



# Dingo

Dingo 项目计划书

DingoLab  
May, 2016

# Dingo Dingo 项目计划书

康赣鹏

DingoLab

ganpengkang@gmail.com

May 30<sup>th</sup> 2016

Version 0.1.0.0

西安

DingoLab

# 目录

<b>1</b>	<b>范围</b>	<b>1</b>
1.1	标识	1
1.2	系统概述	1
1.3	文档概述	1
1.4	术语定义	1
<b>2</b>	<b>引用文档及参考资料</b>	<b>2</b>
<b>3</b>	<b>策划背景概述</b>	<b>2</b>
<b>4</b>	<b>项目开发活动的总体实施计划</b>	<b>2</b>
4.1	软件开发过程	2
4.2	项目开发的总体计划	3
4.2.1	软件开发方法	3
4.2.2	软件需求的处理	3
4.2.3	决策理由的记录	3
<b>5</b>	<b>详细软件开发活动实施计划</b>	<b>3</b>
5.1	项目策划和监控	3
5.2	软件开发环境建立	3
5.2.1	后端	3
5.2.2	数据库	4
5.2.3	前端	4
5.3	软件需求分析	4
5.4	软件设计	5
5.4.1	软件设计任务分布	5
5.4.2	前后端 API 调用的实现	6
5.5	软件实现和测试	7
<b>6</b>	<b>项目组织和资源</b>	<b>7</b>
6.1	项目组织	7
6.2	项目资源	7
6.2.1	人力资源	7
6.2.2	职责分配	7

6.3 项目依赖关系分析 . . . . . 7

6.4 技术方法和工具 . . . . . 7

6.5 成本估计 . . . . . 8

# 1 范围

## 1.1 标识

本文档为 pdf 文件适用于各个操作系统

## 1.2 系统概述

- 文档适用的系统为 (window mac linux 等等)
- 软件用途: 用户通过我们提供 app 软件客户端, 可查看是否有人在快递集散地, 并通过我们的软件协商实现交易, 消费掉这最后一公里。
- 软件特性: 主要由后端和客户端组成, 客户端在 wp 平台上开发, 后端主要用 Haskell 语言开发数据库开发软件为 PsotgreSQL
- 软件尚处于开发阶段开发方主要由 4 名学生组成
- 有关文档: 产品需求文档、API 文档、后端文档、前端文档

## 1.3 文档概述

本文档主要阐述了开发项目的背景, 项目在国内外市场测分析, 项目主要开发建设的内容和项目实施的技术方案。另外, 对项目人员的组织, 资源的分配以及客户的交流做了初步分析

## 1.4 术语定义

1. API 文档: 确定前后端接口的文档
2. 后端文档: 项目后端设计文档
3. 前端文档: 项目前端设计文档
4. 测试文档: 测试程序的是否能全面完成功能的文档
5. 产品需求文档: 分析一个产品要实现多少功能的文档

## 2 引用文档及参考资料

Haskell Wiki	Hackage Document	PostgreSQL 官方手册
msdn 开发者文档	知乎日报 uwp 版	博客园客户端 uwp 版

## 3 策划背景概述

随着网购逐渐的渗透于我们的生活, 快递也成为我们生活必不可少的一部分。快递方便快捷, 为我们的生活减轻了不少负担。然而在许多地区, 因为各种各样的原因, 快递还是不能实现把商品直达家庭的目标。存在着“最后一公里”, 这与快递这种服务诞生的理念相悖。

根据我的调查, 在中国的大部分高校。因为快递的种类繁杂, 人员众多, 以及宿舍的集体生活, 大部分快递都选择在学校租赁一个快递集散地, 让学生自取。这造成一种现象就是: 每当下课铃响后, 学校的两个地方就会挤满人, 一个是饭堂, 一个就是快递集散地。将宿舍和快递集散地这段路程利用起来非常有必要。

还有, 在中国的大部分农村, 手机通讯, 网络的搭建已经开始普及了, 但运输快递的网络雏形还没有出现。大部分的村子, 想要领取快递必须到附近的镇子上, 耗时还费劲。快递公司若要拓展这部分市场, 必须投入大量人力物力, 还很难收回成本。这时候, 我们可以利用的就是人民自身的力量, 填补这部分市场的空缺。

在国外, 许多地广人稀的地方通过类似于快递集散地的驿站, 快递的最后一公里还是得客户去消化。这证明, 无论是在国内还是国外, “最后一公里” 得问题还是没有得到有效得解决, 这部分市场的空间很大。

## 4 项目开发活动的总体实施计划

### 4.1 软件开发过程

软件生命周期模型为敏捷开发的生命周期模型, 软件开发方法为敏捷开发。开发思路按照: 需求 → 分析 → 设计 → 编码 → 测试的基本思路进行, 并在此基础上做出改进。将架构设计细分成三个部分, 针对每个部分进行详细设计 → 编码 → 单元测试。其中两个部分完成后进行集成测试, 所有部分完成后, 进行系统测试。减少错误的风险, 同时保证效率。

## 4.2 项目开发的总体计划

### 4.2.1 软件开发方法

参照敏捷开发的思路，整个开发过程用了 teambition 这个工具实现了任务的发布，团队的交流和任务阶段结束后的总结的目的。使得团队能够达到迅速高效的交流沟通。

