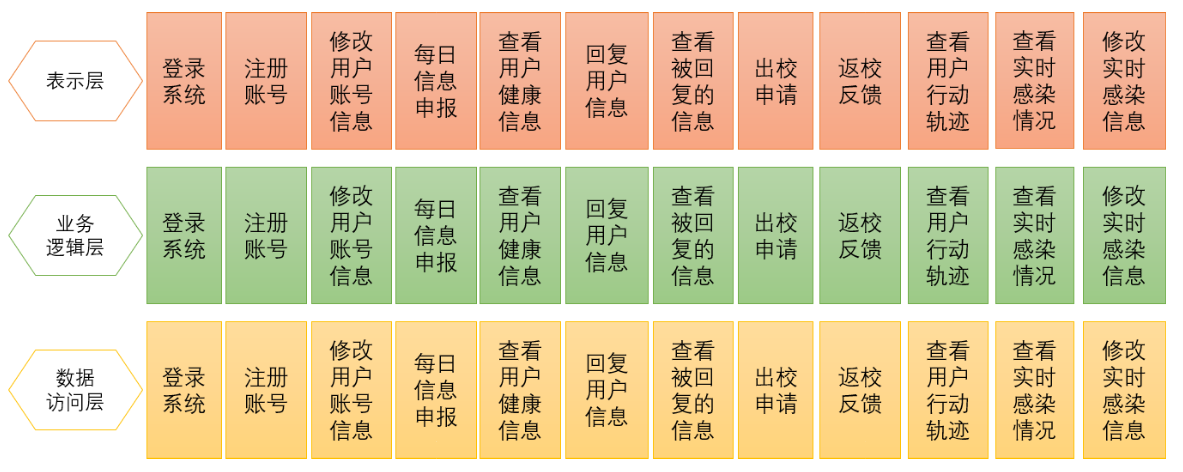
回复用户信息业务的场景分析：

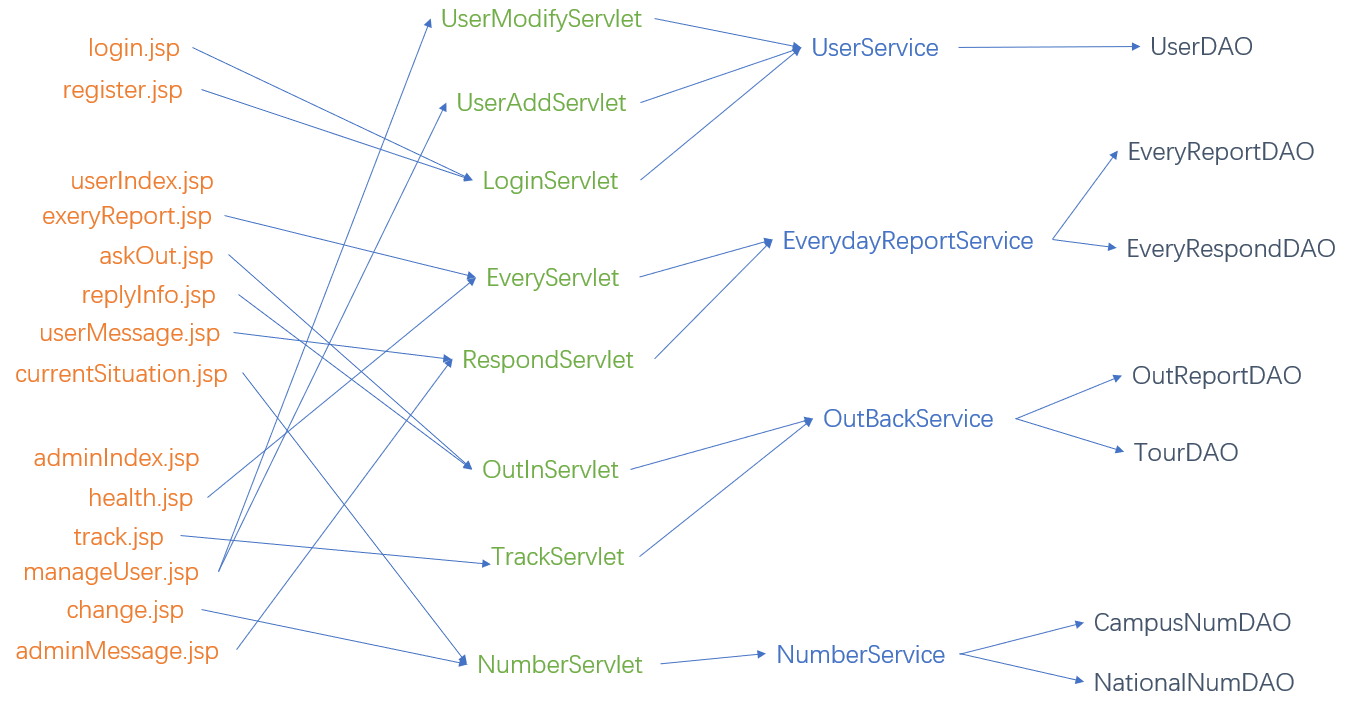


三、系统设计

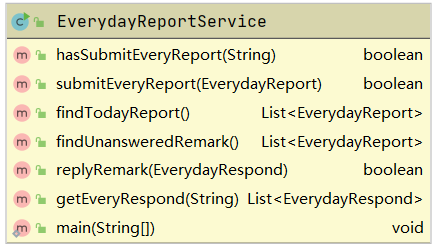
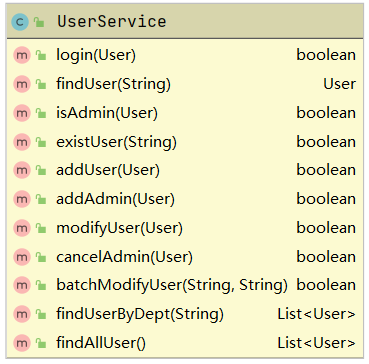
1.基于MVC的软件架构

