班级序号\_\_32\_\_\_

**广州城市理工学院**

**2021-2022学年度第2学期**

**课程名称： 数据结构大作业**

**题　　目： 网约车管理系统**

**专业班级： 软件工程5班**

**年　　级： 20**

**姓　　名： 沈渺森**

**学　　号：202010098202**

|  |  |  |  |
| --- | --- | --- | --- |
| 一级指标 | 二级指标 | 分值 | 评分及成绩 |
| 得分 |
| 作品完成度  （占50%） | 1.工作量达标，功能点完整，独立完成 | 50 |  |
| 2.设计合理性，逻辑条理性 |
| 3.功能测试 |
| 4.数据处理水平 |
| 5.特色鲜明，有设计亮点 |
| 文档撰写（30%） | 1.文字描述规范，语言通顺，结构完整 | 30 |  |
| 2.思路描述清晰，功能测试结果完整 |
| 3.结果分析合理 |
| 答辩成绩 （占20%） | 1.能够对设计进行合理说明 | 20 |  |
| 2.能较好的回答答辩所提问题，解释合理清晰 |
| 合计（百分制） | | |  |
| 总 评 | □优秀 □良好 □中等 □及格 □不及格 | | 签名 |

概述.

程序通过ajax实现异步实时刷新网页，实现前后端分离。构造顺序表，并实现Iterator接口来快速遍历。后台通过io流读取文本，并储存到顺序表中。乘客可以输入目的地和上车地点，实时叫车。

在用户开始打车，遍历储存坐标的顺序表，储存最多十个在范围的点后，进行冒泡排序，根据先根据距离，若是相同则以注册时间对比。

开发环境：Window10，jdk1.8,IDEA. Jquery3.1.1,

1. 程序概要设计

若30秒无车辆扩大搜索时间和范围

1. 程序的总功能流程

90秒后无车则通知重新打车

实时打车

用户

订单查看

个人信息查看

在线心理测试系统

订单，用户，司机信息查看

管理员

个人信息查看

订单查看（获取的价格有抽成）

管理员

开始营运，随机移动

1. 功能实现程度

乘客：乘客可以输入目的地和起点，实时打车。在匹配中，根据司机的远近程序，注册时间进行排序，为乘客筛选最优的司机接单。在一点时间内没有司机接单，自动为乘客扩大搜索范围，以便乘客出行，乘客完成订单后可以查看订单，查询个人信息。

司机：司机随机生成坐标点，后每秒自动运动，接单后前往目的，并自动重新开始接单。如果完成订单后驶出营运范围，就讲司机的位置查询定位到花都广场。司机可以查看平台抽成后的收益，查询个人信息。

管理员：管理员可以获取完整的订单，完整的乘客和司机的个人信息方便管理。。

二.程序详细设计

1. 设定车辆活动范围

通过randomLon生成随机数，因为调用了百度地图的接口，大约计算了花都广场为中心的花都区半径

经度为double lon = ((double) (1132280 + randomLon.nextInt(2000))) / 10000;维度为double lat = ((double) (232470 + randomLon.nextInt(1600))) / 10000;。生成的坐标，再随机增加一定范围，实现自我运动后重新，在顺序表中进行坐标更新。坐标更新后，遍历顺序表重新输出文本进行缓存，If语句判断运动后的点是否在范围，或不在则重新回到花都广场。

1. 实现网约车车辆注册、驾驶员注册和乘客注册等功能

一个登录的jsp页面加loginServlet进行访问LoginDAO登录验证，表单传值判断登录类型。

登录成功并跳转到该用户的欢迎界面

3．驾驶员登陆功能，实现车辆出车、车辆接单和运营等功能；

司机登录后，点击开始运营，1.html 每秒自动通过ajax访问PointServlet，PointServlet根据前端上传的id，在坐标顺序表中查询，若无此id，则初始化生成随机坐标点，如果存在，就进行坐标点更新。Ajax异步实时刷新汽车位置。通过j记录当前遍历位置，调用顺序表的update方法

public String updateSQlist(String str) {

indata();

