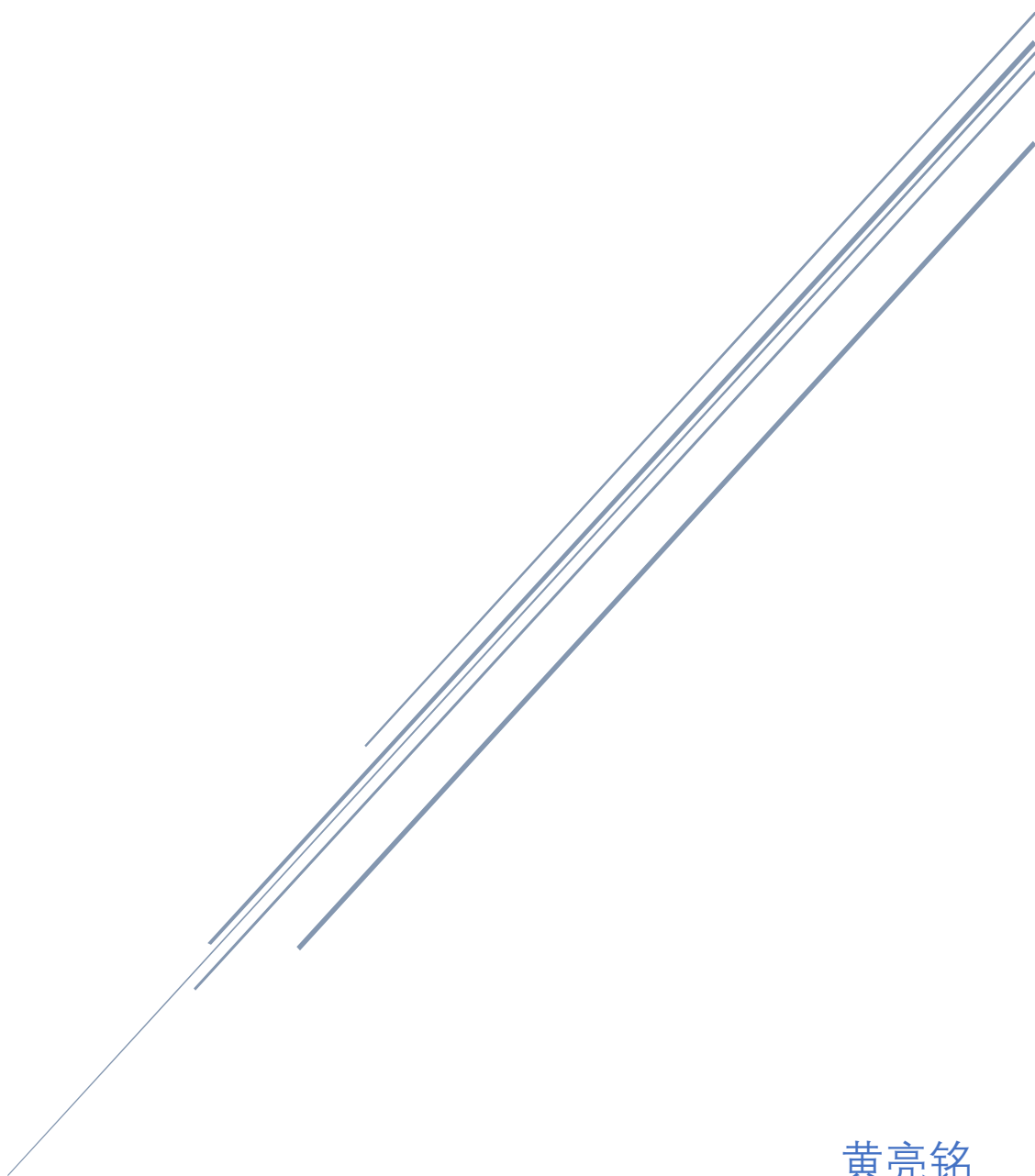


# 账易通

项目管理文档



黄亮铭

2022155028

## 目录

1 项目章程.....	2
1.1 项目基本信息.....	2
1.2 项目目标.....	2
1.3 项目范围.....	2
1.4 项目开发框架.....	3
2 项目资源管理.....	4
2.1 人力资源.....	4
2.1 物力资源.....	4
2.3 项目交付产品清单.....	5
3 项目进度管理.....	5
3.1 项目阶段.....	5
3.2 关键里程碑与交付物.....	6
4 项目风险管理.....	7
4.1 风险识别.....	7
4.2 具体风险分析.....	7
4.3 风险监控与控制.....	8
5 项目变更管理.....	8
5.1 历史变更记录.....	8
5.2 变更流程.....	8
6 项目完成情况总结.....	9
6.1 时间总结.....	9
6.2 成本总结.....	10
6.3 交付结果总结.....	10
7 项目总体总结.....	11
7.1 成功经验总结.....	11
7.2 不足教训总结.....	12

# 1 项目章程

## 1.1 项目基本信息

- 项目名称：账易通记账 APP
- 项目负责人：黄亮铭
- 项目基本分工：
- 项目类型：计算机软件开发

学号	姓名	负责内容
2022155028	黄亮铭	需求分析、功能模块设计、功能模块开发、系统原型设计、系统前后端对接、系统测试、整体规划、撰写及整理文档

图 1 项目成员与基本分工

- 项目计划开始日期：2024 年 11 月 23 日
- 项目实际开始日期：2024 年 11 月 30 日
