中山大學
SUN YAT-SEN UNIVERSITY

中山大学计算机学院

软件工程课程项目

LifeMaster系统建模报告

项 目 名 称: LifeMaster

组 员 姓 名: 刘昊、彭怡萱、马福泉

: 林炜东、刘贤彬、刘明宇

专 业: 软件工程

课 程 教 师: 郑贵锋

起 始 日 期: 2025年3月1日

结 束 日 期: 2025年7月6日

学 院: 计算机学院

目录

1 系统概述

LifeMaster是一个集成待办事项管理、记账管理和手账管理功能的个人生活管理系统。本报告通过UML建模图形系统地描述了系统的结构、行为和交互关系，包括类图、顺序图、用例图、活动图和状态图等多个维度的建模分析。

1.1 建模目的

通过UML建模，我们可以：

- **结构化设计**：清晰地定义系统的静态结构和动态行为
- **需求分析**：准确理解和表达系统的功能需求
- **沟通工具**：为开发团队提供统一的理解基础
- **设计验证**：确保系统设计的完整性和一致性

2 类图（UML）

2.1 类图概述

类图描述了LifeMaster系统中的核心实体及其相互关系。系统主要包含用户管理、任务管理、记账管理和手账管理四个核心模块。

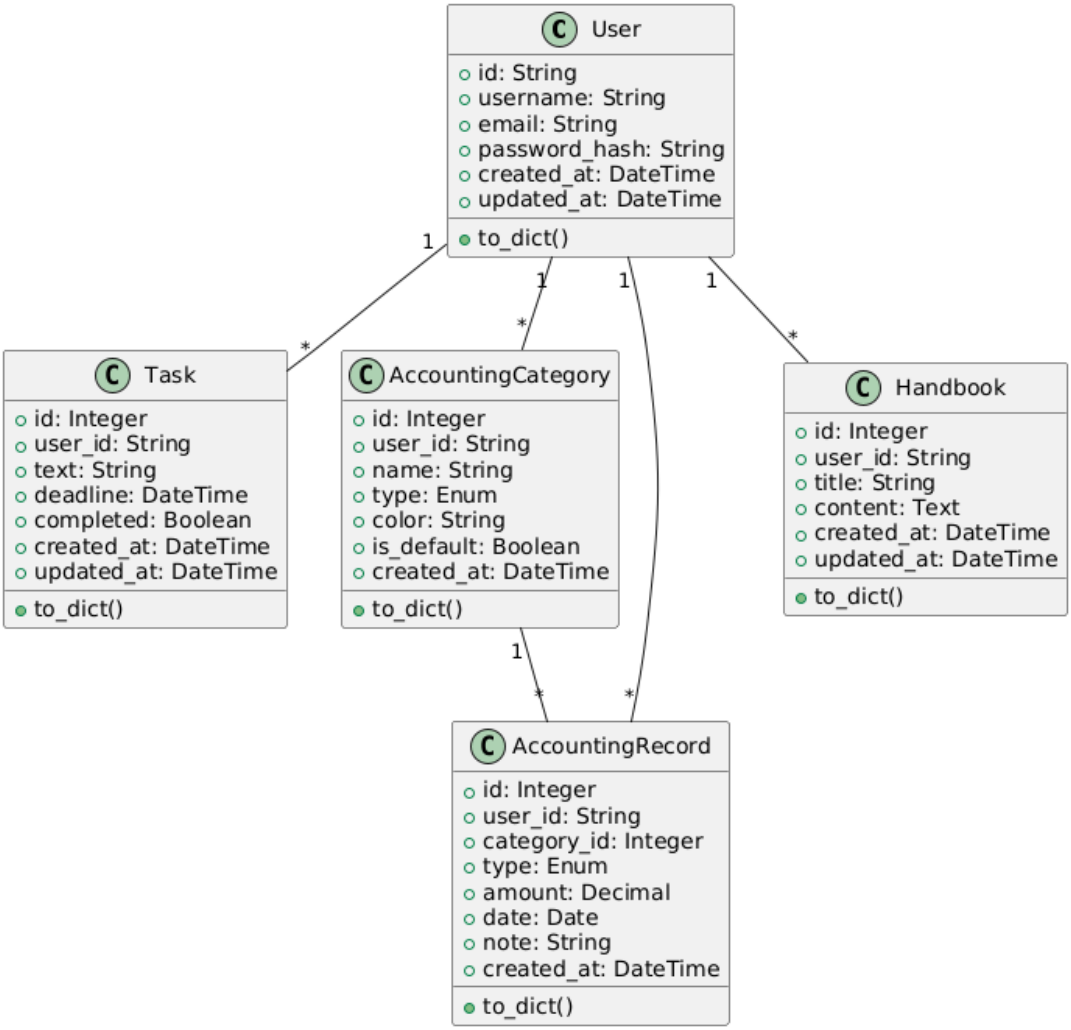

图 1: LifeMaster系统类图

2.2 核心类说明

2.2.1 User类（用户类）

User类是系统的核心用户实体，包含用户的基本信息和权限管理。

主要属性：

- `id: String` - 用户唯一标识符
- `username: String` - 用户名
- `email: String` - 电子邮箱地址
- `password_hash: String` - 密码哈希值
- `created_at: DateTime` - 账户创建时间
- `updated_at: DateTime` - 最后更新时间

主要方法：

- login(username, password): Boolean - 用户登录验证
- logout(): void - 用户登出
- updateProfile(userInfo): Boolean - 更新用户信息
- changePassword(oldPass, newPass): Boolean - 修改密码

