**基于局域网的分布式下载系统**

目录

[一、产品设计方案 2](#_Toc61177525)

[1.项目实施可行性报告 2](#_Toc61177526)

[1.1 行业市场分析 2](#_Toc61177527)

[1.2 竞争对手或同类产品分析 2](#_Toc61177528)

[1.3 自身条件分析 3](#_Toc61177529)

[2. 产品定位及目标 4](#_Toc61177530)

[2.1 产品特征 4](#_Toc61177531)

[2.2 目标市场 4](#_Toc61177532)

[2.3 目标市场特征 4](#_Toc61177533)

[2.4 目标市场需求 4](#_Toc61177534)

[3. 产品内容策划 4](#_Toc61177535)

[3.1 应用流程规划 4](#_Toc61177536)

[3.2 设计与测试规范 9](#_Toc61177537)

[3.3 开发日程表 9](#_Toc61177538)

[4. 技术解决方案 9](#_Toc61177539)

[4.1 概述 9](#_Toc61177540)

[4.2 系统设计 10](#_Toc61177541)

[5. 推广方案 10](#_Toc61177542)

[5.1 线上推广 10](#_Toc61177543)

[5.2 线下推广 10](#_Toc61177544)

[6. 运营规划书 11](#_Toc61177545)

[6.1 培育阶段 11](#_Toc61177546)

[6.2 成长阶段 11](#_Toc61177547)

[6.3 稳定阶段 11](#_Toc61177548)

[6.4 盈利模式 11](#_Toc61177549)

[二、产品实现方案 11](#_Toc61177550)

[1.系统的主要功能 11](#_Toc61177551)

[1.1带宽合并 11](#_Toc61177552)

[1.2任务分块 11](#_Toc61177553)

[2.UI界面设计 12](#_Toc61177554)

[3.关键技术和技术难点 12](#_Toc61177555)

[4.用户体验记录和分析 12](#_Toc61177556)

[5.已完成的改进和存在的问题 12](#_Toc61177557)

[三、测试大纲和测试报告 12](#_Toc61177558)

[四、产品安装和使用说明 12](#_Toc61177559)

# 一、产品设计方案

## 1.项目实施可行性报告

### 1.1 行业市场分析

#### 1.1.1 行业简介

这个APP是下载器型的APP，主要功能就是下载。现在，网络已经成为了人们现代生活中不可或缺的一部分。网络的传输速度也成为了人们密切关注和追求的性能指标之一。

#### 1.1.2 市场特征

大学一般拥有独自的校园网络，已缴纳网费的学生能够通过校园网络进行上网。通过校园网络上网的学生之间带宽是独立的，即相互间的上网速度互不影响，但校园网络的带宽往往过低，无法满足学生对于大文件下载的需求。而处于同一局域网的其他账号，往往处于空闲状态。如果局域网中其他账号同意共享带宽，那么就可以借助共享的带宽帮助下载，提高下载文件的速度，更加快速地完成下载任务，缩短等待时间。

因此我们可以知道这个市场发展迅速，受众庞大，能在短期内获得利润回报；但是，其发展的蓬勃性也伴随着负面矛盾。

#### 1.1.3 发展环境

大环境下是移动互联网和智能手机的普及，小环境下是人们随着时间推移对快速下载文件的需求。

#### 1.1.4 发展趋势

功能性上朝着专业化发展；推广性上朝着打造长周期的、良好的用户体验发展。

### 1.2 竞争对手或同类产品分析

#### 1.2.1 关于分布式下载的关键词搜索指数的分析

5118.com：

****

chinaz.com

****

由此可见我们这款下载APP的市场竞争不那么激烈，说明我们现在有着较大的市场。

#### 1.2.2 关于类似的下载类APP

与我们产品类似的是P2P下载，但是P2P是对等的是所有下载同一文件的用户间相互分享数据，互惠互利，例如我们都要下载同一个软件，所以我们可以共享下载的资源，而该系统则是直接分享带宽，仅仅是为了帮助别人，不一定是互助的关系。因此我们APP在校园网下载文件时更加便捷。

### 1.3 自身条件分析

#### 1.3.1 产品性质特点

该软件是面向在校大学生的，以共享带宽，快速下载为主题的，以为广大同校师生提供更方便快速的下载渠道为目的平台，同时也是一个属于广大学生互帮互助的平台。产品的主要特色是较强的便利性和亲和性。

