软 件 工 程 导 论

**实**

**验**

**报**

**告**

**题目：**

**学 院**

**年 级**

**组 员**

**年 月 日**

**[1.引言 1](#_Toc116050378)**

[1.1编写目的 1](#_Toc116050379)

[1.2项目背景 1](#_Toc116050380)

[1.3术语说明（必填） 1](#_Toc116050381)

[1.4参考资料 1](#_Toc116050382)

**[2.任务概述](#_Toc116050383)** [2](#_Toc116050383)

[2.1待开发软件的一般描述 2](#_Toc116050384)

[2.2用户特征 2](#_Toc116050385)

[2.3运行环境 2](#_Toc116050386)

**[3.数据描述](#_Toc116050387)** [2](#_Toc116050387)

[3.1静态数据 2](#_Toc116050388)

[3.2动态数据 2](#_Toc116050389)

[3.3内部生成数据 2](#_Toc116050390)

[3.4业务功能模块数据流图（必填） 3](#_Toc116050391)

[3.5数据词典（必填） 3](#_Toc116050392)

**[4.功能需求（必填）](#_Toc116050393)** [3](#_Toc116050393)

**[5.性能需求（必填）](#_Toc116050394)** [3](#_Toc116050394)

[5.1并发性 3](#_Toc116050395)

[5.2容错要求 3](#_Toc116050396)

[5.3时间特性 3](#_Toc116050397)

[5.4适应性 3](#_Toc116050398)

**[6.软件属性需求 4](#_Toc116050399)**

[6.1正确性 4](#_Toc116050400)

[6.2可靠性 4](#_Toc116050401)

[6.3效率 4](#_Toc116050402)

[6.4完整性 4](#_Toc116050403)

[6.5易使用性 4](#_Toc116050404)

[6.6可维护性 4](#_Toc116050405)

[6.7可测试性 4](#_Toc116050406)

[6.8复用性 4](#_Toc116050407)

[6.9安全保密性 5](#_Toc116050408)

[6.10可理解性 5](#_Toc116050409)

[6.11可移植性 5](#_Toc116050410)

[6.12互联性 5](#_Toc116050411)

**[7.其他需求（非必须） 5](#_Toc116050412)**

**[8.附录 5](#_Toc116050413)**

[8.1尚未解决的问题（可选） 5](#_Toc116050414)

[8.2注解（可选） 5](#_Toc116050415)

**需求分析文档**

需求分析也称为软件需求分析、系统需求分析或需求分析工程等，是开发人员经过深入细致的调研和分析，准确理解用户和项目的功能、性能、可靠性等具体要求，将用户非形式的需求表述转化为完整的需求定义，从而确定系统必须做什么的过程。

# 1.引言

## 1.1编写目的

1.阐明开发本软件的目的

2.说明编写本软件说明书的目的

3.指明软件需求说明书所预期的读者

## 1.2项目背景

## 1.3术语说明（必填）

|  |  |  |
| --- | --- | --- |
| **序号** | **术语** | **所指对象或含义** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 1.4参考资料

列出本文档引用的所参考的资料，文档的编号,标题、版本号和发行日期、出版单位或资料来源。

# 2.任务概述

## 2.1待开发软件的一般描述

描述项目开发意图、所应达到的目标、作用范围以及市场前景等。

## 2.2用户特征

描述最终用户应具有的受教育水平、工作经验及技术专长。

## 2.3运行环境

描述软件的运行环境，包括硬件平台、硬件要求、操作系统和版本，以及其他的软件或与其共存的应用程序等。

# 3.数据描述

## 3.1静态数据

|  |  |  |
| --- | --- | --- |
| 数据名称 | 数据类型 | 数据说明 |
| 用户名 | 字符串 | 用户注册时使用的唯一标识符。 |
| 密码 | 字符串 | 用户注册时设置的安全凭证。 |
| 收支类别 | 枚举 | 包括收入和支出两种类型。 |
| 金额 | 浮点数 | 记录每笔交易的具体数额。 |
| 日期 | 日期 | 每笔交易发生的具体日期。 |
| 预算 | 浮点数 | 特定时间段的预算目标 |

