

智能出行车辆一体化服务管理系统
软件体系结构描述文档



小组成员： 涂远鹏-1652226

刘铸煌-1652313

黎盛烜-1652310

雷成钤-1652307

指导老师： 王继成

目录

1. 引言	1
1.1. 编制目的	1
1.2. 词汇表	1
1.3. 参考资料	1
2. 产品概述	1
3. 体系结构模型	2
3.1. 整体架构描述	2
3.2. 用户层面模块分解	6
3.3. 业务层面模块分解	6
3.4. 数据层面模块分解	9
3.5. 界面模块设计	10
3.6. 运行时组件	11
3.7. 物理部署	12
4. 模块之间的映射	13
4.1. 调用关系映射	13
4.2. 数据模型	13
5. 系统体系结构设计思路	15

1. 引言

1.1. 编制目的

本报告详细完成对智能出行车辆一体化服务管理系统的概要设计,达到指导详细设计和开发的目的,同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写,是了解系统的导航。

1.2. 词汇表

词汇名称	词汇含义	备注
ITVISMS	Intelligent Travel Vehicle Integrated Service Management System	智能出行车辆一体化服务管理系统
Bmob	云端服务器	在线云端服务器
Dd	Data dictionary	数据字典
TDT	Task Decomposition Technology	任务分解技术
Dpc	Data processing cycle	数据处理流程
Dpss	Data processing system security	数据处理系统安全性
OF	Operational Feasibility	操作可行性

1.3. 参考资料

- 1) 郭霖. 第一行代码[M]. 人民邮电出版社:郭霖, 2014. 69-307
- 2) IEEE[1471-2000]标准
- 3) 骆斌. 软件工程与计算(卷二)——软件开发的技术基础[M]. 机械工业出版社:骆斌, 2016. 82-118, 403-410
- 4) 郭志宏. Android 应用开发详解[M]. 电子工业出版社. 2010.
- 5) 余志龙, 陈昱勋, 郑名杰, 陈小凤, 郭秩均. Google Android SDK 开发范例 大全[M]. 人民邮电出版社. 2009.
- 6) 李宁. Android/OPhone 开发完全讲义[M]. 中国水利水电出版社. 2010.

2. 产品概述

现有的地图软件能够提供一定程度的停车场定位服务,但是无法提供停车场内部更精确的车位信息,而我们设计的系统整合了地图定位提供的导航服务和停车场管理处的车位实时信息,相比现有系统的服务,能够节省车主在停车场内部寻找车位的时间,避免导航到停车场却发现已满的情况。

另外,目前的地图软件没有一个整合车辆服务点的平台,用户只能查找到服务点后电话联系商家或到现场咨询,浪费时间,通过这个系统能在线上完成查询-预约-导航的整个流程,节省时间成本。对于商家来说,能够在线上完成车辆服务的预约管理,合理分配服务资源。

智能车辆出行一体化系统就是为了满足上述需求而开发的系统, 他包含一个数据集中服务器和若干个移动客户端，数据服务器用于将所有的用户数据集中便于维护和管理，用户则通过客户端完成相关的服务，客户端和服务端采用 TCP/IP 协议实时通信的方式完成数据的交换和维护。