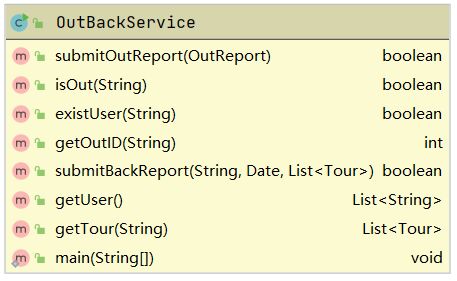
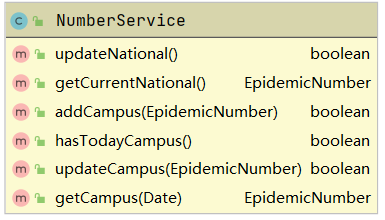
（1）整体MVC架构图



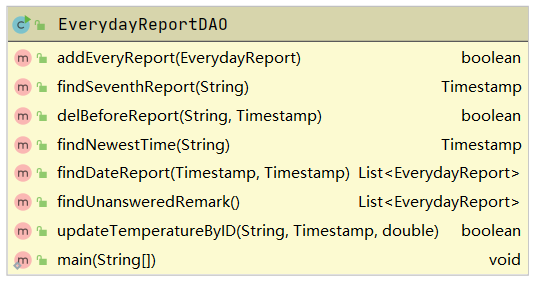
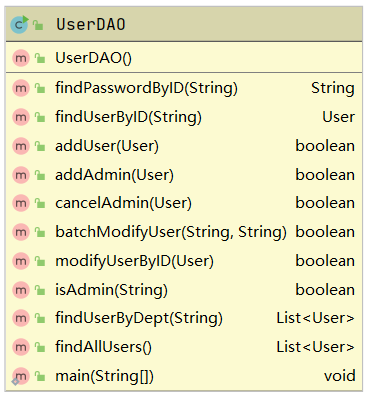
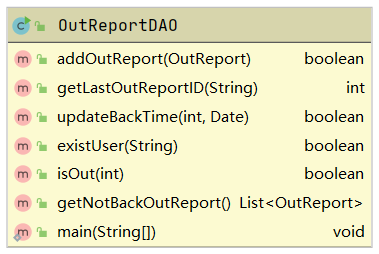
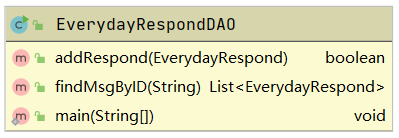


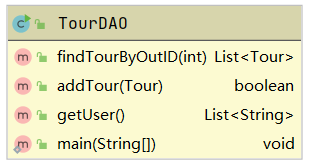
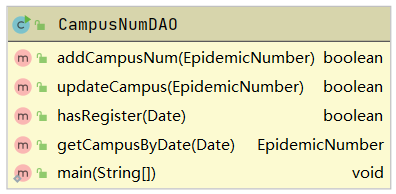
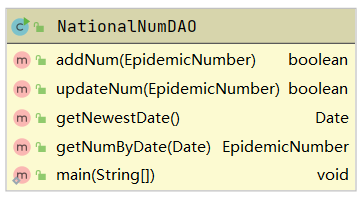
（2）业务逻辑层类图



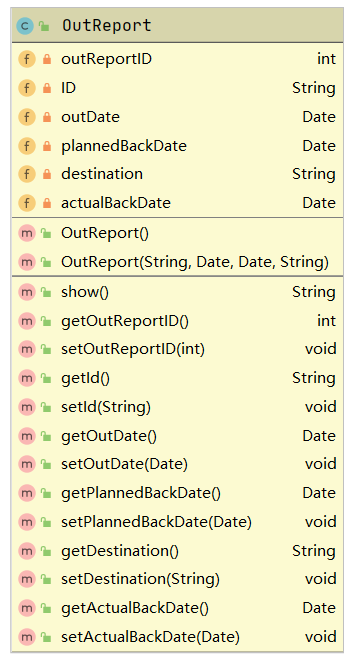
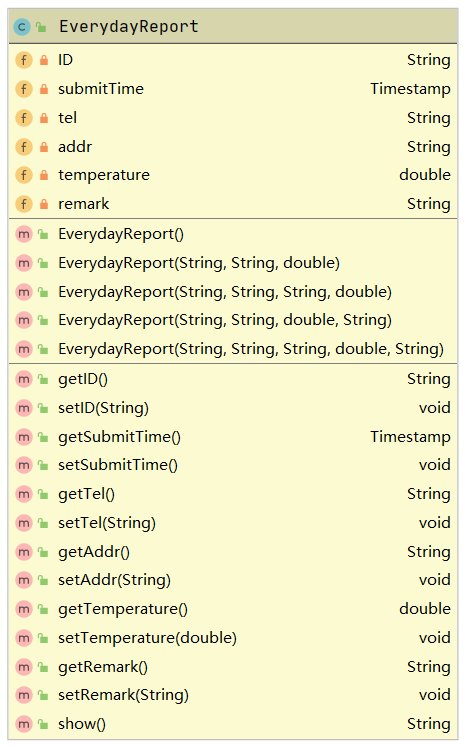
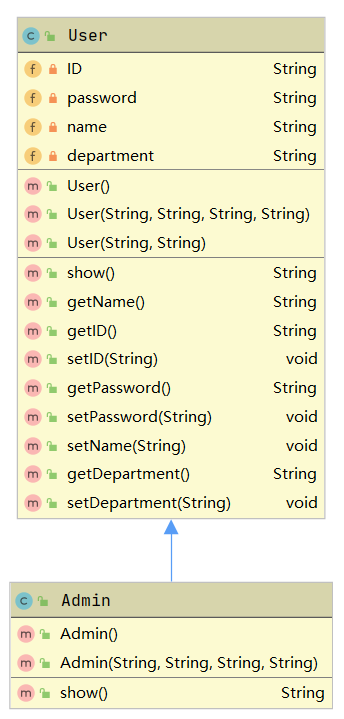
 

