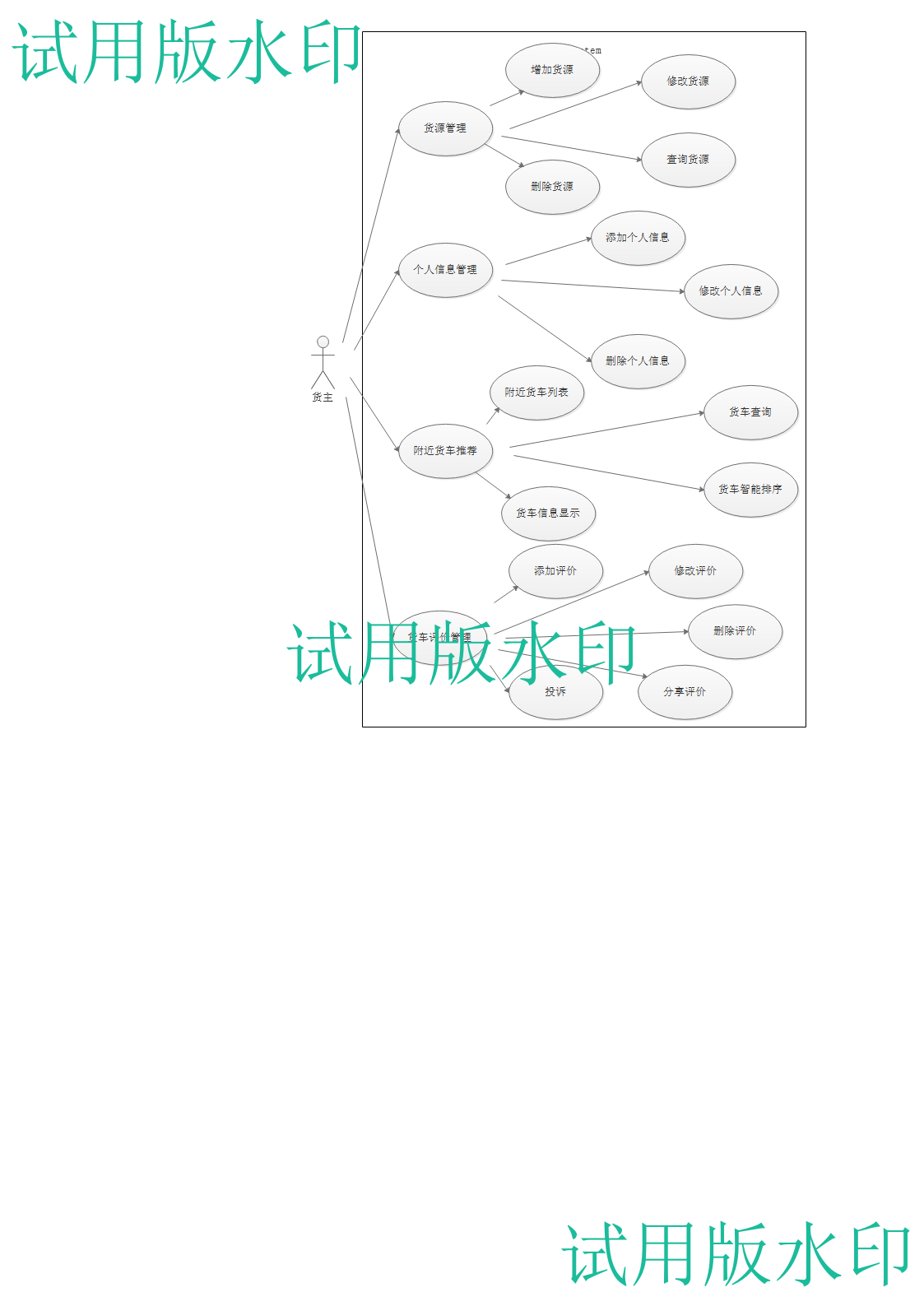
模块功能描述:



本模块主要是对货主所使用的功能实现。主要包括四大模块：货源信息管理，个人信息管理，附近货车推荐，货车评价管理。

1. 货源信息管理

货主进入“货主管理”模块，使用货源信息管理功能，可以添加自己运货的需求，输入货物种类，货物出发地，货物目的地，运货时间，期望价格等等基本信息。货源信息成功添加之后，运货司机变可以了解到货主的运货需求。货主还可以修改已发布的信息，删除发布信息，查询发布的信息等等。

1. 个人信息管理

货主进入“个人信息管理”模块，管理维护自己的基本信息。可以添加货主的信息，比如姓名，性别，联系方式等等。可以添加多个货主的信息，适应实际的运货需求。货主可以修改基本信息，删除基本信息。

