川川帮详细设计文档

**1. 引言**

**1.1 背景**

在大学，由于繁忙的学习和课余活动，每天会有很多很多需要别的同学帮助的需求产生。常见的需求比如：拿快递，跑腿打印取文件，又或者是需要出手自己的闲置物品避免浪费，亦或是需要有同学一同跑步、看电影等等。很多时候，我们只能一次又一次拜托同学，欠下很多人情；又或是重新安排自己的时间去专门做这些事情。

还有的同学，课程较少，每天充裕的时间比较多，希望能够做一些相对轻松的事情赚取一些零花钱。

那么这两部分同学之间，缺乏一种沟通的渠道。我们将建立一个能够发布需求，接收需求的平台，让这两部分同学之间有一个良性的沟通渠道，让有需求的同学花较小的代价完成自己的需求；让空闲时间的同学能够利用自己的时间赚取零用钱。

**1.2 目的**

该文档首先给出项目的整体结构和功能结构概貌，试图从总体架构上给出整个系统的轮廓。同时对功能需求、性能需求进行了详细的描述。便于用户、开发人员进行理解和交流，反映出用户问题的结构，可以作为软件开发工作的基础和依据、确认测试和验收的依据以及软件维护的参考资料。

本文档是用于定义项目范围，明确开发需求，并为后期的分析设计、代码实现和测试提供指导。

该文档是为了明确数据结构以及系统结构，说明系统整体的概要设计，为之后系统编码实现提供支持。

**2. 总体设计**

**2.1 系统描述**

该项目名为“川川帮”，基于微信小程序平台进行开发，微信小程序本身依附在微信这个用户量庞大的流量平台中，而微信本身是一个强社交平台，使得小程序在社交分享方面有天然的优势，这也使得可以通过小程序来实现低成本的用户增长。

我们的项目主要聚焦于解决四川大学校内同学可能产生的需求上，经过小组成员分析以及同学填写问卷的形式，我们总结出了三种急需解决的需求：1.闲置物品交易；2.校内需求解决；3.活动安排（如找寻同伴看电影，参加各种活动等）

川川帮是一个依托于微信平台的小程序项目，对应不同类型的需求进行分类展示，使得用户界面更加纯粹，增强软件易用性，利用前后端分离的方式，是的开发更加高效。我们希望通过“川川帮”构建一个校内同学需求解决的渠道，双方各取所需，实现共赢。

**2.2 项目功能**

该项目以发布需求任务，接取需求任务两者为核心功能，还包括对需求的搜索，发布者可对该需求任务完成度进行评价，对于心仪的需求任务或者闲置物品进行收藏，后续用户量较大时可加入信用机制，根据需求任务完成情况对于用户信用进行量化，并以此为依据进行接取需求任务时的限制条件，以上的功能共同组成了川川帮。

1. 发布需求任务
2. 用户选择需求类别（闲置物品交易；需求解决；活动安排）
3. 用户输入具体标题，描述以及相应的附属条件（价格，时限等）
4. 对于数据处理后存入数据库
5. 接取需求任务
6. 用户选定想要接取的需求任务
7. 用户选择接取任务
8. 用户提供验证信息（对于某些隐私性较高任务如拿快递，该步骤必须）
9. 评价任务完成度

（1）在需求任务确认完成后，用户对于该次任务进行评价，以此为依据对执行者信用产生影响

1. 收藏需求信息
2. 用户看到心仪的物品或需求任务，但暂时无法完成，可选择收藏
3. 用户可在个人中心找到收藏内容
4. 查看个人信息
5. 用户进入个人中心
6. 用户可查看当前提供的验证信息以及历史的需求任务
7. 搜索需求任务
8. 用户提供关键词进行搜索
9. 针对显示的关键词内容进行选择

**2.3 运行环境**

项目基于微信小程序端进行开发，由于微信的生态体系非常强大，这就节省了大量的推广成本，并且微信小程序具有丰富的功能和出色的使用体验，同时也降低了用户的使用难度。微信小程序在各个平台的表现十分良好，对于开发者而言成本较低。

