

<校园巴士 APP>

软件需求规约

版本 <2.0>

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

修订历史记录

日期	版本	说明	作者
2018.6.2.	<0.0>	<详细信息>	王一林
2018.6.3	<1.0>		励颖
2018.6.4	<1.1>		姚子航
2018.6.5	<2.0>		王鑫伟

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

目录

1. 简介	4
1.1 目的	4
1.2 背景	4
1.3 定义、首字母缩写词和缩略语	4
1.4 参考资料	4
1.5 概述	5
2. 整体说明	5
3. 具体需求	6
3.1 功能	6
3.1.1 <Use case 图>	6
3.1.2 <注册用例规约>	6
3.1.3 <登录用例规约>	7
3.1.4 <管理个人信息用例规约>	7
3.1.5 <预约座位用例规约>	8
3.1.6 <取消预约用例规约>	9
3.1.7 <查询预约记录用例规约>	9
3.1.8 <查询巴士实时位置用例规约>	10
3.1.9 <查询巴士班次信息用例规约>	10
3.1.10 <查询巴士线路信息用例规约>	11
3.1.11 <管理预约信息用例规约>	11
3.1.12 <发布系统公告用例规约>	12
3.1.13 <管理用户信息>	12
3.1.14 <管理巴士信息>	12
3.2 易用性	13
3.2.1 <易用性需求一>	13
3.3 可靠性	13
3.3.1 <可靠性需求一>	13
3.4 性能	14
3.4.1 <性能需求一>	14
3.5 可支持性	14
3.5.1 <Java 编码规范>	14
3.5.2 <Android 端编码规范>	15
3.5.3 <Web 后端编码规范>	17
3.6 设计约束	18
3.6.1 <设计约束一>	18
3.7 联机用户文档和帮助系统需求	19
3.8 接口	19
3.8.1 用户界面	19
3.8.2 硬件接口	22
3.8.3 软件接口	22
3.8.4 通信接口	22
3.9 适用的标准	22

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

软件需求规约 (简化版)

1. 简介

1.1 目的

项目为基于安卓的校园巴士 APP，文档的目的在于提供关于校园巴士 APP 的需求功能的详述以及系统性能参数的说明。

1.2 背景

上海交通大学目前建有徐汇、闵行、黄浦等 6 个校区，基于不同校区间师生互相交流的切实需求，以及部分校区本身占地面积较大的客观现实，先后推出“校区往返校园巴士”和“校内环线校园巴士”供师生乘坐。

但随之推广，校园巴士也暴露出了一些问题。比如，由于校区往返校园巴士是优先满足教师和博士研究生的，所以常常会发生本科生排了二十分钟的队，最终因为巴士坐满而无法上车，只能另寻其他交通方式前往徐汇校区。对于校内往返校园巴士，同学则常常会因为不知道站点所设的地方以及班次时间，发生特地走到某个站点发现刚刚开走一班车，而下一班又要等许久的情况，这样反而会给我们造成困扰。

因此，鉴于校园巴士对于交大学生的重要意义，以及考虑到它目前的种种弊端，我们希望开发一款实用、易用的校园巴士 APP，克服校园巴士现有的使用问题，以满足学生对校园巴士更舒适体验的需求

1.3 定义、首字母缩写词和缩略语

暂无

1.4 参考资料

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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

