**项目总结报告**

日期：2023年1月11日

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 第11组 | 项目名称 | 数据流动态分析平台 |
| 编程语言 | Java + JavaScript + HTML | 开发平台和框架 | IntelliJ IDEA + React + SpringBoot |

|  |  |
| --- | --- |
| **项目工作小结** | |
| **1.是否实现了项目立项时的所有需求？列出实现的新增需求和未实现的需求。**  **答：**  本次项目由于各种风险的存在，没有实现项目立项时的所有需求。  **新增需求**  由于缺少实际的流数据源，我们需要使用 Flink 来模拟流数据源信息的发送。  **未实现需求**  数据流动态分析平台的流程编排低代码化；数据流动态分析平台流程编排中的 Loop 与If-Else 操作。  **2.采用哪种架构风格？哪些设计模式？**  **答：**  **架构风格**  微服务的架构风格。项目通过将应用程序设计为一组松散耦合的协作服务，每个服务都实现了一部分的相关功能。服务之间使用HTTP/REST 等同步协议或异步协议进行通信，可以彼此独立地进行开发和部署。  **设计模式**  抽象工厂模式、模板模式以及职责链模式。在抽象工厂模式中，我们定义了一个接口用于创建算子对象，而无需明确指定具体类。抽象工厂将对象的实例化交给了子类。在模板模式中，我们首先定义一个算子的父类，并将某些步骤推迟到算子的子类中实现。在职责链模式中，多个算子依次处理数据源。形成一个链条。  **3.技术方案有哪些亮点？**  **答：**  我们技术方案的亮点，体现在使用代码生成技术，对数据流和算子进行抽象。数据流，实际上表现为字符串流，它是无结构的数据，对于后续的数据处理与编排来说，是非常难以使用的。  我们通过使用 freemaker 来实现字符串到 java class 的转化。freemaker 是一款模板引擎，多被用于生成 html 、电子邮件、配置文件等。首先，我们获取用户提供的数据流属性信息，通过使用 freemarker 模板引擎，读取 ftl 模板文件和 fm 配置文件， 解析并生成 java class。  **4.是否做了单元测试？是否做了系统功能测试？是否做了性能测试？是否做了兼容性等其他非功能测试？**  **答：**  项目进行了单元测试、系统功能测试以及性能测试。对于项目兼容性，数据流处理平台是WEB项目，通过浏览器进行访问，支持IE6、Edge、Chrome等浏览器。另外，我们也对部署进行了相关测试。项目的具体测试方法以及测试内容，详见测试计划文档。 | |
| **项目组成员对项目的贡献度（%）** | |
| Naomi（27%）：系统功能点分解、系统各模块架构设计、前端 UI 设计、前端界面实现  李梦瑶（27%）：后端业务逻辑实现、数据流算子抽象、数据流算子编排、数据库部署与搭建  佘智龙（27%）：后端代码框架搭建、后端业务逻辑实现、数据流算子编排、数据库部署与搭建  胡昊源（19%）：项目管理与安排、迭代计划与迭代审查、系统功能性测试、系统非功能性测试 | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 9563 |
| 复用他人代码行数： | 3271 |

|  |
| --- |
| **经验、教训和建议** |
| 本次软件工程项目开发工作，带给我们很多经验与教训。从技术上来说，我们积累了开发经验，拓展了技术栈，熟悉了软件工程的常见开发流程。从项目管理上来说，本次项目最大的教训，是我们需要安排合适的人去做其擅长的事情，在项目开发过程中，学习新的框架，使用新的技术，反而增加了项目的时间成本，从而导致对需求的各种实现不够完善与全面。 |

项目组各成员签字：胡昊源 Naomi 李梦瑶 佘智龙