运行环境：ios系统以及安卓系统

软件环境：微信小程序

使用阿里云服务器以及数据库。

主要使用的技术有：Javascript，Django

JS负责业务逻辑的实现，页面则由WXML和WXSS语言（类似CSS）来实现。

**2.4 系统性能**

**一、数据性能**

系统应满足如下技术指标：

1. 数据类型支持

系统主要支持一般结构性事务数据包括文本，图片等，后续可考虑加入GPS定位数据等。

2. 数据库性能要求

根据本系统数据的特点,采用标准MYSQL语句，以便将来的扩展和移植。

系统将采用数据库建模工具，根据系统功能模块的设计，构建出整个数据库。在构建数据库时，也会定义好数据库表的约束、关联以及索引。

针对系统的具体特点和系统要求，我们在进行数据库方案设计时对数据库平台提出下列性能方面的要求：

o 支持对称处理和多线程技术，支持XML/CORBA，支持数据分区；

o 高级语言、汉化功能先进，易于方便使用，支持汉字，GB18030标准；

o 支持主流的网络协议，如TCP/IP、IPX/SPX、NETBIOS、DECNET、SNA等。

o 能支持同构、异构网络分布操作，支持松散耦合及海量并行处理；

o 有足够的并发控制，授权控制和事务处理能力及恢复能力；

o 与异种数据源有良好的可互操作性；

o 具有可靠的数据安全保密措施以及故障恢复能力；

o 具有很强的容错能力，错误恢复能力，错误记录及预警能力，具备异地容灾能力；

o 具有完整的安全性（帐号安全，系统级权限，对象安全性，审查等），细粒度化的访问控制，适合于多层环境的安全模式的能力；

**二、 并发性**

1. 数据库并发

数据库支持超过500个用户的并发访问能力。

2. 传输并发

系统业务功能包括附件和图片的传输的时候，需提供稳定快速的传输效率，以及支持多附件多图片并发上传和下载的能力。

**三、响应特性**

1. 查询响应

一般数据查询响应时间<3秒。

**四、架构特性**

1. 可靠性

系统需提供7\*24的不间断服务，实现数据的准确无误传输。

2. 稳定性

系统需合理的利用资源，保证前后台数据操作的效率，以及在数据响应和界面承载方面都要达到不会出现界面混乱、数据报错、触发按钮功能缺失、操作频繁或者快速容易崩溃的问题。