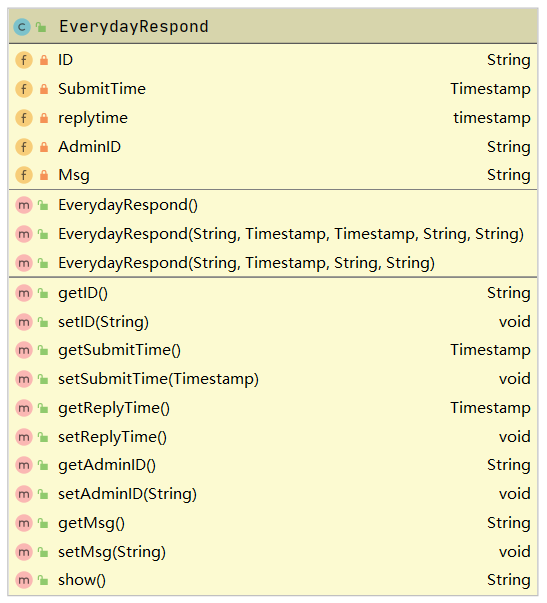
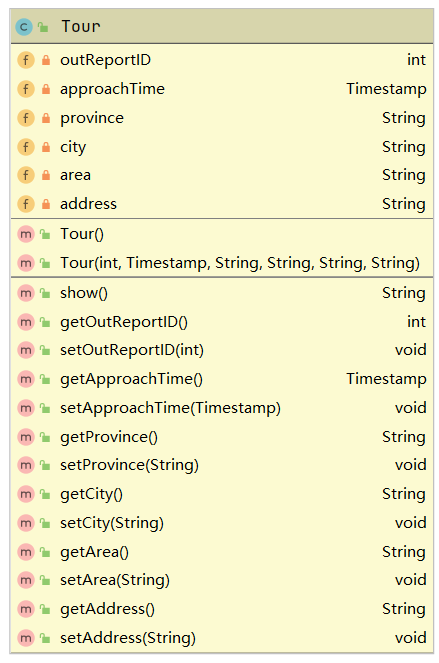
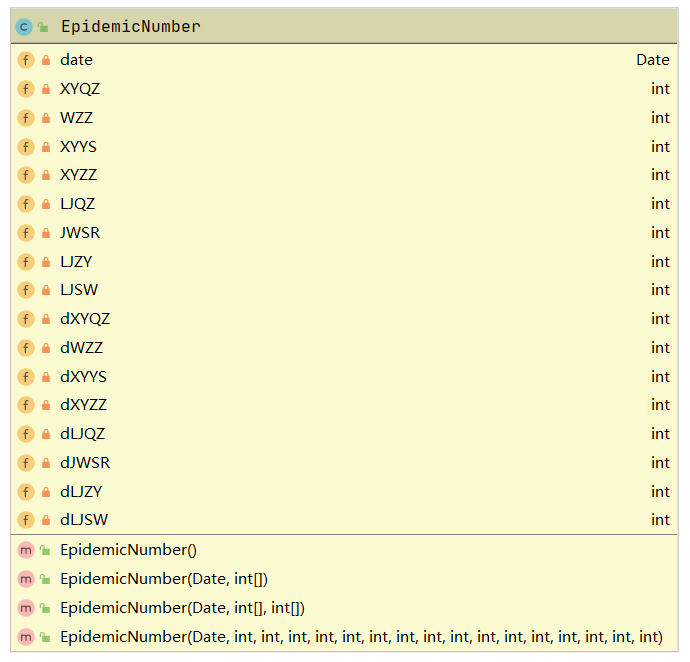
（3）数据访问层类图