- 项目计划结束日期：2024 年 12 月 23 日
- 项目实际结束日期：2024 年 12 月 23 日
- 项目预算：人民币 100 元

## 1.2 项目目标

本项目的目标是开发一款简洁无广告、功能全面和操作便捷的记账小程序，为用户提供一种高效管理日常收支的工具，帮助用户轻松管理个人财务，实现收支记录、预算设置、财务分析等功能。

## 1.3 项目范围

项目范围说明是项目管理中的一个重要说明，用于明确项目的目标、交付物、相关方期望以及项目执行的范围。

图 2 是我们团队的项目范围。

范围条目	详细描述
用户沟通与管理	与用户进行定期沟通，获取反馈并及时调整项目方向。确保项目的目标与用户的期望保持一致，最大程度满足用户需求。
用户界面与体验设计	确保用户在系统中获得清爽、直观、完整、智能的任务系统体验。

最小化轻量设计	以最小化的设计理念，简化用户界面和功能，提供直观、高效的用户体验。
系统设计与开发	使用 Gin 框架、Gorm 框架和 Uniapp 进行设计、开发项目的核心功能，包括增删改查收支记录和收支记录可视化。
上线部署与维护	制定上线部署计划，确保系统平稳上线并向用户提供服务。提供系统的后续维护和支持，及时处理可能出现 bug 和用户反馈。
风险管理	进行风险评估和管理，识别可能影响项目进展的风险，并采取措施降低风险发生的可能性和影响。
项目文档管理	管理并更新项目文档，确保项目的信息和进展得以记录和传递。

图 2 项目范围

## 1.4 项目开发框架

本项目采用增量开发模型，将系统功能逐步分解为多个增量进行开发。在每个增量中，先进行需求分析和设计，然后进行开发、测试，最后将可执行的增量交付给用户使用（图 3）。

