<交大蕉蕉APP>

软件需求规约

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2019.6.1. | <0.0> | 初稿 | 林江浩 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 简介 4](#_Toc517096442)

[1.1 目的 4](#_Toc517096443)

[1.2 背景 4](#_Toc517096444)

[1.3 定义、首字母缩写词和缩略语 4](#_Toc517096445)

[1.4 参考资料 4](#_Toc517096446)

[1.5 概述 5](#_Toc517096447)

[2. 整体说明 5](#_Toc517096448)

[3. 具体需求 6](#_Toc517096449)

[3.1 功能 6](#_Toc517096450)

[**3.1.1** **<Use case 图>** 6](#_Toc517096451)

[**3.1.2** **<注册用例规约>** 7](#_Toc517096452)

[**3.1.3** **<登录用例规约>** 7](#_Toc517096453)

[**3.1.4** **<管理个人信息用例规约>** 8](#_Toc517096454)

[**3.1.5** **<预约座位用例规约>** 8](#_Toc517096455)

[**3.1.6** **<取消预约用例规约>** 9](#_Toc517096456)

[**3.1.7** **<查询预约记录用例规约>** 10](#_Toc517096457)

[**3.1.8** **<查询巴士实时位置用例规约>** 10](#_Toc517096458)

[**3.1.9** **<查询巴士班次信息用例规约>** 11](#_Toc517096459)

[**3.1.10** **<查询巴士线路信息用例规约>** 11](#_Toc517096460)

[**3.1.11** **<管理预约信息用例规约>** 11](#_Toc517096461)

[**3.1.12** **<发布系统公告用例规约>** 12](#_Toc517096462)

[**3.1.13** **<管理用户信息>** 12](#_Toc517096463)

[**3.1.14** **<管理巴士信息>** 13](#_Toc517096464)

[3.1.15 **<管理巴士线路信息>** 13](#_Toc517096465)

[3.2 易用性 13](#_Toc517096466)

[3.3 可靠性 14](#_Toc517096467)

[3.4 性能 14](#_Toc517096468)

[3.5 可支持性 14](#_Toc517096469)

[**3.5.1** **<Java编码规范>** 14](#_Toc517096470)

[**3.5.2** **<Android端编码规范>** 16](#_Toc517096471)

[**3.5.3** **<Web后端编码规范>** 18](#_Toc517096472)

[3.6 设计约束 19](#_Toc517096473)

[3.7 联机用户文档和帮助系统需求 19](#_Toc517096474)

[3.8 接口 20](#_Toc517096475)

[**3.8.1** **用户界面** 20](#_Toc517096476)

[**3.8.2** **硬件接口** 23](#_Toc517096477)

[**3.8.3** **软件接口** 23](#_Toc517096478)

[**3.8.4** **通信接口** 23](#_Toc517096479)

[3.9 适用的标准 23](#_Toc517096480)

软件需求规约 (简化版)

# 简介

## 目的

项目名为交大蕉蕉APP，是一款交大的二手交易信息平台，此文档的目的在于提供关于交大蕉蕉APP的需求功能的详述以及系统性能参数的说明。

## 背景

二手交易是指消费者将购买于企业或代理商、零售商后使用一段时间，或者未使用过的商品，作为一种新的商品再进行交易流通的行为。大学生这一群体，生活区域相近，消费心理相似，购买的需求商品具有极高的重合度，有着极高的二手交易的潜在需求。譬如在开学季，大学生会寻求以低价购入相应的教材，在一个学期的使用后重新以二手书的形式重新售出；在毕业季，毕业生的大部分书籍、生活用品、自行车都可以以二手交易的形式售出，否则也只能当做废品丢弃或回收。

