|  |
| --- |
| **校园C2C平台《Fetcher》**  **项目概要设计**  **V1.0.0** |

|  |
| --- |
| **Fetcher项目组** |

**版 本 历 史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本/状态 | 作者 | 日期 | 备注 |
| 1.0.0 | 《Fetcher》项目组 | 2018-08-22 | 创建 |
| 1.0.0 | 《Fetcher》项目组 | 2018-08-23 | 添加了部分插图 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**目 录**

[第一部分 引言 6](#_Toc25457)

[一、编写目的 6](#_Toc6977)

[二、读者对象 6](#_Toc20182)

[三、术语与缩写解释 6](#_Toc1021)

[1、C2C 6](#_Toc4615)

[2、Fetcher 6](#_Toc18396)

[3、Bigbrother 7](#_Toc3383)

[4、DB服务器 7](#_Toc13447)

[5、APP服务器 7](#_Toc5282)

[四、参考资料 8](#_Toc32157)

[第二部分 项目概述 9](#_Toc17527)

[一、项目描述 9](#_Toc20851)

[二、项目功能描述 9](#_Toc4373)

[1、账户登录 10](#_Toc15584)

[2、账户注册 10](#_Toc24092)

[3、 默认登录 10](#_Toc7813)

[4、发布带哥意向 10](#_Toc31523)

[5、选择匹配的大师兄意向并生成订单 11](#_Toc11563)

[6、开始订单 11](#_Toc6938)

[7、订单结算（乙方） 11](#_Toc29226)

[8、评价大师兄 12](#_Toc17298)

[9、发布大师兄意向 12](#_Toc24571)

[10、生成订单或取消意向 12](#_Toc23438)

[11、带哥开始订单后查询订单现状 13](#_Toc11871)

[12、订单结算（甲方） 13](#_Toc32130)

[13、评价带哥 13](#_Toc12484)

[第三部分 设计约束 14](#_Toc3758)

[一、需求约束 14](#_Toc16591)

[1、本系统应当遵循的技术标准 14](#_Toc28157)

[2、软、硬件环境标准 14](#_Toc26163)

[3、接口/协议标准 14](#_Toc24320)

[4、用户界面标准 14](#_Toc28219)

[5、软件质量 14](#_Toc14799)

[二、隐含约束 15](#_Toc30902)

[第四部分 校园C2C平台《Fetcher》方案设计 16](#_Toc28363)

[一、《Fetcher》客户端相关业务模块 17](#_Toc27935)

[1、帐户认证模块 17](#_Toc9624)

[1.1、帐户登录 17](#_Toc8766)

[1.2、账户注册 17](#_Toc14540)

[1.3、默认登录 18](#_Toc5011)

[2、业务逻辑模块 18](#_Toc7199)

[2.1、发布乙方意向（乙方选择发布意向） 19](#_Toc6980)

[2.2、选择匹配的甲方意向并生成订单 19](#_Toc26921)

[2.3、开始订单（乙方选择开始订单） 19](#_Toc7279)

[2.4、订单结算（乙方或甲方选择结束订单） 20](#_Toc28393)

[2.5、评价甲方 20](#_Toc10506)

[2.6、发布甲方意向（甲方选择发布意向） 22](#_Toc17067)

[2.7、生成订单或取消意向 22](#_Toc18354)

[2.8、乙方开始订单后查询订单现状 22](#_Toc1291)

[2.9、订单结算（甲方或乙方选择结束订单） 22](#_Toc24867)

[2.10、评价乙方 22](#_Toc25639)

[3、UI模块 22](#_Toc23077)

[4、通信模块 23](#_Toc27708)

[5、数据库模块 23](#_Toc17973)

[二、《Fetcher》服务端相关业务模块 23](#_Toc573)

[1、数据处理模块 23](#_Toc16443)

[1.1、服务器初始化 23](#_Toc46)

[1.2、请求判断 23](#_Toc8702)

[1.3、身份验证 23](#_Toc26949)

[1.4、返回商品信息 23](#_Toc22588)

[1.5、意向处理 24](#_Toc16533)

[1.6、意向匹配 24](#_Toc23431)

[1.7、交易结算 24](#_Toc10146)

[2、通信模块 25](#_Toc2731)

[2.1、发送消息 25](#_Toc28914)

[2.2、接收消息 25](#_Toc8106)

[3、数据库模块 25](#_Toc32708)

[3.1、操作交易记录数据 25](#_Toc10212)

[3.2、操作用户数据 25](#_Toc28337)

[3.3、操作带哥意向数据 25](#_Toc21360)

[3.4、操作大师兄意向数据 25](#_Toc21937)

[3.5、操作带哥数据 25](#_Toc27044)

[3.6、操作大师兄数据 26](#_Toc6419)

[3.7、操作商品数据 26](#_Toc22176)

[第五部分 E-R实体设计 27](#_Toc20831)

[一、E-R实体结构图 27](#_Toc16930)

[二、实体描述 27](#_Toc6982)

[1、accountInfo实体描述 27](#_Toc25155)

[2、userInfo实体描述 28](#_Toc22533)

[3、itemInfo实体描述 28](#_Toc4292)

[4、itemExInfo实体描述 28](#_Toc27110)

[5、wantedInfo实体描述 28](#_Toc8210)

[6、wantInfo实体描述 29](#_Toc5573)

[7、tradeRecordInfo实体描述 29](#_Toc5428)

[8、fetcherInfo实体描述 30](#_Toc14787)

[9、bbInfo实体描述 30](#_Toc22064)

[第六部分 总体设计 31](#_Toc19086)

[一、校园C2C平台《Fetcher》逻辑架构设计 31](#_Toc6326)

[1、成熟的C/S架构 31](#_Toc21477)

[2、专业的Apache MINA服务端框架 32](#_Toc31509)

[3、基于React-Native框架的App业务平台 32](#_Toc24129)