3. 兼容性

小程序天然的具有良好的各个平台的兼容性。

4. 灵活性

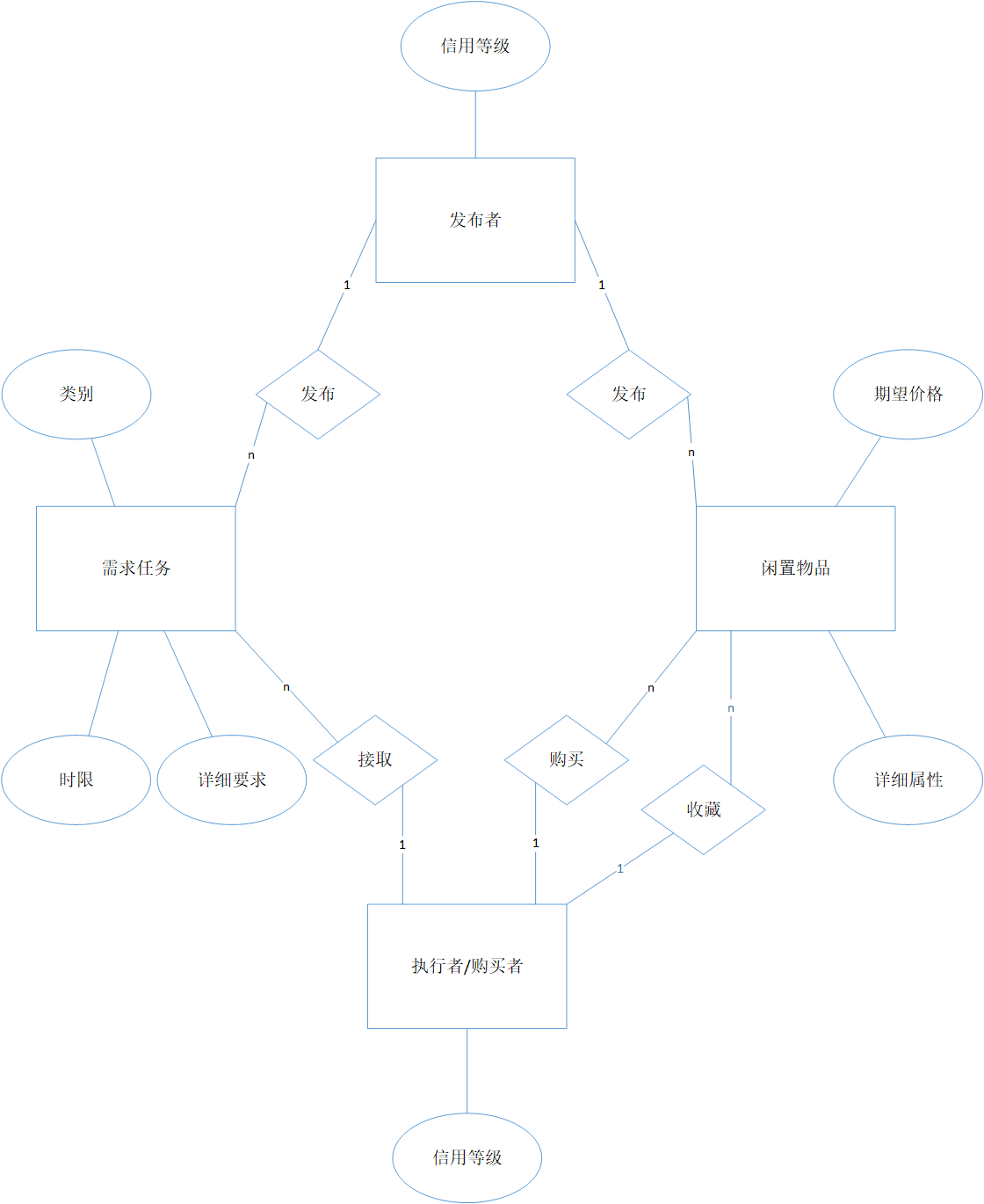
能够适应主流手机屏幕尺寸，该项目灵活性较高，适用于川大校园内部需求的解决，后续可进一步迭代加入更多的功能，也可以针对性的解决新增的需求。

**2.5 基本设计**

本项目为基于微信小程序的大学生在线互助平台，有需求的同学能够在小程序中发布自己的需求；有余力的同学能够利用自己的课余时间帮助其他同学，同时获得小额的报酬。本项目主要有发布需求，查看需求，接收需求，完成订单等功能，前端主要使用WXML、WXSS与JavaScript语言作为开发工具；后端主要使用Python语言中的FLASK框架、Golang语言中的Gin框架进行后台开发，使用Go+Docker的架构实现了微服务的部署，从而用更小的内存消耗，保证良好的新能以及可拓展性。

小程序端需要实现良好的多界面交互，以及流畅的动画，因此采用了渲染层与逻辑层分离的框架，逻辑进程与渲染进程分离，二者均与微信Native进程通信，从而在保证业务逻辑分离的同时，保证安全稳健的服务。在进行业务请求时，通过微信Native进程发出HTTPS请求，在保证信息安全的同时也保证了通讯的稳定性。

服务端采用了微服务架构，具有良好的可拓展性。当业务逻辑较少时，能够使用少量的容器完成整套的请求流程；当业务繁忙时，通过预留的脚本，可以一键式的在阿里云等云服务平台上，通过Dockerfile迅速部署容器，从而缓解主服务压力，保证了后台服务的稳定。在后台请求的核心调度中，我们采用了Flask框架与Nginx服务器，能够使用最小的代码量高效的完成任务的分配工作；后台的核心服务部署在了Docker容器中，与云Redis数据库，云MySQL数据库结合，实现了高并发以及每秒十万次的读写速度。