而目前，交大师生进行二手交易的主要渠道主要有三：二手交易群、个人在朋友圈或空间发布交易信息、以闲鱼为代表的传统二手交易APP。

以“上海交大二手交易群”为例，此交易群的群成员已经超过3000人（其中不乏教职工的存在），并且正在建立更多的分群，大群平均每日发布的二手交易信息接近百条，若是碰上开学季、毕业季等高峰时期，每日的交易信息可以接近千条。可见交大当下对二手交易的需求极大，且还有不断增长的趋势。

但随着二手交易群的不断推广，以及二手交易行为的频率不断上升，以上三种交易渠道的弊端也逐渐凸显，比如，二手交易群没有历史检索、商品分类、智能搜索等功能，朋友圈发布消息传播性极低，而以闲鱼为代表的传统二手交易APP又具有非本土化和无法发布求购信息的不足。

综上，鉴于二手交易在交大师生群体中存在极大的需求空间，以及考虑到当下以二手交易群为代表的二手交易途径的种种弊端，通过现代化技术手段优化传统群发盲寻的二手群交易模式，，开发一款实用、易用的二手交易信息平台APP，克服二手交易群现有的使用问题，以满足交大师生对二手交易更便捷体验的需求，是十分有必要的。

## 定义、首字母缩写词和缩略语

暂无

## 参考资料（待填写）

Java Code Convention : <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

## 概述

该文档主要分为三个部分，第一部分即简介，主要对该文档的目的及项目的背景进行简要介绍；第二部分为整体说明，从整体的视角描述项目的效果、功能、特征、约束等；第三部分为具体需求，详细描述了项目的功能需求、质量属性、设计约束等。

# 整体说明

交大蕉蕉APP产品以达成上海交通大学内与二手交易相关的最好的网站/APP为目标，支持卖家发布出售信息、卖家发布求购信息、智能检索、自动匹配买卖双方等功能，后续可能会朝校园租赁服务等方向扩展功能。

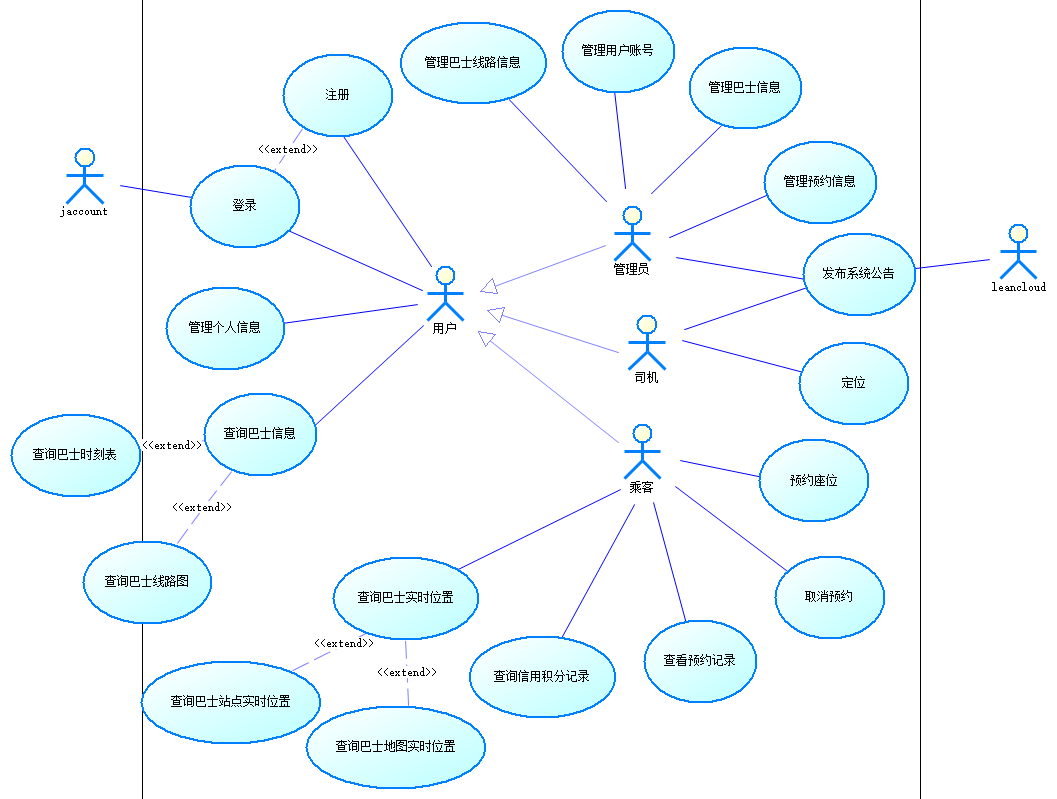
交大蕉蕉APP针对的用户群体广泛且稳定，为上海交通大学的在校师生，主要分布于闵行、徐汇两大校区，它的开发紧紧依赖于学生对更好的二手交易体验的需求，并以SJTU项目组的技术能力和开发经验为约束瓶颈。