String str1 = "";

String splitPoint[] = new String[2];

String ID =str.split(" ")[2];

for (Object o :poSqlist) {

int j =0;

splitPoint = o.toString().split(" ");

if (splitPoint[2].equals(ID)) {

poSqlist.unpdate(str,j);

}

j++;

}

for (Object o : poSqlist) {

str1 += (String) o;

str1 += "\r\n";

}

outdata(str1);

str1 = "";

return str1;

}

乘客登陆功能，实现约车等功能；

调用百度的new BMap.Geocoder()类的getPoint后，通过回调函数获取poi的point，因为回调函数还不是很明白，使用不作说明。获取point的经度lng，维度lat。用ajax上传到servlet。

同时进行约车倒计时，如果90秒内有返回，则隐藏div弹窗。

while ((DriverString = reader.readLine()) != null) {

pointSplit= DriverString.split(" ");

if (new Point().getDistance(Double.parseDouble(a),Double.parseDouble(b),Double.parseDouble(pointSplit[0]),Double.parseDouble(pointSplit[1]) )<3\*I ){

correct[length]=DriverString +" "+new Point().getDistance(Double.parseDouble(a),Double.parseDouble(b),Double.parseDouble(pointSplit[0]),Double.parseDouble(pointSplit[1]));

length++;

}

}匹配车辆通过先获取十个在范围内的车辆信息，包括距离，司机id，坐标。用冒泡进行排序。返回最优的目标司机，将司机的坐标更改为乘客坐标，实现上车

if (mindistance.equals("")){

System.out.println("正在为你扩大范围");

mindistance= timer(a,b,i\*2,time\*2);

if (mindistance.equals("")){

return "暂无车辆,请重新打车";

}

}else {

return mindistance;

}

return mindistance;

}

如果30秒没有车辆，就扩大搜索时间time，扩大搜索距离i；重新查找车辆。在排序过程中，先比较distance距离，如果距离相等，则比较注册时间。

if(distance1==distance2)

{

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");

Date sd1=df.parse(Time);

Date sd2=df.parse(Time1);

if (sd1.after(sd2)){

String temp=Return[j];

Return[j]=Return[j+1];

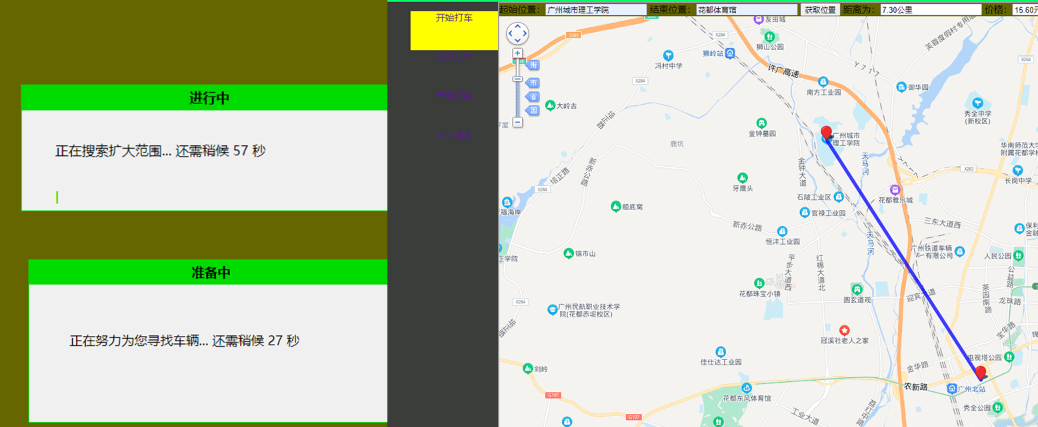
Return[j+1]=temp;

}

Java [Date类](https://so.csdn.net/so/search?q=Date%E7%B1%BB&spm=1001.2101.3001.7020)的after()方法测试日期是否在指定的日期之后，如果在就返回true，把该司机放在后面。最后返回Return[0]作为最优司机接单，后台把该司机坐标更改为乘客坐标，并在前端通知乘客已经上车。

