**项目总结报告**

日期：2023.6.18

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 12 | 项目名称 | 交大点评 |
| 编程语言 | Js, css, html, java, MySQL | 开发平台和框架 | Intellij idea |

|  |
| --- |
| **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。  否，  由于进度原因，我们未能实现项目立项时的所有需求。  实现的新增需求：无。  未实现的需求：Order food点菜，我们未能实现学生通过交大点评进行线上点餐的功能；  Manage comments，我们未能实现管理员在社区对于用户评论进行管理的功能  Manage merchants info，我们未能实现管理员对于商家信息进行管理的功能  2.采用哪种架构风格？哪些设计模式？  逻辑视图采用分层架构风格，主要由三层组成，分别是Application层（应用层），Business Service层（领域中间件）和Middleware层（分布式中间件）。  部署视图采用客户端-服务器架构风格，描述了web 服务器，application server，database，客户端之间的联系  实现视图采用的是三层架构风格，分为表示层，业务逻辑层和数据访问层。  架构风格方面具体详见软件架构文档。  产品信息管理系统的创建和管理：工厂方法模式。  在该模式中，抽象产品类菜品信息类表示菜品对象，具有一些常见的操作，如获取描述和获取食谱等方法。抽象工厂类菜品上传系统表示工厂接口，用于创建产品对象。其子类上传描述，配料登记和上传口味信息各自实现了该工厂方法以创建不同的产品。  菜品推荐功能：策略模式。在该模式中，菜品推荐类表示上下文类，负责维护一个指向推荐策略对象的引用。而推荐策略则表示策略抽象类，定义了一个执行推荐方法用于实现具体的算法。其子类时下热门、等则分别实现了具体的算法。  评论和社交相关功能：观察者模式。在该模式中，Subject表示被观察者类，具有attach、detach和notifyObservers等方法，其中attach和detach用于添加和移除观察者对象，notifyObservers用于通知所有的观察者对象进行状态更新。而Observer则表示观察者类，定义了update方法用于接收被观察者通知并进行处理。  设计模式方面具体详见设计模式文档。  3.技术方案有哪些亮点？  1、使用了推荐匹配算法为用户提供个性化服务。  2、为不同用户生成不同的默认头像。  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？  完成了单元测试。  完成了系统功能测试。  软件测试部分具体详见各测试报告。 |
|  |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 姓名 | 需求 | 设计 | 编码 | 测试 | 项目管理\* | 小计 | | 彭奕涛 | 5 | 4.5 | 4.5 | 4.5 | 5.75 | 24.25 | | 陈一鸣 | 5 | 4.5 | 3 | 2.75 | 3.5 | 18.75 | | 雷宇昊 | 5 | 4.25 | 6 | 6.25 | 5.75 | 27.75 | | 朱亦腾 | 5 | 5.75 | 6.5 | 6.25 | 5.75 | 29.25 |   注： 项目管理包括项目计划、报告、沟通与协调等。 |

|  |  |
| --- | --- |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 3499 |
| 复用第三方代码行数： | 0 |
| 类的个数： |  |

|  |
| --- |
| **经验、教训和建议** |
| 问题、变更和返工：   1. 进度过慢，部分计划中的功能来不及实现。变更：暂缓或取消“社群”等功能的实现。 2. 数据量过小，通常的推荐算法难以正常应用。变更：由寻找合适的推荐算法改为参考网上的推荐算法自己实现符合实际的算法。 3. 小组成员因身体问题等因素无法按时完成计划内容。变更：调整计划，将部分复杂功能的实现延后。   经验与教训：   1. 数据库设计需要慎重考虑，一旦修改牵涉到的部分较多。 2. 定期线下沟通与开发能使效率，相比纯线上开发大大提高。 3. 多人共同开发应事先对接口、版本等进行协调，以免由于环境不同导致出错。 4. 计划中要预留出足够的时间来应对突发情况，包括组员身体因素等。 |

项目组各成员签字：彭奕涛 朱亦腾 雷宇昊 陈一鸣