交大蕉蕉APP开发时间预计为2个月，采用敏捷开发的Scrum软件过程，项目组为4人。

# 具体需求

## 功能（待填写）

### **<Use case 图>**



用例图

### **<注册用例规约>**

概述：所有用户都可以在校园巴士APP上进行注册

执行者：乘客

前置条件：无

基本流：

1. 系统提示用户输入用户名、密码、确认密码、手机、邮箱、身份、验证码
2. 用户输入系统要求的信息，并点击注册按钮
3. 系统提示注册成功

备选流：

* 1. 注册用户名重复、密码和确认密码不相同、验证码输入错误 -> 系统提示信息，返回2
  2. 用户点击登录按钮 -> 用例结束，进入登录用例

特殊说明：

1. 暂定不可随意注册管理员和司机账号，新管理员由旧管理员授权，司机的信息由管理员导入数据库，并与相应巴士班次绑定。

### **<登录用例规约>**

概述：所有用户都可以登录校园巴士APP

执行者：所有用户

前置条件：用户的账号已经存在

后置条件：无

基本流：

1. 系统提示用户选择通过第三方（jaccount）登录或账号密码登录
2. 系统提示用户输入用户名、密码和验证码，并选择身份（学生、老师、司机、管理员）
3. 用户输入用户名、密码和验证码，并选择身份
4. 系统确认用户名存在且与密码、身份匹配，并且验证码正确，成功登录

备选流：

* 1. 用户名不存在 -> 系统提示“用户名不存在”，用例结束，进入注册用例
  2. 用户名与密码、身份不匹配 -> 系统提示信息错误，返回到3
  3. 验证码输入不匹配 -> 系统提示“验证码错误”，返回到3

### **<管理个人信息用例规约>**

概述：所有用户都可以管理自己的个人账户，包括修改个人信息、上传头像、修改密码等

前置条件：用户已登录

后置条件：无

事件流：

1. 用户进入个人信息界面
2. 用户点击“修改”按钮，在客户端提供的个人信息页面上修改个人信息
3. 用户点击确认按钮
4. 系统检验用户信息合法，保存修改，并重新加载

备选流：

3.1 用户选择退出 -> 系统弹窗是否放弃修改，若是则用例结束，否则返回到2

4.1 系统检验用户修改信息不合法 -> 返回到2

### **<预约座位用例规约>**

概述：乘客（学生、老师）可以在校园巴士有空座的情况下预约巴士的座位

前置条件：乘客已登录

后置条件：无

事件流：

1. 系统进入预约座位界面，用户选择巴士路线
2. 用户选择预约选定巴士的具体班次
3. 用户选择该巴士该班次的具体座位，并点击预约按钮
4. 系统反馈具体预约信息，包括日期、班次、座位号

备选流：

* 1. 用户点击返回 -> 回到Home界面，用例结束
  2. 用户选择的巴士今日班次已全部发出，或因特别原因停班 -> 系统提示“xxx巴士今日已停班”，返回到1
  3. 用户未登录 -> 系统提示“登录后预约”，用例结束，进入登录用例
  4. 如用户选择的班次已发车，则界面不显示；如未发车但已无座，预约按钮灰置
  5. 用户选择返回 -> 返回到1
  6. 如用户选择预约的时间段与其未出行的预约重合 -> 系统提示错误，返回到2