#### 1.3.2 产品竞争力分析

能够满足在校学生下载文件的需求。该系统则是无论下载的是什么，只要用他人同意，就可以借助他人的带宽分部分的下载文件，而用户也可以选择自己共享的带宽大小和共享自己带宽的设备，产品所提供的下载方式正是在校学生所需的贴近大学生生活和学习的产品，这是异于下载APP的广泛性的一个特点。

## 2. 产品定位及目标

### 2.1 产品特征

该APP的基础功能是能够通过局域网连接其他终端，建立连接，并且分配下载任务，能够自动检测掉线终端，并且对其任务进行再分配。同一局域网下，若其他账号同意共享带宽，则可利用其共享的带宽帮助下载，提高下载文件的速度，并且用户可设置密码来阻止未知设备使用自己的带宽。操作简单，界面良好。

### 2.2 目标市场

面向所有高校的大学生。

### 2.3 目标市场特征

由于无线访问接入点提供的接入能力是有限的，所以如果给某一个用户的带宽高了，势必影响对其他用户的服务，所以学校一般都会要求进行基于每个用户的带宽限制。接入校园网人数过多，不论是动态限制还是静态限制，每个学生最终得到的带宽都过低，无法满足大文件下载的需求，下载几百M以上的文件，快则几分钟，慢则半小时。

### 2.4 目标市场需求

由于网速过低，学生在线看视频总是要不断加载，出现卡顿等现象，十分影响体验，而下载之后观看可以解决看视频卡顿的问题，但由于网速过低，下载所花费的时间过长，或者下载的过程中中断了要不断重新下载，十分麻烦。若此时能够快速下载完，学生就能够马上体验极佳地观看视频。Android内置的下载和iOS 上的Safari下载都是单线程的，速度缓慢，使用我们的app则可分布式下载，体验飞一般的感觉。