3. 体系结构模型

3.1 整体架构描述

3.1.1 逻辑视角

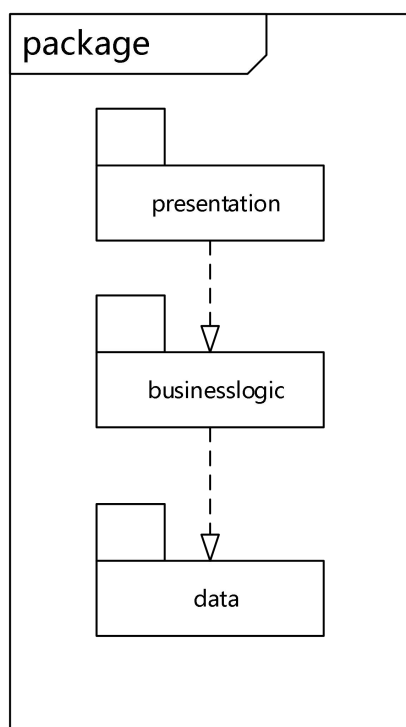
使用分层风格，将系统分为 3 层：

展示层（presentation）

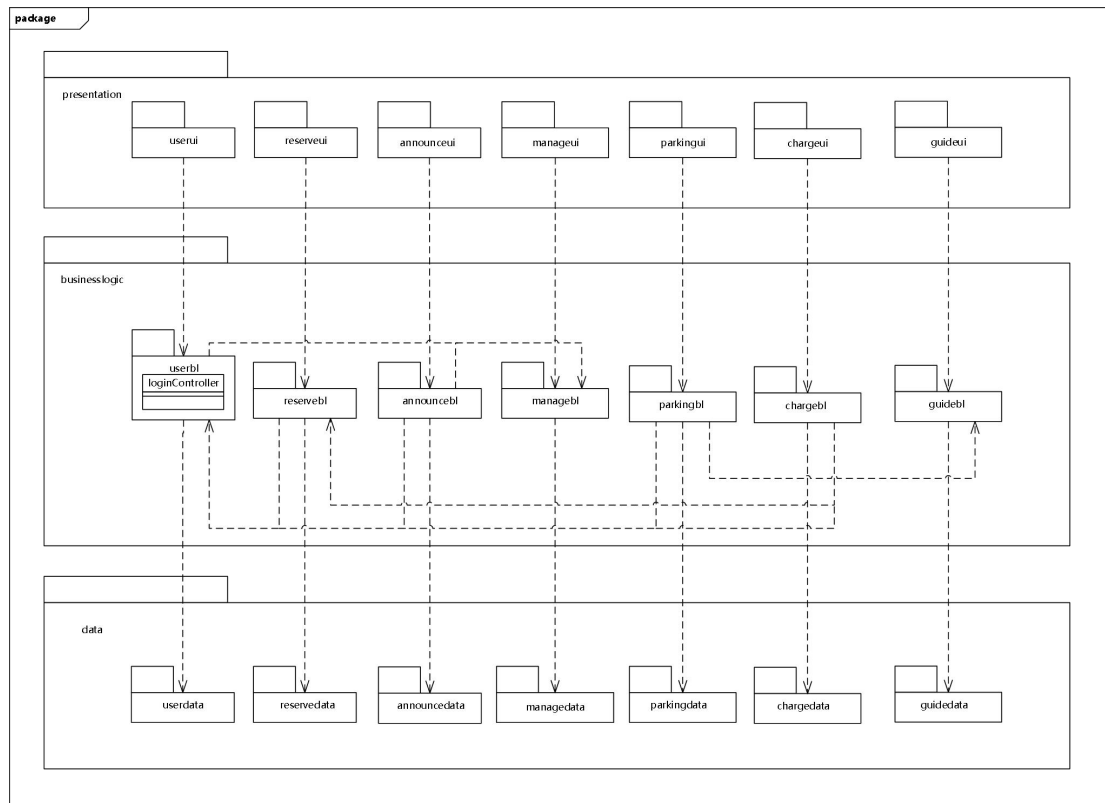
业务逻辑层（businesslogic）

数据层（data）

其中，展示层包含 GUI 页面的实现，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如下：



参照体系结构风格的包图表达逻辑视角



软件体系结构逻辑设计方案

3.1.2 组合视角

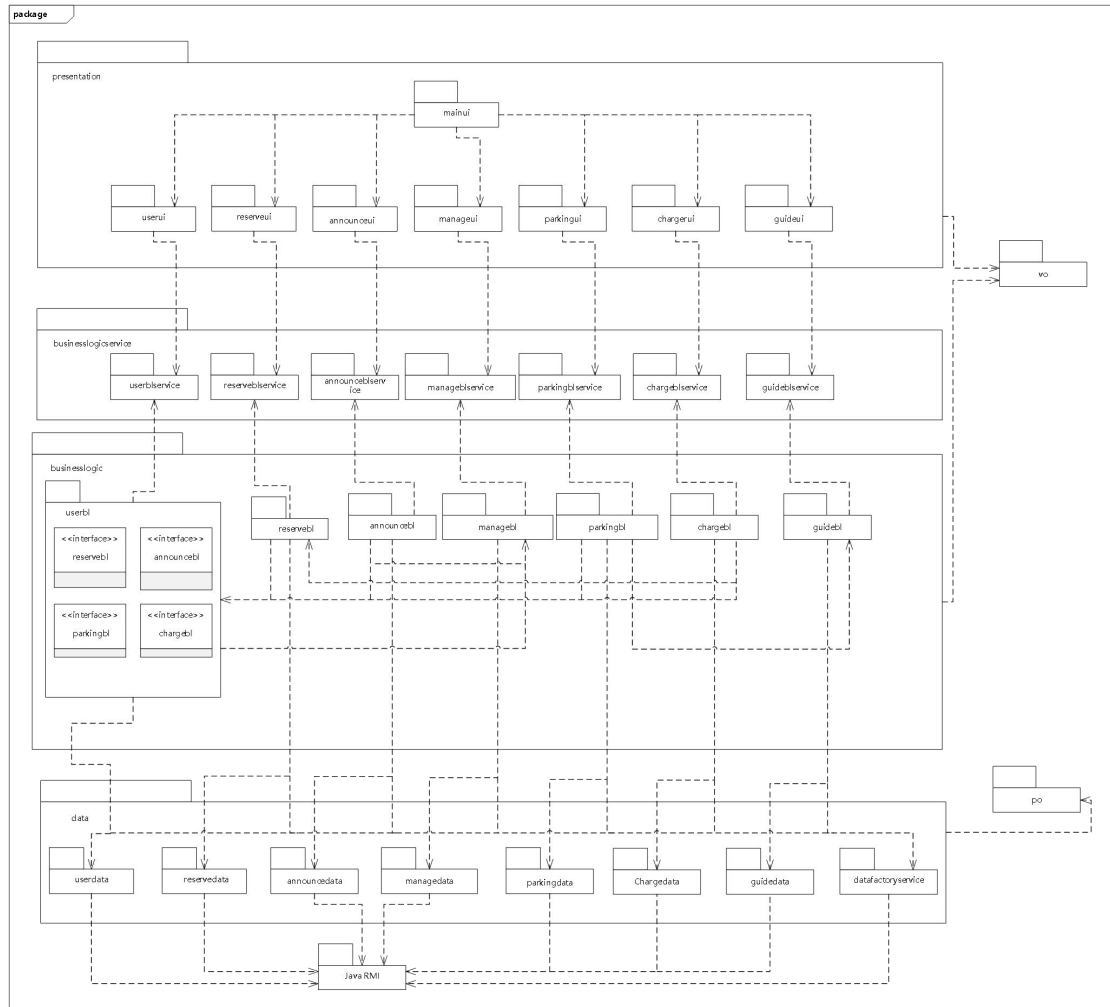
3.1.2.1 开发包图

智能出行车辆一体化服务管理系统的最终开发包设计表如下：

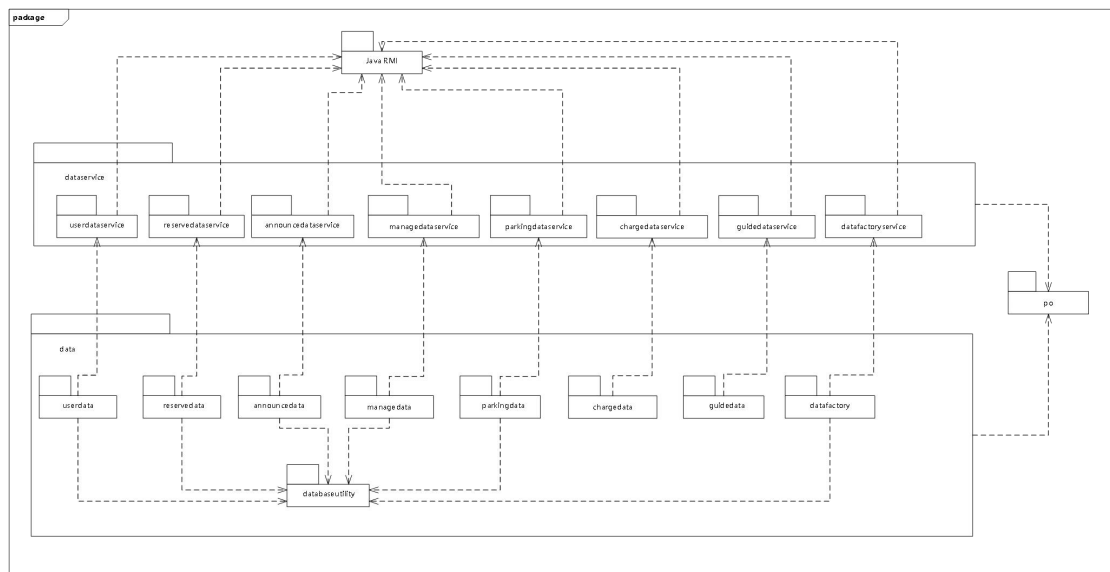
开发（物理）包	依赖的其他开发包
mainui	userui, reserveui, announceui, manageui, parkingui, chargeui, guideui, vo
userui	userblservice, 界面类库包
userblservice	
userbl	UserInterface, UserDataClient, UserPO
userdataservice	Java RMI, po
userdata	RMI, po, databaseutility
reserveui	reserveblservice, 界面类库包, vo
reserveblservice	
reservebl	reserveblservice, reservedataservice, po, userbl
reservedataservice	Java RMI, po
reservedata	databaseutility, po, reserveblservice
announceui	announceblservice, 界面类库包
announceblservice	
announcebl	announceblservice, announcedataservice, po, userbl, managebl
announcedataservice	Java RMI, po
announcedata	Java RMI, po, databaseutility

manageui	manageblservice, 界面类库包
manageblservice	
managebl	manageblservice, managedataservice, vo
managedataservice	Java RMI, po
managedata	Java RMI, po, databaseutility
parkingui	parkingblservice, 界面类库包
parkingblservice	
parkingbl	parkingblservice, parkingdataservice, po, userbl, guidebl
parkingdataservice	Java RMI, po
parkingdata	RMI, po, databaseutility
guideui	guideblservice, 界面类库包
guideblservice	
guidebl	guideblservice, guidedataservice, vo
guidedataservice	Java RMI, po
guidedata	Java RMI, po, databaseutility
vo	
po	
utilitybl	
界面类库包	
Java RMI	
databaseutility	JDBC