3.1 用户打开一个未完成的预约（预约信息入口） -> 根据上次退出前情况，进入3

3.2 用户选择取消 -> 用例结束

3.3 用户选择保存 -> 保存预约信息，可在预约信息中查看，进入3.1

4.1 因系统错误预约失败（同时预约或者断网） -> 重新执行4或返回到3，并提示用户重新点击预约

特殊说明：

1. 设计原则上，校园巴士APP需要维持目前老师“一定能坐上车”的特权地位，并且需要考虑到未安装APP的同学的坐车需求，老师、持有APP并预约的同学、未预约的同学三类人群的优先级地位待定。
2. 鉴于老师的行程大多比较固定，可能实行保留座位制度，即为老师预留若干座位，预约座位数由系统根据历次乘坐该班巴士的老师数量决定。

### **<取消预约用例规约>**

概述：乘客可以在发车之前取消某次预约

前置条件：乘客已完成至少一次预约座位用例，且已登录

后置条件：无

事件流：

1. 用户进入预约记录界面
2. 用户点击一次“未出行”的预约，并点击“取消预约”按钮
3. 系统提示取消预约成功

备选流：

* 1. 用户未登录 -> 用例结束，进入登录用例
  2. 用户点击“已出行”的预约 -> 用例结束，用户可查看该预约的详细情况
  3. 用户点击“返回” -> 返回到1
  4. 用户选择取消预约离巴士发车不足半小时 -> 系统提示取消预约失败并说明原因，返回到1

3.1 由于网络错误等，取消预约失败 -> 系统提示取消预约失败，返回到1

其他：

1. 仿照高铁售票规则，学生取消预约应当有时间限制，比如发车前半小时之内无法再取消预约

### **<查询预约记录用例规约>**

概述：用户在预约后可以查询自己的预约记录

前置条件：用户已登录

后置条件：无

事件流：

1. 用户进入个人信息界面
2. 用户点击预约记录，查看已出行和未出行的预约记录

备选流：

2.1 用户未登录 -> 用例结束，进入登录用例

### **<查询巴士实时位置用例规约>**

概述：所有用户都可以查询校车目前的实时位置，包括站点实时位置和地图实时位置

前置条件：无

后置条件：无

事件流：

1. 用户进入实时位置查询界面，选择查询的巴士，即查询的方式（实时地图查询、实时站点查询）
2. 用户进入实时地图位置查询
3. 用户进入实时站点位置查询

备选流：

* 1. 用户点击地图上的具体站点 -> 提示站点名，途经时间（已过的标灰），上一站以及下一站，返回到2
  2. 用户点击地图上的巴士 -> 提示巴士发车时间，方向，司机姓名及联系方式，返回到2

3.1 用户点击具体站点 -> 提示到达该站点的下一班车还要多久，返回到3

其他：

1. 查询巴士实时位置（地图及站点）是项目的最大难点之一

### **<查询巴士班次信息用例规约>**

概述：所有用户都可以查询巴士的时刻表信息

前置条件：无

后置条件：无

事件流：

1. 用户进入巴士时刻表查询界面
2. 用户选择需要查询的巴士的类型和方向
3. 用户点击查询巴士的时刻表

备选流：

2.1 用户选择取消 -> 返回到1

3.1 用户选择取消 -> 返回到1

### **<查询巴士线路信息用例规约>**

概述：所有用户都可以查询巴士的线路图信息

前置条件：无

后置条件：无

事件流：

1. 用户进入巴士线路查询界面

1. 用户选择需要查询的巴士的类型和方向
2. 用户点击查询巴士的线路图

备选流：

2.1 用户选择取消 -> 返回到1

3.1 用户选择取消 -> 返回到1

### **<管理预约信息用例规约>**

概述：管理员可以对本巴士的预约信息进行管理

前置条件：管理员已登录

后置条件：无

事件流：

1. 管理员点击管理预约信息，进入界面
2. 管理员选择自己的某一班次查看具体预约信息，包括预约人的姓氏和手机号

管理员可以选择编辑公告，完成后点击“发布”