[二、物理架构设计 32](#_Toc16825)

[1）服务端 32](#_Toc27341)

[2）客户端 32](#_Toc17866)

[三、技术架构运用 33](#_Toc18)

[1、React-Native框架 33](#_Toc17835)

[第七部分 用户界面设计 34](#_Toc21758)

[一、界面风格 34](#_Toc3676)

[第八部分 运行环境和部署 37](#_Toc21810)

[一、运行环境 37](#_Toc27922)

[1、服务器环境 37](#_Toc20668)

[2、客户机器环境 37](#_Toc4747)

[3、开发环境要求 37](#_Toc31403)

[二、系统性能要求 37](#_Toc7641)

第一部分 引言

### 一、编写目的

编写本说明书的目的是为了准确阐述项目具体业务需求和需求边界，本说明书的作者是校园C2C平台《Fetcher》项目组，本说明书的确认者是项目组长，本说明书的读者是项目所有直接干系人。

本说明书是指导项目实施的重要指导性文件，也是用户最后进行验收（终验）的依据，说明书中内容一旦确认双方将以此为基础开展工作。如果需要变更说明书内容，必须走变更流程，变更必须得到甲乙双方书面确认，最后变更内容将作为本文的一部分，在项目实施过程中得以体现。

### 二、读者对象

该文档的读者为用户代表、软件分析人员、开发管理人员和测试人员。

### 三、术语与缩写解释

#### 1、C2C

**C2C**是Customer（Consume）to Customer（Consumer）的缩写，即消费者个人间的电子商务行为。比如一个消费者有一台电脑，通过网络进行[交易](https://baike.baidu.com/item/%E4%BA%A4%E6%98%93/32757" \t "https://baike.baidu.com/item/c2c/_blank)，把它出售给另外一个消费者，此种交易类型就称为C2C电子商务。C2C领域现已形成了四足鼎立之势：[淘宝](https://baike.baidu.com/item/%E6%B7%98%E5%AE%9D" \t "https://baike.baidu.com/item/_blank)、[易趣](https://baike.baidu.com/item/%E6%98%93%E8%B6%A3" \t "https://baike.baidu.com/item/_blank)、[拍拍](https://baike.baidu.com/item/%E6%8B%8D%E6%8B%8D" \t "https://baike.baidu.com/item/_blank)、[有啊](https://baike.baidu.com/item/%E6%9C%89%E5%95%8A" \t "https://baike.baidu.com/item/_blank)。其四大标准分别为：Website Aesthetics （网页审美）、Usability （可用性）、Content （内容为王）、SEO （搜索引擎优化）。

#### 2、Fetcher

**Fetcher**字面上的意思是取物的人，而在我们校园C2C平台“Fetcher”上则是作为乙方——服务提供方，我们称之为“带哥”。这个平台作为中介平台、公共服务平台的核心交易物就是：以校园内某地为终点的移动人员的空余装载量，而Fetcher可在该平台进行提交服务项目、捕获服务订单等操作以完成Bigbrother发布的订单。

#### 3、Bigbrother

**Bigbrother**字面上的意思是大师兄，而在我们校园C2C平台“Bigbrother”上则是作为甲方——服务需求方，我们称之为“大师兄”。Bigbrother可在该平台进行选择服务类别、制定服务细节等操作以发布订单，等待Fetcher的服务。

（为帮助理解在本说明文档中的余下部分，以“乙方”代指Fetcher，“甲方”代指Bigbrother）

#### 4、DB服务器

**DB服务器**是Database服务器的缩写，即数据库服务器。数据库服务器由运行在局域网中的一台/多台计算机和数据库管理系统软件共同构成，数据库服务器为客户应用程序提供数据服务。数据库服务器建立在数据库系统基础上，具有数据库系统的特性，且有其独特的—面。主要功能如下：

1）数据库管理功能，包括系统配置与管理、数据存取与更新管理、数据完整性管理和数据安全性管理。

2）数据库的查询和操纵功能 ，该功能包括数据库检索和修改。

3）数据库维护功能，包括数据导入/导出管理，数据库结构维护、数据恢复功能和性能监测。

4）数据库并行运行，由于在同一时间，访问数据库的用户不止一个，所以数据库服务器必须支持并行运行机制，处理多个事件的同时发生。

**5、APP服务器**

**APP服务器**（APP Server）是一个让用户接收从一个远端网络[存取服务器](https://baike.baidu.com/item/%E5%AD%98%E5%8F%96%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/APP%E6%9C%8D%E5%8A%A1%E5%99%A8/_blank)上的象征密，其促使一个用户码改变的工具。App Server可以自动管理并发性、事务、对象分布、[负载均衡](https://baike.baidu.com/item/%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1" \t "https://baike.baidu.com/item/APP%E6%9C%8D%E5%8A%A1%E5%99%A8/_blank)、安全性和资源管理等等系统级功能。简单的来说就是App Server是管理服务端组件的，它给服务端组件提供了一个全功能可靠的运行环境。

### 四、参考资料

《系统需求分析》

第二部分 项目概述

### 一、项目描述

现实生活中，物理距离常常会限制了人们饮食类的购物欲望，尤其是在当代校园中，我们身边总会听到这样的需求——客户希望不用自己跑腿就能通过其他人的服务以获得相应的需求物品，但实际上，一方面客户可利用的资源非常有限，大部分被仅限于身边的舍友或同学，而且前提是对方也有相同的需求或意愿，这又大大降低了用户获得此项服务的可能性；其次，用户在不断询问搜索提供该服务的人群的过程中，不断地消耗客户的时间和精力。另一方面，随着网购已成为生活常态，快递数量日益增多，这个物理距离上存在足够的交通流量提供足够低价的代购跑腿服务。另外，已有部分高校正在开发相应的APP以解决上述类似问题。