智能出行车辆一体化服务管理系统客户端开发包和服务端开发包如下图所示：



智能出行车辆一体化服务管理系统客户端开发包图



智能出行车辆一体化服务管理系统服务器端开发包图

3.2. 用户层面模块分解

3.4.1. 用户层面模块的职责

下表所示为用户层面模块的职责

模块	职责
Mainframe	界面 Frame，负责界面的显示和界面的跳转

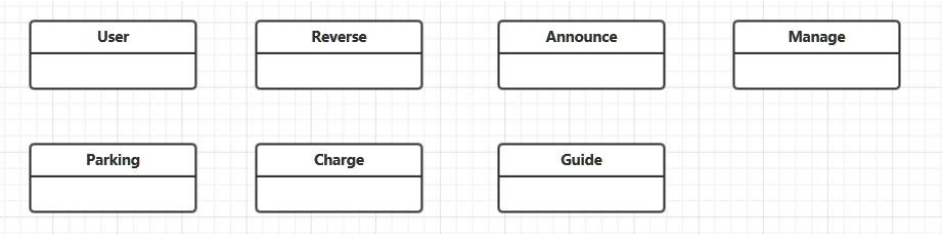
3.4.2. 用户层面模块的接口规范

用户层面模块的接口规范如下表所示

提供的服务（供接口）		
Mainframe	语法	Init(args:String[])
	前置条件	无
	后置条件	显示 Frame，以及 LoginPanel
需要的服务（需接口）		
服务名		服务
businesslogicservice.LoginBLService		登陆界面的业务逻辑接口
businesslogicservice.*BLService		每个界面都有一个相应的业务逻辑接口

3.3. 业务逻辑层模块分解

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如，User 对象负责处理登录界面的业务逻辑；Reverse 对象负责处理预定界面的业务逻辑。



3.3.1. 业务逻辑层模块的职责

业务逻辑层的模块的职责如下表所示：

模块	职责
Userbl	实现身份认证和用户登录界面需要的服务
Reservebl	实现预定界面需要的服务
Announcebl	实现更改信息发布需要的服务
Managebl	提供数据管理需要的服务
Parkingbl	提供停车导航需要的服务
Chargebl	提供费用计算需要的服务
guidebl	提供导航需要的服务

3.3.2. 业务逻辑层的接口规范

Userbl 模块接口规范

提供的服务（供接口）

User.login	语法	Public ResultMessage login(long id,String password)
	前置条件	Password 符合输入规则
	后置条件	查找是否存在相应的 user，根据输入的 password 返回登录验证的结果
需要的服务（需接口）		
服务名		服务
DataBaseFacory,getUserDataBase		得到 User 数据库的服务的引用
UserDataService.insert(UserPo po)		在数据库中插入 UserPo 对象

Reservebl 模块接口规范

提供的服务（供接口）		
Reserve.addMember	语法	Public ResultMessage addMember(long id)
	前置条件	启动一个服务预定回合
	后置条件	在一个服务预定回合中，增加预定的用户信息
Reserve.addReserve	语法	Public ResultMessage addReserve (long id,long quantity)
	前置条件	启动一个服务预定回合
	后置条件	在一个服务预定回合中，增加预定的场地信息
Reserve.getTotal	语法	Public ResultMessage getTotal(long id,long quantity)
	前置条件	已添加用户信息和预定服务信息
	后置条件	返回此次预定服务中需要支付的总额
Reserve.endReserve	语法	Public ResultMessage endReserve (long id,long quantity)
	前置条件	已支付
	后置条件	结束此次服务预定回合，持久化更新涉及的领域对象的数据
需要的服务（需接口）		
服务名		服务
ReserveDataservice.find(long id)		根据 id 进行查找单一持久化对象
ReserveDataservice.insert(ReservePo po)		插入单一持久化对象
ReserveDataservice.delete(ReservePo po)		删除单一持久化对象
ReserveDataservice.update(ReservePo po)		更新单一持久化对象
ReserveDataservice.init(ReservePo po)		初始化单一持久化对象
ReserveDataservice.finish(ReservePo po)		结束持久化数据库的使用

Announcebl 模块接口规范

提供的服务（供接口）		
Announce.addMember	语法	Public ResultMessage addMember(long id)
	前置条件	启动一个场地信息更新回合
	后置条件	在一个场地信息回合中，增加场地信息的用户信息
	语法	Public ResultMessage addReserve (long