### 4.2.2 软件需求的处理

根据开发软件的实际需求编写文档，让前端和后端工程师根据需求计算工作量。发布每个部分的内容。当工程师完成内容后，由测试工程师，做出单元测试，看是否符合软件需求。

### 4.2.3 决策理由的记录

通过纪录记录决策。

## 5 详细软件开发活动实施计划

### 5.1 项目策划和监控

目标：用户通过我们提供 app 软件客户端，可查看是否有人在快递集散地，并通过我们的软件协商实现交易，消费掉这最后一公里。

规模：项目尚处于开发阶段，规模较小

工作量、

关键计算机资源等估计。

本条也包括进度的导出方法等。

### 5.2 软件开发环境建立

#### 5.2.1 后端

**操作系统** Window 7(及以上)，Linux（内核 4.0 版本及以上），OS X 10.9（及以上）

**开发语言** Haskell

**依赖软件** Glasgow Haskell Compiler 7.10（及以上，Windows 要求 7.10.3 以上），Cabal 1.22（及以上），Stack 1.0.0（及以上）

### 5.2.2 数据库

**数据库管理软件** PostgreSQL

**使用语言** PL/SQL 与 PL/C 参考后端文档

### 5.2.3 前端

**操作系统** Windows 10 Build 10586 版本（及以上）

**集成开发环境** Visual Studio 2015 Update1（及以上）

**软件开发套件** Windows10 SDK 10586（及以上）可参考前端文档

## 5.3 软件需求分析

用户注册及用户管理。

- 手机号
- 验证码)
- 密码
- 找回密码
- 实名认证
- 头像
- 昵称
- 用户详细的家庭住址
- 上传若干张多角度的自行车照片，供其他用户参考
- 要求用户选择可以搭载的快递大小和重量
- 需要考虑快递是否是易碎物品，若打碎，要有责任承担者或者不能碰水

用户通过软件呼叫，实现搭载其他用户自行帮忙携带快递的目的，费用由二者商议决定。

- 利用手机内的 gps 实现对每个用户的定位，实时计算最适合用户搭载的自行车对象。



- 用户通过界面发出请求，系统按优先级（优先级判断）排列出可搭载的自行车对象，返回给界面，让用户做出选择。
  - 用户和自行车车主可交流商议费用
  - 利用第三方 api 实现交易
  - 发布人信息
  - 任务状态：已完成 / 已接单 / 未接单
  - 任务详细内容 A—>B A 地址 B 地址任务描述：从 A 到 B 搭自行车或者送快递。送快递要有快递的适当描述特别要注明是否是易碎物品不能碰水（不能拆快递）
  - 快递类别：电子设备书本水果生活用品信件
  - 任务注意事项：由用户填写
  - 交易：代收快递的车主要有实名认证才能接单，单到了确认无误后第三方再把钱打给接单者
  - 防止垄断：限制每个人接单的次数
  - 任务流水号
- 设置账号等级及评价体系。
- 用户信用跟账号等级直接挂钩，账号等级为优先级判断提供素材
  - 用户交易完后可对自信车车主进行评价，包括等级评价、文字评价和图片评价，车主可以回复从而进行申诉。其他用户可直接查看该车主的所有评论

用户间的信息交流

可参考需求分析文档

## 5.4 软件设计

### 5.4.1 软件设计任务分布

分为前端和后端分别由前端设计师和后端设计师开发。

## 5.4.2 前后端 API 调用的实现

### 用户管理

- 用户注册
- 用户认证
- 查询认证状态
- 用户登入
- 用户登出
- 查询用户信息
- 获取用户的头像
- 用户信息修改
- 修改密码
- 收货地址增删改

### 任务管理（代收快递）

- 任务发布
- 任务的删改
- 获取任务
- 协商价格
- 获取任务信息
- 支付与确认
- 可代收状态的增改
- 可代收状态的删除与完成
- 可代收状态查询

**消息通告** 具体实现可参考 API 文档

## 5.5 软件实现和测试

参考单元测试文档，集成测试文档和系统测试文档

## 6 项目组织和资源

### 6.1 项目组织

### 6.2 项目资源

#### 6.2.1 人力资源

投入人数：4 人产品经理、前端工程师、后端工程师、测试工程师

#### 6.2.2 职责分配

**产品经理** 负责整个项目的开发计划及软件需求文档的编写。

**前端工程师** 负责前端的设计（包括界面，交互，信息存储等）和前端设计文档的编写。

**后端工程师** 负责后端的设计（包括数据库，业务流程、信息处理等）和后端设计文档的编写。

**测试工程师** 针对每个任务单元测试和测试文档的编写。

### 6.3 项目依赖关系分析

产品经理负责发布任务和任务结束时间，并督促工程师，保证任务进度。前端和后端工程师要进行定期的讨论，确定任务进度。当一个任务阶段完成后，测试工程师应完成单元测试文档，并对当前单元进行测试。以此为基本流程，进行项目开发。

### 6.4 技术方法和工具

**文档写作** L<sup>A</sup>T<sub>E</sub>X 与 Markdown。

**图形设计** L<sup>A</sup>T<sub>E</sub>X 与 Visio

## 6.5 成本估计

**时间成本** 开发周期为一个半月

**开发成本** 数据库存储于小型服务器，在 docker 容器里运行