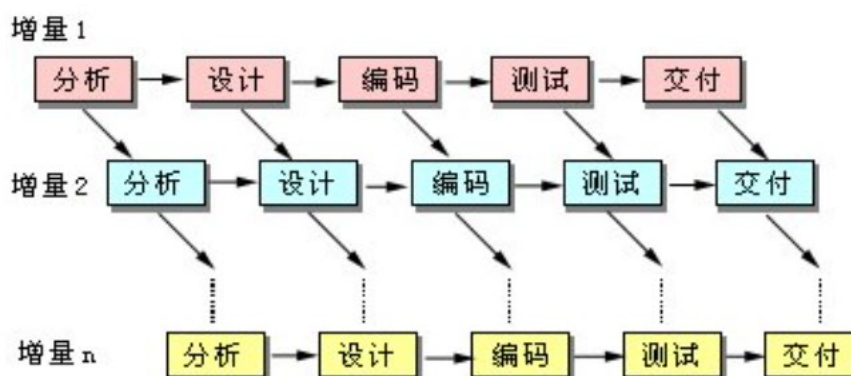


图 3 增量开发模型核心

用户在使用过程中提出反馈，开发团队根据反馈对后续增量进行调整和优化。这种开发模型的优点如下：

- ✓ **快速交付价值：**每个增量都能为用户提供部分可用功能，使项目价值能够尽早体现，增强用户信心。
- ✓ **及时响应需求变更：**能够根据用户反馈及时调整后续增量的开发内容，更好地满足用户需求。
- ✓ **降低风险：**将项目分解为多个小增量，降低了开发过程中的技术风险和需求变更风险。

## 2 项目资源管理

### 2.1 人力资源

人力资源是项目的重要资源之一。一个项目的成功，必须拥有一支具备相应技能、经验和专业知识的团队，他们能够高效地完成项目中的各个任务。而且，除了人员的专业素质之外，还必须关注团队成员的数量、工作分配、组织结构以及培训需求等方面，这些都是确保团队协作顺畅和执行力的关键因素。

项目组成员以及技能如图 4 所示。

职责	要求
项目负责人	要求有项目管理经验、团队组织能力、掌握沟通技巧、熟悉风险管理
前端开发人员	熟悉 Uniapp 框架、Vue.js、HTML、CSS、JavaScript 等前端技术，具备良好的用户界面设计和交互设计能力。
后端开发人员	掌握 Go 语言、Gin 框架、Gorm 框架、数据库（MySQL）操作、服务器端技术（如 Linux 环境配置、Nginx 部署等），了解数据结构和算法。
测试人员	熟悉测试方法和工具（如 Postman、Jmeter 和 等），掌握自动化测试技术，具备缺陷跟踪和分析能力。

图 4 职责与要求

根据职责要求和项目组成员已掌握的技术，本项目的分工如下所示。

- **项目负责人：**黄亮铭（技能：项目管理经验、团队领导能力、沟通技巧、风险管理）
- **前端开发人员：**黄亮铭（技能：熟悉 Uniapp，具备良好的用户界面设计和交互设计能力）
- **后端开发人员：**黄亮铭（技能：掌握 Go 语言、Gin 框架、Gorm 框架，熟悉数据库）
- **测试人员：**黄亮铭（技能：掌握常用测试工具 PostMan）

### 2.1 物力资源

- **开发设备：**为开发人员配备性能良好的个人计算机，满足开发工具运行和代码编写需求。
- **软件工具**
  - **前端开发工具：**HBuilderX，集成了编译器等功能，方便前端代码的编写、调试和打包。
  - **后端开发工具：**GoLand，用于 Go 语言项目的开发，提供代码编辑、编译、调试等功能。

- 数据库管理工具：MySQL Workbench，用于数据库的设计、管理和数据操作。
- 测试工具：Postman，用于后端 API 接口的测试，模拟 HTTP 请求并验证响应结果。
- **服务器资源**：根据项目上线后的预估用户量和数据量，选择合适的云服务器提供商，租赁服务器资源，用于部署项目后端服务和数据库。

## 2.3 项目交付产品清单

- **系统实现**：账易通记账小程序的源代码，包括前端和后端代码。
- **文档资料**
  - 项目需求文档：详细记录用户需求、功能规格和业务流程。
  - 系统原型文档：在设计初期，按照第一部分提交的用户需求提供原型报告，并详细准确地描述系统原型。
  - 系统设计文档：包含系统架构设计、数据库设计、模块设计等内容。
  - 系统测试文档：包括测试计划、测试用例、测试报告等，记录测试过程和结果。
  - 系统说明文档：为用户提供系统使用指南，包括功能介绍、操作步骤、常见问题解答等。
  - 项目管理文档：记录项目计划、进度、资源管理、风险管理、变更管理等过程文档。

## 3 项目进度管理

项目进度管理是确保项目按计划进行的关键活动之一。它涉及到计划的创建、执行的监控、进度的测量和必要时的调整。通过以下的进度管理计划，可以有效地监控项目的进展，及时发现和解决问题，确保项目按计划顺利进行。