备选流：

如果管理员未登录 -> 用例结束，进入登录用例

### **<发布系统公告用例规约>**

概述：管理员司机都可以发布系统公告，用户可以在消息界面接收到

前置条件：管理员/司机已登录

后置条件：无

事件流：

1. 管理员/司机点击“发布公告”，系统显示发布公告界面
2. 管理员/司机点击编辑框，输入公告具体内容
3. 管理员/司机点击“发布”功能，系统刷新界面

备选流：

* 1. 管理员/司机未登录 -> 用例结束，进入登录用例

3.1 管理员/司机选择退出 -> 系统提示是否退出，是则返回到1，否则返回到3

### **<管理用户信息>**

概述：管理员可以查看用户信息

前置条件：管理员已登录

后置条件：无

事件流：

1. 管理员点击“用户管理”
2. 页面显示用户列表
3. 管理员点击某个用户查看完整信息

备选流：

3.1 管理员返回上级界面 -> 返回到2

其他：

### **<管理巴士信息>**

概述：管理员可以管理巴士信息

前置条件：管理员已登录

后置条件：无

事件流：

1. 管理员点击“巴士信息管理”

2. 页面显示巴士班次列表

3. 管理员点击某个巴士班次查看完整信息

备选流：

* 1. 管理员返回上级界面 -> 返回到2

### **<管理巴士线路信息>**

概述：管理员可以管理巴士线路信息

前置条件：管理员已登录

后置条件：无

事件流：

1. 管理员点击“巴士线路管理”

2. 页面显示巴士线路列表

3. 管理员点击某个巴士线路进行修改

4. 管理员提交线路修改

备选流：

* 1. 管理员返回上级界面 -> 返回到2

## 易用性

普通用户和高级用户高效地执行发布信息操作所需的培训时间不超过10分钟

用户界面的风格具有高度的一致性

用户交互实现向导和帮助的70%覆盖

## 可靠性

平均故障间隔时间（MTBF）为7天

平均修复时间（MTTR）为24小时

一年中交大蕉蕉APP正常运行时间至少达到98%

手机客户端的崩溃率不超过3%

每千行代码的错误数目为 4bug/KLOC

## 性能

交大蕉蕉APP对事务的平均响应时间为1s，最长响应时间为4s

交大蕉蕉APP每秒响应100个用户请求

交大蕉蕉APP在内存、磁盘、通信等方面的资源利用情况为比较好

交大蕉蕉APP最大并发用户数为100

## 可支持性（待填写）

### **<Java编码规范>**

**3.5.1.1.命名**

局部变量、静态变量使用驼峰命名法

类首字母大写

包名小写

常量字母均大写，由字母与下划线构成

**3.5.1.2. 类文件**

变量定义, 属性名即可

构造函数,

重写的函数,

自己的函数

 getX…(), setX…(),

initXXX(), isXXX(), checkXXX(), getXXX()

 processXXX(),  displayXXX(), saveXXX(),

resetXXX(), clearXXX(), removeXXX()

**3.5.1.3.注释（重要）**

* 文件注释

 /\*

  XXX

\*/

  写在class头顶，import下面，写清楚这个文件的功能

* 段落注释

  /\*

 XXX------------------------------------------------------------------------

  \*/

* 函数注释

 /\*

  XXX

 \*/

写在每个函数的头顶

* 代码注释

//

### **<Android端编码规范>**

**3.5.2.1. 文件命名**

* Layout文件

\*activity\_模块名\_activity名\*

\*fragment\_模块名\_activity名\_fragment名\*

* 包名和文件夹命名

对于比较简单的文件夹退化成文件文件夹名用小写，文件名每个单词首字母大写

* Java文件

\*\*\*Activity\*\*\* 文件：所有单词的第一个字母大写

\*\*\*Fragment\*\*\* 文件：所有单词的第一个字母大写，所属的Activity名+Fragment名+Fragment

\*\*\*Adapter\*\*\*文件：所有单词的第一个字母大写，要以Adapter结尾以适配的类型开头，比如ListView、Pager、GridView、ExpandableList，中间是这个Adapter的功能或者这个控件所属的 Activity或Fragment