2.2.2 Task类（任务类）

Task类管理用户的待办事项，支持任务的创建、编辑、删除和状态管理。

主要属性：

- id: Integer - 任务唯一标识符
- user_id: String - 所属用户ID
- title: String - 任务标题
- description: String - 任务描述
- deadline: DateTime - 截止时间
- status: TaskStatus - 任务状态（待完成/已完成/已过期）
- priority: Priority - 优先级（高/中/低）
- created_at: DateTime - 创建时间
- updated_at: DateTime - 更新时间

主要方法：

- create(taskInfo): Task - 创建新任务
- update(taskInfo): Boolean - 更新任务信息
- delete(): Boolean - 删除任务
- markCompleted(): Boolean - 标记任务为已完成
- setDeadline(date): Boolean - 设置截止时间

2.2.3 AccountingCategory类（记账分类类）

AccountingCategory类管理记账的分类信息，为记账记录提供分类支持。

主要属性：

- id: Integer - 分类唯一标识符
- user_id: String - 所属用户ID
- name: String - 分类名称
- type: CategoryType - 分类类型（收入/支出）

- color: String - 分类颜色标识
- icon: String - 分类图标
- created_at: DateTime - 创建时间
- updated_at: DateTime - 更新时间

主要方法：

- create(categoryInfo): AccountingCategory - 创建新分类
- update(categoryInfo): Boolean - 更新分类信息
- delete(): Boolean - 删除分类
- getRecordsByCategory(): List<AccountingRecord> - 获取该分类下的记录

2.2.4 AccountingRecord类（记账记录类）

AccountingRecord类管理具体的收支记录信息。

主要属性：

- id: Integer - 记录唯一标识符
- user_id: String - 所属用户ID
- category_id: Integer - 所属分类ID
- amount: Decimal - 金额
- description: String - 记录描述
- date: Date - 记录日期
- created_at: DateTime - 创建时间
- updated_at: DateTime - 更新时间

主要方法：

- create(recordInfo): AccountingRecord - 创建记账记录
- update(recordInfo): Boolean - 更新记录信息
- delete(): Boolean - 删除记录
- getStatistics(period): Statistics - 获取统计信息

2.2.5 Handbook类（手账类）

Handbook类管理用户的手账内容，支持文本、图片等多媒体内容。

主要属性：

- `id: Integer` - 手账唯一标识符
- `user_id: String` - 所属用户ID
- `title: String` - 手账标题
- `content: Text` - 手账内容
- `images: String` - 图片路径列表
- `tags: String` - 标签列表
- `mood: Integer` - 心情评分（1-5）
- `date: Date` - 手账日期
- `created_at: DateTime` - 创建时间
- `updated_at: DateTime` - 更新时间

主要方法：

- `create(handbookInfo): Handbook` - 创建手账
- `update(handbookInfo): Boolean` - 更新手账内容
- `delete(): Boolean` - 删除手账
- `addImage(imagePath): Boolean` - 添加图片
- `setMood(moodLevel): Boolean` - 设置心情评分

2.3 类之间的关系

- **User与Task**：一对多关系，一个用户可以有多个任务
- **User与AccountingCategory**：一对多关系，一个用户可以创建多个记账分类
- **User与AccountingRecord**：一对多关系，一个用户可以有多个记账记录
- **User与Handbook**：一对多关系，一个用户可以创建多个手账
- **AccountingCategory与AccountingRecord**：一对多关系，一个分类可以包含多个记录