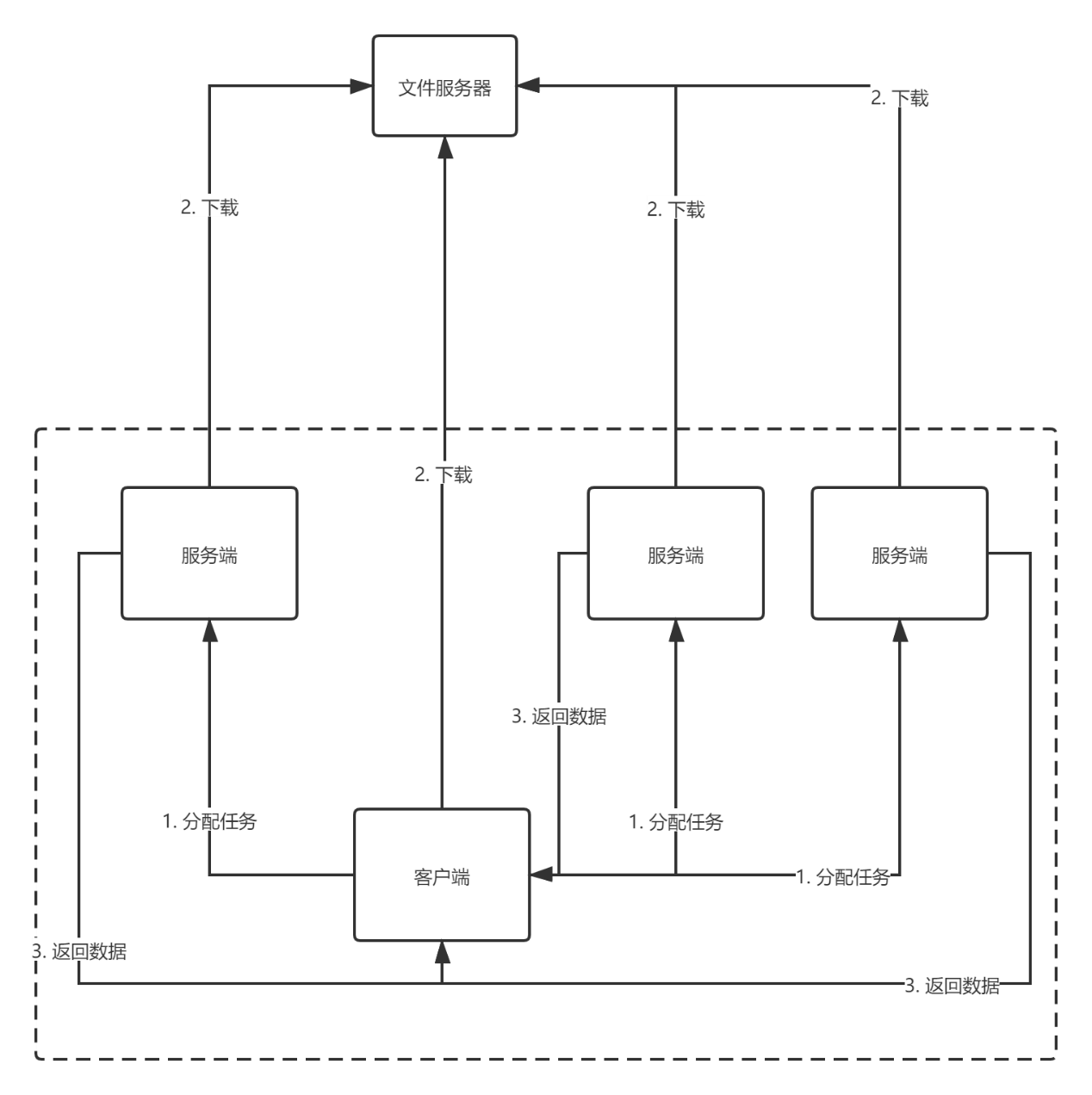
## 3. 产品内容策划

### 3.1 应用流程规划

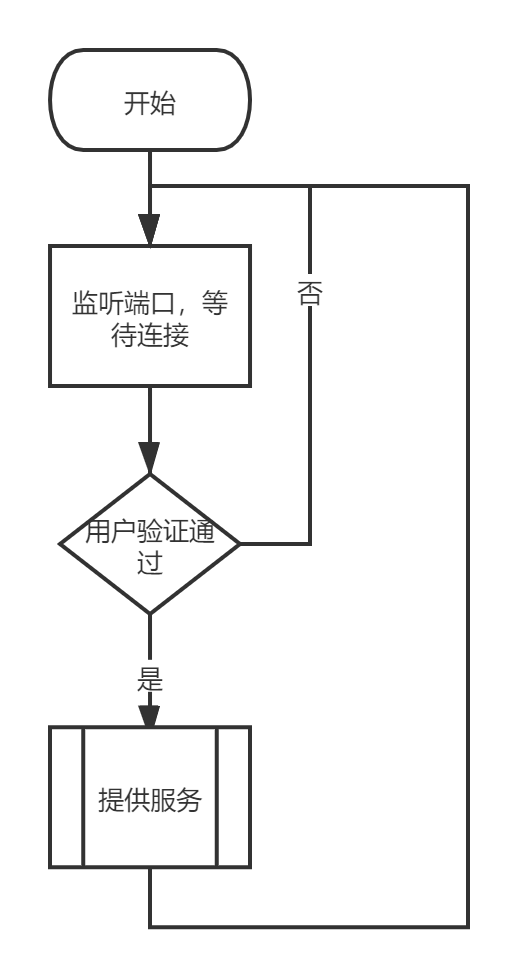
应用主要分为服务端和客户端两个模块，其中服务端和客户端集成在同一个安卓应用中，用户可以根据需要开启两个模块的功能。