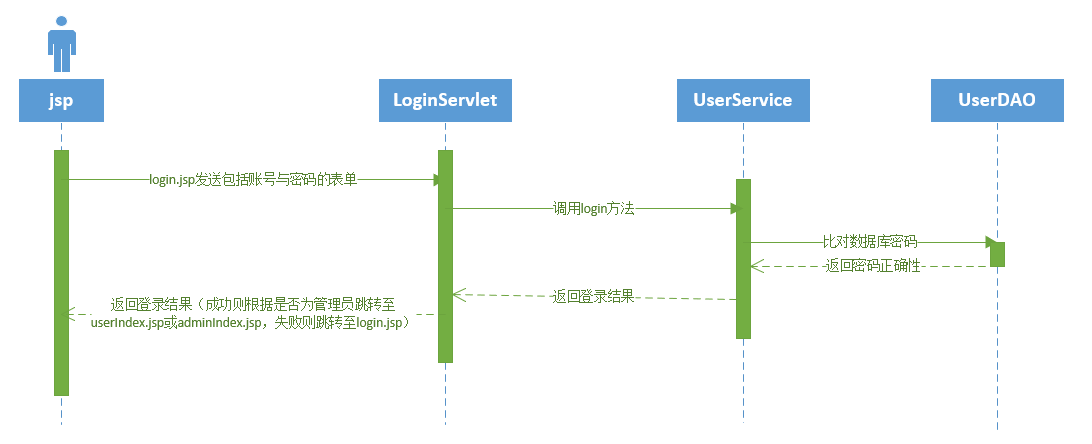
（4）javabean类图



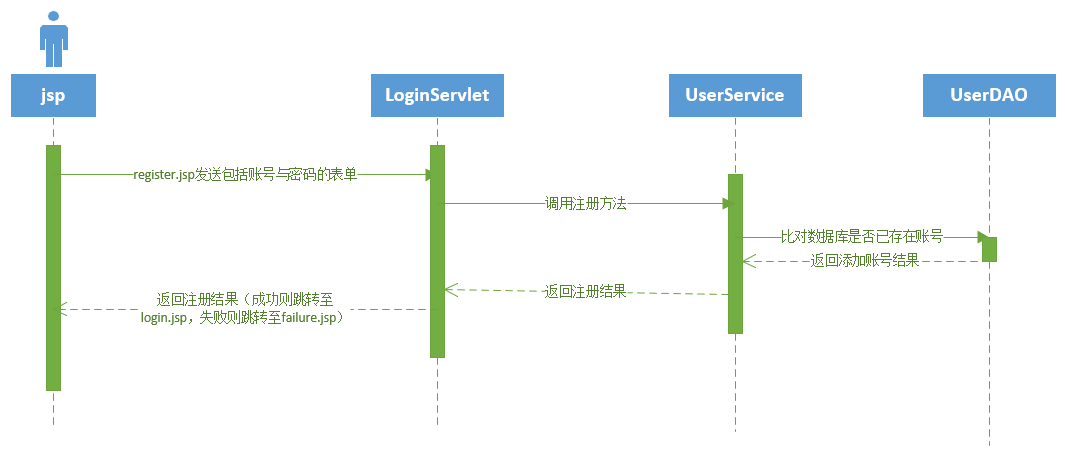
  

2.各业务具体流程

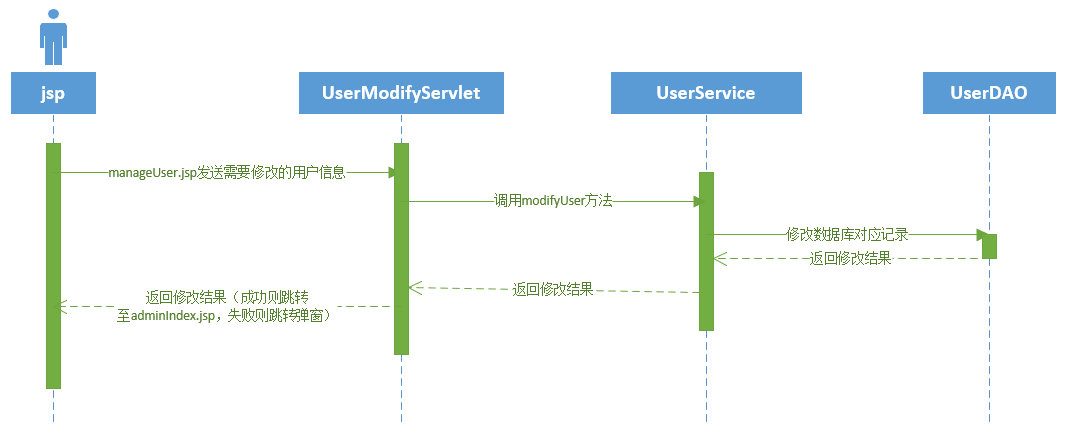
（1）登录系统业务



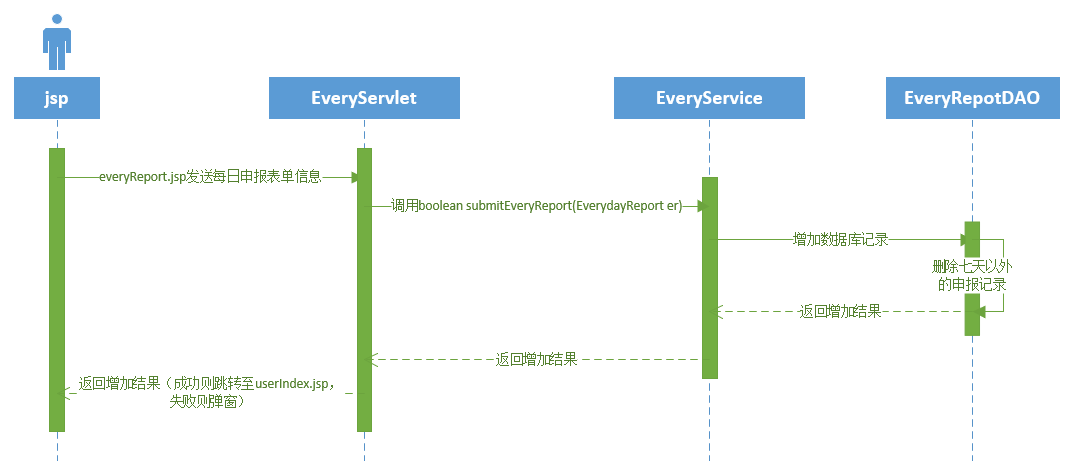
（2）注册账号业务



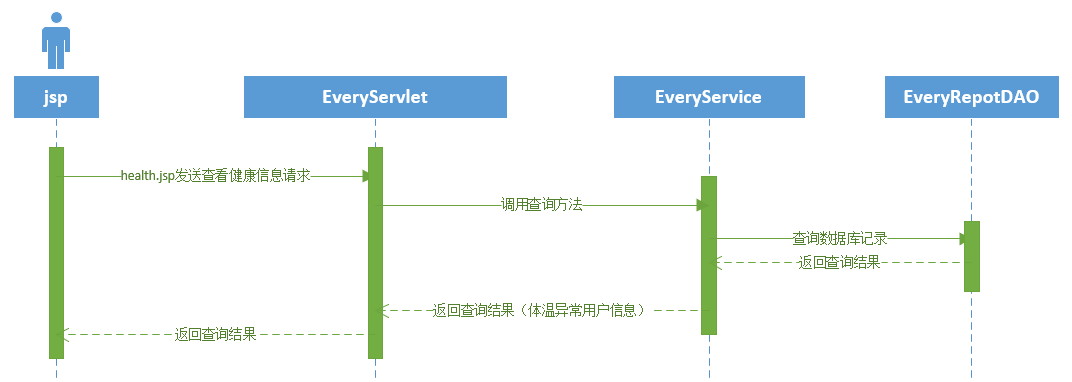
（3）修改用户信息业务



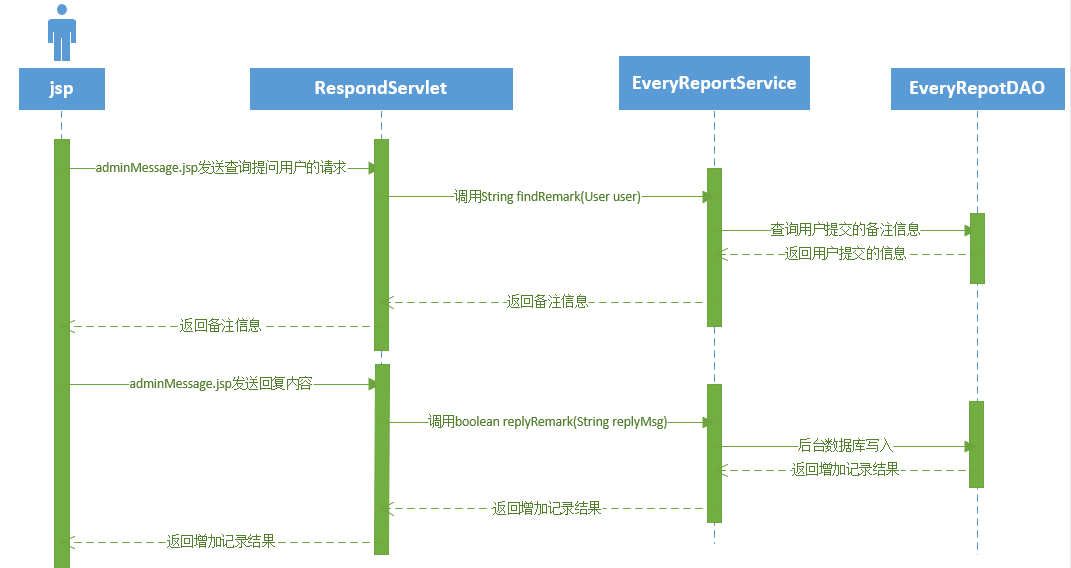
（4）每日信息申报业务



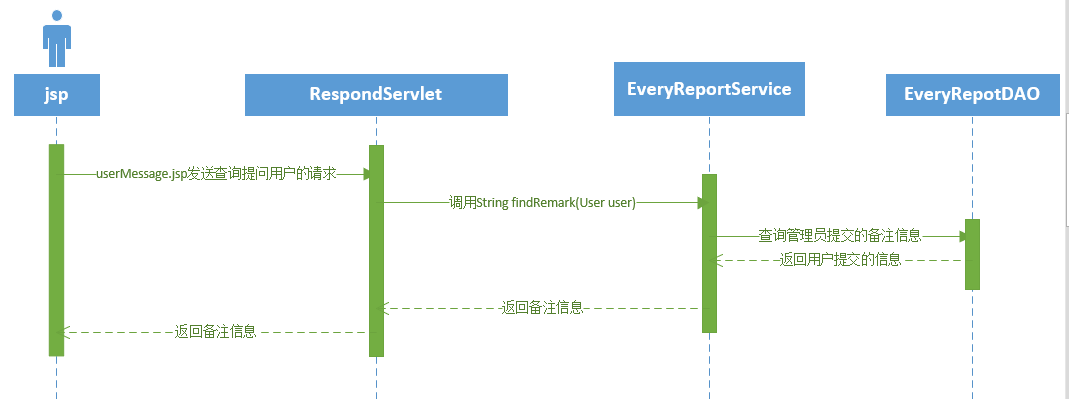
（5）查看用户健康信息业务



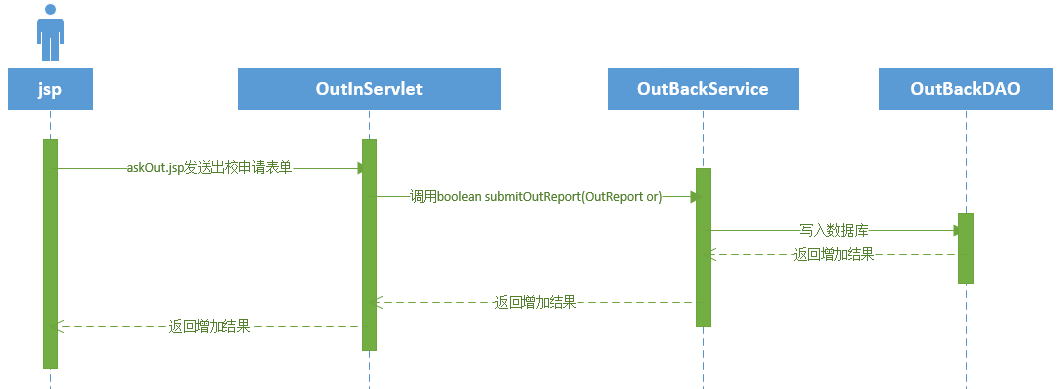
（6）回复用户信息业务



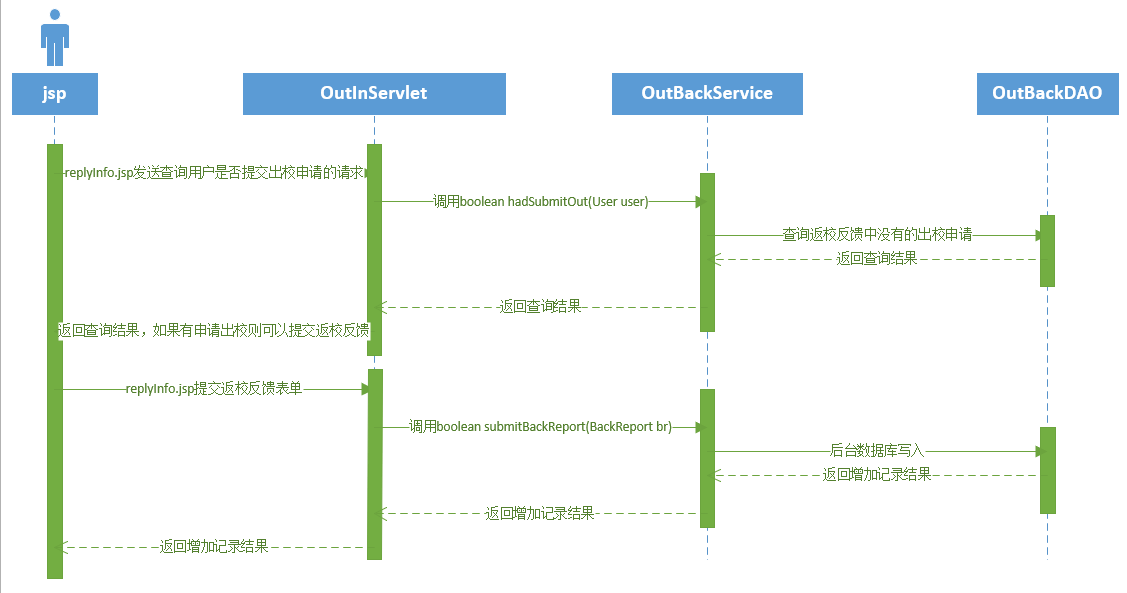
（7）查看被回复的信息



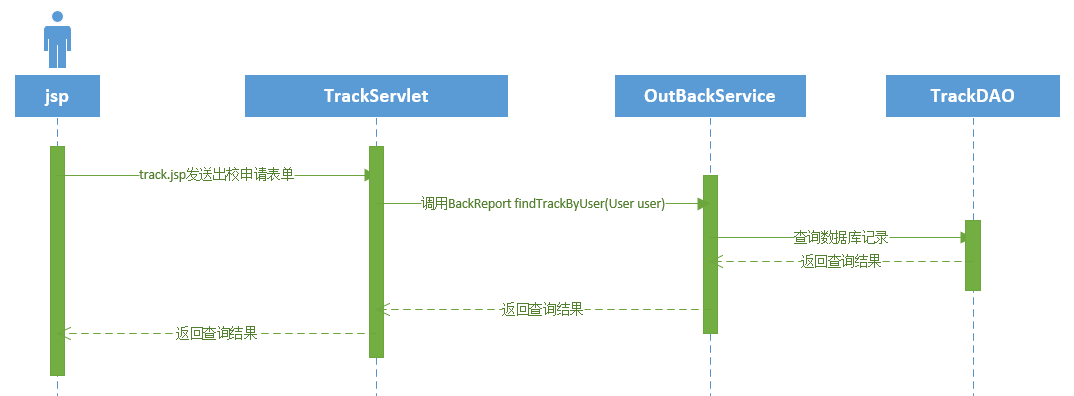
（8）出校申请业务



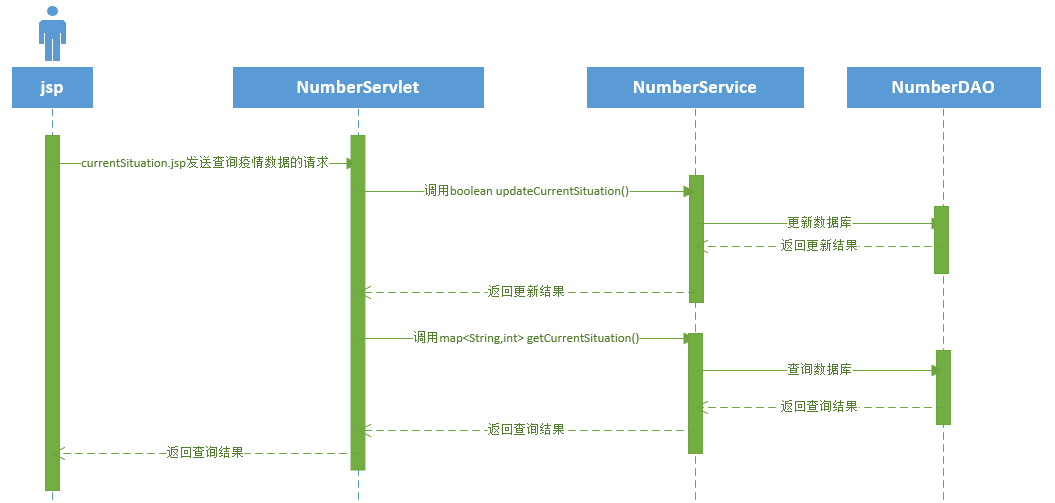
（9）返校反馈业务



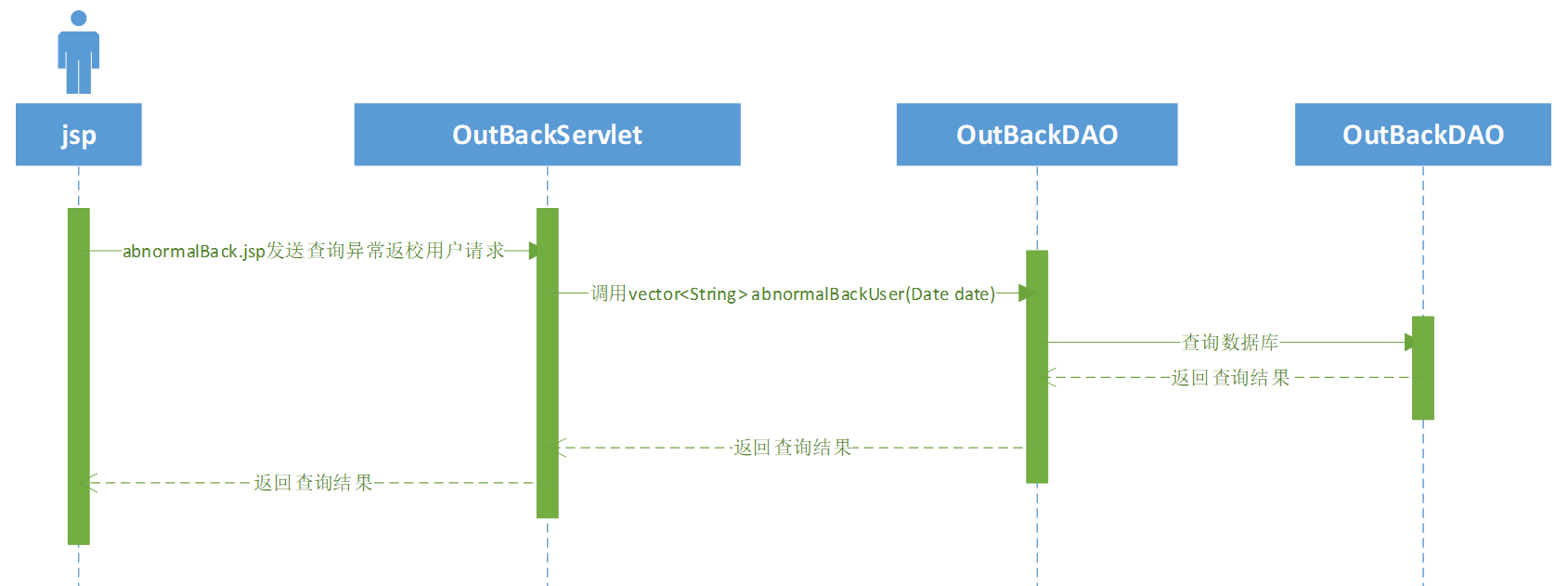