## 3.2动态数据

|  |  |  |
| --- | --- | --- |
| 数据名称 | 数据类型 | 数据说明 |
| 总资产 | 浮点数 | 用户当前拥有的资金总额。 |
| 总负债 | 浮点数 | 用户当前拥有的负债总额。 |
| 净资产 | 浮点数 | 用户当前拥有的净资产总额。 |
| 各类收入 | 浮点数 | 用户的各类收入数据 |
| 各类支出 | 浮点数 | 用户的各类支出数据 |

## 3.3内部生成数据

|  |  |  |
| --- | --- | --- |
| 数据名称 | 数据类型 | 数据说明 |
| 总资产 | 数值型 | 计算用户的所有资产总额 |
| 总负债 | 数值型 | 计算用户的全部负债总额 |
| 净资产 | 数值型 | 资产减去负债得到的净值 |
| 收支差额 | 数值型 | 某一时间段内收入与支出的差额 |

## 3.4业务功能模块数据流图（必填）

列举出所开发的软件系统实现的业务功能模块的数据流图，可采用判定树和判定表对数据流图中数据加工进行说明。

顶层数据流图：

导出报表

注册

用户

用户

个人理财管理系统

## 

用户信息

用户账号信息

D1

第0层数据流图

1

信息输入

注册用户账号

用户

用户

用户财务情况

D3

输入登录信息

用户财务情况

导出报表

3

2

录入财务信息

登录用户账号

本地文件

D5

6

设定预算

保存数据

D2

登录信息

登录信息

设定预算

全部信息

全部信息

查看信息

5

查看用户信息

D4

保存到本地文件

4

D5

D1

用户账号信息

本地文件

第一层数据流图

用户信息

第0层数据流图

1.1

输出

信息输入

1

输出错误信息

注册用户账号

用户

D3

输入登录信息

用户

用户财务情况

用户财务情况

2.1

导出报表

D5

输出

3

2

D5

输出错误信息

登录用户账号

录入财务信息

登录信息

登录信息

本地文件

D5

6

设定预算

保存数据

D2

设定预算

5

查看信息

4

D4

全部信息

保存到本地文件

查看用户信息

D5

本地文件

## 3.5数据词典（必填）

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 定义 | 类型 | 备注 |
| 用户名 | 用户用于登录的身份标识符 | 字符串 | 唯一 |
| 密码 | 用户的安全凭证 | 字符串 | 不对外公开 |
| 收入 | 用户获得的资金流入 | 数值型 | 正数 |
| 支出 | 用户的资金流出 | 数值型 | 正数 |
| 日期 | 发生交易的具体日期 | 日期型 | 格式统一 |
| 备注 | 对交易的额外说明 | 字符串 | 可选 |
| 总资产 | 用户拥有的所有财产价值之和 | 数值型 | 包括现金、存款、投资等 |
| 总负债 | 用户欠他人的债务总额 | 数值型 | 包括贷款、信用卡账单等 |
| 净资产 | 用户的实际财富 | 数值型 | 总资产减去总负债 |
| 预算 | 用户计划的未来一段时间内的资金安排 | 数值型 | 包括收入预算和支出预算 |
| 报表 | 用户财务数据的整理和呈现形式 | 文件 | Excel或CSV格式 |

# 4.功能需求（必填）

1. 系统详细描述

个人理财管理系统时在帮助用户在本地环境中管理其财务数据，并进行合理的资产分配和消费控制。该系统不依赖网络，所有数据均存储在本地文件中，以确保用户的财务信息安全可控。

该系统能够为用户提供一个集成的财务管理平台，帮助用户记录日常的收入和支出情况，跟踪资产负债状况，并为未来的支出设定预算目标。用户可以通过该系统便捷地查看个人财务的历史数据及趋势，以支持他们做出更为理性的财务决策。