1. 附近货车推荐

货主进入“附近货车推荐”模块，帮助用户快速找到自己需要的货车。

1. 附近货车列表

货主可以获得附近货车的信息，以列表的形式展现给用户。列出用户附近的货车车辆，具体信息包括：车型，位置，载重，车牌号等等。用户点击其中一条即可查看详细信息。

1. 货车信息查询

货主可以查询自己需要的货车，输入查询关键字，比如“面包车、海淀区”，app就会给用货主查询海淀区的面包车，并将所有的查询结果以列表的形式逐渐返回给用户，用户下拉，便可获得更多。如果用户点击一条，便显示这条货车的基本信息。

1. 货车智能排序

用户点击“智能排序”，便可以获得自己需要的货车列表，比如适合自己的货物重量的车型，适合自己运送货物种类的货车条件等等。

1. 货车信息显示

用户点击“货车信息显示”，便可以显示该条货车的所有详细信息，比如车牌号，司机姓名，货车类型，货车现在位置等等。

1. 货车评价管理

货主进入“评价管理”模块。货主可以对评价信心进行管理。

1. 添加评价

货主在完成一次交易之后，可以对此次运货提供评价。比如对货车司机的服务评分，对运货进行描述，是否守时，价格是否合理等等。这些评价可以让app参考对该司机进行评分。

1. 删除评价
2. 修改评价
3. 分享评价

货主可以将自己的评价分享给自己的好友，其作用相当于给好友推荐司机，如果此评价是正向的，相当于将该好友推荐该司机。如果评价负向，相当于给好友提示，不要雇佣该司机。

1. 投诉

货主可以对该次运货行为进行投诉，投诉将发给后台，后台将对该评价进行核实。如果投诉属实，将对该司机账号进行处理。

功能实现：

1. 货源管理
2. 添加货源信息

货主填写完货物重量，货物类型，运货时间，出发地，目的地之后，点击提交。App调用addGoods(Goods)方法向服务器发送添加货物的请求。服务器收到该请求的路由“/goods/add”之后传递给相应的路由并且进行处理，将该条信息添加到数据库中，并且向客户端返回添加的结果，成功或失败。

1. 修改货源信息

货主首先选择要修改的信息，修改完要修改的属性之后，向服务器端发送“/goods/update”请求，服务端接受该请求并传递给相应的路由，调用更新该信息的方法updateGoods(Goods)更新该条数据。

1. 查询货源信息

货主输入查询的关键字，如“6吨、海淀区”，向服务端发送“/goods/query”请求，服务端接受请求之后将该请求传递给相应的路由，该路由中调用查询货源的方法，queryGoods(String)方法，查询符合要求的货源，并将该货源以列表的形式返回给客户端并且显示。

1. 删除货源信息

货主选择要删除的货源，获得该货源的ID，向服务端发送“/goods/delete”请求，并且传递给服务端货源ID，服务端获得该请求并且将该请求传递给相关的路由，该路由调用deleteGoods(String)方法从数据库中删除该条记录。并且将删除结果返回给客户端并且显示。

1. 个人信息管理
2. 添加个人信息

货主输入姓名，性别，联系方式的个人信息之后，点击“添加”按钮，向服务端发送“/info/add”请求，服务端接受该请求之后，将该请求传递给相应的路由，该路由调用addInfo(Info)方法，将该用户的个人信息添加到数据库中，并向客户端返回添加结果。

1. 修改个人信心

货主选择要修改的信息，点击“修改”按钮，此时向服务端发送“/info/update”请求，服务端接受到该请求之后，将该请求传递给相应的路由，该路由调用修改方法updateInfo(Info)方法，将该修改后的个人信息保存到数据库中，并且将修改的结果返回给客户端。

1. 删除个人信息

用户选择要删除的个人信息，获得该信息的ID，点击删除按钮，向服务端发送“/info/delete”请求，服务端接受该请求并且将该请求传递给相应的路由，该路由调用deleteInfo(String)方法，将该用户的信息从数据库中删除。

1. 附近列车推荐
2. 附近货车列表

货主点击“获取附近货车”按钮，获取货主此时的位置坐标，此时向服务端发送“/nearby/getCars”请求，服务端接受该请求之后将该请求传递给相应的路由，该路由调用getNearbyCars（Position）方法，查询出附近的列车信息，以列表的形式返回给客户端并且显示。

1. 货车信息查询

货主输入要查询的条件，输入关键字，比如“小面包，海淀区”，此时向服务端发出“/nearby/queryCars”请求，服务端接受该请求之后将其传递给相关的路由，该路由调用queryCar(String)方法，查询符合要求的货车信息，比如“在海淀区运货的小货车”，然后将该信息已列表的形式返回给客户端并显示。

1. 货车智能排序

货主点击“智能排序”按钮，此时app调用我们根据实际情况设计的排序算法，对获取的货车进行排序，列出最适合该货主的货车列表，将排序后的列表重新显示在客户端。

1. 货车信息显示

用户点击某一条货车信息，此时跳转到一个新的页面，该页面专门用来展示该货车的信息，会将该货车的全部详细信息按格式显示，例如，货车号，货车司机姓名，货车类型，货车载重，货车位置等等。

