**项目开发计划书**

**综合运营平台**

目录

一．引言3

1.1 编写目的3

1.2 背景3

二．项目概述3

2.1 工作内容3

2.2 主要参加人员3

2.3 产品4

2.3.1 程序4

2.3.2 文件5

2.3.3 非移交产品5

三．实施计划5

3.1 工作任务的分解与人员分工5

3.2 接口5

3.3 进度5

3.4 关键问题6

一． 引言

1.1 编写目的

软件开发周期较长，涉及环节较多，开发过程复杂，如果没有一份详尽的开发计划，项目整体进展会受到阻碍。为了使软件开发过程合理有序，将开发各个阶段目标，参与人员，工作分配进行详细说明，有助于团队合作有条不紊地进行，严格按照项目开发计划执行开发流程也可以很大程度上避免突发情况造成的项目延期。以项目开发计划作为开发向导，可以明确目标，步骤化结构化开发总过程，对于项目的进展有利无害。

本项目开发计划用于总体上指导金融服务系统综合运营平台顺利进行并最终得到通过评审的项目产品。本项目开发计划面向项目组全体成员。

1.2 背景

综合运营平台是金融服务系统中的面向商家的前端部分，编写该网站为金融服务平台提供商家运营平台，实现虚拟账户系统和清结算系统等的可视化。综合运营平台旨在便捷商家进行账务的管理，对于商家的个人账户信息，交易信息进行规范化管理。作为金融服务平台，相比个人用户，商家用户所占的比重往往更大，对于交易信息的需求也更高，为方便交易对账等财务管理，面向商家的信息综合运营管理平台存在着必要性。由于本项目只是金融服务系统的一部分，因此综合运营平台还需要借助虚拟账户系统，清结算系统及数据区块链存储等后端作为基础才可以顺利运行。

二． 项目概述

2.1 工作内容

1. 完成需求分析与概念设计
2. 进行软件详细设计
3. 使用Axure完成产品页面设计
4. 依据产品demo搭建网站
5. 使用MQ进行项目测试
6. 编写部署文档和用户手册，交付项目

2.2 主要参加人员

（排名按姓氏排序）

1. 段柯宇
2. 李光昊
3. 李昕航
4. 孟庆博
5. 尹蕾

2.3 产品

2.3.1 程序

**名称：《金融服务平台-综合运营平台》**

|  |  |  |  |
| --- | --- | --- | --- |
| 前端技术 | angularJS | bootstrap | jQuery |
| 缓存框架 | Redis |  |  |
| 消息队列 | rabbitMQ |  |  |

* angularJS：是一个JavaScript框架，它是一个以JavaScript编写的库，angularJS把应用程序数据绑定到HTML元素，可以克隆、重复、显示和隐藏HTML元素，支持输入验证
  + AngularJS 把应用程序数据绑定到 HTML 元素。
  + AngularJS 可以克隆和重复 HTML 元素。
  + AngularJS 可以隐藏和显示 HTML 元素。
  + AngularJS 可以在 HTML 元素"背后"添加代码。
  + AngularJS 支持输入验证。
* bootstrap：是一个用于快速开发web应用程序和网站的前端框架。bootstrap是基于HTML、CSS、JavaScript的
  + **基本结构**：Bootstrap 提供了一个带有网格系统、链接样式、背景的基本结构。
  + **CSS**：Bootstrap 自带以下特性：全局的 CSS 设置、定义基本的 HTML 元素样式、可扩展的 class，以及一个先进的网格系统。
  + **组件**：Bootstrap 包含了十几个可重用的组件，用于创建图像、下拉菜单、导航、警告框、弹出框等等。
  + **JavaScript 插件**：Bootstrap 包含了十几个自定义的 jQuery 插件。您可以直接包含所有的插件，也可以逐个包含这些插件。
  + **定制**：可以定制 Bootstrap 的组件、LESS 变量和 jQuery 插件。
* jQuery：一个轻量级的、流行的JavaScript库，主要包含以下功能：
  + HTML 元素选取
  + HTML 元素操作
  + CSS 操作
  + HTML 事件函数
  + JavaScript 特效和动画
  + HTML DOM 遍历和修改
  + AJAX
  + Utilities

2.3.2 文件

* 用户使用说明书
* 部署文档
* 可执行程序

2.3.3 非移交的产品

* 源代码
* 项目开发计划
* 需求分析文档
* 软件设计说明书
* 测试报告

三．实施计划

3.1 工作任务的分解与人员分工

项目开发主要步骤及各阶段任务参与人员分配如下：

1. 项目开发计划编写（李昕航，段柯宇）
2. 需求分析文档编写（尹蕾，李光昊，孟庆博，李昕航，段柯宇）
3. 软件设计说明书编写（李昕航，段柯宇）
4. 软件源代码开发（段柯宇，李昕航）
5. 软件测试与分析（李光昊，孟庆博）
6. 测试分析文档编写（李光昊、孟庆博）
7. 部署文档和用户手册编写（段柯宇，李昕航，尹蕾）

3.2 接口

项目主要为金融服务平台的前端部分，因此接口需要与金融服务平台的后端部分进行对接。本项目直接对接的部分为虚拟账户系统和清结算系统，对于前端的各种信息调取操作都需要与后端账户进行信息对接，因此预留出面向虚拟账户系统和清结算系统的信息查询的接口。

虚拟账户系统接口（李昕航）：完成用户登录的信息验证，用户开户和权限分配等管理操作以及交易信息流水查询。

清结算系统接口（段柯宇，尹蕾）：完成对账文件的解析与可视化，对账户清结算手续费进行汇总显示。

3.3 进度

项目整体按照以下部署进行：

1. 需求分析（4.16-4.24）：完成目标为需求分析文档编写结束。
2. 代码设计（4.25-4.29）：完成目标为软件设计书编写结束。
3. 编码实现（4.30-5.27）：需要资源为临时服务器及后端数据库，完成目标为网页各页面及基本功能编写完成。
4. 项目测试（5.28-6.3）：需要资源为后端虚拟账户系统和清结算系统，可用rabbitMQ代替，完成目标为项目基本功能可正常运行。
5. 项目部署对接及移交（6.4-6.10）：将金融服务平台的各部分进行整合部署，各部分进行完整对接并进行整体测试，无意外情况发生则移交项目投入使用。

3.4 关键问题

项目存在的主要难点与关键问题：

1. 网页页面的设计。由于综合运营平台涉及功能较多，页面关系较为复杂，相互连接与整体的风格设计都需要进行考虑，同时为解决页面重复加载和页面信息保存困难等问题，页面之间的信息传递和恢复都需要进行精细的设计。
2. 前后端对接与信息交互。综合运营平台只是前端页面，所有实例功能都需要调用后端接口进行实现，而前后端进行对接实现功能需要大量的信息交互，限定信息的种类与模式，保证信息在传输过程中的安全性和健壮性等都是项目的难点。
3. 页面信息安全。网页的页面含有大量信息，而对于金融服务平台来说信息安全至关重要，如何保证页面的信息不泄露，页面无法被恶意访问，确保页面信息安全，是项目投入使用需要解决的关键问题。