系统还具备任务提醒功能，例如对未完成的收支记录进行提示，帮助用户确保财务数据的完整性。总体而言，本系统时在通过直观易用的界面，为用户提供便捷、高效的个人财务管理服务。

1. 系统功能

用户通过系统可以执行的操作包括：注册个人账户、登录系统、录入并保存收支数据、查看和跟踪资产负债概况、设定预算目标、导出财务统计报表。记录的数据包括：收入与支出类别、金额、日期、备注、资产负债信息、预算目标等。

* 1. 系统功能时序

当用户首次使用系统时，用户需要注册个人账户。成功注册后，用户可以通过登录功能进入系统，并开始管理其财务数据。

时序流程如下：

1. 注册：用户输入用户名、密码等信息，完成账户注册。

2. 登录：用户使用注册信息登录系统。

3. 录入财务数据：用户登录后，录入日常的收入与支出信息，包括金额、类别、日期等。

4. 保存数据：用户完成数据录入后，系统将财务数据保存到本地文件中。

5. 查看资产概况：用户可以查看个人的资产、负债以及整体的财务概况。

6. 设定预算：用户可以为未来的支出设定预算目标，系统会跟踪实际收支情况与预算的差距。

7. 导出报表：用户可以选择导出财务数据，生成Excel或CSV格式的报表。

输入 → 注册 → 登录 → 录入财务数据 → 保存 → 查看 → 设定预算 → 导出报表

* 1. 功能要求

功能-1: 用户注册

详细要求：

1. 用户注册时需要输入用户名和密码。

2. 系统自动验证用户名的唯一性，确保不存在重复用户名。

3. 密码必须符合复杂性要求，包括至少8个字符，且包含字母和数字。

4. 用户输入的信息验证通过后，系统生成用户账户并将数据保存至本地文件。

5. 系统需提供注册成功的反馈信息，若注册失败，则提示用户失败原因（如用户名已存在、密码不符合要求等）。

6. 用户名与密码的存储需采用安全的本地文件存储方式，便于后续登录的使用。

7. 由于系统为全程断网环境，所有用户数据的验证和存储均在本地进行，不依赖任何外部服务或网络连接。

功能-2: 用户登录

详细要求：

1. 用户需要输入已注册的用户名和密码进行登录。

2. 系统在本地验证用户输入的用户名和密码是否匹配。

3. 如果用户名或密码不正确，系统需提供相应的错误提示信息，并允许用户重新输入。

4. 若登录成功，系统需提供登录成功的反馈信息，并进入主界面以供用户执行其他操作。

5. 登录过程需在本地进行，不依赖网络连接，所有数据验证操作均通过本地存储的数据文件进行。

6. 为防止多次登录失败带来的安全障碍，系统在多次连续登录失败后需要进行短时间的锁定，提示用户稍后再试。

功能-3: 录入财务数据

详细要求：

1. 用户登录成功后，可以录入日常的收入和支出数据。

2. 录入的数据包括：类别（收入或支出）、金额、日期、备注等。

3. 系统需提供类别选择列表，方便用户快速选择收入或支出的分类（如食品、住房、娱乐等）。

4. 用户录入的数据需实时显示在界面上，以便用户确认输入的准确性。

5. 录入完成后，用户可以选择保存，系统将数据保存到本地文件中。

6. 系统需对录入的数据进行基本验证，例如金额必须为正数，日期必须是有效日期。

7. 录入的财务数据需保证存储的完整性和可跟踪性，方便用户后续查询和分析。

功能-4: 查看资产概况

详细要求：

1. 用户可以查看当前的资产总览，包括个人资产、负债以及整体的财务状况。

2. 系统需对用户的财务数据进行汇总和分类展示，帮助用户直观了解其财务健康状况。

3. 展示内容包括：总资产、总负债、净资产、各类收入和支出的汇总数据等。

4. 用户可以通过系统界面方便地切换不同的视图，例如月度、季度或年度的财务数据概况。