在这个大环境下，校园C2C平台“Fetcher”应运而生。《Fetcher》作为一个公共服务平台，可以为大师兄（甲方）提供代取物件的服务，为乙方提供派件的订单以获得创收，解决了上述资源有限、效率低下、成功率低等一系列问题，为校园生活提供极大的便利。应用场景中派送终点可以为教学楼+宿舍+院楼+体育馆（实际上只要交通流量满足需求，就可以全面覆盖校园），其次内容可以拓展到食堂菜品+超市商品+文具书本+校外奶茶/肯德基（或校门代取）。最后，我们可以再整合其他的C2C服务到平台上，例如代取快递，代运物品等。总之这个平台的核心交易物就是：以校园内某地为终点的移动人员的空余装载量。

### 二、项目功能描述

用户业务可根据对象的区别分为带哥业务和大师兄业务。

带哥和大师兄均包含账户认证业务：账户注册、默认登录、账户登录。

带哥业务包含：发布带哥意向、选择匹配的大师兄意向并生成订单、开始订单、订单结算、评价大师兄。

大师兄业务包含：发布大师兄意向、生成订单或取消意向、带哥开始订单后查询订单现状、订单结算、评价带哥。

#### 1、账户登录

账户登陆用于甲、乙方用户登录账户以进入平台执行操作。

该功能流程步骤为：

1. 判断本地是否存在账户信息
2. 存在账户信息调用“默认登录”，不存在调用“账户注册”
3. 进入登录界面
4. 用户输入用户名、密码
5. 系统上传用户名、密码，服务器匹配
6. 匹配成功进入系统，匹配失败提示账户不存在

#### 2、账户注册

账户注册用于为新用户在该平台注册新账号，新的账号信息将写入系统数据库，新用户注册后方可使用APP提供的功能。

该功能流程步骤为：

1）用户输入手机号、用户名、密码

2）系统上传用户手机号、用户名、密码给服务器，服务器保存信息并分配唯一用户ID

3）账户注册完成，在本地保存账户信息

#### 默认登录

默认登录用于已注册用户的直接登录，新用户无法使用该功能。

该功能流程步骤为：

1）系统上传账户信息与服务器账户表进行匹配

2）匹配成功进入系统，匹配失败提示账户不存在需要重新注册账户

#### 4、发布带哥意向

发布带哥意向用于乙方发布派单意向至系统，以匹配相应的甲方需求。

该功能流程步骤为：

1）选择目的地区域（教学楼、图书馆、宿舍）

2）选择详细目的地、到达时间、特殊要求

3）系统上传带哥意向信息给服务器，服务器查询大师兄意向，有匹配意向则把意向返回系统调用“选择匹配的大师兄意向并生成订单”，无匹配意向则返回无给系统系统提示挂起或取消