3 顺序图（UML）

顺序图描述了系统中各个对象之间的时序交互过程。以下是LifeMaster系统的关键业务流程顺序图。

3.1 用户登录流程

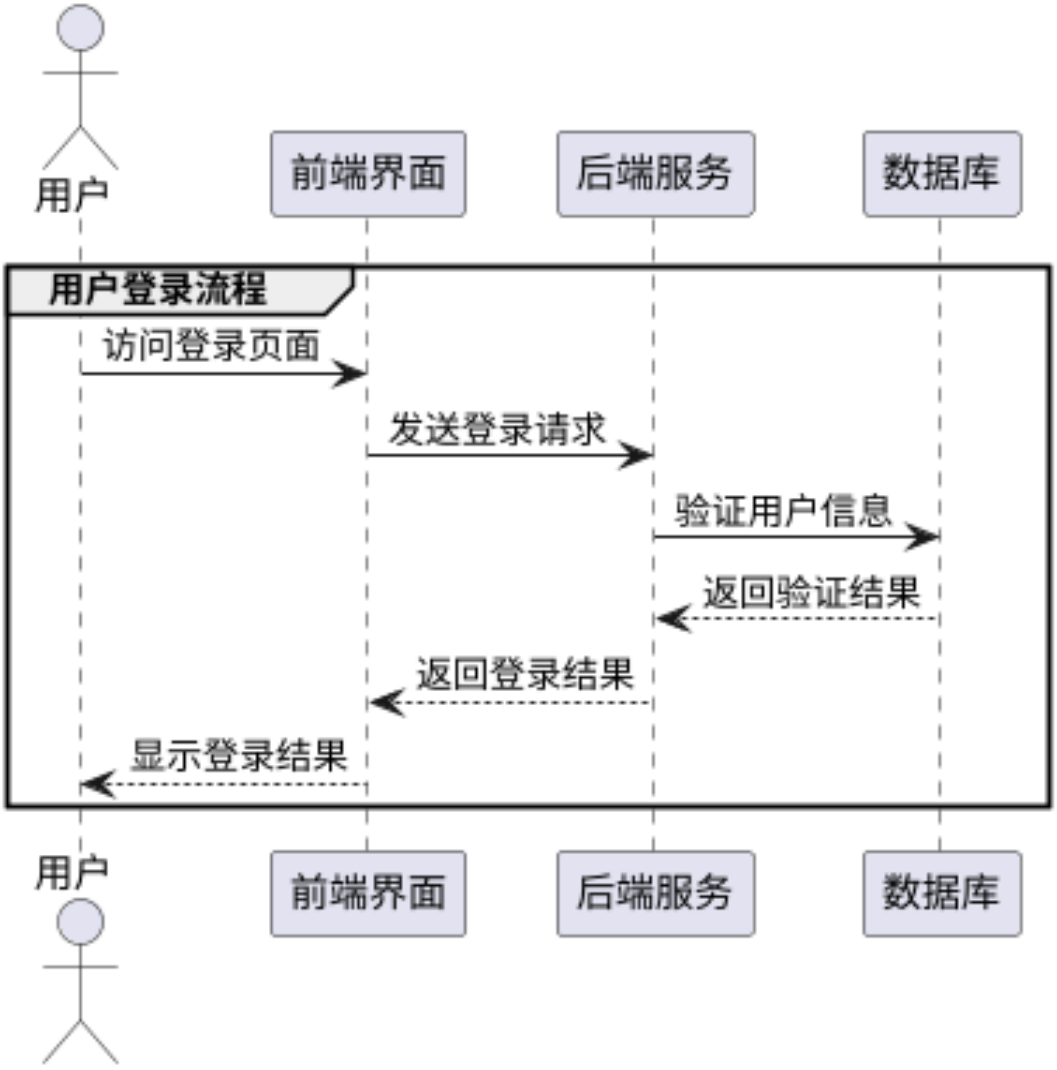

图 2: 用户登录流程顺序图

流程说明：

1. 用户访问登录页面，前端显示登录界面
2. 用户输入用户名和密码，点击登录按钮
3. 前端发送登录请求到后端API
4. 后端接收请求，查询数据库验证用户信息
5. 数据库返回用户验证结果
6. 如果验证成功，后端生成JWT token
7. 后端将token返回给前端
8. 前端接收token，存储到本地存储
9. 前端显示登录成功消息并跳转到主页面