5. 系统需支持资产和负债的图形化展示，提供柱状图、饼图等可视化工具，以增强用户对财务状况的理解。

6. 查看资产概况的过程在本地进行，不依赖网络连接，所有展示的数据均来源于用户本地存储的文件。

7. 用户可以通过筛选条件（如日期范围、类别）查看特定时间段的财务状况，以便进行更详细的分析。

功能-5: 设定预算

详细要求：

1. 用户可以为未来的特定时间段（如月度或年度）设定预算目标。

2. 系统需提供收入和支出分类的预算设定，用户可以分别为不同类别设定预算金额。

3. 用户设定预算时，系统需提供过去的历史数据参考，帮助用户做出合理的预算决策。

4. 预算设定完成后，系统将预算数据保存至本地文件，便于后续跟踪和比较。

5. 系统需实时跟踪用户的实际收支情况，并将其与预算目标进行对比，帮助用户了解收支是否符合预算。

6. 系统需在用户超出预算时提供提醒，帮助用户更好地控制支出。

7. 预算设定功能在本地进行，不依赖网络，所有数据的存储和处理均通过本地文件完成。

功能-6: 导出报表

详细要求：

1. 用户可以选择导出财务统计数据，生成Excel或CSV格式的报表，以便于后续的查看和分析。

2. 导出的报表需包含用户的收入、支出、资产和负债等详细信息，方便用户进行全面的财务总结。

3. 用户可以选择特定的时间段来导出数据，例如月度、季度或年度的财务报表。

4. 系统需提供导出前的预览功能，让用户确认导出的数据内容是否符合需求。

5. 导出的文件应保存在用户指定的本地路径，文件命名应包含导出日期，以便于管理和查找。

6. 报表导出过程中，系统需保证数据的完整性和准确性，避免数据丢失或损坏。

7. 导出报表的功能在本地进行，不依赖网络，所有数据的处理和生成均通过本地文件完成。

1. 本系统的业务逻辑围绕用户的个人财务管理需求展开，提供全方位的财务管理解决方案。系统通过本地化的数据存储与处理，确保用户的数据安全并为其提供便捷的财务管理体验。系统的业务逻辑流程如下：首先，用户需要通过注册功能创建一个唯一的个人账户，并通过用户名和密码进行身份验证以登录系统。登录成功后，用户可以进行财务数据的录入操作，包括日常的收入与支出数据，同时系统会对用户输入的数据进行验证，以确保其准确性和完整性。用户完成财务数据的录入后，系统将这些数据存储在本地文件中，并为用户提供资产概况的查看功能。通过资产概况，用户可以清晰地看到其当前的财务状况，包括资产、负债及净资产等信息。系统还通过多样化的图形化工具，帮助用户更直观地了解自己的财务健康状况。为了帮助用户进行更好的财务规划，系统提供了预算设定功能。用户可以为未来的特定时间段设定收入和支出的预算目标，系统会根据历史数据给出合理的建议，并实时跟踪用户的收支情况。若用户的实际支出超出了设定的预算，系统会给出相应的提醒，以帮助用户更好地控制支出。在此以外，系统还提供了导出报表的功能，用户可以将自己的财务数据导出为Excel或CSV格式，以便于后续的分析和管理。导出前，系统会提供数据的预览功能，确保用户导出的数据符合需求，所有的处理和生成操作均在本地进行，确保数据的安全性和完整性。

# 5.性能需求（必填）

## 5.1并发性

由于该系统是面向个人用户设计，并且所有操作均在本地环境中完成，不涉及多用户同时使用，因此无需处理并发用户的请求。然而，系统仍需确保在用户执行多种操作（如录入数据、查看资产、导出报表）时能保持高效的响应速度，不会出现卡顿或延迟。

## 5.2容错要求

系统应具备基本的容错能力。例如，在用户操作失误（如输入非法数据、意外退出系统等）时，系统应能捕获异常并提示用户纠正。此外，系统需提供数据备份机制，确保本地文件保存时的数据完整性，避免因意外关闭或文件损坏导致的数据丢失。