#### 3.1.1 工作流程

局域网内服务端和客户端配合，能够很大程度地加快下载速度。



#### 3.1.2 服务端主要流程

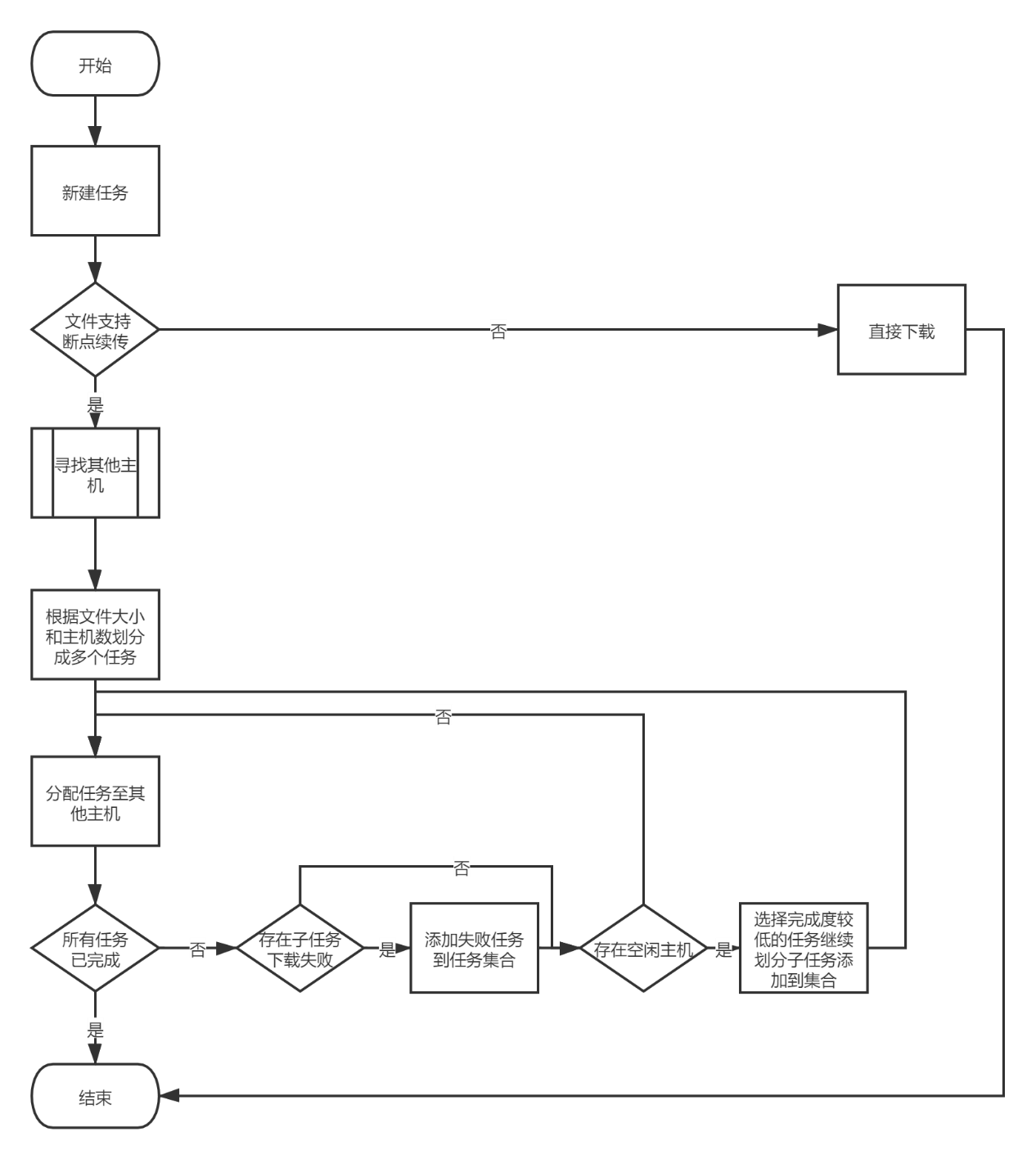
服务端旨在为别人提供下载服务，帮助其他用户更快地完成下载任务。服务端模块中提供了用户鉴权的功能，能够阻止未知用户的连接，并且可以对连接的用使用的带宽进行限制。