项目整体ER图

* 前端响应流程

前端采用视图逻辑分离架构，依赖核心的Wechat Native进程作为通讯媒介，提供良好的动画效果以及逻辑控制。

当逻辑层有网络请求时，请求会发送到Native队列中。Native进程随后会发送HTTPS请求到指定服务器中。服务器进行相应后，与客户端Native进程进行握手，然后进程将任务返回到逻辑层进行处理。

图片包含 屏幕截图

描述已自动生成

* 后端响应流程

多个客户端发出多条请求后，通过中间件服务，过滤掉可能的垃圾请求，请求进入均衡服务器请求队列中；均衡服务器会从Redis数据库中，实时读取微服务核心状态，并将请求分配到最近的空闲容器中。

微服务核心接收到请求后，会迅速进行相应，并且直接将响应结果返回客户端。数据库采用阿里云RDS数据库，降低了数据库的搭建与维护成本，提高了数据的可读性以及安全性。

图片包含 屏幕截图

描述已自动生成

**3. 模块设计**

**3.1 UI设计**

该项目作为微信小程序，开发时使用了ColorUI框架作为项目的UI框架，并据此编写了具体的代码，展示如下图所示：

用户进入首页后，可进行需求的查看，包括校内需求，活动组队以及闲置交易三个需求，不同的需求模块根据易用性的要求，用户利用滑动屏幕的方式进行具体查看。

第二个页面可进行双方消息的交流以及系统通知如“您发布的需求被成功接取”等，实现双方基于小程序的沟通，更方便的能够解决需求。

最后一个为基础设置界面，可在此提供校园卡便于身份验证，或者是对小程序开发人员提出意见，清空缓存内容，查看历史记录等等。

上述四图为小程序具体界面，后续将进一步完善各个界面的功能点。

**3.2 接口设计**

## 川川帮接口定义

1. 获取所有需求内容 getReqList (post)

2. 获取所有活动组队内容 getActList (get)

3. 获取所有商品列表 getGoodsList (get)

4. 查看详情 checkGoodsDetail (get) 序列化传递参数id

5. 获取顶部广告(热点)内容 getTopItem (get)

6. 发布需求内容 releaseItem (post)

7. 获取个人通知(如需求接取完成)列表 getPersonalMsg (get)

8. 用户发送信息 sendMsg (post)

9. 请求聊天内容 getAllMsg (post)

10. 获取聊天对象信息 getFriendInfo (post)

11. 提交身份验证(包括修改个人信息) submitIdCheck (post)

12. 获取个人历史记录(包括发布、接取以及买卖) getHistory (get)

13. 添加内容至收藏 addItemLike (post)