#### 5、选择匹配的大师兄意向并生成订单

选择匹配的大师兄意向并生成订单用于乙方发布意向后系统智能匹配符合乙方条件且合适的甲方订单，乙方可选择一个或多个订单，选择确认后即可生成订单。

该功能流程步骤为：

1）系统接收服务器返回的大师兄意向

2）带哥勾选大师兄意向并确认

3）系统生成订单，向服务器上传订单信息，服务器保存订单信息并更新带哥和大师兄意向状态

#### 6、开始订单

开始订单用于乙方选择开始订单。

该功能流程步骤为：

1）展示需购买的物品和大师兄地点

2）系统上传带哥位置

#### 7、订单结算（乙方）

订单结算用于乙方或甲方选择结束订单，为避免不必要的纠纷，平台采用面对面交易——乙当面出示支付宝二维码供甲方扫码付款。

该功能流程步骤为：

1）进入订单结算界面

2）系统上传订单信息，服务器更新订单状态，完成订单

#### 8、评价大师兄

评价大师兄用于乙方对甲方的订单进行评价以供其他用户或平台参考，为保证乙方隐私，评价信息将以匿名的形式延迟或乱序地发送给甲方。

该功能流程步骤为：

1）离开订单结算界面，进入评价界面

2）选择评价标签

3）输入更多评价

4）上传评价，服务器更新大师兄信息

#### 9、发布大师兄意向

发布大师兄意向用于甲方发布订单意向，甲方将向系统提供服务的必要信息以便系统匹配合适的乙方以接收订单请求。

该功能流程步骤为：

1）选择目的地、时间

2）选择物品和数量

3）系统上传大师兄意向并挂起

#### 10、生成订单或取消意向

生成订单或取消意向用于向展示甲方显示订单信息。

该功能流程步骤为：

1）系统接收服务器信息，意向被更新已生成订单

2）展示订单信息和带哥信息

3）意向结束前半小时还未生成订单

4）提示大师兄无人接单，选择继续等待或取消意向

5）取消意向或意向时间结束，系统上传取消信息，服务器更新大师兄意向状态

#### 11、带哥开始订单后查询订单现状

带哥开始订单后查询订单现状用于甲方实时查看订单信息和乙方信息以保证服务的安全性。

该功能流程步骤为：

1）展示订单信息和带哥信息

2）接收带哥位置并展示

#### 12、订单结算（甲方）

该功能流程步骤为：

1）进入订单结算界面

2）系统上传订单信息，服务器更新订单状态，完成订单

#### 13、评价带哥

评价大师兄用于甲方对乙方的订单完成服务进行评价以供其他用户或平台参考，为保证甲方隐私，评价信息将以匿名的形式延迟或乱序地发送给乙方。

该功能流程步骤为：

1）离开订单结算界面，进入评价界面

2）选择评价标签

3）输入更多评价

4）上传评价，服务器更新带哥信息

第三部分 设计约束

### 一、需求约束

#### 1、本系统应当遵循的技术标准

数据命名的规则遵循Java命名惯用格式。

#### 2、软、硬件环境标准

本系统采用C/S架构。

客户端采用React-Native框架，数据库部分使用sqlite。部署于Android 7.0及以上，IOS 10及以上。

服务端采用Apache MINA框架，数据库部分使用MySQL。部署于Ubuntu Server 18.04.1 LTS。

#### 3、接口/协议标准

本系统通过自定协议实现客户端和服务端器之间的数据通信。

#### 4、用户界面标准

使用React-Native相关三方类库，并进行界面全部美工优化。

#### 5、软件质量

1）正确性

系统必须交易能够被正确处理；

2）健壮性

系统应能够7\*24小时无故障运行；

3）效率性

系统可以支持100个终端同时发起业务，处理业务的时间不超过10秒钟；

4）易用性

界面应采用图形化操作方式，便于业务人员操作；

5）安全性

报文中的关键数据域以密文的方式传输；

6）可扩展性

应该充分考虑到将来交易的修改或增加，避免需求变更时大规模修改程序。

7）网络体系结构

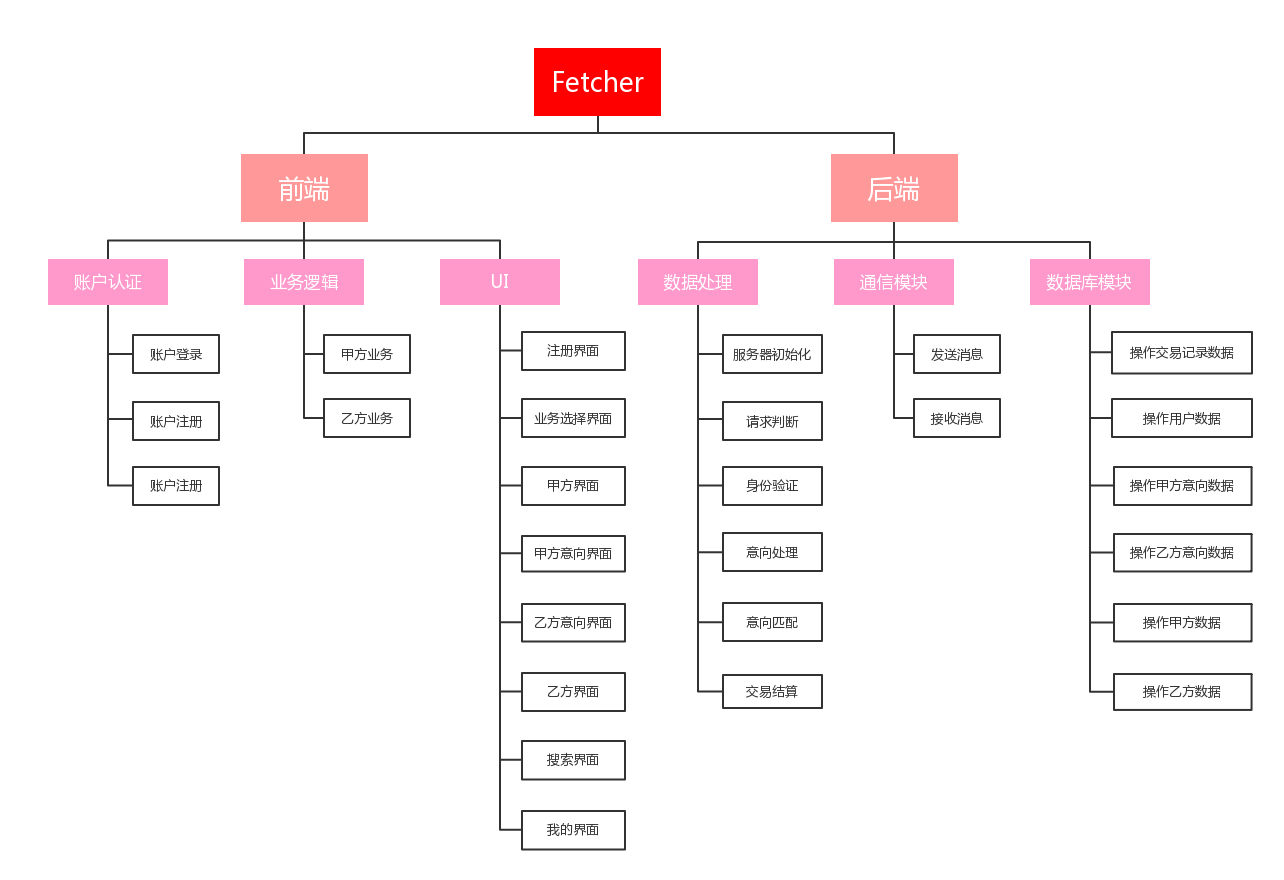
### 网络部署图二、隐含约束

1）用户具有基本的手机知识，项目提供的App界面能被用户几近完全理解；

2）用户了解并遵循App所搭建平台的参与规则。

3）软件可以流畅运行在Android 7.0及以上、IOS 10及以上的设备中；

4）应该把有可能变动的参数存放到配置文件或数据库中，保证修改参数的灵活性。

第四部分 校园C2C平台《Fetcher》方案设计

### 一、《Fetcher》客户端相关业务模块

#### UML时序图1、帐户认证模块

**1.1、帐户登录**

1. 判断本地是否存在账户信息
2. 存在账户信息调用“默认登录”，不存在调用“账户注册”
3. 进入登录界面
4. 用户输入用户名、密码
5. 系统上传用户名、密码，服务器匹配
6. 匹配成功进入系统，匹配失败提示账户不存在