Announce.addReserve		id,long quantity)
	前置条件	启动一个场地信息回合
	后置条件	在一个场地信息回合中, 增加场地信息的场地信息
Announce.addMessage	语法	Public ResultMessage addMessage(long id,long quantity)
	前置条件	已添加用户信息和场地信息
	后置条件	添加一个场地更新信息
Announce.end Repairupdate	语法	Public ResultMessage endRepairupdate (long id,long quantity)
	前置条件	信息更新完成
	后置条件	结束此次报修信息更新回合, 持久化更新涉及的领域对象的数据
需要的服务 (需接口)		
服务名		服务
Announce Dataservice.find(long id)		根据 id 进行查找单一持久化对象
Announce Dataservice.insert(ReservePo po)		插入单一持久化对象
Announce Dataservice.delete(ReservePo po)		删除单一持久化对象
Announce Dataservice.update(ReservePo po)		更新单一持久化对象
Announce Dataservice.init(ReservePo po)		初始化单一持久化对象
Announce Dataservice.finish(ReservePo po)		结束持久化数据库的使用

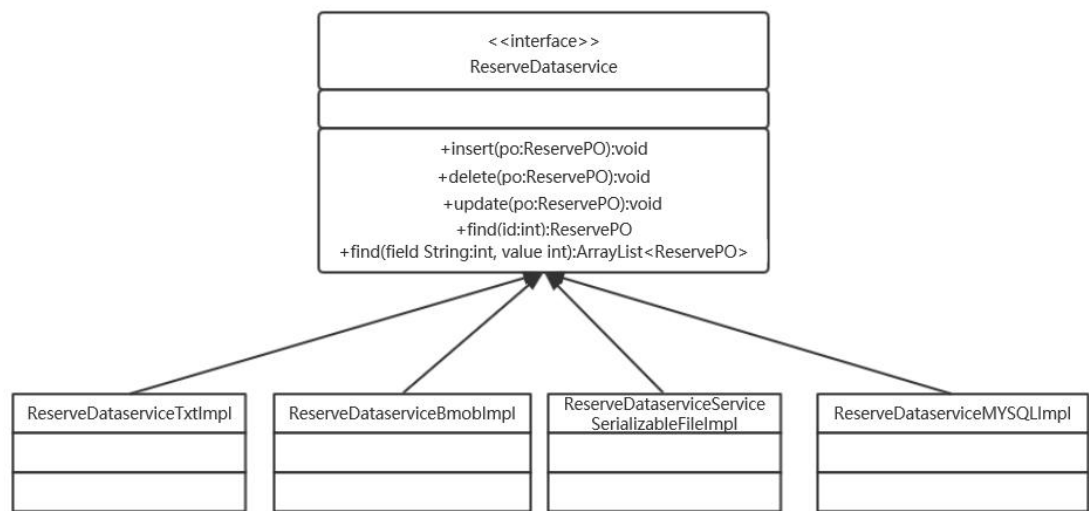
Guidebl 模块接口规范

提供的服务 (供接口)		
Guide.addMember	语法	Public ResultMessage addMember(long id)
	前置条件	启动一个导航回合
	后置条件	在一个导航回合中, 增加用户信息
guide.addCommodity	语法	Public ResultMessage addCommodity(long id,long quantity)
	前置条件	启动一个导航完成回合
	后置条件	在一个导航回合中, 增加停车点信息信息
Duide.end	语法	Public ResultMessage end Unsubscribe (long id,long quantity)
	前置条件	导航已完成
	后置条件	结束此次导航回合, 持久化更新涉及的领域对象的数据
需要的服务 (需接口)		
服务名		服务
guideDataservice.find(long id)		根据 id 进行查找单一持久化对象
guideDataservice.insert(ReservePo po)		插入单一持久化对象
guiderDataservice.delete(ReservePo po)		删除单一持久化对象
guideDataservice.update(ReservePo po)		更新单一持久化对象
guideDataservice.init(ReservePo po)		初始化单一持久化对象

guideDataservice.finish(ReservePo po)	结束持久化数据库的使用
---------------------------------------	-------------

3.4. 数据层模块分解

数据层主要给业务逻辑层提供数据访问，包含对持久化与阶段性数据的增、删、改、查。Reserve 业务逻辑需要的服务由 ReserveDataservice 接口提供。Announce 业务逻辑需要的服务由 AnnounceDataservice 接口提供。User 业务逻辑需要的服务由 UserDataService 接口提供。Manager 业务逻辑需要的服务由 ManagerDataservice 接口提供。Parking 业务逻辑需要的服务由 ParkingDataService 接口提供，Guide 业务逻辑需要的服务由 GuideDataservice 接口提供。Charge 业务逻辑需要的服务由 ChargeDataservice 接口提供。由于七类业务均以数据库的增删改查为主，所以以 Reserve 业务为例阐述，其余三类服务不再赘述。由于持久化数据和阶段性数据保存可能存在多种形式：TXT 文件，序列化文件，数据库等类型，所以抽象化了数据服务。Reserve 业务数据层模块的描述具体如下所示：