### 3.1 项目阶段

项目规划是项目管理的基础，它包括项目的启动、执行和收尾等各个阶段的详细计划。在项目计划中，需要明确每个阶段的任务、负责人、起止时间和前置条件等信息。图 5 展示了本项目各个阶段的任务、负责人、时间、前置条件等，以帮助成员更好地管理项目的进度。

阶段	时间	主要任务	负责人	前置条件
需求分析阶段（第 1 周）	2024/11/23 ~2024/11/30	完成市场调研，分析同类记账应用的优缺点，收集用户需求和期望。编写需求规格	黄亮铭	无

		说明书，明确系统功能、性能、界面等要求，确定项目范围。		
设计阶段 (第 1 周)	2024/11/26 ~2024/11/ 30	进行系统架构设计，确定前端、后端和数据库的技术选型和架构模式。完成数据库设计，绘制 E - R 图，设计表结构、字段和关系，制定数据库操作规范。设计用户界面，制作系统原型，与用户沟通确认界面布局和交互流程，输出原型设计文档。	黄亮铭	需求分析 完成
开发阶段 (第 2 周、 第 3 周)	2024/12/01 ~2024/12/ 15	前端开发人员根据设计文档进行页面开发，实现功能模块和交互效果。后端开发人员搭建开发环境，开发 API 接口，实现业务逻辑。	黄亮铭	系统设计 完成
测试阶段 (第 4 周)	2024/12/16 ~2024/12/ 23	测试人员制定测试计划，编写测试用例，包括功能测试、性能测试、兼容性测试、安全测试等。执行测试用例，记录测试结果，提交缺陷报告，跟踪缺陷修复情况，进行回归测试。对测试结果进行分析总结，评估系统质量，确保系统满足上线要求。	黄亮铭	系统开发 完成
维护阶段	持续进行	监控系统运行状态，及时处理故障和问题，定期进行数据备份和恢复。根据用户反馈和业务需求变化，制定更新计划，进行系统升级和优化，发布新版本	黄亮铭	系统测试 完成

图 5 项目阶段详情

3.2 关键里程碑与交付物

设置关键里程碑和交付物，以便更好地监控项目进度和质量。关键里程碑和交付物将在项目的不同阶段起到关键作用，帮助团队监控项目的进展，确保按计划推进。本项目的关键里程碑和交付物如图 6 所示。

关键里程碑	交付物
需求分析阶段结束	项目章程、需求规格说明书
设计阶段结束	系统原型报告、系统设计报告
开发阶段结束	系统源代码、实现报告、系统说明书
测试阶段结束	模块测试通过报告

图 6 项目关键历程与交付物

## 4 项目风险管理

### 4.1 风险识别

在项目执行过程中，可能会面临一些潜在的风险，为了及时应对这些风险，需要在项目初期进行全面的风险识别。本项目为小型项目，潜在的风险比较少，主要有以下风险（图 7）。

风险类型	风险描述
技术风险	Uniapp 框架的某些功能可能存在兼容性问题，影响程序在不同设备上的正常运行。
时间风险	需求变更频繁，导致开发进度延迟。项目组成员变动，导致项目某一方向开发进度延迟。
竞争风险	市场上可能出现类似功能且具有竞争力的记账应用，吸引用户，降低本项目产品的市场份额。
安全风险	程序可能存在安全漏洞，被黑客利用，导致用户信息被窃取或系统被破坏。

图 7 项目风险类型

### 4.2 具体风险分析

对于每个风险，本团队进行详细的分析，包括可能的影响、概率和紧急程度（图 8）。同时，本团队针对每一个风险都提出了应对方案（图 8）。

风险类型	影响	概率	紧急程度	应对策略
技术风险	功能无法正常使用，影响用户体验，导致用户流失；增加开发成本和时间。	中	高	在开发前进行技术调研和预研，对关键技术进行验证；建立技术问题解决机制，及时请教专家或参考开源项目经验。
时间风险	项目交付延迟，错过市场最佳时机，增加成本。	高	高	在项目计划中预留一定时间应对需求变更，严格控制需求变更流程，确保变更经过评估和审批。成员变更时项目组负责人负责监督交接，确保交接顺利。
竞争风险	市场份额下降，收益减少。	中	中	持续关注市场动态，分析竞争对手产品特点，突出本项目产品的优势，如简洁性、个性化分析等。