#### 3.1.3 客户端主要流程

客户端通过TCP协议和服务器进行连接，可以自由的控制任务的开始、暂停、恢复。

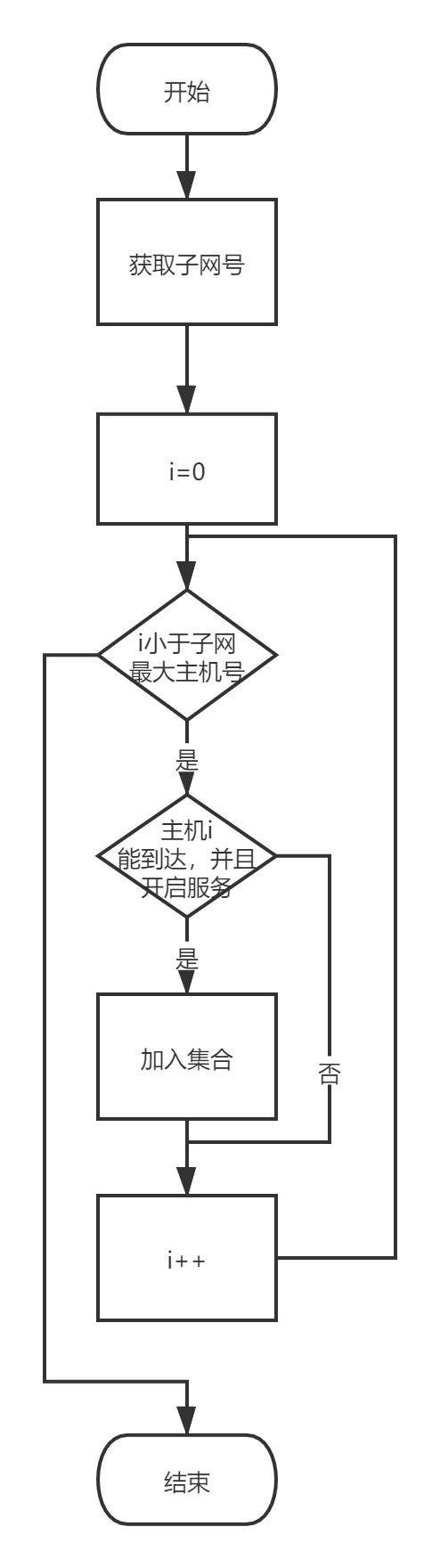
通过判断文件是否支持断点续传，从而决定下载的方式。

1. 若支持断点续传则将任务划分，并且将任务分配给其他主机。
2. 等待其他主机完成任务，如果有空闲主机，那么可以自动从完成度较低的任务中继续划分进行分配。若有主机断开连接导致任务失败，则将其任务重新分发。
3. 开始、暂停、恢复等命令通过一套自定义的协议来进行交流。



#### 3.1.4 局域网扫描

局域网扫描能够自动的搜寻其他主机，加强了程序的易用性。同时能够在公共场合使用陌生人的服务。



### 3.2 设计与测试规范

#### 3.2.1 编码规范

1. 使用Android Studio开发。
2. 采用Android Studio默认的代码风格。

#### 3.2.2 APP设计规范具体内容

1. 采用Material Design设计，遵循统一的准则。

#### 3.2.3 测试规范

1. 测试用例编写应严格根据PRD、用户故事及测试需求功能分析点进行，要求覆盖全部需求功能点。
2. 测试用例编写应该制订统一的模板进行，并约定模板的使用方法。
3. 测试用例中要写清楚测试的操作步骤和预期结果，能被不同测试人员理解和执行。
4. 测试用例中要明确一级模块、二级模块和三级功能。
5. 注重测试用例的可复用性，即在以后相似系统的测试过程中可以重复使用。

### 3.3 开发日程表

1. UI设计（2周）
2. 划分小组，同时开发多个模块（4周）
3. UI界面（4周）
4. 应用测试、修正以（1周）
5. 撰写文档（1周）

## 4. 技术解决方案

### 4.1 概述

本方案为—基于局域网的分布式下载系统（以下简称本系统）提供技术内容解释。

### 4.2 系统设计

#### 4.2.1 系统模式

本系统采用分布式系统模式，因为其具有数据处理层面的分布性，（只有自己的资源是本地的，其余的处理终端及其资源非本地）但各个终端又具有自治性（各个终端之间可通过通信线路传递信息，也能独立工作，即不影响原设备独立性），且通过通信进程实现全局性（各个终端之间通过通信进程统一调配资源，实现分布式下载中的数据传输）。

#### 4.2.2 系统架构

本系统是安卓原生 APP，部署平台为Android，采用Kotlin语言开发，开发工具为Android Studio。

#### 4.2.3 主要问题及解决手段

1. 应对开发过程中的代码对接难题，采用不同模块专人开发的策略，避免引起代码冲突，同时使用Git统一管理版本。
2. 在后续开发过程中的难点会在下一版本中逐一提供解决方案。

#### 4.2.4 产品介绍

本系统旨在以不影响正常网络功能使用为前提，通过充分利用局域网内的闲置带宽获取互联网资源，并经局域网数据传输后实现下载提速，提升下载效率。

## 5. 推广方案

### 5.1 线上推广

1. 建立公众号，在朋友圈、微信群推广。
2. 论坛、贴吧、微博推广。
3. 应用商店推广：如酷安、豌豆荚等。
4. 建立奖励措施：鼓励用户将产品推广给亲朋好友，依据成功推广的数量给予响应的、产品功能上的奖励。