3.4.1. 数据层模块的职责

模块	职责
ReserveDataservice	持久化数据库的接口，提供集体载入、集体保存、增、删、改、查服务
ReserveDataserviceBmobImpl	基于在线云服务器数据库的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务
ReserveDataserviceTxtImpl	基于 TXT 文件的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务
ReserveDataserviceServiceSerializableFileImpl	基于序列化文件的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务
ReserveDataserviceMySQLImpl	基于 MySQL 本地数据库的持久化数据库的接入，提供集体载入、集体保存、增、删、改、查服务

3.4.2. 数据层模块的接口规范

提供的服务（供接口）		
ReserveDataservice.find	语法	Public void ReservePO find(long id) throws RemoteException
	前置条件	无
	后置条件	按照用户 ID 在数据中进行查找并返回查找结果 ReservePO
ReserveDataservice.insert	语法	Public void ReservePO insert(long id) throws RemoteException
	前置条件	相同的 ID 的 po 在 Mapper 中不存在
	后置条件	在数据库中增加一个 po 记录
ReserveDataservice.delete	语法	Public void ReservePO delete(long id) throws RemoteException
	前置条件	在当前数据库中存在与删除 ID 相同的 ID 号的 po
	后置条件	删除当前 ID 的一个 po
ReserveDataservice.update	语法	Public void ReservePO update(long id) throws RemoteException
	前置条件	在当前数据库中存在与更新 ID 相同的 ID 号的 po
	后置条件	按照用户 ID 在数据库中进行更新当前 ID 的一个 po
ReserveDataservice.init	语法	Public void ReservePO init() throws RemoteException
	前置条件	无
	后置条件	初始化持久化数据库
ReserveDataservice.finish	语法	Public void ReservePO finish() throws RemoteException
	前置条件	无
	后置条件	结束持久化数据库的使用

3.5. 界面模块设计

界面跳转设计：

普通车主用户：

主界面为功能选择界面，用户可以选择预约记录查询、取车、存车，主界

面左侧侧滑栏可以选择个人信息查询、密码修改、账户余额查询、反馈意见、关于我们、注销账号、系统公告查看、历史评论分别跳转个人信息查询界面、密码修改界面、账户余额查询界面、反馈意见填写界面、关于我们界面、登录界面、系统公告查看界面、历史评论查看界面，主界面预约记录查询、取车、存车分别跳转预约记录查询界面、未取车记录查看界面、周边停车场查看界面。

取车界面点击对应的预约记录跳转停车场导航界面，周边停车场查询界面点击对应的停车场跳转对应停车场信息查看界面。

取车导航界面在导航结束后跳转支付界面，支付界面若余额不足跳转充值界面，余额充足则跳转主界面或停车场评论界面，个人信息界面点击查看位置跳转 GPS 位置显示界面。

停车场信息查看界面点击查看公告跳转对应停车场公告查看界面，点击预约跳转服务预约界面，点击预约跳转对应停车场导航界面，存车导航界面在导航结束之后跳转存车界面，存车界面点击存车跳转主界面。

停车场管理员用户：

主界面为服务选择界面，选择的服务包含更改信息发布、信息检查，费用勘误、评论回复，分别跳转更改信息填写界面、信息查看界面、费用查看界面、评论回复界面。更改信息发布界面点击对应信息跳转信息填写界面，费用查看界面点击对应记录跳转费用修改界面。

服务点管理员用户：

主界面为服务选择界面，选择的服务包含更改信息发布、信息检查，服务项目介绍、项目信息修改、评论回复，分别跳转更改信息填写界面、信息查看界面、服务查看界面、评论回复界面。信息查看界面点击对应信息跳转信息修改界面，项目信息查看界面点击对应项目信息修改界面。

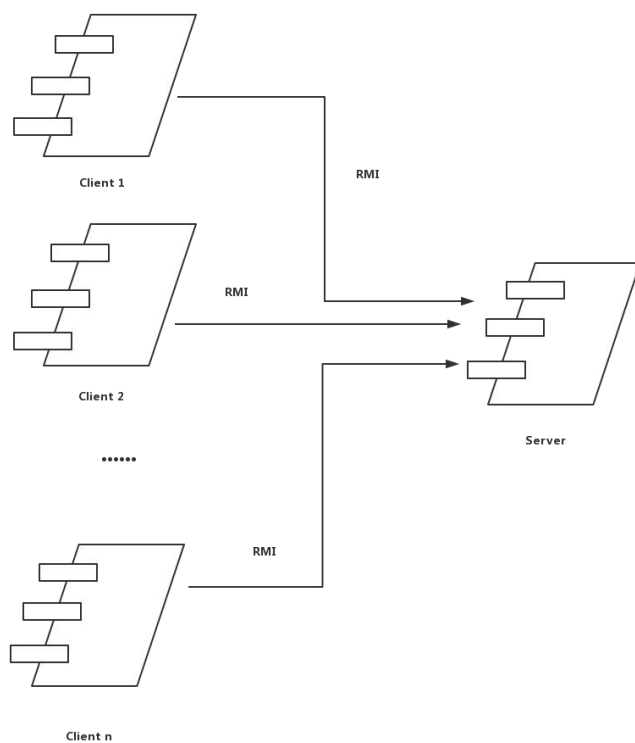
界面风格：

界面采用的风格为简约风格，以单一色调为主，背景设置为动画背景。界面直观对用户透明，用户接触该软件之后对界面上的功能一目了然，无需培训便可使用该软件。界面设计采用以线性布局 LinearLayout 为主，相对布局 RelativeLayout 为辅的设计方法。字体为灰底黑字，部分为灰底红字，字体大小为宋体，大小为 14sp，部分显示带边框，对齐方式为居中对齐。