1.5 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。

2. 整体说明

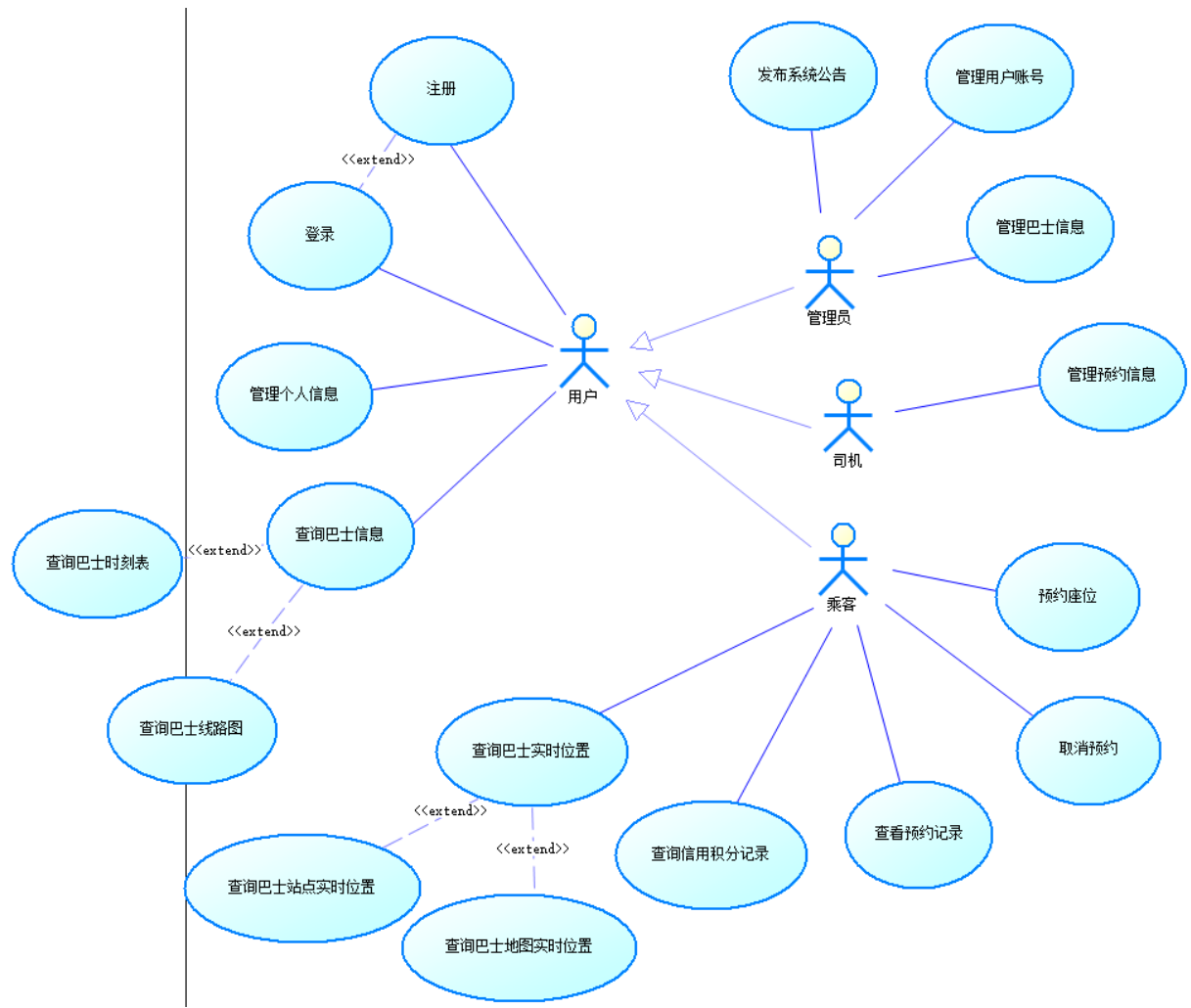
校园巴士 APP 产品以达成上海交通大学内与校园巴士相关的最好的网站/APP 为目标，支持校园巴士的预约、实时位置查询、线路图/时刻表查询等功能，后续可能会朝自定义线路等方向扩展功能；校园巴士 APP 针对的用户群体广泛且稳定，主要为在上海交通大学的在校师生，主要为闵行、徐汇两大校区，它的开发紧紧依赖于学生对更好的校车体验的需求，并以 SJTU 项目组的技术能力和开发经验为约束瓶颈；校园巴士 APP 开发时间预计为 2 个月，采用敏捷开发的 Scrum 软件过程，项目组为 4 人。