（10）查看用户行动轨迹业务



（11）查看实时感染情况业务

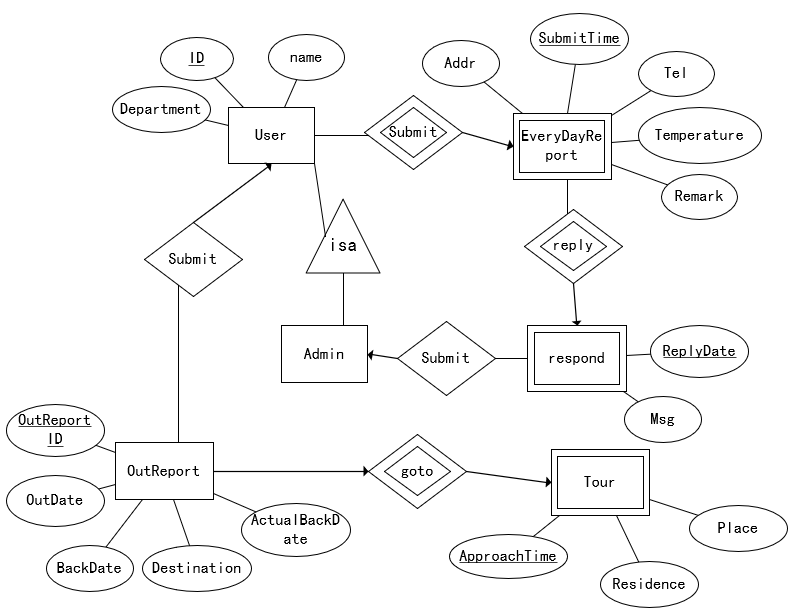


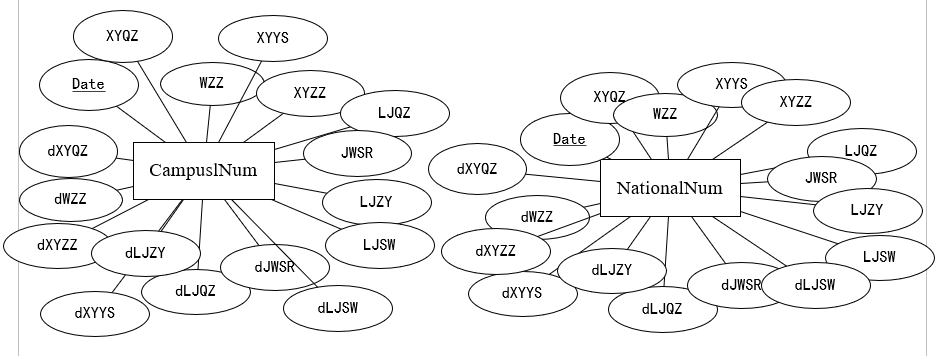
（12）修改实时感染信息业务



3.数据库设计

（1）逻辑结构设计





（2）物理结构设计

CREATE TABLE User(

ID VARCHAR (20) PRIMARY KEY,

Pswd VARCHAR (50) NOT NULL,

Name VARCHAR (50) NOT NULL,

Department VARCHAR(50),

IsAdmin BIT)

CREATE TABLE EverydayReport(

ID VARCHAR(20) NOT NULL,

SubmitTime DATETIME NOT NULL,

Tel VARCHAR(20) NOT NULL,

Addr VARCHAR(100),

Temperature DOUBLE NOT NULL,

Remark VARCHAR(200),

PRIMARY KEY (ID,SubmitTime),

FOREIGN KEY (ID) REFERENCES User (ID),

UNIQUE (ID,SubmitTime)

)

CREATE TABLE EverydayRespond(

ID VARCHAR(20) NOT NULL,