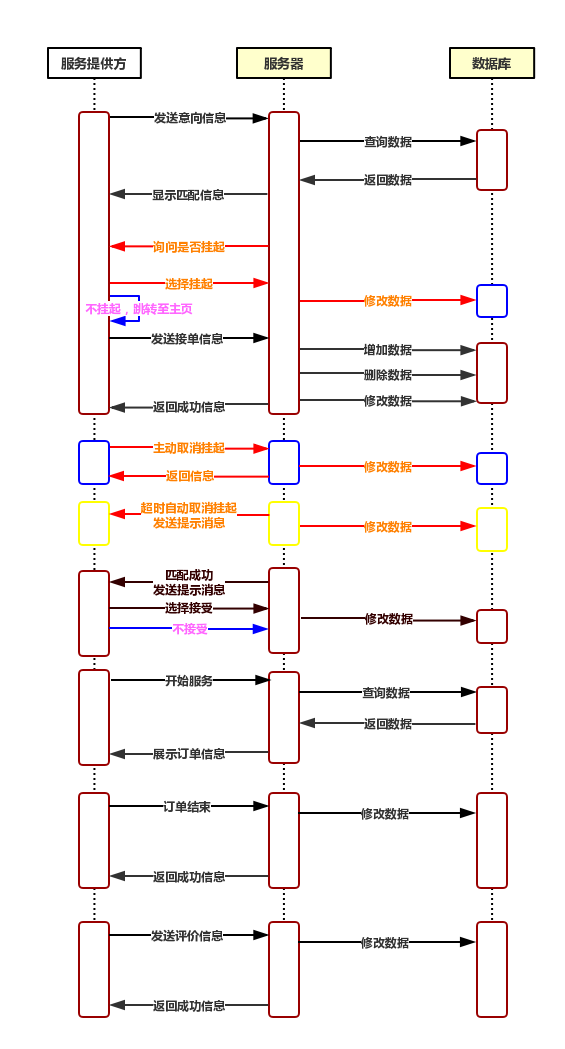
**1.2、账户注册**

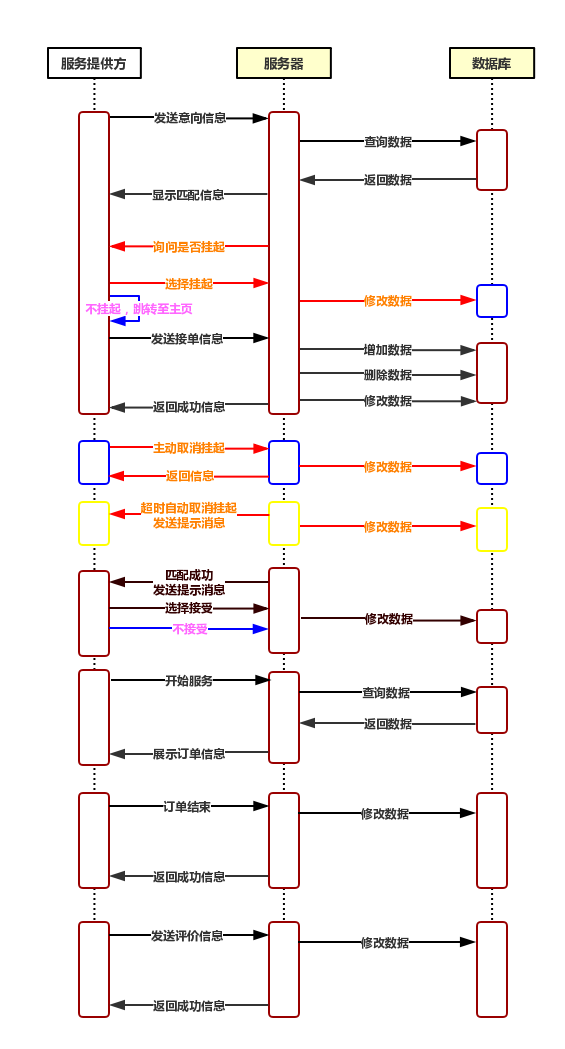
1. 用户输入手机号、用户名、密码
2. 系统上传用户手机号、用户名、密码给服务器，服务器保存信息并分配唯一用户ID
3. 账户注册完成，在本地保存账户信息

**1.3、默认登录**

1. 系统上传账户信息与服务器账户表进行匹配
2. 匹配成功进入系统，匹配失败提示账户不存在需要重新注册账户

#### 2、业务逻辑模块

乙方业务：发布乙方意向、选择匹配的甲方意向并生成订单、开始订单、订单结算、评价甲方

甲方业务：发布甲方意向、生成订单或取消意向、乙方开始订单后查询订单现状、订单结算、评价乙方

**2.1、发布乙方意向（乙方选择发布意向）**

1. 选择目的地区域（教学楼、图书馆、宿舍）
2. 选择详细目的地、到达时间、特殊要求
3. 系统上传乙方意向信息给服务器，服务器查询甲方意向，有匹配意向则把意向返回系统调用“选择匹配的甲方意向并生成订单”，无匹配意向则返回无给系统系统提示挂起或取消