**3.5.2.2. Java内部代码命名**

采用驼峰命名法

**3.5.2.3 Java文件代码结构**

* Activity文件

变量定义

构造函数

（菜单）

onCreate()函数

其它重写函数

onCreate()里的每一个部分拆分成一个函数

自己的工具函数

* Fragment文件

变量定义

传值的函数和构造函数

onCreate()

onCreateView()

其它重写函数

onCreateView()里的每一个部分拆分成一个函数

自己的工具函数

* Adapter文件

 变量定义

 构造函数

重写函数

  自己的工具函数

* 其他

引用第三方库要慎重，避免应用大容量的第三方库，导致客户端包非常大

单元测试（逻辑测试、界面测试）

使用静态变量方式实现界面间共享要慎重

 java代码中不出现中文，最多注释中可以出现中文

activity中在一个View.OnClickListener中处理所有的逻辑

### **<Web后端编码规范>**

**3.5.3.1. 文件命名**

* 包名和文件夹命名

对于比较简单的文件夹退化成文件文件夹名用小写，文件名每个单词首字母大写

* Java文件

\*\*\*Service 文件：所有单词的第一个字母大写

\*\*\*Dao 文件：所有单词的第一个字母大写

\*\*\*Controller文件：所有单词的第一个字母大写，

**3.5.3.2. Java内部代码命名**

采用驼峰命名法

**3.5.3.3 包层次结构结构**

Controller层 - controller包

Service层 - service包

Dao层 - dao 包

Entity实体 - entity包

非实体java类 - model包

帮助类 - utils包

**3.5.3.4.其他**

引用第三方库要慎重，避免应用大容量的第三方库，导致客户端包非常大

单元测试（逻辑测试、界面测试）

java代码中不出现中文，最多注释中可以出现中文

## 设计约束（待填写）

编码语言：Java

流程需求：所有代码都应fork到自己仓库，然后通过Pull Request提交到自己开发分支，不允许直接push到master

开发环境： Intellij IDEA，Android Studio

构建工具：Maven，Gradle

安装环境：android5.0以上版本系统

版本控制工具：Git

远程版本库：Github

数据库系统：MySQL

后端框架：Spring framework

## 联机用户文档和帮助系统需求

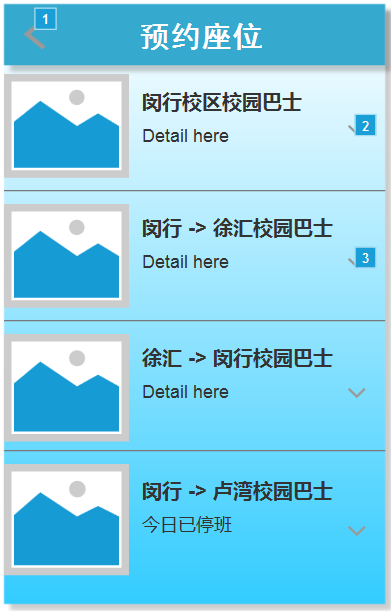
暂无

## 接口

### **用户界面**

### **硬件接口**

暂无

### **软件接口**

用户使用Android5.0以上系统版本的手机客户端图形界面进行操作，管理员使用浏览器的图形界面进行后台管理

### **通信接口**

传输层协议为TCP

网络层协议为IP

前后端通信使用基于HTTP协议的Restful API

## 适用的标准