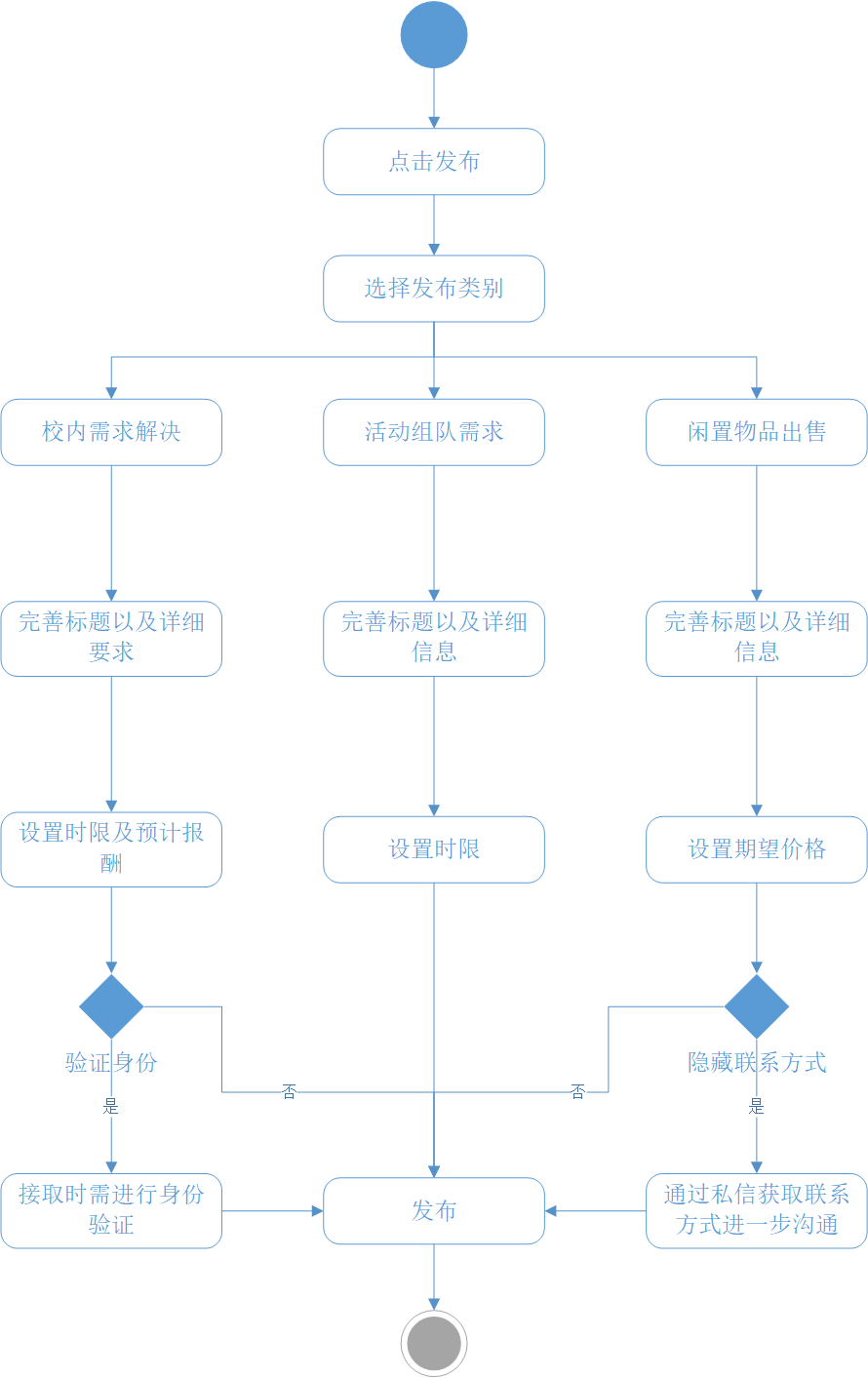
14. 查看收藏内容 checkLikeList (get)

**3.3 发布模块**

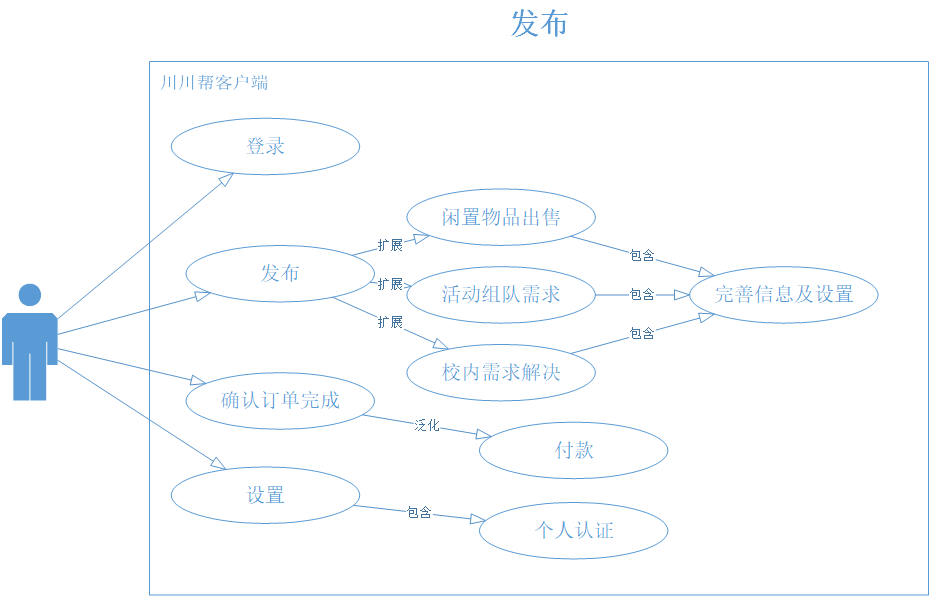
用户选择发布类别为校内需求解决后，将具体的要求地点等补充完善，选择时限以及预计的报酬，如帮忙取快递，则时限可设置为两天，报酬为3元，之后由于涉及到相应的隐私问题，所以针对性的加入了是否需要验证接取者身份的步骤，当用户选择是时则需验证接取者的身份信息，方可将自己的信息进行透露。