**2.2、选择匹配的甲方意向并生成订单**

1. 系统接收服务器返回的甲方意向
2. 乙方勾选甲方意向并确认
3. 系统生成订单，向服务器上传订单信息，服务器保存订单信息并更新乙方和甲方意向状态

**2.3、开始订单（乙方选择开始订单）**

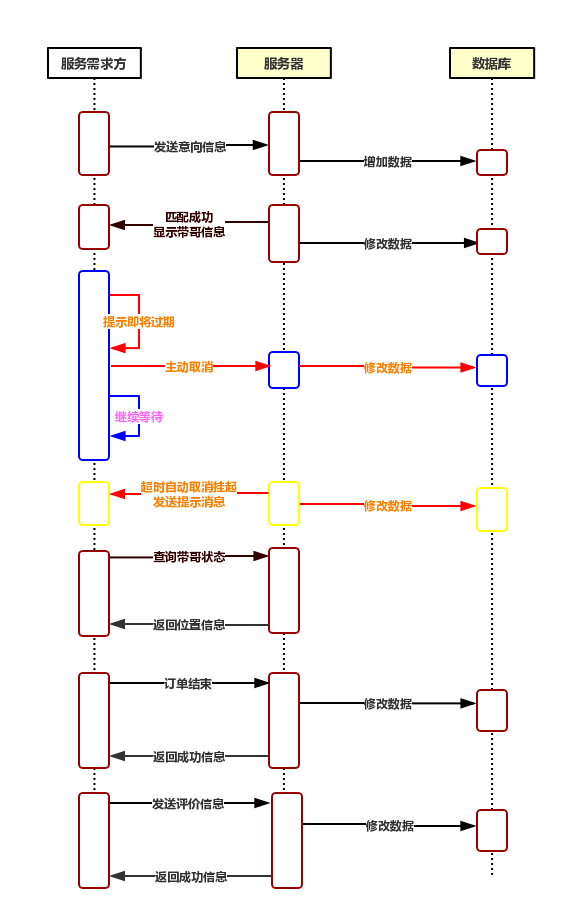
1. 展示需购买的物品和甲方地点
2. 系统上传乙方位置

**2.4、订单结算（乙方或甲方选择结束订单）**

1. 进入订单结算界面
2. 系统上传订单信息，服务器更新订单状态，完成订单

**2.5、评价甲方**

1. 离开订单结算界面，进入评价界面
2. 选择评价标签
3. 输入更多评价
4. 上传评价，服务器更新甲方信息



**2.6、发布甲方意向（甲方选择发布意向）**

1. 选择目的地、时间
2. 选择物品和数量
3. 系统上传甲方意向并挂起

**2.7、生成订单或取消意向**

1. 系统接收服务器信息，意向被更新已生成订单
2. 展示订单信息和乙方信息
3. 意向结束前半小时还未生成订单
4. 提示甲方无人接单，选择继续等待或取消意向
5. 取消意向或意向时间结束，系统上传取消信息，服务器更新甲方意向状态