SubmitTime DATETIME NOT NULL,

ReplyTime DATETIME NOT NULL,

AdminID VARCHAR(20) NOT NULL,

Msg VARCHAR(100) NOT NULL,

PRIMARY KEY (ID,SubmitTime),

FOREIGN KEY (ID,SubmitTime)S REFERENCES EverydayRespond(ID,SubmitTime),

UNIQUE(ID,SubmitTime))

CREATE TABLE OutReport(

OutReportID INT UNSIGNED NOT NULL AUTO\_INCREMENT ,

ID VARCHAR(20) NOT NULL,

OutDate DATE NOT NULL,

PlannedBackDate DATE NOT NULL,

Destination VARCHAR(100) NOT NULL,

ActualBackDate DATE,

PRIMARY KEY (OutReportID),

FOREIGN KEY (ID) REFERENCES User (ID),

UNIQUE(ID,OutDate,PlannedBackDate))CREATE TABLE Tour(

OutReportID INT UNSIGNED NOT NULL,

ApproachTime DATETIME NOT NULL,

Province VARCHAR(50) NOT NULL,

City VARCHAR(50) NOT NULL,

Area VARCHAR(50) NOT NULL,

Address VARCHAR(100) NOT NULL,

PRIMARY KEY(OutReportID,ApproachTime),

FOREIGN KEY (OutReportID) REFERENCES OutReport (OutReportID))

CREATE TABLE NationalNum(

Date DATE UNIQUE PRIMARY KEY,

XYQZ INT UNSIGNED NOT NULL,

WZZ INT UNSIGNED NOT NULL,

XYYS INT UNSIGNED NOT NULL,

XYZZ INT UNSIGNED NOT NULL,

LJQZ INT UNSIGNED NOT NULL,

JWSR INT UNSIGNED NOT NULL,

LJZY INT UNSIGNED NOT NULL,

LJSW INT UNSIGNED NOT NULL,

dXYQZ INT NOT NULL,

dWZZ INT NOT NULL,

dXYYS INT NOT NULL,

dXYZZ INT NOT NULL,

dLJQZ INT NOT NULL,

dJWSR INT NOT NULL,

dLJZY INT NOT NULL,

dLJSW INT NOT NULL)

