# 项目设计文档

## 文档修改历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 版本描述 | 版本号 |
| 张博 | 2020.03.20 |  | V1 |
| 翟翊翎 | 2020.04.10 | 增加4.2中的后端逻辑结构介绍 | V2 |

## 1、引言

### 1.1、编制目的

本报告详细完成对COIN系统的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

### 1.2、词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| COIN | 知识图谱可视化系统 | 无 |

### 1.3、参考资料

1.知识图谱可视化系统用例文档

2.知识图谱可视化需求规格说明文档

3.《软件工程与计算（卷三）团队与软件开发时间》

## 2、产品描述

参考知识图谱可视化系统用例文档和知识图谱可视化系统软件需求规格说明文档中对产品的概括，描述。

## 3、系统结构设计概述

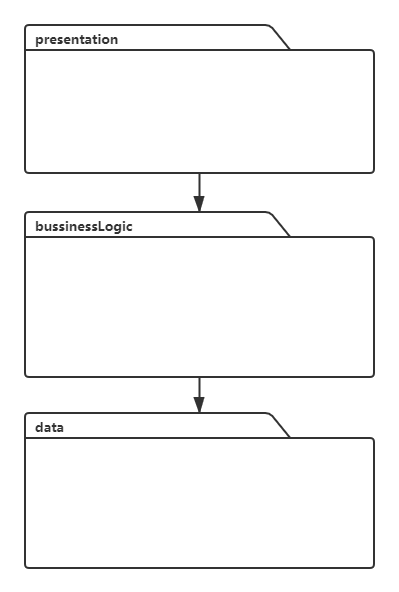
由于Web开发的特殊性，COIN系统的用户端不需要进行开发，只需用户自行安装浏览器即可，服务器端各层的职责如下：

|  |  |
| --- | --- |
| 层 | 职责 |
| 启动模块 | 负责启动系统 |
| 展示层 | COIN系统的客户端用户界面 |
| 接口模块 | 负责客户端和服务端的通信及数据传递 |
| 业务逻辑层 | 对用户界面的输入进行响应并执行业务处理 |
| 数据服务层 | 抽象出的数据操作接口 |
| 数据层 | 负责数据的持久化和访问 |

## 4、结构视角

### 4.1业务逻辑层的分解

COIN系统中，选择了MVC体系结构风格，但因为Web开发的特殊性，以及选用的开发技术（Spring Boot框架）在配置上的简化，使得在进行逻辑设计时，只需要将系统分为3层（展示层、业务逻辑层、数据层），即可示意整个高层抽象。展示层包含GUI页面的实现和用户行为的响应，业务逻辑层包含业务逻辑处理的实现，数据层负责数据的持久化和访问



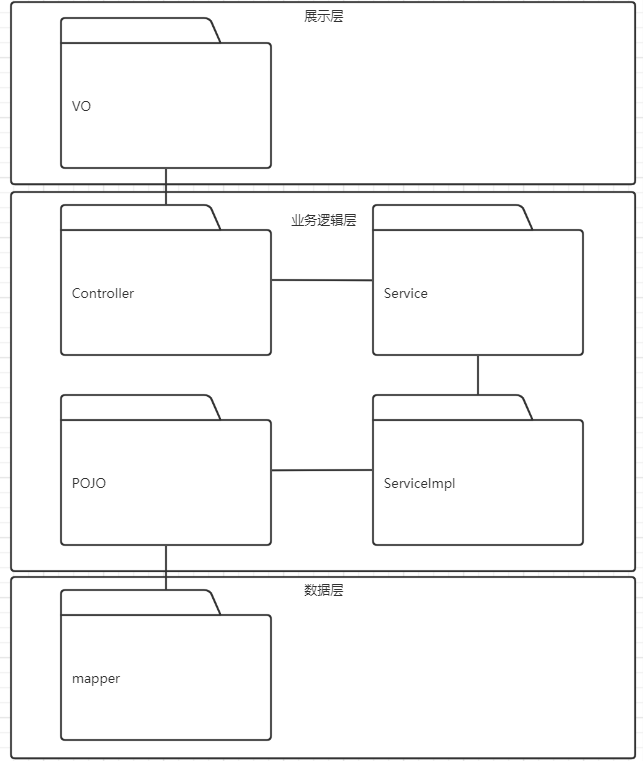
### 4.2各模块介绍

系统可以部署在以下三个物理层次：展示层、业务逻辑层、数据层。展示层用于用户访问系统，业务逻辑层用于业务控制和逻辑，数据层部署和存储系统中相关数据。

后端系统架构中的对象分为6类，主要涉及业务逻辑层和数据层：

1. controller：负责处理REST API请求
2. service：业务逻辑层接口
3. serviceImpl：业务逻辑层实现
4. pojo：简单java对象，用于数据层和业务逻辑层之间数据传递
5. vo：值对象，用于业务逻辑层和展示层（前端）数据传递
6. mapper：用于数据库持久化操作

系统中的组件和组件接口：



|  |  |  |  |
| --- | --- | --- | --- |
| 接口ID | 连接组件 | 接口信息 | |
| I1 | 连接VO和Controller | 语法 | Return(Response)  Interface(Request) |
| 前置条件 | 用户输入正确 |
| 后置条件 | 处理控制请求组件并响应 |
| I2 | 连接Controller和Service | 语法 | Return(result)  Interface() |
| 前置条件 | 无 |
| 后置条件 | 对应的Service执行业务逻辑 |
| I3 | 连接Service和POJO | 语法 | Return(dataset)  Interface(command) |
| 前置条件 | 无 |
| 后置条件 | 对应的POJO调用Mapper获取数据层数据并返回数据集 |

具体接口参见接口文档