**项目总结报告**

日期：2024年6月19日

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 7 | 项目名称 | 课表小助手 |
| 编程语言 | Dart | 开发平台和框架 | Android Studio & VSCode, Flutter, sqflite |

|  |
| --- |
| **项目工作小结** |
| 1.采用哪种架构风格？哪些设计模式？  架构风格：采用了Model-View-ViewModel (MVVM) 架构模式，将应用程序的逻辑层、视图层和数据层进行了明确的分离，提高了代码的可维护性和可测试性。  设计风格：在数据层采用单例模式，确保数据库实例唯一；实体类的设计采用工厂方法，课程类Course、任务类Task、事务类Matter都是抽象类Event的子类，EventWidget为抽象产品，其子类的各种视图为具体产品；在从教务处获取课表时采用适配器模式，将API返回的数据转换为客户端可以理解的CourseList对象格式，从而无缝对接。  2.技术方案有哪些亮点？  数据持久化：使用 SQLite 进行本地数据存储，即使应用关闭或设备重启，数据仍然保留，增强了用户体验；  状态管理：使用Provider进行全局项目管理，状态变化能自动更新UI，提升了代码的可维护性和可测试性；  WebView 集成：用户可以直接在应用内完成jAccount登录授权，而无需切换到浏览器，本地应用与外部系统无缝交互；  大量使用异步操作，避免阻塞主线程，在处理网络请求或数据库操作时，UI 仍然可以保持流畅，提升应用性能。  3.是否做了单元测试？是否做了系统的功能测试、性能测试、兼容性测试和易用性测试等？  后端完成了对所有重要类全面的单元测试，包括测试正常情况、边界情况以及成员函数结果的正确性，代码覆盖率也达到了90%以上，能够确保对象行为的正确性。  前端完成了“添加课程组件”的单元测试，包括输入合法情况、输入非法情况，并通过expect函数检查变量值是否符合预期，代码覆盖率达到95%以上。  4.是否采用大模型来辅助开发？对开发生产率约有百分之几的提升？  在项目开发过程中，采用了大模型ChatGPT-4和github copilot插件来辅助开发，在快速生成代码片段、优化代码结构、提供调试建议等方面提供了显著的帮助，使得开发生产率约提升了100%。 |
| **项目组成员对项目的贡献度（%）** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 姓名 | 需求 | 设计 | 编码 | 测试 | 项目管理\* | 小计 | | 张倍宜 | 20% | 35% | 45% | 35% | 40% | 35% | | 陈瑞涵 | 30% | 30% | 40% | 35% | 30% | 33% | | 顾一帆 | 50% | 35% | 15% | 30% | 30% | 32% | |  |  |  |  |  |  |  |   注： 项目管理包括项目计划、报告、沟通与协调等。 |

|  |  |
| --- | --- |
| **软件规模** | |
| 前端的代码行数（不包括注解行、空行和复用代码）： |  |
| 后端的代码行数（不包括注解行、空行和复用代码）： |  |

|  |
| --- |
| **经验、教训和建议** |
|  |

项目组各成员签字：