3.2 财务管理流程

3.2.1 记账分类管理

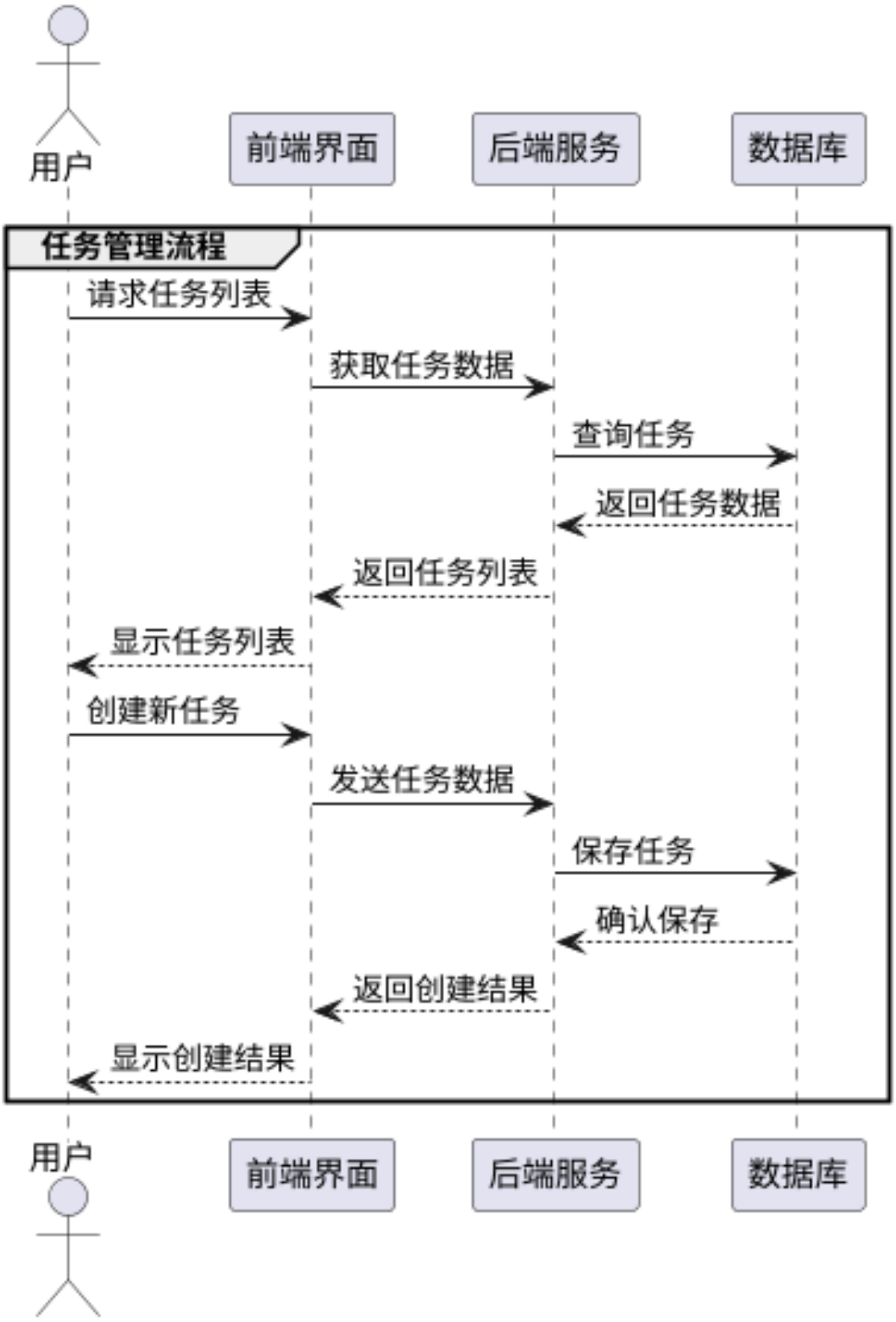

图 3: 记账分类管理顺序图

流程说明：

1. 用户点击”分类管理”按钮
2. 前端发送获取分类列表的请求
3. 后端从数据库查询用户的所有分类
4. 数据库返回分类数据列表
5. 后端处理数据并返回给前端
6. 前端接收数据并渲染分类列表界面
7. 用户可以查看、编辑或删除分类

3.2.2 记账记录管理

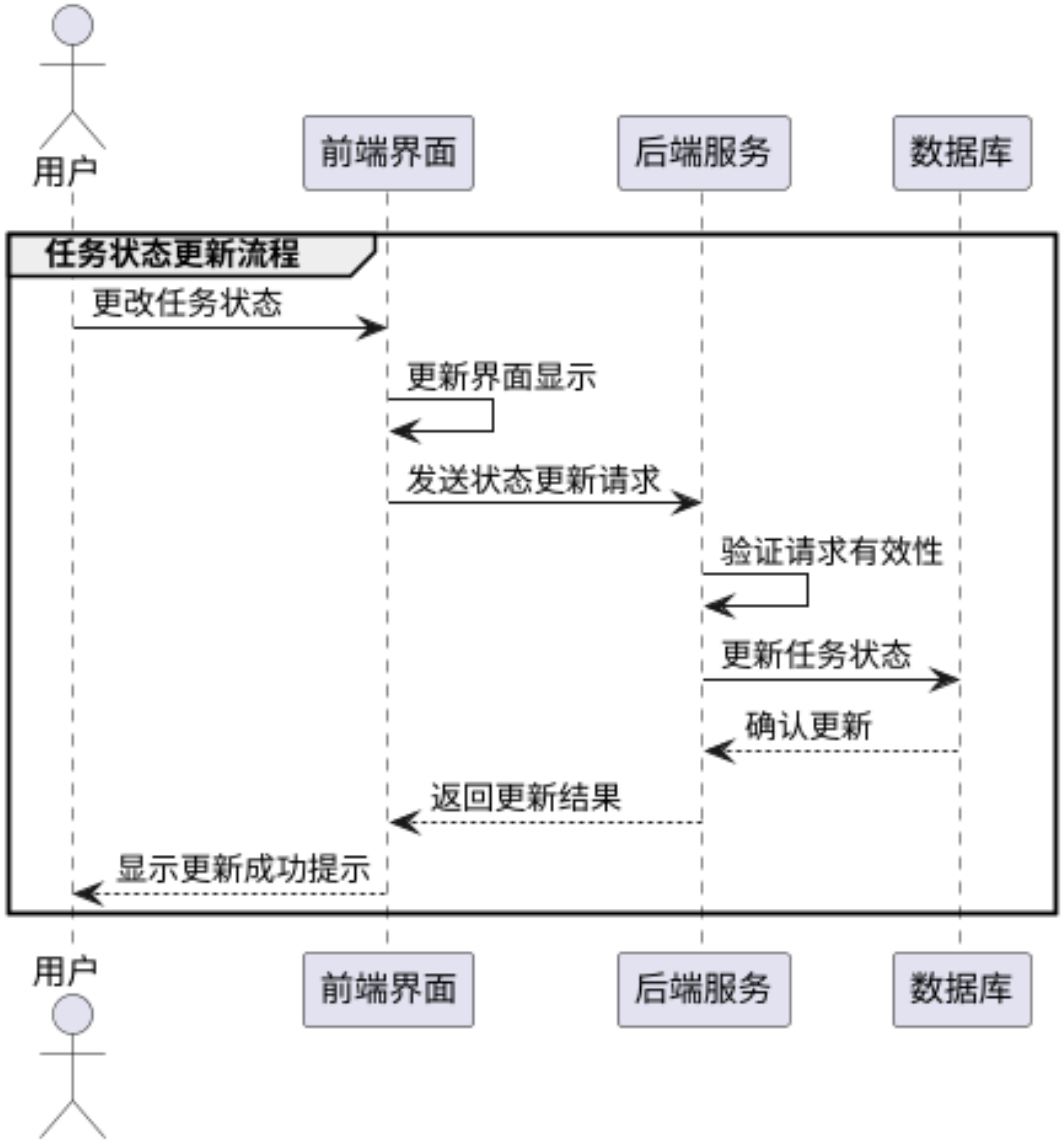

图 4: 记账记录管理顺序图

流程说明：

1. 用户填写收支记录表单（金额、分类、描述等）
2. 前端进行表单数据验证（必填项、格式检查）
3. 验证通过后，前端提交记录数据到后端
4. 后端接收数据，执行业务逻辑验证
5. 后端将验证通过的数据存入数据库
6. 数据库确认存储完成，返回操作结果
7. 后端将操作结果返回给前端
8. 前端显示操作成功消息，更新界面