三．使用情况

1．乘客打车，以及弹窗

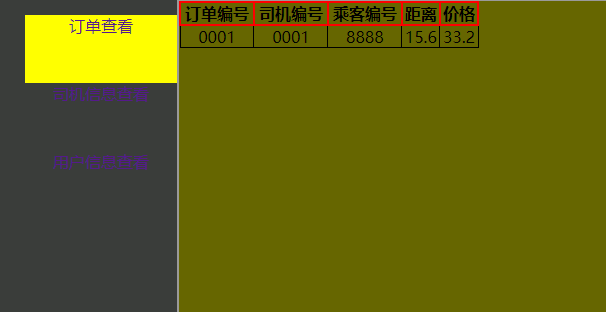


输入乘客可以输入目的地和上车地点，显示价格和距离，点击生成订单，实时叫车。30秒后无车辆自动扩大搜索半径。如果有车辆接单则通知乘客，你已经上车，跳转到导航2.html页面。实时显示当前位置。完成订单后，乘客点击查看订单可以实时查看历史所以订单。点击个人信息可以进行查看。页面增加了高亮显示，点击功能，该功能模块会进行高亮

2．司机营运，查询收益



司机点击开始营运则上传当前坐标点，每秒一次，点击结束营运则停止与服务器发送消息，已经完成的订单可以点击查看订单查看到，获取的价格是乘客支付价格的七成，平台抽三成。点击个人信息可以进行查看。页面增加了高亮显示，点击功能，该功能模块会进行高亮。

1. 平台管理员查看订单，顾客实名信息，司机实名信息。

管理员只能查看信息，不能对用户和司机信息进行修改。管理员可以完整获取订单中的司机编号，乘客编号以及实际价格。以便于平台的运营。

五．小结

本次数据结构的网约车管理系统，让我学到了很多，首先本来想写成gui的，但考虑到本学期学习了不少html+css。故使用了web的形式完成。但在web中，数据共享和传递并不是很方便，还要对页面进行实时更新，于是查看资料，采用一部分JavaScript方便前端的数据更新，同时运用MVC结构，提高代码的可读性，在写代码途中，强迫自己建立Bean、DAO、Servlet的分类。其次一开始计划使用顺序表来方便在不同编号的司机中采用二分法寻找范围内的司机。但最后写成了采用文本储存，顺序表储存在内存中进行遍历。就应该更改为链表。因为链表在查找方面更快，即使创建一个内部类实现Iterator接口，来帮助查找也使代码过于冗杂。但因为坐标写入中，一直对数据进行修改，增加。考虑到这只是小规模的遍历，就继续采用顺序表来执行。第三：tomcat发布war文件后，运行的文件地址是以tomcat/bin的所在目录查找的，于是了解到了this.getClass().getClassLoader().getResource().getPath() ;通过这个方法，在项目项目更换服务器后也可以方便获取实际路径。第四，为了实现地图，原本想用二维数组，或者建立图来实现，然后寻找最小路劲获取司机，但由于web项目可以实现接口，使用我就调用了百度地图的api，在这个过程中，查看百度的api文献，获取他自带的方法等等。这一过程即痛苦也快乐。

本次大作业都是由本人独立完成，在完成过程中遇到过各种各样的问题，首先是一个课外知识的扩展，对课本内容的巩固，在思考如何实现功能时，会先把功能写出来，后面再主动去搜索实现代码的精简。一些方法和类，写代码的过程中，深刻认知到了不同数据结构之间的特点，同时也增强了个人学习能力与实践能力，这是我在本次作业中的最大收获。独立查询资料，完善信息，提高功能的完整度，过程很艰苦，但完成之后的喜悦更加甜美。虽然很多功能冗余度高，一直进行优化，对代码进行精简，对于代码的书写过程还是不满意。在将来的学习中，会尝试用更多新知识来重写此次大作业，做到温故而知新。

六．参考目录

1.React-BMapGL文档 ：<http://huiyan.baidu.com/github/react-bmapgl/>