TodoList 应用项目管理材料

1. 项目概述

项目名称：TodoList 待办事项应用
项目类型：移动应用（Android）
技术栈：Flutter + GraphQL + Go
目标用户：需要管理个人任务和日程的用户
项目周期：2023年4月1日 - 2023年5月10日（约6周）
团队规模：2人

2. 项目组织结构

2.1 团队角色与职责

角色	职责	人员配置
全栈开发者 A	项目协调、前端开发、UI实现	1人
全栈开发者 B	后端开发、数据库设计、API开发	1人

注：由于团队规模小，每位成员需要承担多个角色的职责

2.2 沟通计划

沟通类型	频率	参与者	目的
进度同步会议	每周2次	全体成员	同步进度和解决问题
代码审查	按需	全体成员	确保代码质量
测试反馈	每周	全体成员	发现并修复问题

3. 项目计划

3.1 开发环境

- **开发环境**：Windows 10 操作系统
- **开发工具**：
  - Android Studio

    - Visual Studio Code
    - Google Android Emulator
- **测试设备**：华为P50 pro 鸿蒙4.2.0系统
- **版本控制**：Git

3.2 开发阶段

阶段	时间范围	主要工作	交付物
需求分析与设计	4月1日-4月5日	确定需求范围、设计系统架构和UI	需求文档、设计文档
基础开发	4月6日-4月15日	搭建项目框架、实现基本功能	基础代码框架
功能开发	4月16日-4月30日	开发核心功能、实现数据同步	功能模块代码
测试与优化	5月1日-5月8日	功能测试、性能优化、Bug修复	测试报告、优化后代码
项目交付	5月9日-5月10日	最终测试、文档整理、项目交付	完整应用和文档

3.3 迭代计划

总迭代次数：3次
每次迭代：约2周

迭代1（4月1日-4月15日）

- 需求分析与原型设计
- 搭建项目基础结构（前端和后端）
- 实现基本的待办事项增删改查功能
- 完成基础UI组件开发

迭代2（4月16日-4月30日）

- 实现待办事项分类功能
- 实现按状态、类别和时间筛选功能
- 实现本地数据存储
- 开发GraphQL API

迭代3（5月1日-5月10日）

- 实现数据同步机制
- 测试和修复Bug
- 性能优化
- 文档完善和项目交付

3.4 进度跟踪

进度跟踪通过Git仓库的commit历史进行详细记录，包括：

- 每日代码提交
- 功能完成标记
- Bug修复记录
- 版本迭代标签

4. 优先功能实现计划

优先级定义

- **P0**：核心功能，6周内必须实现
- **P1**：重要功能，时间允许时优先实现
- **P2**：次要功能，资源允许时考虑实现

功能优先级列表

ID	功能	优先级	预计完成日期
F-01	添加、删除待办事项	P0	4月10日
F-02	修改待办事项状态（完成/未完成）	P0	4月12日
F-03	为待办事项添加类别标签	P0	4月15日
F-04	为待办事项添加时间	P0	4月17日
F-05	为待办事项添加地点	P1	4月20日
F-06	按时间顺序查看未完成待办事项	P0	4月22日
F-07	按类别筛选未完成待办事项	P0	4月25日
F-08	筛选某一时间范围内的待办事项	P1	4月28日
F-09	查看全部已完成/未完成的待办事项	P0	4月30日
F-10	本地待办事项与云端同步	P1	5月5日

5. 技术实现计划

5.1 技术选型

层级	技术/框架	选择理由
前端框架	Flutter	跨平台开发，可以后续扩展到iOS
前端状态管理	Provider	轻量级状态管理，适合此规模应用
后端语言	Go	高性能，适合构建GraphQL API
API	GraphQL	灵活的查询，减少网络请求
数据库	SurrealDB	支持本地和云端部署
本地存储	SharedPreferences	适合存储用户设置和小量数据

5.2 简化架构设计

前端架构

- **UI层**：Flutter widgets
- **业务逻辑层**：Provider状态管理

- **数据层**：GraphQL客户端和本地存储

后端架构

- **API层**：GraphQL服务
- **业务逻辑层**：Go服务
- **数据存储层**：SurrealDB

6. 风险管理

6.1 主要风险

风险项	影响程度	应对策略
进度延误	高	优先实现核心功能，非核心功能可延后
技术难点	中	提前研究关键技术点，必要时调整技术方案
设备兼容问题	中	定期在实际设备上测试，及时解决兼容性问题
数据同步复杂性	高	简化同步机制设计，先确保基本功能可用

6.2 应急计划

- 如遇技术难题，可暂时采用替代方案确保基本功能可用
- 时间紧张时，可降低非核心功能的优先级
- 建立每日备份机制，确保代码安全

7. 交付计划

7.1 交付内容

- 完整的应用源代码
- 需求与设计文档
- 用户手册
- 项目管理材料

7.2 交付方式

- 通过Git仓库交付源代码

- 通过文档管理系统交付文档
- 提供APK安装包

8. 项目成功标准

- 所有P0级功能完成并通过测试
- 应用在华为测试手机上运行流畅
- 基本的数据同步功能可用
- 用户界面友好且符合设计要求