3.1环境配置与技术

3.1.1概述

我们依据B/S架构,使用前后端完全分离的方式进行开发.我们使用了Spring Boot, Hibernate-jpa,vue等框架,使用了Java,Kotlin,JavaScript等开发语言,使用了git,maven,postman等辅助工具辅助开发,在Windows,Android双平台上实现了数据库互通的投顾系统.

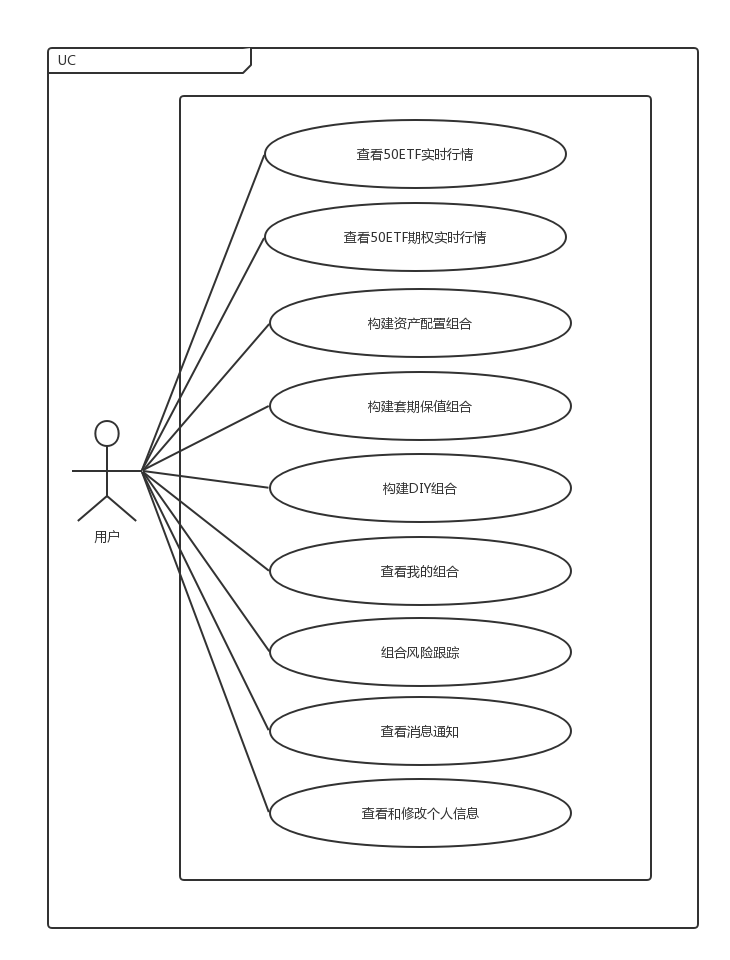
具体技术特性和优势等在此不做赘述,详情见维基百科.完整技术栈罗列如下.

3.1.2技术栈

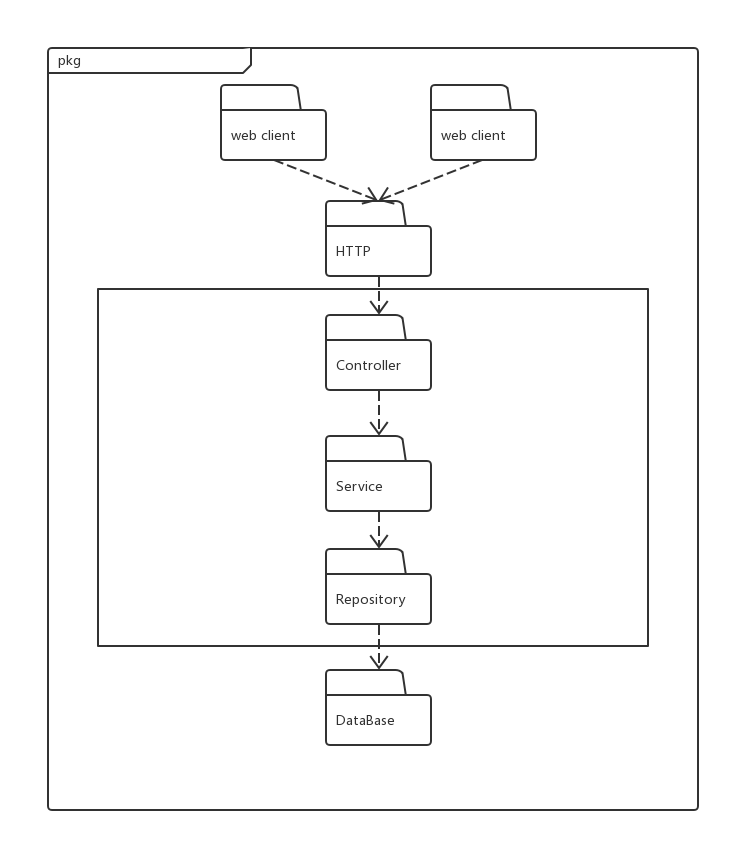
|  |  |
| --- | --- |
| 一、使用框架 | |
| Spring Boot | 服务器端框架 |
| Hibernate-jpa | 数据库框架 |
| Vue+Vue Router+Vuex | web端框架 |
| 二、环境搭建 | |
| Windows10 | 开发环境 |
| Linux | 服务器环境 |
| Android | 移动端平台 |
| Mysql | 数据库支持 |
| JRE8 | Java运行环境 |
| Chrome | Web端测试环境 |
| 三、编程语言 | |
| Java | 服务器端、安卓端开发语言 |
| JavaScript | Web端开发语言 |
| XML | 安卓端开发语言 |
| HTML | Web端开发语言 |
| 四、开发工具 | |
| IntelliJ Idea | Java开发IDE |
| Web Storm、VS Code | web端开发IDE |
| Android Studio | 安卓端开发IDE |
| Git | 版本控制工具 |
| Maven | Java项目依赖整合工具 |
| NPM | JS 包管理工具 |
| Postman | 接口测试工具 |