**2.8、乙方开始订单后查询订单现状**

1. 展示订单信息和乙方信息
2. 接收乙方位置并展示

**2.9、订单结算（甲方或乙方选择结束订单）**

1. 进入订单结算界面
2. 系统上传订单信息，服务器更新订单状态，完成订单

##### 2.10、评价乙方

1、离开订单结算界面，进入评价界面

2、选择评价标签

3、输入更多评价

4、上传评价，服务器更新乙方信息

#### 3、UI模块

注册界面、业务选择界面、乙方界面、甲方界面、乙方意向界面、甲方意向界面、我的界面、搜索界面

#### 4、通信模块

#### 5、数据库模块

### 二、《Fetcher》服务端相关业务模块

#### 1、数据处理模块

##### 1.1、服务器初始化

1、判断是否为首次启动服务器

2、若为首次启动，初始化实体对象

3、向数据库发送请求，创建表单并设置属性默认值

##### 1.2、请求判断

1、解析数据包

2、判断请求类型

##### 1.3、身份验证

1、获取客户端登录请求

2、调用数据库模块操作用户数据功能，查询userInfo表中的UserName与Password

3、匹配成功，则用户登录成功，建立连接

4、获取客户端注册请求

5、调用数据库模块操作用户数据功能，查询UserName

6、若不存在相同UserName，则建立新用户，通过通信模块发送注册成功信息，否则返回失败信息。

##### 1.4、返回商品信息

1、获取客户端加载商品请求

2、调用数据库模块操作用户数据功能，查询ItemInfo表

3、通过通信模块向客户端返回Item信息

##### 1.5、意向处理

1、接收大师兄意向请求

2、将从通信模块中获取的信息通过数据库模块 操作带哥意向数据功能写入wantInfo表中

3、接收带哥意向请求

4、将从通信模块中获取的信息通过数据库模块操作大师兄意向数据功能写入wantedInfo表中

5、接收带哥接单请求

6、通过数据库模块将wantedInfo设为结束，wantInfo设为待完成

7、依据订单里程计算价格，并更新tradeRecordInfo，记录交易信息

##### 1.6、意向匹配

1、获取带哥挂起请求时

2、通过数据库模块将带哥设置为挂起状态

3、有符合条件的大师兄意向请求接入，将该意向信息通过通信模块发送给带哥，取消其挂起状态

##### 1.7、交易结算

1、带哥和大师兄都发出完成请求

2、将wantInfo设为结束状态，更新tradeRecordInfo信息

3、交易双方相互评价，计算平均评价，更新fetcherInfo与BBInfo信息

#### 2、通信模块

##### 2.1、发送消息

依据协议，将数据处理模块中产生的数据打包，通过apache mina提供的通道向目标客户端进行发送。

##### 2.2、接收消息

通过apache mina接收客户端数据包，将其交给数据处理模块进行请求判断。

#### 3、数据库模块

##### 3.1、操作交易记录数据

依据数据处理模块需求，对tradeRecordInfo表进行增删改查操作。

##### 3.2、操作用户数据

依据数据处理模块需求，对userInfo表进行增删改查操作。

##### 3.3、操作带哥意向数据

依据数据处理模块需求，对wantedInfo表进行增删改查操作。

##### 3.4、操作大师兄意向数据

依据数据处理模块需求，对wantInfo表进行增删改查操作。

##### 3.5、操作带哥数据

依据数据处理模块需求，对fetcherInfo表进行增删改查操作。

##### 3.6、操作大师兄数据

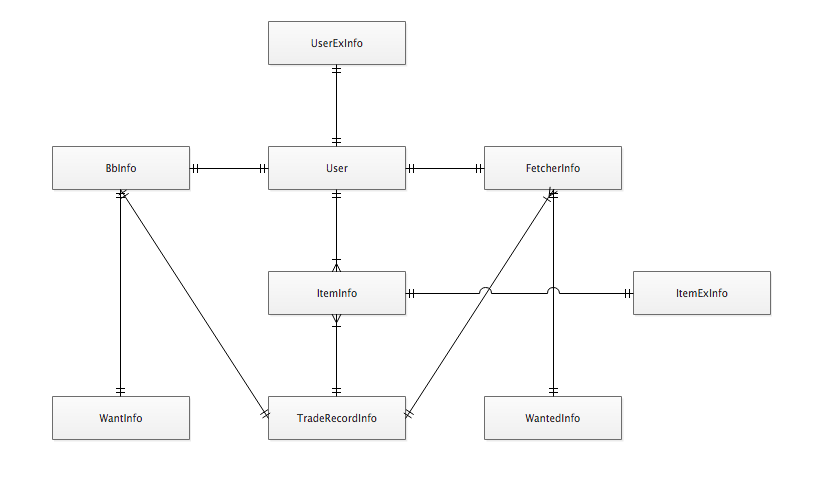
依据数据处理模块需求，对BBInfo表进行增删改查操作。

##### 3.7、操作商品数据

依据数据处理模块需求，对ItemInfo表进行增删改查操作。

第五部分 E-R实体设计

### 一、E-R实体结构图



### 二、实体描述

#### 1、accountInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | UserID | 用户ID | Int |
| 2 | UesrName | 用户名 | Char[] |
| 3 | NickName | 昵称 | Char[] |
| 4 | Psw | 密码 | Char[] |

#### 2、userInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | Gender | 性别 | Boolean |
| 2 | PhoneNumber | 手机号 | Char[] |
| 3 | ExtraInfo | 其他信息 | Char[] |

#### 3、itemInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | ItemID | 货品ID | Int |
| 2 | Name | 货品名称 | Char[] |
| 3 | Weight | 重量(g) | Int |
| 4 | Size | 大小 | Char[] |
| 5 | Price | 价格 | Int |
| 6 | Location | 地点 | Char[] |
| 7 | Type | 类型 | Int |

#### 4、itemExInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | ExtraInfo | 货品其他信息 | Char[] |

#### 5、wantedInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | FetcherID | 乙方ID | Int |
| 2 | WangtedID | 意向ID | Int |
| 3 | StartingPoint | 起点 | Char[] |
| 4 | Destination | 目的地 | Char[] |
| 5 | ArriveTime | 到达时间 | Char[] |
| 6 | AcceptBigSize | 是否结束大件 | Boolean |
| 7 | AcceptSE | 是否提前到达 | Boolean |
| 8 | State | 状态 | Int |
| 9 | Rate | 评价 | Int |

#### 6、wantInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | BBID | 甲方ID | Int |
| 2 | WantID | 意向ID | Int |
| 3 | Destination | 目的地 | Char[] |
| 4 | AddressDetail | 详细地址 | Char[] |
| 5 | ArriveTime | 希望到达时间 | Char[] |
| 6 | AcceptND | 是否接受不上门 | Boolean |
| 7 | Type | 类型 | int |
| 8 | ItemList | 货物列表 | Char[][] |
| 9 | Note | 备注 | Char[] |
| 10 | Image | 图片 | Char[] |
| 11 | Sate | 状态 | Int |
| 12 | Rate | 评价 | Char[] |

#### 7、tradeRecordInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | TradeID | 交易ID | Int |
| 2 | FetcherID | 乙方ID | Int |
| 3 | BbID | 甲方ID | Int |
| 4 | WantedID | 意向ID | Int |
| 5 | WantIDList | 意向ID | Int |
| 6 | StartTime | 开始时间 | Char[] |
| 7 | FinishTime | 结束时间 | Char[] |
| 8 | State | 状态 | Int |

#### 8、fetcherInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | UserID | 用户ID | Int |
| 2 | TotalDistance | 总路程 | Int |
| 3 | TotalTime | 总时间 | Char[] |
| 4 | Counter | 送达次数 | Int |
| 5 | TurnOver | 总钱数 | Int |
| 6 | Rate | 信誉评级 | Int |

#### 9、bbInfo实体描述

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **英文名** | **中文名** | **数据类型** |
| 1 | UserID | 用户ID | Int |
| 2 | TotalDistance | 总路程 | Int |
| 3 | TotalTime | 总时间 | Char[] |
| 4 | Counter | 送达次数 | Int |
| 5 | TurnOver | 总钱数 | Int |
| 6 | Rate | 信誉评级 | Int |

第六部分 总体设计

### 微服务架构图-3一、校园C2C平台《Fetcher》逻辑架构设计

#### 1、成熟的C/S架构

即[Client/Server](https://baike.baidu.com/item/Client/Server" \t "https://baike.baidu.com/item/CS%E6%9E%B6%E6%9E%84/_blank) (客户机/服务器) 结构，是大家熟知的软件[系统体系结构](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E4%BD%93%E7%B3%BB%E7%BB%93%E6%9E%84" \t "https://baike.baidu.com/item/CS%E6%9E%B6%E6%9E%84/_blank)，通过将任务合理分配到Client端和Server端，降低了系统的通讯开销，需要安装客户端才可进行管理操作。

客户端和服务器端的程序不同，用户的程序主要在客户端，服务器端主要提供数据管理、数据共享、数据及系统维护和[并发控制](https://baike.baidu.com/item/%E5%B9%B6%E5%8F%91%E6%8E%A7%E5%88%B6" \t "https://baike.baidu.com/item/CS%E6%9E%B6%E6%9E%84/_blank)等，客户端程序主要完成用户的具体的业务。

开发比较容易，操作简便，且具有安全性强、事务处理逻辑明晰和交互性良好的特点。

#### 2、专业的Apache MINA服务端框架

Apache的MINA（Multipurpose Infrastructure Networked Applications）是一个网络应用框架，可以帮助用户开发高性能和高扩展性的网络应用程序。