用户选择发布类别为活动组队时，需完善标题以及详细的信息方便后续的搜索，之后设置相应的时限，便于双方进行后续的联系。

用户选择发布闲置交易信息时，需完善标题以及物品详细的介绍，最后需要填写预期的价格，之后判断是否需要隐藏自己的联系方式，如果是则买家需要通过私信的方式来进行进一步沟通，之后便完成发布。



发布模块活动图



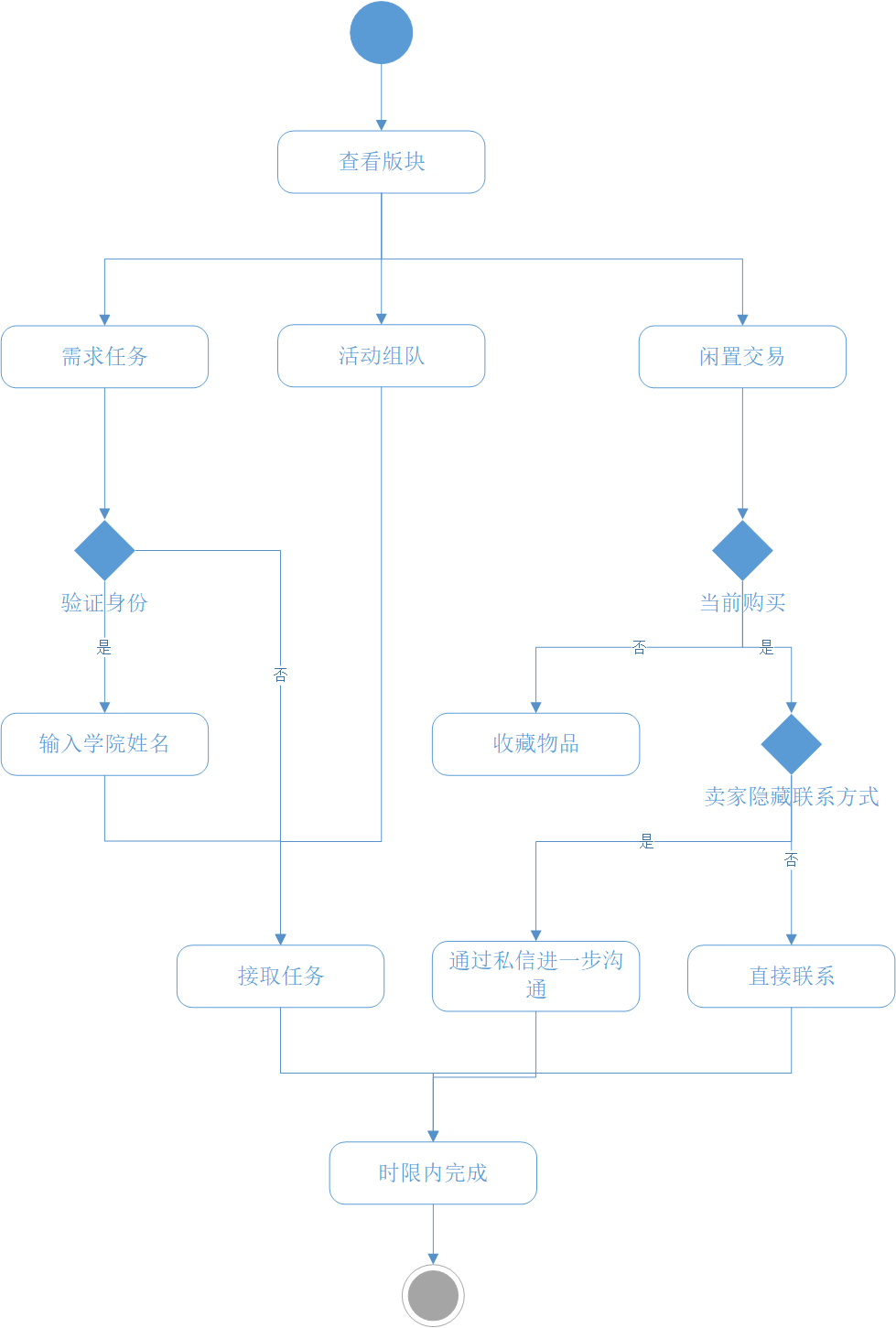
发布模块用例图

**3.4 接取模块**

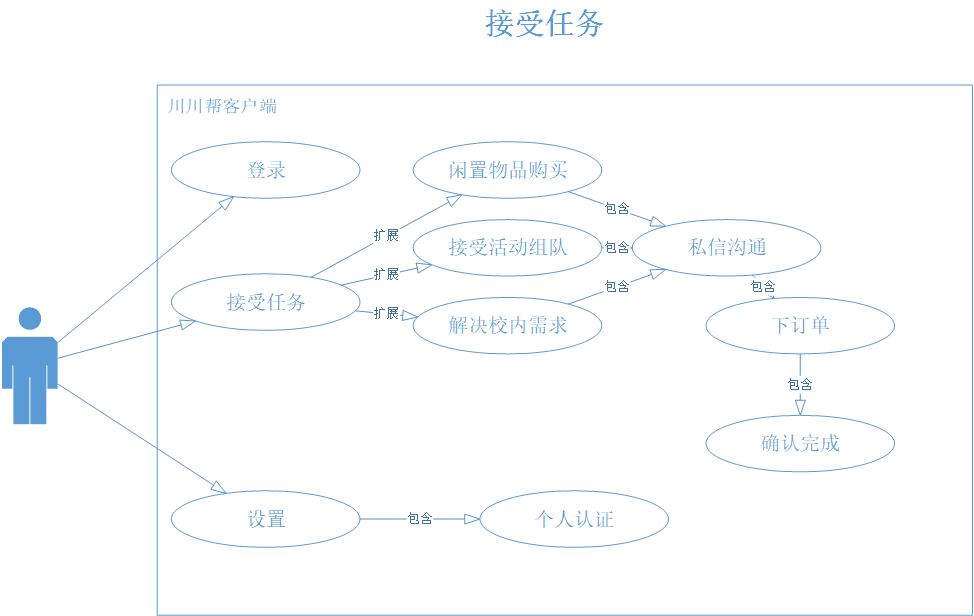
当用户需要接收任务时，需要先进行登录，并且已进行过个人认证，以方便后期的交易活动顺利进行。

找到自己心仪的任务或物品时，可以对订单进行评价或者对发起人进行私信沟通。如果决定接受则可以下订单。

在订单结束之后接收者确认订单结束提醒发起者确认，若是有报酬则会转到接收者账上。



接取模块活动图



接取模块用例图

接取模块界面设计

用户在需求解决版块时，针对不同的需求类型，可能出现不同的要求，比如帮忙拿快递这一点上，由于涉及到较为隐秘的私人信息，所以会要求接需求任务的同学提供验证信息，作为保证，同时针对较为简便的需求，可不进行需求验证，如教学楼送伞。

用户在活动组队版块时，可直接根据需求的详细信息判断时间条件是否合适，接下来就直接接取任务，双方建立联系。

用户在闲置交易版块时，可对有意向的物品进行收藏，系统判断是否需要当前进行购买，用户可将物品收藏，在个人信息版块后续进行再次查看。