使用扁平化设计，弱化视觉透视，降低了光影纹理材质阴影对设计的影响，将界面设计回归到艺术的根本，图形和色块。它在交互领域主要体现在易理解，易操作，易用性和简单化上。

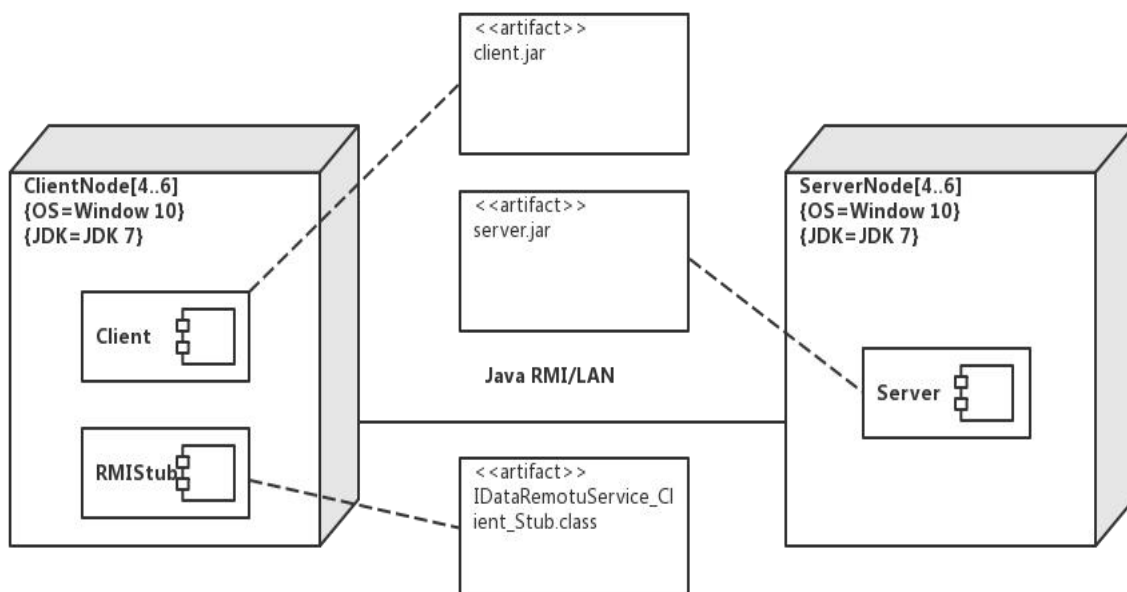
3.6. 运行时进程

在智能出行车辆服务一体化管理系统中，将有多个客户端进程和一个服务端进程，其进程图如下所示。结合部署图，客户端进程是在客户端机器上运行，服务端进程在服务器端机器上运行。



3.7. 物理部署

智能出行车辆服务一体化管理系统中客户端构件放在客户端机器上,服务器端构件放在服务器端机器上。在客户端节点上,部署 `RMISub` 构件。由于 `Java RMI` 构件属于 `JDK 10` 的一部分。所以在系统 `JDK` 环境已配置好的情况下,不需要再独立部署。部署图如下所示。



4. 模块之间的映射

4.1. 调用关系映射

车主用户：

(1)主模块调用存车模块、取车模块、服务点选择模块、预约记录查询模块，密码修改模块，个人信息查询模块、公告查询模块、用户反馈模块，登录模块、注册模块、账户余额查询模块,查看历史评论模块，查看服务进度查看模块，服务结算模块。

(2)个人信息查询模块调用 GPS 定位显示模块，账户余额查询模块调用充值模块。

(3)存车模块调用周边停车场查询模块，停车场查询模块调用服务预约选择模块，服务预约模块调用停车场导航模块。取车模块调用停车记录查询模块，后者调用停车场导航模块、，取车导航模块调用支付模块,支付模块调用充值模块，评论模块。