风险类型	影响	概率	紧急程度	应对策略
安全风险	用户数据泄露，损害用户利益和公司声誉，面临法律风险。	低	高	定期进行安全审计和漏洞扫描，及时修复发现的安全问题；加强系统访问控制，设置用户权限，防止非法访。

图 8 具体风险分析

4.3 风险监控与控制

本团队建立了比较完善的风险控制机制，在每周的项目会议上，团队成员均需汇报自己负责模块的风险状态，包括已识别风险的变化情况和新风险的发现。此外，本团队也会定期对系统进行性能监测、安全扫描等，及时发现潜在的风险。

本团队也建立了相对完善的风险应对措施。根据风险监控结果，评估已实施应对措施的有效性，对效果不佳的措施及时调整或更换。针对新出现的风险或风险变化情况，制定新的应对策略，确保风险始终处于可控范围内。

5 项目变更管理

5.1 历史变更记录

本项目的变更主要发生在需求部分。在人员方面，项目维持了原班人马，没有人员变更情况。本项目的需求变更情况如图 9 所示。

变更编号	变更日期	变更描述	变更影响	变更状态
CR-001	2024/11/25	团队决定删减预算设置部分的需求	该模块尚未开始设计和开发，无变更影响	已变更
CR-002	2024/12/6	客户希望记录可视化界面的饼状图改为柱状图。	模块已经开始开发，需要重新进行设计，影响了项目的整体进度	已变更

图 9 项目需求变更情况

5.2 变更流程

本团队建立了细致的变更流程，对变更造成的影响进行全方位的评估，确保变更的负面影响在团队的可控范围。

本项目的变更流程如下所示。

- 1) **提出变更请求：**项目团队成员或客户发现需要变更时，填写变更申请表，详细说明变更内容、原因、影响等信息
- 2) **变更评估：**项目负责人组织相关人员（如开发人员、测试人员、业务专家等）对变更请求进行评估，分析变更对项目范围、进度、成本、质量等方面的影响。
- 3) **变更审批：**根据评估结果，由项目负责人决定是否批准变更。对于重大变更，需告知整个项目团队，整个项目团队根据项目整体情况权衡利弊后做出决策。
- 4) **变更实施：**批准的变更由相应的开发人员或团队成员负责实施，确保变更的正确执行。在实施过程中，要做好版本控制和文档更新工作。
- 5) **变更验证：**测试人员对变更实施后的结果进行测试，验证变更是否达到预期效果，是否引入新的问题。如发现问题，及时反馈给开发人员进行修复。
- 6) **变更关闭：**当变更通过验证，确认无误后，将变更相关信息记录归档，关闭变更流程。同时更新项目计划、需求文档、设计文档等相关文件，确保项目文档与实际情况一致。

## 6 项目完成情况总结

### 6.1 时间总结

在项目进行的过程中，时间总结是对项目各个阶段计划和实际完成情况的对比，以便评估项目的时间管理效果（图 10）。

项目阶段	开始日期	计划完成日期	实际完成日期	完成情况
需求分析阶段	2024/11/23	2024/11/25	2024/11/26	延期完成，需求规格说明书得到团队和相关方认可。
设计阶段	2024/11/26	2024/11/30	2024/11/29	提前完成，系统架构、数据库设计和原型设计符合预期。
开发阶段	2024/12/01	2024/12/15	2024/12/18	基本按时完成，部分功能开发因技术难题稍有延迟，但通过加班赶工确保了关键功能按时交付。
测试阶段	2024/12/16	2024/12/23	2024/12/21	提前完成，通过多种测试方法，发现并修复了大部分缺陷，系统达到上线标准