3.2.3 账务统计分析

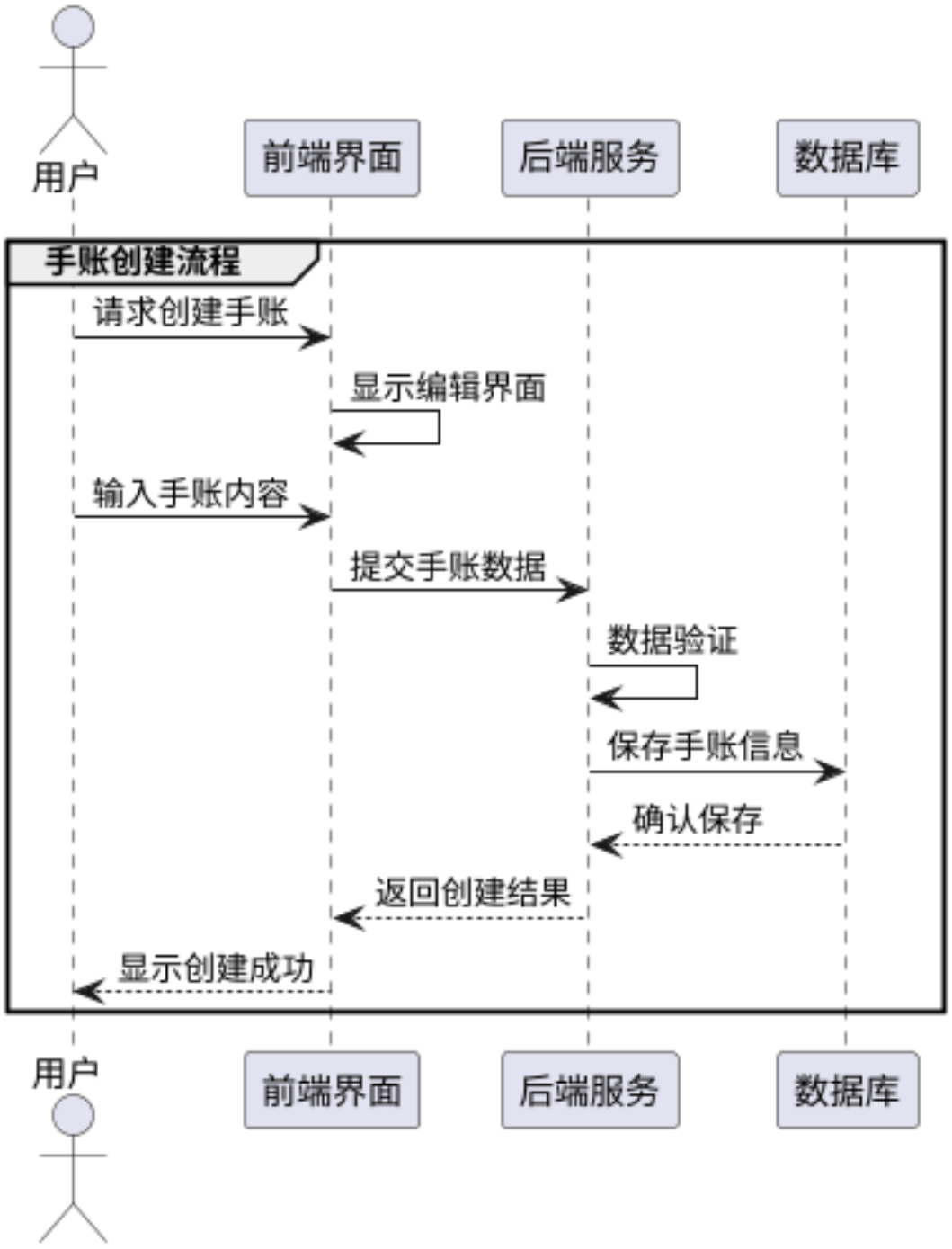

图 5: 账务统计分析顺序图

流程说明：

1. 用户选择查看财务报表
2. 前端发送统计数据请求（指定时间范围）
3. 后端执行数据库聚合查询操作

4. 数据库返回统计分析结果
5. 后端处理统计数据，计算各项指标
6. 后端将处理后的数据返回前端
7. 前端使用Chart.js生成可视化图表
8. 向用户展示包含图表的财务报表

3.3 手账创建流程

手账创建是用户记录生活的重要功能，支持文本、图片和心情记录。

流程说明：

1. 用户点击"新建手账"按钮
2. 前端显示手账编辑界面
3. 用户输入手账标题、内容、选择心情等
4. 用户可选择上传图片文件
5. 前端验证输入数据的完整性
6. 前端提交手账数据到后端API
7. 后端验证数据格式和权限
8. 后端将手账数据保存到数据库
9. 数据库确认保存成功
10. 后端返回创建成功的响应
11. 前端显示创建成功提示并刷新列表

3.4 任务状态更新

任务状态更新是高频操作，需要保证界面响应性和数据一致性。

流程说明：

1. 用户点击任务状态切换按钮（如完成按钮）
2. 前端立即更新界面显示（乐观更新）
3. 前端发送状态更新请求到后端
4. 后端验证请求的合法性和权限
5. 后端更新数据库中的任务状态
6. 数据库确认更新操作完成
7. 后端返回更新结果到前端
8. 如果更新失败，前端回滚界面状态
9. 前端显示相应的成功或失败提示