停车场管理员：

(1)主模块调用评论回复模块、费用勘误模块、发布更改信息模块、信息检查模块。

服务点管理员：

(1) 主模块调用评论回复模块、发布更改信息模块、信息检查模块、项目信息出入更改模块、服务项目介绍模块。

4.2. 数据模型信息视角

4.2.1 数据持久化对象

系统的 po 类就是对应的相关的实体类，在此只做简单介绍

*UserPo 类包含用户的系统编号、用户名、登录密码、支付密码、电话号码、邮箱、账户余额、性别

*ManagerPo 类包含停车场和服务点管理员的编号、登录密码、性别、手机号码、所属服务点/停车场编号

*ReservePo 类包含用户的编号、车牌号、所选服务、预约的停车场编号、预约停车场名

*AnnouncePo 类包含公告发布时间、公告发布人员编号、公告详情

*ParkinglotPo 类包含停车场编号、经度、纬度、停车场名、小型车 位数、中型车位数、大型车位数、地址

其中以持久化对象 User 的定义为例，其定义如下、其余 Po 类定义与 User 类定义类似：

```
public class _User extends BmobObject implements Comparable<_User>{
    private String username;
    private String password;
    private boolean mobilePhoneNumberVerified;
    private String mobilePhoneNumber;
    private int Money;
    private String Gender;
    private boolean emailVerified;
```

```

private String email;
public void setUsername( String username ) {
    this.username = username;
}
public void setPassword( String password) {
    this.password = password;
}
public void setmobilePhoneNumberVerified( boolean mobilePhoneNumberVerified ) {
    this.mobilePhoneNumberVerified = mobilePhoneNumberVerified;
}
public void setMobilePhoneNumber( String mobilePhoneNumber ) {
    this.mobilePhoneNumber = mobilePhoneNumber;
}
public void setMoney( int Money ) {
    this.Money = Money;
}
public void setGender( String Gender ) {
    this.Gender = Gender;
}
public void setEmailVerified( boolean emailVerified ) {
    this.emailVerified = emailVerified;
}
public void setEmail( String email ) {
    this.email = email;
}
public String getUsername() {
    return username;
}
public String getPassword() {
    return password;
}
public boolean getmobilePhoneNumberVerified() {
    return mobilePhoneNumberVerified;
}
public String getmobilePhoneNumber() {
    return mobilePhoneNumber;
}
public int getMoney() {
    return Money;
}
public String getGender() {
    return Gender;
}
public boolean getEmailVerified() {

```



```

        return emailVerified;
    }
    public String getemail() {
        return email;
    }
    @Override
    public int compareTo(_User userClassJava) {
        return 0;
    }
}

```

4.2.2. Txt 持久化格式

Txt 数据保持格式以 ReservePO.txt 和 ParkinglotPO.txt 为例,

ReservePO.txt 每行对应预约记录编号、预约时间、预约用户编号、服务点编号、服务点名称、所选服务、处理状态, 中间以冒号隔开如下所示:

```

001: 2018-08-07: 145624: 13654: 同济大学嘉定校区学生事务中心-停车场: 小型
车位、行车检查: 已处理
002: 2019-05-07: 142624: 13655: 同济大学-停车场: 中型车位、行车检查: 未处
理
003: 2019-05-08: 145625: 13656: 通达馆停车场: 大型车位、洗车: 已处理
004: 2019-05-09: 145626: 13657: 路侧停车位(斗门路): 小型车位、行车检查: 已
处理
005: 2019-05-11: 145627: 13658: 同济大学(嘉定校区)图书馆-停车场: 小型车位、
行车检查: 未处理
.....

```

ParkinglotPO.txt 每行对应停车场编号、经度、纬度、停车场名、小型车 位数、中型车 位数、大型车位数、地址:

```

001: 31.292277: 121.222051: 同济大学嘉定校区学生事务中心-停车场: 28: 10:15:
曹安公路 4800 号同济大学嘉定校区附近
002: 31.29479: 121.218345: 同济大学(嘉定校区)开物馆-停车场: 28:10:15: 上海市
嘉定区曹安公路 4800 号同济大学嘉定校区内
003: 31.292526: 121.223806: 停车场(嘉三路): 29: 10:15: 同丁路与春归路交叉口
西南 150 米
004: 31.293017: 121.219048: 同济大学(嘉定校区)图书馆-停车场: 28:10:15: 曹安
公路 4800 号
005: 31.290851: 121.512346: 同济大学-停车场: 29:10:15: 国康路 11 同济大学
.....

```

4.2.3. 云服务器数据库表

数据库中包含 Parkinglot 表、Reserve 表、ReserveHistory 表、User 表、Comment 表、Service 表、Announce 表

5. 系统体系结构设计思路

5.1. 用户界面的特殊细节

- 1) 用户登录界面时，若 login 返回的结果为 NoUser，发出错误提示，并询问是否需要注册当前账号。
- 2) 用户登录界面时，若 login 返回的结果为 GetUserDataFail，输出提示“当前与服务器连接失败，请稍后再试”。
- 3) 用户添加预定信息时，若 reserve.getTime 返回为 EXISTED，表明该用户同一时间已存在预约信息，不提供另外服务。
- 4) 停车场管理人员更新信息时，将信息更新结果反馈到预定的用户端。
- 5) 车辆服务点管理人员更新信息时，将信息更新结果反馈到预定的用户端。