### 5.2 线下推广

1. 在校园内派发宣传单。

2. 在人流大且可免费宣传的地方张贴海报，简单有效地介绍产品的特点。

## 6. 运营规划书

本产品为用户提供免费服务，并且考虑到我们产品的对象为在校大学生，商业运营模式如下。

### 6.1 培育阶段

完善产品功能。对少量用户的集中运营，对象主要为较为熟悉的同学、朋友。能够掌握用户的详细信息，便于收集用户需求，改进产品功能。方式：社群：建立并活跃微信、qq群。

### 6.2 成长阶段

扩大用户群。将产品推广到全校，并尝试在应用商店上线，让产品接触更多样的用户。

### 6.3 稳定阶段

加入盈利模式。直接从用户获取，或者以用户易于接受的形式植入广告。

### 6.4 盈利模式

1. 吸引用户：优化UI，并为用户提供不同风格的主题，可考虑收费

2. 植入广告

# 二、产品实现方案

## 1.系统的主要功能

### 1.1带宽合并

充分利用局域网内的空闲带宽资源以提升下载速度。

### 1.2任务分块

将任务分为若干个小块，每个小块独立下载，互不影响。适当设置块大小，间接达到了断点续传的效果。

## 2.UI界面设计

### 2.1Task

在任务界面可以看到正在下载中和已完成的任务，点击加号输入下载链接新建任务。通过长按删除任务。

### 2.2Host

在主机界面可以点击加号输入socket添加设备，通过开关启用或停用设备。同时在此界面中也可以查看已经添加的设备。通过长按删除任务。任务有不同状态，在启用任务时，会自动连接代理，即连接中状态（黄色）。连接完成后会进入已连接状态（绿色）。如果发生错误，则会进入错误状态（红色），并出现感叹号，点击感叹号进入重连。

### 2.3设置

在以上两个界面中可以点击右上角的三点进入设置界面。

## 关键技术和技术难点

### 3.1设计数据库

像许许多多的软件一样，我们也需要用到数据库。故设计一个具备增删改查功能的基础数据库来管理数据便成为了我们的首要难题。

考虑到SQLite代码繁琐，因此我们选择了使用ORM框架来实现。在许多框架中，我们选择了Google的Room库来进行构建。Room能与Kotlin、LiveData等协同使用，并且Room相比其他ORM框架，它在编译期就会验证所有的@Query和@Entity，可以确保编译通过后的运行阶段基本不会出错。

### 3.2断点续传与任务分块

初次实现时，我们将这两个子功能独立编写，但在测试中我们发现，当任务分块把任务分的足够小时，其实也已经实现了断点续传。于是我们便将这两个子功能合并到一起，简化了代码的复杂度。

### 3.3实现代理服务器

代理服务器的实现较为困难，需要熟读RFC，因此，我们没有自己实现代理服务，而是采用了网上已有的库littleshoot。

## 用户体验记录和分析

我们寻找了10位同学来对我们的App进行评测，大致概括为3点：

1. UI界面不够友好
2. App有时会崩溃
3. 主机需要自动添加，希望能够自动扫描

## 已完成的改进和存在的问题

已完成：

1. 代理服务
2. 设计数据库
3. 添加、修改、删除服务器
4. 添加、修改、删除任务
5. 获取任务信息、测试文件是否支持断点续传
6. 将任务进行分块，下载文件
7. 核心功能已经完成，在使用多服务器的时候速度提升明显。

存在的问题：

1. 小组内开发的交流不多，代码写得比较混乱
2. 下载器基本功能已完成，但还有待完善，如暂停功能、局域网扫描的功能

# 三、测试大纲和测试报告

## 1.测试大纲

### 1.1基础核心及UI测试

对刚编码完的基础功能进行测试，同时测试UI。

### UI优化测试

在上一次测试后的UI基础上优化UI，并再次进行测试。

### 1.3子功能测试

对开发的子功能进行测试。

### 1.4软件核心功能测试

对软件进行完整测试，包括添加终端、新建任务、下载测速等等。

## 测试报告

### 1.1基础核心及UI测试

基础核心：下载功能

在本次测试中发现了许多bug，但都一一修复。UI界面基本满足要求，但不够完善和美观。

### 1.2 UI优化测试

对优化后的UI进行了测试，效果良好。期间进行过多次测试和优化。

### 1.3子功能测试

子功能：任务的管理（增删查）、终端的管理（增删）

子功能的测试结果令人满意，除了少数bug外，可以满足需求。完善后再次进行测试，没有问题。

### 1.4软件核心功能测试

核心功能：分布式下载

经对比测试后，可以发现下载速度明显提高，但在过多设备时提速不明显，原因猜测是路由器负载过高。

# 四、产品安装和使用说明

将apk文件用手机自带的打包安装程序打开并进行安装。打开程序后，首先点击Host界面，输入终端对应的socket添加终端，通过点击终端列表项中某个项的开关按钮来启用或停用该项终端，当该项出现online字符且变为绿色时，说明运作正常。此时可以点击Task界面，输入下载链接后会自动创建任务并开始下载，接着便可以享受到高速下载了。