4 用例图（UML）

用例图描述了系统的功能需求以及各类用户与系统的交互关系。

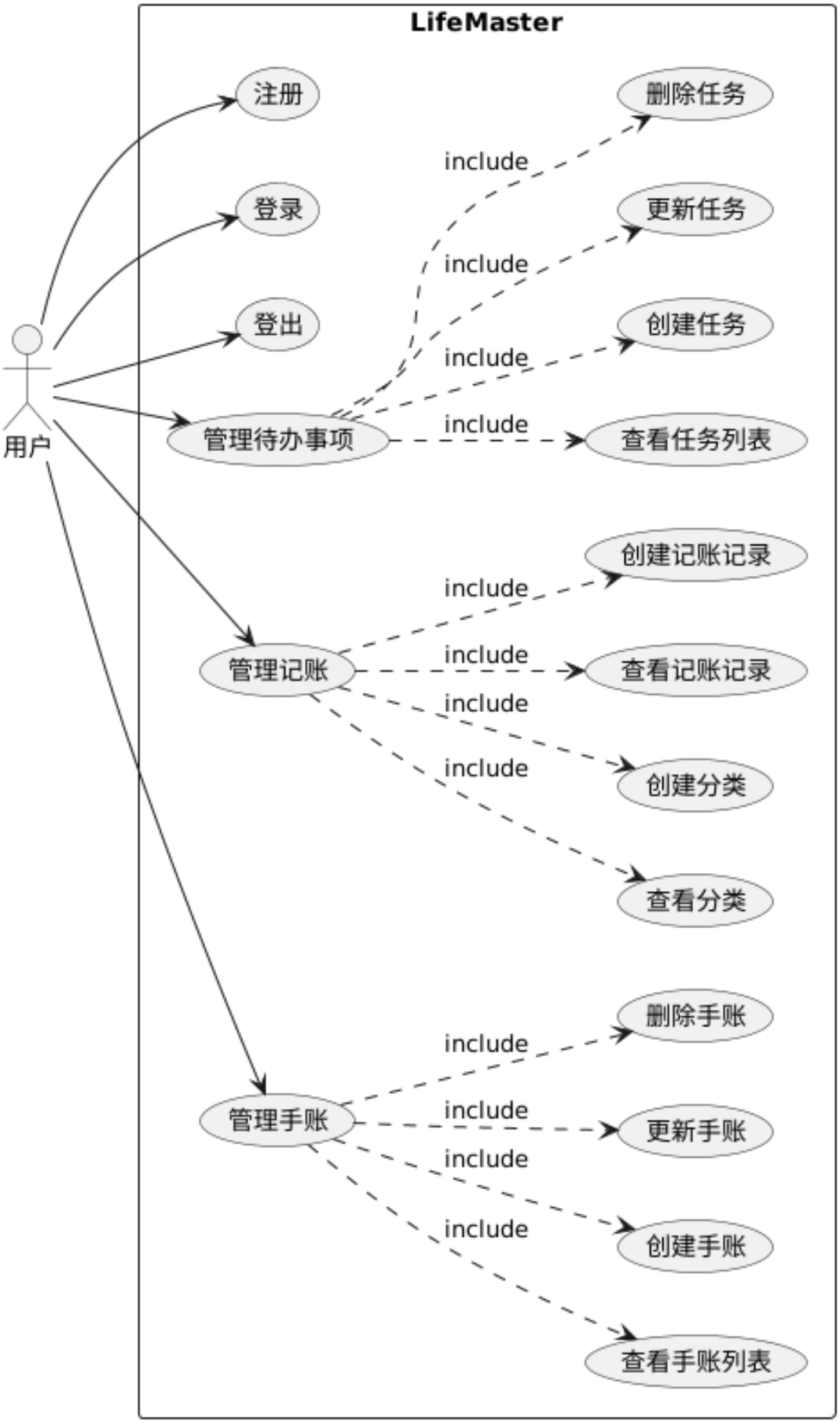

图 6: LifeMaster系统用例图

4.1 主要用例说明

4.1.1 用户管理用例

- **用户注册**：新用户创建账户
- **用户登录**：已注册用户通过凭证访问系统
- **用户登出**：结束当前会话
- **修改个人信息**：更新用户基本信息
- **修改密码**：变更登录密码

4.1.2 任务管理用例

- **查看任务列表**：显示用户的所有任务
- **创建任务**：添加新的待办事项
- **编辑任务**：修改任务内容、截止时间等
- **删除任务**：移除不需要的任务
- **标记任务完成**：更新任务状态
- **设置任务优先级**：调整任务的重要程度

4.1.3 记账管理用例

- **查看记账记录**：浏览历史收支记录
- **添加收支记录**：记录新的收入或支出
- **编辑记录**：修改已有的记账信息
- **删除记录**：移除错误的记录
- **管理分类**：创建、编辑、删除记账分类
- **查看统计报表**：分析收支情况和趋势
- **导出数据**：将记账数据导出为文件

4.1.4 手账管理用例

- **查看手账列表**：浏览所有手账记录
- **创建手账**：写新的日记或笔记
- **编辑手账**：修改手账内容
- **删除手账**：移除不需要的手账
- **上传图片**：为手账添加图片内容
- **设置心情**：记录当天的心情状态
- **添加标签**：为手账添加分类标签

4.2 用例关系

- **包含关系**：用户登录是所有功能用例的前提条件
- **扩展关系**：设置提醒是创建任务的可选扩展
- **泛化关系**：添加收入记录和添加支出记录都是添加记录的特化