CREATE TABLE CampusNum(

Date DATE UNIQUE PRIMARY KEY,

XYQZ INT UNSIGNED NOT NULL,

WZZ INT UNSIGNED NOT NULL,

XYYS INT UNSIGNED NOT NULL,

XYZZ INT UNSIGNED NOT NULL,

LJQZ INT UNSIGNED NOT NULL,

JWSR INT UNSIGNED NOT NULL,

LJZY INT UNSIGNED NOT NULL,

LJSW INT UNSIGNED NOT NULL,

dXYQZ INT NOT NULL,

dWZZ INT NOT NULL,

dXYYS INT NOT NULL,

dXYZZ INT NOT NULL,

dLJQZ INT NOT NULL,

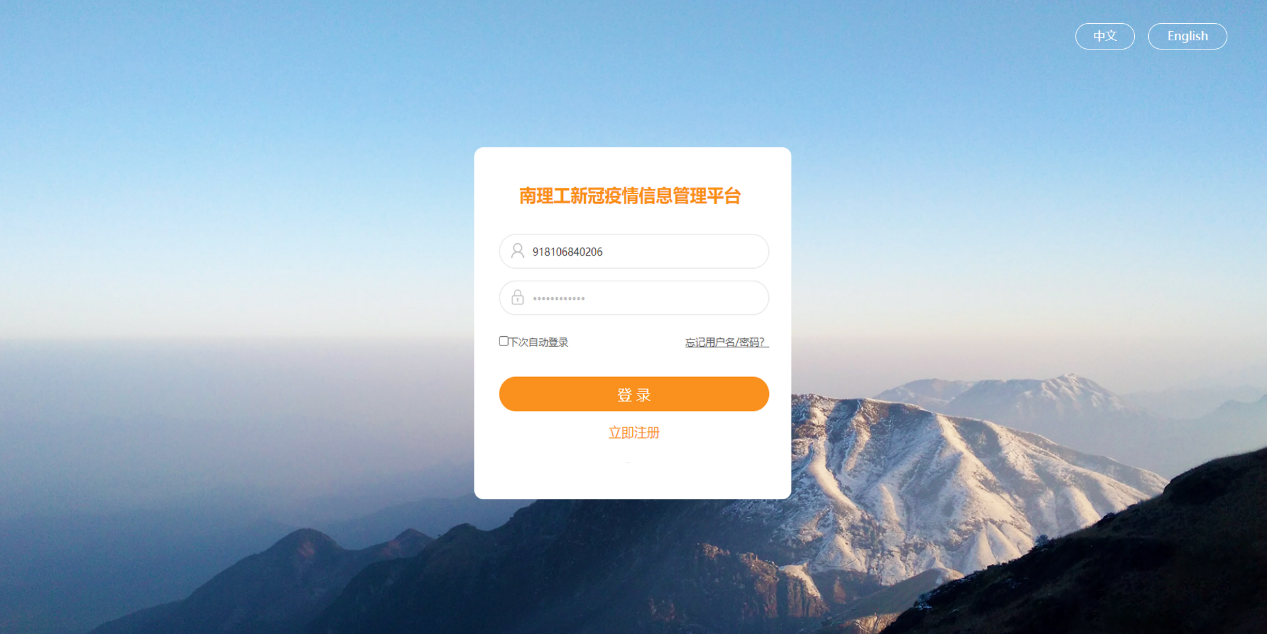
dJWSR INT NOT NULL,

dLJZY INT NOT NULL,

dLJSW INT NOT NULL)

4.页面预览

（1）login.jsp



（2）userIndex.jsp



（3）adminIndex.jsp



（4）manageUser.jsp



（5）everyReport.jsp



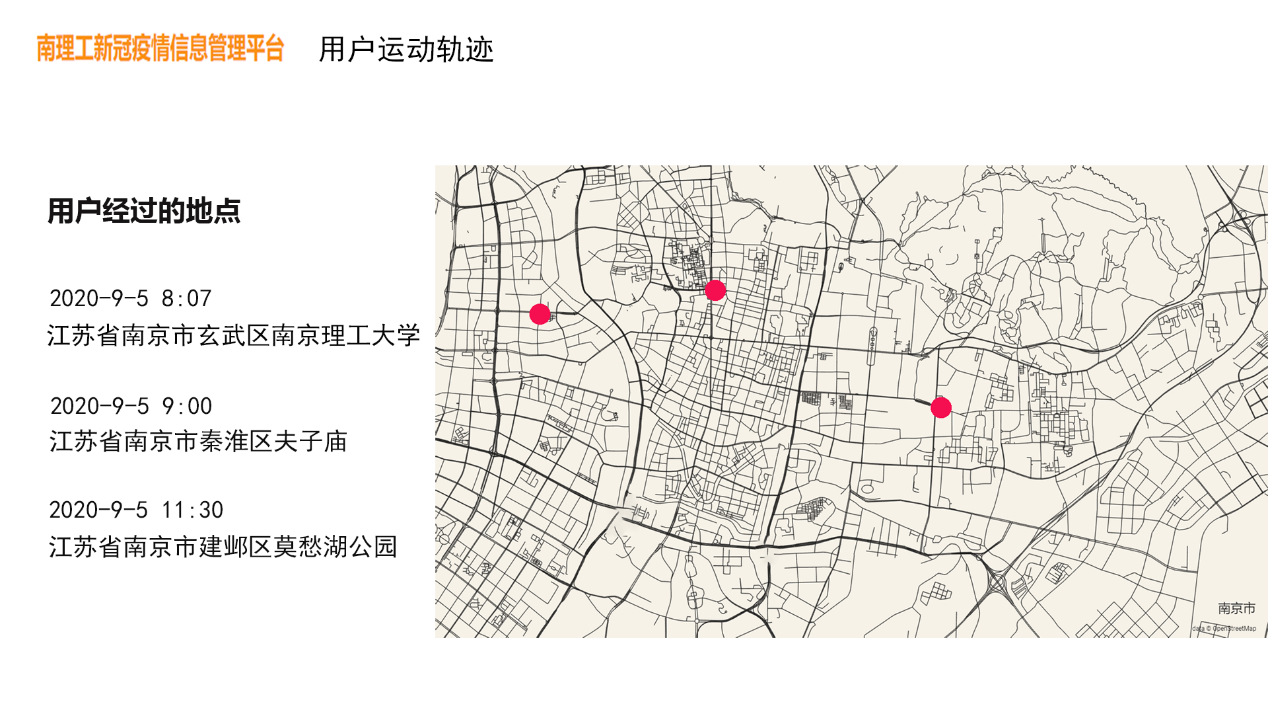
（6）replyInfo.jsp



（7）askOut.jsp



（8）track.jsp



（9）currentSituation.jsp