## 5.3时间特性

系统在数据处理和反馈上应具有较高的实时性。具体表现为：

* 数据录入后，应立即保存至本地文件并反馈给用户；
* 资产概况、预算跟踪等查询操作应在3秒以内响应；
* 导出报表功能应在10秒内完成数据生成与导出操作。

## 5.4适应性

系统应能在不同操作系统（如Windows、macOS、Linux）上无缝运行，且对屏幕分辨率和硬件配置无特殊要求。在不同硬件性能的计算机上，系统仍能提供一致的用户体验。

# 6.软件属性需求

## 6.1正确性

系统需确保所有功能模块的实现符合需求说明，输入数据后能正确保存、展示和处理。系统必须对用户的所有操作做出正确的响应，且在不同场景下保证数据一致性，避免出现错误的计算或显示。

## 6.2可靠性

系统应在长时间运行时保持稳定，不因持续操作而出现崩溃或数据丢失。数据文件的读写过程应安全可靠，避免因异常操作或设备问题导致数据损坏或丢失。定期提供数据备份功能以提高数据恢复的能力。

## 6.3效率

系统需要在资源使用上保持高效，特别是本地数据文件的读写速度。即使处理大量的财务数据，系统也应能在合理的时间内完成操作（如查看历史数据、导出报表等）。同时，系统在CPU、内存等计算资源的占用上应保持在较低水平，以确保在各类硬件环境下的流畅运行。

## 6.4完整性

系统需具备完整的功能，从用户注册、登录到数据录入、查询、预算设定、导出报表等各个功能模块都应实现并测试完备。同时，系统必须具备错误处理能力，在用户输入不合法数据或系统出现异常时能给予适当的提示并恢复正常状态。

## 6.5易使用性

系统需提供直观的用户界面，所有操作流程应简单易懂，用户无需经过复杂的培训即可上手使用。界面应简洁明了，用户能轻松找到所需功能并快速完成操作。系统需提供明确的操作提示与反馈，提升用户体验。

## 6.6可维护性

系统需具备良好的代码结构与注释，便于后续开发人员进行维护与功能扩展。所有模块的设计应遵循模块化原则，便于单独测试与修改。系统应支持版本更新和bug修复，且数据的存储与处理逻辑应具备可扩展性。

## 6.7可测试性

系统各个功能模块应支持单元测试与集成测试，确保在不同输入条件下的稳定性和正确性。特别是数据输入、存储和导出功能，需有完善的测试案例以验证其在各类场景下的表现。测试框架应简单易用，便于快速定位和修复问题。

## 6.8复用性

系统的部分模块（如用户注册与登录、数据存储与导出等）应设计为通用模块，便于在未来其他项目中复用。这不仅提升开发效率，还能减少重复劳动。

## 6.9安全保密性

系统需采用加密方式存储用户的敏感信息（如密码），确保用户数据的安全性。所有本地存储的数据文件应经过加密处理，以防止未经授权的访问或篡改。此外，系统在多次登录失败后需进行短时锁定，防止暴力破解。

## 6.10可理解性

系统的操作应清晰易懂，提供必要的帮助文档或提示信息，帮助用户在操作过程中遇到问题时快速找到解决方案。开发人员也应确保代码可读性，代码组织规范，带有清晰的注释，便于后续的理解和维护。

## 6.11可移植性

系统应具有良好的跨平台兼容性，能够在不同操作系统和硬件环境中顺利运行。特别是在不同计算资源和屏幕分辨率的设备上，系统应自适应调整界面和性能，确保一致的用户体验。

## 6.12互联性

虽然本系统为离线应用，不依赖网络，但未来可以考虑添加与其他系统（如银行API）的对接功能，用以同步用户的银行交易记录等信息。在确保用户数据安全的前提下，系统应具备扩展为联网应用的潜力。

# 7.其他需求（非必须）

# 8.附录

## 8.1尚未解决的问题（可选）

如需要，可说明软件需求中的尚未解决的遗留问题。

## 8.2注解（可选）

包含有助于理解本文档的信息（例如背景信息、原理）。