5 活动图（UML）

活动图描述了系统中主要业务流程的控制流和活动序列。

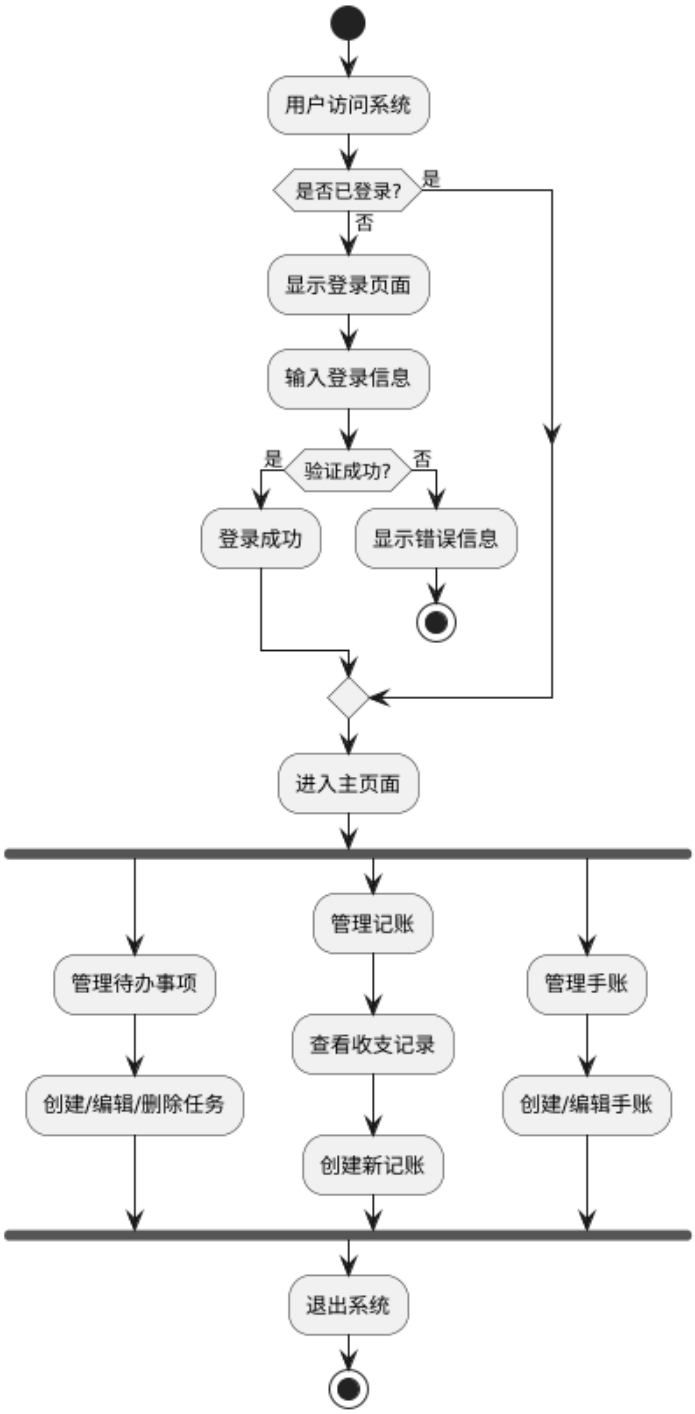

图 7: LifeMaster系统主要活动流程图

5.1 活动流程分解

5.1.1 步骤1：用户访问系统

初始节点：用户打开应用或网页，触发系统访问。

- 系统检查用户设备和浏览器兼容性
- 加载必要的静态资源（CSS、JavaScript等）

- 初始化用户界面框架
- 检查本地存储中的用户认证信息

5.1.2 步骤2：登录验证

判断是否已登录：

- **是** → 验证token有效性，直接跳转至主页面
- **否** → 显示登录页面，要求用户输入凭证

验证登录信息：

- **成功** → 生成新的访问token，进入主页面
- **失败** → 显示具体错误信息（如"用户名不存在"、"密码错误"等）
- 如果连续失败超过3次，触发账户临时锁定机制
- 提供"忘记密码"选项，引导用户重置密码

5.1.3 步骤3：主页面功能操作

用户登录成功后，可以执行以下操作（并行执行）：

管理待办事项：

- 查看任务概览：显示今日任务、即将到期任务等
- 创建任务：填写任务信息（标题、描述、截止时间、优先级）
- 编辑任务：修改现有任务的任何属性
- 删除任务：移除不需要的任务（需确认操作）
- 批量操作：支持批量标记完成、删除等操作

管理记账：

- 查看记录：按时间、分类、金额等条件筛选查看
- 添加记录：选择分类、输入金额、添加备注信息
- 统计分析：查看月度、季度、年度的收支统计
- 分类管理：自定义收入和支出的分类体系
- 预算设置：为不同分类设置月度预算限额

手账管理：

- 浏览手账：按日期、标签、心情等方式查看
- 创建手账：写日记、添加图片、标记心情、设置标签
- 编辑手账：修改已有的手账内容和属性
- 搜索功能：根据关键词搜索手账内容
- 导出功能：将手账内容导出为PDF或其他格式

5.1.4 步骤4：退出系统

正常退出流程：

- 用户主动选择退出操作
- 系统保存所有未提交的更改
- 清理客户端缓存和临时数据
- 注销服务器端的用户会话
- 清除本地存储的认证信息
- 返回登录页面或欢迎页面