1. 评价信息管理
2. 添加评价

货主输入对该次运货的司机评价信息，点击提交。此时向服务端发送“/comments/add”的请求，服务端接受请求之后将其转发给相应的路由，该路由调用addComments(comments)方法，将该评价信息保存在数据库中，并且将添加结果返回给客户端。

1. 修改评价

货主完成对评价信息的修改，点击保存信息，此时向服务器发送“/comments/update”请求，服务端接受该请求之后，将该请求传递给相应的路由，该路由调用函数updateComments(Comments)将该信息从数据库中更新。并将更新结果返回给客户端。

1. 删除评价

货主选择一条自己的评价，此时获取该条评价的ID，此时向服务器端发送“/comments/delete”请求，服务器端接受到该请求之后，将该请求转发给相应的路由，该路由调用函数deleteComments(String)删除该条评价，并将删除结果返回给客户端并显示。

1. 分享评价

货主选择一条评价，app获取该条评价的ID，此时向服务器端发送“/comments/share”请求，服务器端接收到该请求之后将其转发给相应的路由，该路由调用函数shareComments(ID)方法，首先从数据库中查询出该条评论的所有类容，并将该条信息发送给需要被分享的对象。

1. 投诉

货主点击“投诉”按钮，输入投诉的基本信息，此时向服务器端发送“/comments/complaint”请求，服务器端接受该请求后将其转发给相应的路由，该路由调用函数complaint将该投诉信息保存在数据库中等待审核人员审理，若审核结果属实，则对投诉的司机账号进行处理，比如降低其信誉度等级等待。

实现技术：

1. node js

使用node js为服务器端语言。

1. express框架

使用express为框架来完成对不同请求的处理，其中使用Router使得路由的编写更加规范化，每一块相关功能都交给一个Router处理，app.js接受到请求之后就先按照其路由分发给不同的Router。

关键类设计：

1. app.js

app.js主要负责对请求URL的处理，该类获得URL请求之后对其进行分类:

1.“/goods/\*”的URL交给goodsRouter处理,

2.“/info/\*”的URL交给infoRouter处理

3.“/comments/\*”的URL交给commentsRouter处理

4.“/nearby/\*”的URL交给nearbyRouter处理

1. goods.js

goods.js主要负责与货源有关的操作，例如增删改查

1. add(Goods):用来将货源信息添加到数据库中
2. update(Goods):用来更新货源信息
3. queryGoods(String)用来查询带有关键字的货源信息
4. deleteGoods(String):用来删除指定的货源信息:
5. comments.js

comments.js主要负责与货主评论有关的操作

1. add(Comments):用来添加评论
2. update(Comments):用来更新评论
3. delete(String):用来删除指定的评论
4. share(Comments):用来分享评论
5. complain(Complain):用来处理投诉信息:
6. nearbyCar.js

nearbyCar.js主要负责与附近货车推荐有关的功能操作。

1. getNearbyCars(Position):获得该position附近的所有货车信息
2. queryCars(String):查询符合关键字描述的货车信息
3. sorted():对货车列表进行智能排序
4. show():显示某一货车的详细信息
5. dbUtil.js

dbUtils主要负责对数据库sql语句的封装，能够更加简便的对数据库进项增删改查操作

1. save(Object):将该对象保存在数据库中
2. update(Object):将该对象进行更新
3. query(Object):查询符合要求的对象
4. delete(Object):将该对象删除
5. goodsBean.js

goodsBean.js主要对货物信息进行封装。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 中文名 | Setter方法 | Getter方法 | 备注 |
| Name | 货物名称 | setName(String) | getName() | 货物名称 |
| Weight | 重量 | setWeigth(int) | getWeight | 货物的重 |
| transferFee | 运费 | setTransferFee(float) | getTransferFee() | 期望运费 |
| beginPlace | 出发地 | setBeginPlace(String) | getBeginPlace | 无 |
| endPlace | 目的地 | setEndPlace(String) | getEndPlace() | 无 |

1. infoBean.js

infoBean.js主要对用户个人信息进行封装

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 中文名 | Set方法 | Get方法 | 备注 |
| Name | 姓名 | setName(String) | getName() | 货主姓名 |
| Sex | 性别 | setSex(int) | getSex() | 货主性别 |
| Phone | 手机号 | setPhone(int) | getPhone() | 货主联系方式 |

1. commentsBean.js

commentsBeans.js主要对用户的评论进行封装

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性 | 中文名 | Set方法 | Get方法 | 备注 |
| Content | 内容 | setContent(String) | getContent() | 评价内容 |
| Sore | 等级 | setSore(int) | getSore() | 评价星级 |
| Name | 评论货主 | setName(String) | getString() | 评论者 |