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

3. 具体需求

3.1 功能

3.1.1 <Use case 图>



3.1.2 <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

后置条件：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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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：

3.1 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

3.2 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。

3.1.3 <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：

- 4.1 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
- 4.2 用户名与密码、身份不匹配 -> 系统提示信息错误，返回到 3
- 4.3 验证码输入不匹配 -> 系统提示“验证码错误”，返回到 3

3.1.4 <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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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

3.1.5 <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：

- 1.1 用户点击返回 -> 回到 Home 界面，用例结束
- 1.2 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
- 2.1. 用户未登录 -> 系统提示“登录后预约”，用例结束，进入登录用例
- 2.2. 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
- 2.3. 用户选择返回 -> 返回到 1
- 2.4. 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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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。

3.1.6 <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：

- 1.1 用户未登录 -> 用例结束，进入登录用例
- 1.2 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
- 2.1 用户点击“返回” -> 返回到 1
- 2.2 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

3.1.7 <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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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

3.1.8 <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：

- 2.1 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
- 2.2 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

3.1.9 <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：

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

2.1 用户选择取消 -> 返回到 1

3.1 用户选择取消 -> 返回到 1

3.1.10 <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
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

3.1.11 <管理预约信息用例规约>

概述：司机可以对本巴士的预约信息进行管理

前置条件：司机已登录

后置条件：无

事件流：

1. 司机点击管理预约信息，进入界面
2. 司机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

司机可以选择编辑公告，完成后点击“发布”

备选流：

如果司机未登录 -> 用例结束，进入登录用例

3.1 司机未完成公告编辑或未保存 -> 返回到 1

特殊说明：

1. 司机可以设置座位是否可被预约（比如座椅损坏，老师长期预约专座等）
2. 界面有公告栏，司机可以编辑公告栏，预约该巴士该班次的乘客能够看到司机写的公告

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

3.1.12 <发布系统公告用例规约>

概述：管理员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：管理员已登录

后置条件：无

事件流：

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

备选流：

- 1.1 管理员未登录 -> 用例结束，进入登录用例
- 3.1 管理员选择退出 -> 系统提示是否退出，是则返回到 1，否则返回到 3

3.1.13 <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：

3.1.14 <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”

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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：

- 3.1 管理员返回上级界面 -> 返回到 2

3.2 易用性

3.2.1 <易用性需求一>

普通用户和高级用户高效地执行预约操作所需的培训时间不超过 10 分钟

普通用户在确定乘坐班次后可以在 5 分钟之内完成预约

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

3.3 可靠性

3.3.1 <可靠性需求一>

平均故障间隔时间（MTBF）为 7 天

平均修复时间（MTTR）为 24 小时

GPS 定位及巴士实时位置显示的精确为 5m

一年中校园巴士 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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

3.4 性能

3.4.1 <性能需求一>

校园巴士 APP 对事务的平均响应时间为 1s，最长响应时间为 4s

校园巴士 APP 每秒响应 100 个用户请求

校园巴士 APP 在内存、磁盘、通信等方面的资源利用情况为比较好

3.5 可支持性

3.5.1 <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()

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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

```
//
```

3.5.2 <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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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()

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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

3.5.3 <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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

***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

3.6 设计约束

3.6.1 <设计约束一>

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

<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

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

3.7 联机用户文档和帮助系统需求

暂无

3.8 接口

3.8.1 用户界面



<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>



<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>



<项目名称>	Version: <2.0>
软件需求规约	Date: <2018.06.05>

3.8.2 硬件接口

暂无

3.8.3 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

3.8.4 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

3.9 适用的标准

标准名称	标准号
软件工程术语	GB/T 11457-89
软件工程标准分类法	GB/T 15538-95
软件开发规范	GB/T 8566-88
软件项目管理	ISO 10006
软件工程产品质量	ISO/IEC 9126
软件维护指南	GB/T 14079-93
软件可靠性和可维护性管理	GB/T 14394-93