异常退出处理：

- 会话超时自动退出
- 检测到异常登录时强制退出
- 网络中断时的客户端状态保护

6 状态图（UML）

状态图描述了系统中关键对象在其生命周期内状态的变化过程。

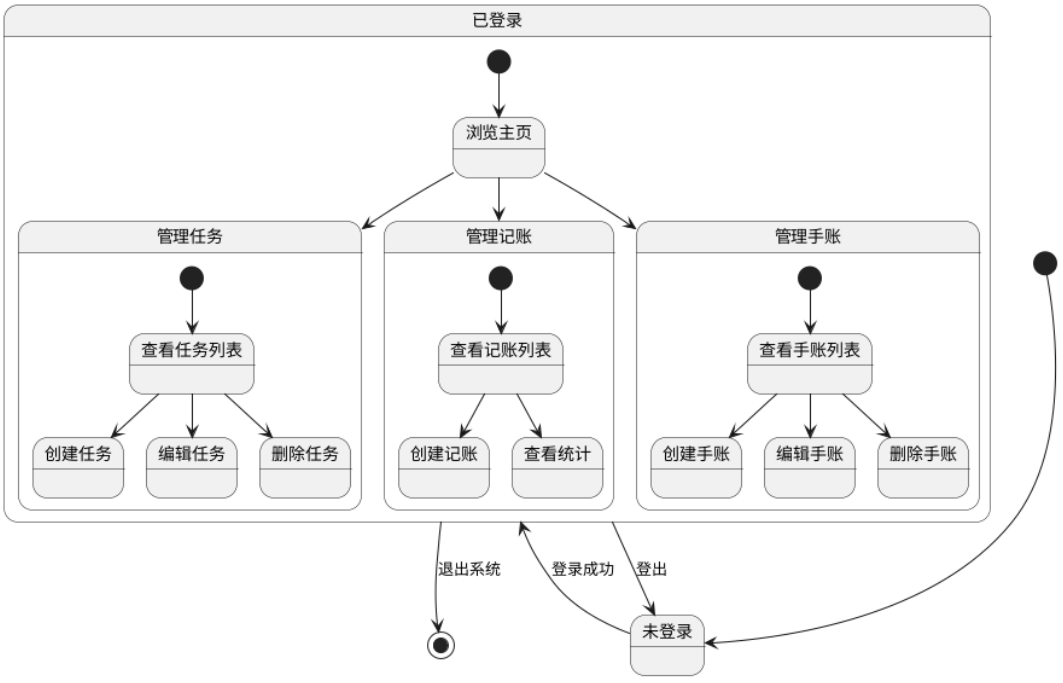

图 8: 用户会话状态图

6.1 核心状态与转换

6.1.1 （1）未登录（初始状态）

触发条件：

- 用户首次访问系统
- 用户主动退出系统
- 会话token过期
- 系统检测到安全异常强制登出

允许操作：

- 查看登录页面
- 尝试登录验证
- 访问用户注册功能
- 查看忘记密码功能
- 浏览系统介绍页面（如果有）

状态特征：

- 无法访问任何受保护的功能
- 不保存任何用户个人数据
- 只能进行公开的、不涉及个人信息的操作

6.1.2 （2）已登录（核心状态）

进入条件：

- 用户成功通过登录验证
- 系统验证用户凭证有效
- 生成有效的访问token

子状态详解：
用户在已登录状态下可以自由切换以下功能模块：
管理任务子状态：

- **查看任务列表：**显示所有任务的概览视图
- **创建任务：**进入任务创建模式，填写必要信息
- **编辑任务：**修改选定任务的属性和内容
- **删除任务：**移除不需要的任务项目

管理记录（记账）子状态：

- **查看记录列表**：浏览历史收支记录
- **创建记录**：添加新的收入或支出记录
- **编辑记录**：修改已有记录的详细信息
- **查看统计**：生成和展示财务分析报表
- **管理分类**：维护收支分类体系

管理手账子状态：

- **查看手账列表**：浏览所有手账记录
- **创建手账**：写新的日记或笔记内容
- **编辑手账**：修改现有手账的内容和属性
- **删除手账**：移除不需要的手账记录

状态转换条件：

- 用户通过导航菜单选择不同功能
- 通过快捷操作直接跳转到特定功能
- 系统根据用户行为自动切换相关状态

6.1.3 （3）退出系统（终止状态）

触发条件：

- 用户主动选择退出登录
- 系统检测到会话超时
- 管理员强制用户下线
- 检测到账户安全异常

处理流程：

- 保存用户当前的工作状态
- 清理客户端的敏感数据
- 注销服务器端的用户会话
- 记录用户退出日志

结果状态：

- 返回”未登录”状态
- 需要重新进行身份验证才能访问系统功能
- 清除所有本地缓存的用户数据

6.2 状态转换的触发事件

- **登录成功事件**：从"未登录"转换到"已登录"
- **退出事件**：从"已登录"转换到"未登录"
- **功能切换事件**：在"已登录"状态的各子状态间转换
- **会话超时事件**：自动从"已登录"转换到"未登录"
- **安全异常事件**：强制从任何状态转换到"未登录"

7 建模总结

7.1 建模成果

通过UML建模，我们完成了LifeMaster系统的全面设计分析：

- **类图**：明确定义了系统的核心实体和它们之间的关系
- **顺序图**：详细描述了关键业务流程的时序交互
- **用例图**：全面梳理了系统的功能需求和用户交互
- **活动图**：清晰展现了业务流程的控制逻辑
- **状态图**：准确描述了用户会话的状态变迁

7.2 设计验证

通过建模过程，我们验证了系统设计的：

- **完整性**：所有核心功能都有对应的建模描述
- **一致性**：各个建模图之间保持逻辑一致
- **可追溯性**：需求与设计之间建立了清晰的映射关系
- **可实现性**：建模设计考虑了技术实现的可行性

7.3 指导意义

本建模报告为LifeMaster系统的后续开发提供了：

- **开发指南**：为程序员提供清晰的实现指导
- **测试依据**：为测试用例设计提供参考标准
- **维护支持**：为系统维护和扩展提供文档支撑
- **沟通工具**：为团队协作提供统一的理解基础

通过系统性的UML建模，我们确保了LifeMaster系统设计的科学性和规范性，为项目的成功实施奠定了坚实的基础。