3.2 体系结构设计

3.2.1 用例视图

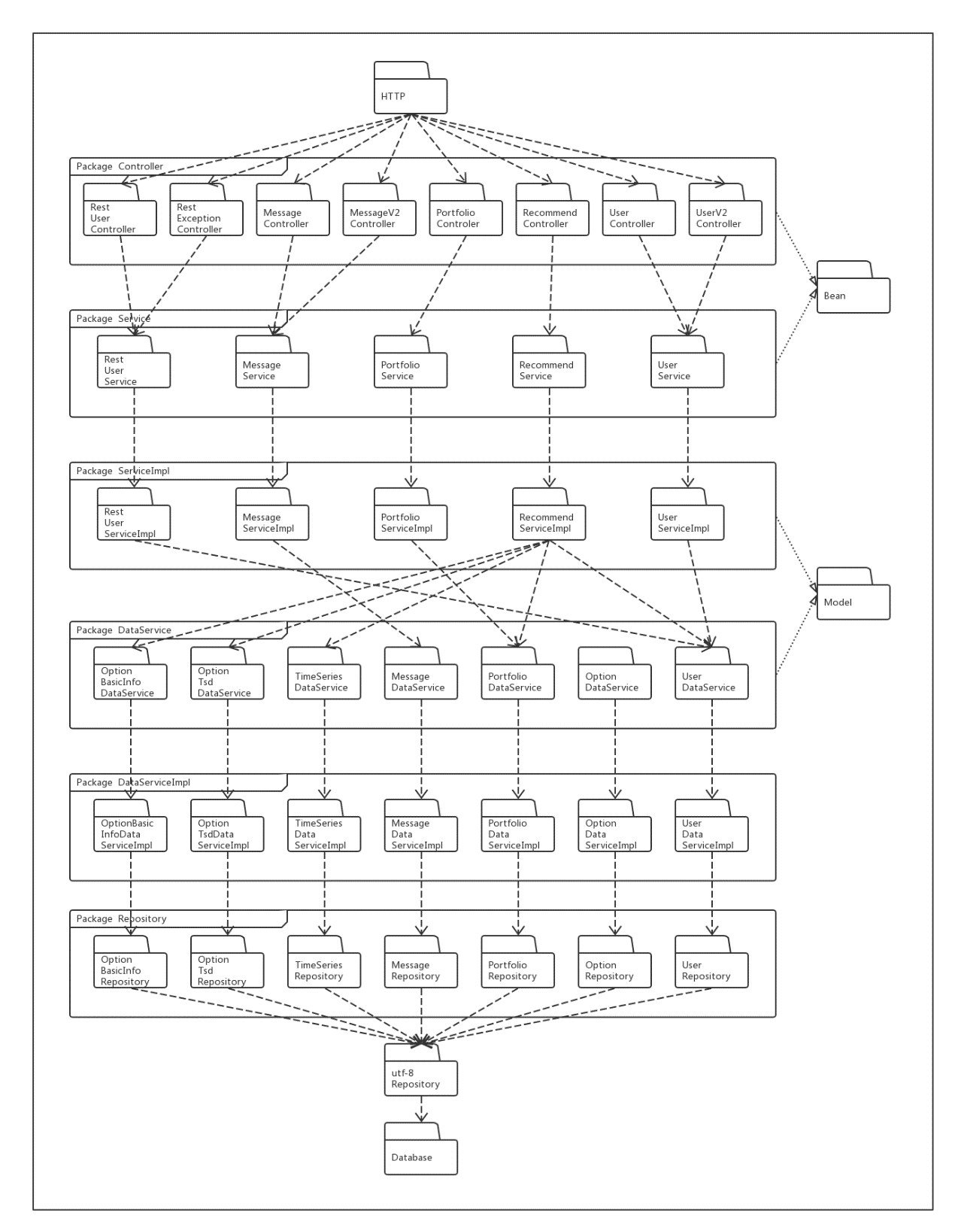


3.2.2 逻辑视图

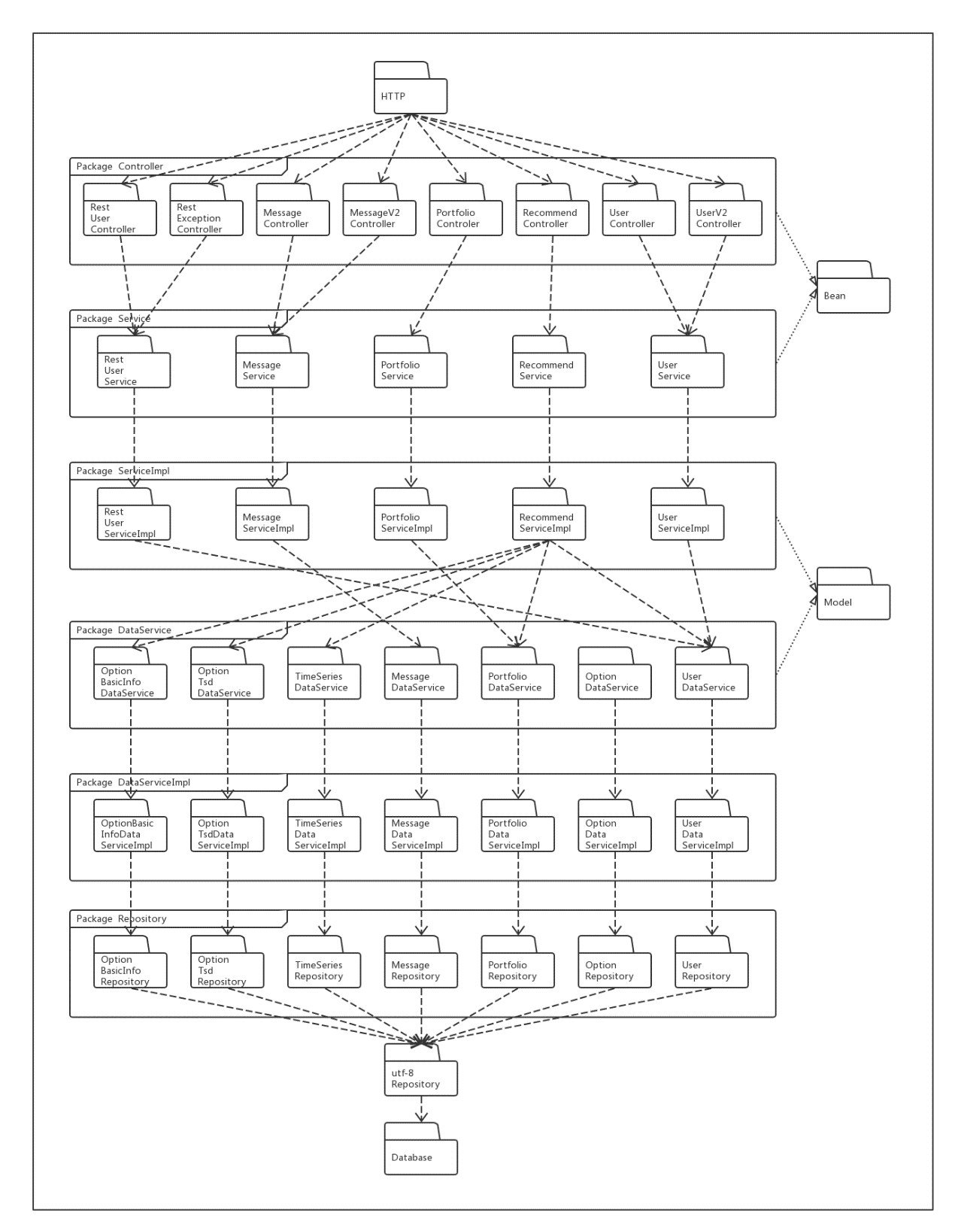


该系统采用了界面层-逻辑层-数据层的分层结构，客户端首先发送HTTP请求，服务器的控制端接受客户端发送的请求，调用相应模块的接口，各模块处理其自身的业务逻辑，从而实现功能需求，返回给用户。客户端与服务器端的数据以JSON的形式传输。逻辑层可以通过数据层的相应接口获取数据，此时，数据层将请求向下分配给各自的数据模块进行处理。三个层次相互配合共同使用，从而实现客户要所需功能需求及非功能需求。

3.2.3 开发视图



3.2.4 进程视图



3.2.5 部署视图