续表

项目阶段	开始日期	计划完成日期	实际完成日期	完成情况
维护阶段	2024/12/24	无	无	已上线，运行稳定，根据用户反馈及时进行了优化和维护。

图 10 时间总结

6.2 成本总结

在项目进行的过程中，成本总结是对项目投入与产出的对比，有助于评估项目的经济效益（图 11）。

项目阶段	成本类型	预算（元）	实际花费（元）	成本情况
需求分析阶段	人力成本	30	6	低于预算，主要原因是需求分析过程较为顺利，团队成员效率较高。
设计阶段	人力成本	15	3	低于预算，设计阶段未出现重大问题，设计工作按计划完成。
开发阶段	人力成本	30	3	低于预算，虽然开始时间延长，但是仍然在预估的成本之内。
测试阶段	人力成本	5	3	低于预算，测试计划合理，测试过程顺利，未出现大量反复测试情况。
维护阶段	服务器租赁、人力成本	20	0	低于预算，选择了性价比高的云服务器提供商，按流量费用计费
总计	-	100	24	低于预算，项目在成本控制方面表现较好。

图 11 成本总结

6.3 交付结果总结

交付结果总结是对项目最终产出的成果进行评估，确保它们满足项目目标和相关方的期望。图 12 是交付结果总结表格。

项目成果	交付内容	完成情况	评估指标
记账 APP	可运行的应用、源代码	已完成	<b>功能完整性</b> ：实现了收支记录、预算设置、数据统计分析等核心功能，满足用户基本记账需求。 <b>用户体验</b> ：界面简洁直观，操作流畅，用户反馈良好。 <b>性能稳定性</b> ：在不同设备和网络环境下运行稳定，响应速度快。
项目文档	需求分析文档、设计文档、实现文档、测试文档、系统使用说明书	已完成	<b>文档完整性</b> ：涵盖项目各个阶段的关键信息，对系统功能、架构、测试过程等有详细记录。 <b>准确性</b> ：文档内容与实际系统一致，准确反映系统设计和实现。 <b>清晰度</b> ：文档结构清晰，易于理解和查阅，便于后续维护和升级。

图 12 交互结果总结

## 7 项目总体总结

通过对本项目的工程经验经验和不足教训的总结, 项目团队将在未来的项目中不断改进工作方法和流程, 提高项目管理能力和产品质量, 为用户提供更优质的服务。同时, 也为团队成员的个人成长和发展提供有益的参考, 促进团队整体能力的提升。

### 7.1 成功经验总结

- 在项目前期, 通过深入的市场调研和用户需求分析, 准确把握用户痛点和期望, 制定了详细且明确的需求规格说明书。在项目过程中, 严格控制需求变更, 确保所有变更都经过评估和审批, 避免了需求的无序蔓延, 保证了项目开发方向的稳定性。
- 基于项目特点和需求, 选择了 Uniapp、Go 和 MySQL 的技术组合。Uniapp 的跨平台特性大大提高了开发效率, 减少了开发成本; Go 语言的高性能和简洁性适合处理后端业务逻辑, 确保系统的稳定和快速响应; MySQL 作为成熟的关系型数据库, 能够可靠地存储和管理财务数据。同时, 采用 MVC 架构模式, 使系统结构清晰, 各层职责明确, 便于开发、维护和扩展。
- 建立了定期的项目会议制度, 包括每日站会、每周周会等, 确保团队成员及时沟通项目进展、问题和解决方案。
- 测试人员在项目中发挥了重要作用, 从单元测试、集成测试到系统测试, 全面覆盖了系统的各个功能和业务场景。

## 7.2 不足教训总结

- 在测试过程中，虽然进行了多种类型的测试，但仍有一些复杂业务场景和边界情况未被充分测试到，导致上线后出现了一些小问题。未来应制定更完善的测试计划，增加测试用例的多样性和全面性，特别是针对关键业务流程和高风险模块，加强测试力度，提高测试覆盖度。
- 在项目开发后期，发现部分功能在高并发情况下性能表现不佳，需要进行紧急优化。这表明在项目前期和开发过程中，对性能优化的重视程度不够。后续项目应在系统设计阶段就考虑性能因素，如数据库索引优化、缓存策略制定等，并在开发过程中定期进行性能测试和优化，确保系统性能满足实际业务需求。