它提供了一个抽象的、事件驱动的异步API，使Java NIO在各种传输协议（如TCP/IP，UDP/IP协议等）下快速高效开发。

#### 3、基于React-Native框架的App业务平台

引导用户在移动平台上进行业务处理。通过WIFI/4G等联网技术，便于用户方便、快捷、实时处理和操作业务。

移动客户端适用多种硬件设备，支持Android7.0及以上与iOS10及以上系统，移动端创新的操作体验为用户提供更好的使用体验。

### 网络部署图二、物理架构设计

1）服务端

系统服务端可以部署到自建服务器上，系统管理员通过IE实现业务管理操作。

2）客户端

用户通过应用商店等途径下载App安装包到手机（Android-apk，IOS-ipa），安装运行并操作。

### 三、技术架构运用

#### 1、React-Native框架

[](https://baike.baidu.com/pic/react%20native/20307162/0/d53f8794a4c27d1eceb4a58d10d5ad6edcc438ec?fr=lemma%26ct=single)React Native (简称RN)是Facebook于2015年4月开源的跨平台移动应用开发框架，是Facebook早先开源的JS框架 React 在原生移动应用平台的衍生产物，目前支持iOS和安卓两大平台。RN使用Javascript语言，类似于HTML的JSX，以及CSS来开发移动应用，因此熟悉Web前端开发的技术人员只需很少的学习就可以进入移动应用开发领域。

React Native使你能够在Javascript和React的基础上获得完全一致的开发体验，构建世界一流的原生APP。

React Native着力于提高多平台开发的开发效率：仅需学习一次，编写任何平台。(Learn once, write anywhere)

[Facebook](https://baike.baidu.com/item/Facebook" \t "https://baike.baidu.com/item/react%20native/_blank)已经在多项产品中使用了React Native，并且将持续地投入建设React Native。

React Native主要特性如下：

1）原生的iOS组件：React Native主张“Learn once, write everywhere”而非其他跨平台工具一直宣扬的“Write once, run everywhere”。通过React Native，开发者可以使用UITabBar、UINavigationController等标准的iOS平台组件，让应用界面在其他平台上亦能保持始终如一的外观、风格。

2）异步执行：JavaScript应用代码和原生平台之间所有的操作都采用异步执行模式，原生模块使用额外线程，开发者可以解码主线程图像、后台保存至磁盘、无须顾忌UI等诸多因素直接度量文本设计布局。

3）触摸处理：React Native引入了一个类似于iOS上Responder Chain响应链事件处理机制的响应体系，并基于此为开发者提供了诸如TouchableHighlight等更高级的组件。

第七部分 用户界面设计

### 一、界面风格







第八部分 运行环境和部署

### 一、运行环境

#### 1、服务器环境

1）OS: Ubuntu 16.04 LTS

2）CPU: Intel Xeon X3430 2.4GHz（四核四线程）及以上

3）Mem: 16GB及以上

4）HD: 500GB及以上（含数据存储空间）

5）JDK: Java 1.8

6）Application Server：Apache MINA

#### 2、客户机器环境

1）OS: Android 7.0及以上

IOS 10 及以上

2）Mem: 2GB及以上

3）HD: 32GB及以上

#### 3、开发环境要求

|  |  |  |
| --- | --- | --- |
| 项目 | 名称 | 版本 |
| 开发平台 | Windows 10 | - |
| 开发工具 | Intellij  Webstorm  Genymotion | -  -  - |
| 代码管理工具 | Git | - |
| 开发环境 | Java  Android SDK  React-native | JDK 1.8  API 28  0.55.4 |

### 二、系统性能要求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 项目 | 模块 | 级别 | 技术参数 |
| 1 | 设计实现技术指标 | 系统架构 | A | 采用C/S模式三层架构 |
| 2 | 面向对象开发语言与框架 | A | 采用Java、JavaScript开发语言；采用React-Native开源移动App开发框架 |
| 3 | 注释和文档 | A | 符合CMMI软件开发过程标准文档（至少提供：需求、概要、详细设计、测试报告、部署和环境、用户手册），代码注释量>=30% |
| 4 | 模块化和适合实训 | A | SOA设计、模块化，保证系统各模块单元较强的独立性适合实训教学 |
| 5 | 测试覆盖率 | A | 功能覆盖率>=100%，业务覆盖率>=100%，语言覆盖率>=100%，逻辑覆盖率>=80% |
| 6 | 资源利用率要求 | CPU占用率 | B | <=15%利用率（附近标准配置） |
| 7 | 内存使用率 | B | <=15%利用率（附近标准配置） |
| 8 | 响应时间要求 | 服务器 | B | <=100ms（附近标准配置） |
| 9 | 网络 | B | <=100ms（附近标准配置） |
| 10 | 客户端 | B | <=5s（附近标准配置） |
| 11 | 系统稳定性要求 | 成熟性 | A | 真实的用户，成功使用本系统 |
| 12 | 稳定性 | B | 无故障运行时间>=30天，系统恢复时间<=6小时 |
| 13 | 先进性 | A | 采用目前体验最好、最流行的App混合开发框架React-Native |
| 14 | 创新意义 | A | 案例项目具有创新性 |
| 15 | 通信接口要求 | 服务器端 | A | 支持协议定制，支持安全过滤，支持消息队列，多种模式重发，支持成功检测。消息延迟<=2S，丢包<=0.001% |
| 16 | 客户端 | A | 支持协议定制，支持安全过滤，支持消息队列，多种模式重发，支持成功检测。消息延迟<=2S，丢包<=0.001% |
| 18 | 集成部署环境 | 服务器 | A | Ubuntu Server 18.04.1 LTS |
| 19 | 数据库 | A | MySQL、Sqlite |
| 20 | 手持机 | A | Android 7.0及以上、IOS 10及以上 |
| 说明：级别（A:表示非常重要必须达到的技术性能要求,B:表示重要推荐达到的技术性能要求,C：表示非重要可以弱化的技术性能要求.） | | | | |