**需求规格说明书**

**《**翼界探秘系统**》**

**编写日期：**2024年6月20日

**项目组：**

|  |  |  |
| --- | --- | --- |
| **学号** | **姓名** | **角色** |
| **202231060926** | **魏辰睿** | **组长** |
| **202231060923** | **胡承波** | **组员** |
| **202231060920** | **李浩楠** | **组员** |
| **202231060924** | **董成阳** | **组员** |
| **202231060921** | **王皓宇** | **组员** |
| **202231045205** | **董美萱** | **组员** |

**修改日志**

|  |  |  |
| --- | --- | --- |
| 修改者 | 修改日期 | 备注说明 |
| 魏辰睿 | 2024.6.11 |  |
| 胡承波 | 2024.6.15 |  |
| 董成阳 | 2024.6.18 |  |
| 董美萱 | 2024.6.20 |  |

# 

**目录**

[1. 引言 5](#_Toc170501507)

[1.1 目的 5](#_Toc170501508)

[1.2 定义 5](#_Toc170501509)

[1.3 参考资料 5](#_Toc170501510)

[2. 软件总体概述 6](#_Toc170501511)

[2.1 软件标识 6](#_Toc170501512)

[2.2 软件描述 6](#_Toc170501513)

[2.2.1 系统属性 6](#_Toc170501514)

[2.2.2 开发背景 7](#_Toc170501515)

[2.2.3 软件功能 7](#_Toc170501516)

[2.3 用户的特点 8](#_Toc170501517)

[3. 具体需求 8](#_Toc170501518)

[3.1 功能需求 8](#_Toc170501519)

[3.2 性能需求 10](#_Toc170501520)

[3.3 设计约束 11](#_Toc170501521)

[3.4 其它非功能性需求 11](#_Toc170501522)

[3.5 外部接口需求 12](#_Toc170501523)

[3.5.1 用户接口 12](#_Toc170501524)

[3.5.2 硬件接口 12](#_Toc170501525)

[3.5.3 软件接口 12](#_Toc170501526)

[3.5.4 通信接口 13](#_Toc170501527)

[附录 功能模型 14](#_Toc170501528)

# 引言

## 目的

1. **明确软件要求**：详细阐述软件产品所必须满足的功能和非功能需求，确保开发团队和用户对软件的预期有共同的理解。
2. **作为沟通基础**：充当用户和软件开发人员之间的沟通桥梁，帮助双方明确需求并达成一致。
3. **指导设计和开发**：为软件的结构设计和编码工作提供基础和指导，确保开发过程中遵循既定的需求。
4. **需求跟踪和测试**：通过为每个功能需求分配唯一标识，便于需求的跟踪、验证和测试。
5. **性能要求说明**：提供软件性能方面的要求，包括数据处理能力和时间特性等，确保软件满足性能标准。
6. **设计约束和限制**：明确软件开发过程中需遵守的设计约束和限制条件，如技术限制、硬件限制、安全要求等。
7. **用户界面和交互**：定义用户界面需求，包括屏幕布局、报表格式、菜单结构等，以提升用户体验。
8. **外部接口规范**：详细描述软件与其他系统或硬件的接口需求，确保系统的兼容性和集成性。
9. **维护和可扩展性**：确保需求规格说明书支持软件的可维护性和未来可能的扩展。
10. **风险管理**：通过明确需求减少开发过程中的不确定性和风险。

## 定义

* **SRS**: 需求规格说明书（Software Requirements Specification）- 详细描述软件需求的文档。
* **AI**: 人工智能（Artificial Intelligence）- 使计算机模拟人类智能行为的技术。
* **UI**: 用户界面（User Interface）- 人与计算机系统交互的前端部分。
* **API**: 应用程序编程接口（Application Programming Interface）- 软件组件之间通信的一套规则和协议。
* **DFD**: 数据流图（Data Flow Diagram）- 表示系统中数据流动的图形表示法。
* **QFD**: 质量功能展开（Quality Function Deployment）- 一种将用户需求转化为产品设计的方法。

## 参考资料

**《用户需求说明书》**

**《项目开发委托合同书》**

**《技术可行性报告》**

**《项目开发计划》**

**《数据保护法》**

# 软件总体概述

## 软件标识

1. **软件名称**：
   * 全称：翼界探秘系统
   * 描述：一款集成了人工智能技术的移动应用，旨在帮助用户快速识别鸟类并获取相关知识。
2. **软件缩写**：
   * 缩写：YJTM（“翼界探秘”的首字母缩写）
3. **版本号**：
   * 当前版本：1.0
   * 版本命名规则：主版本号.次版本号.修订号（MAJOR.MINOR.PATCH）
4. **软件标识符**（如果适用）：
   * 唯一标识符：YJTM-APP-1001
5. **许可证信息**（如果适用）：
   * 许可证类型：专有软件/开源许可证（具体类型根据实际情况填写）
   * 许可证编号：（如果适用）
6. **开发团队**：
   * 团队名称：翼界探秘开发团队
   * 团队成员：魏辰睿、董成阳、李浩楠、王皓宇、胡承波、董美萱

## 软件描述

### 系统属性

1. **软件名称**：翼界探秘应用（Yingie Exploration App）
   * **版本**：1.0.0
2. **软件类型**：微信小程序
   * **平台**：iOS、Android
3. **独立性**：
   * 本软件作为一个独立的应用程序运行，不依赖特定的硬件或操作系统。
4. **集成性**：
   * 软件设计有API接口，允许与社交媒体平台集成，便于用户分享观察结果。
5. **可扩展性**：
   * 系统采用插件架构，支持未来添加新的功能模块或集成第三方服务。
6. **模块化**：
   * 主要模块包括：用户账户管理、鸟类识别引擎、社交分享、教育资源库、数据记录与分析。
7. **兼容性**：
   * 软件兼容所有主流浏览器的最新两个版本，以及iOS 12.0+和Android 8.0+操作系统。
8. **互操作性**：
   * 支持JSON和XML数据交换格式，确保与其他系统的数据交换无障碍。
9. **用户界面**：
   * 提供直观的图形用户界面，支持触摸屏操作，适配不同屏幕尺寸。
10. **多语言支持**：
    * 初始版本支持中文简体和英文，后续版本计划增加更多语言。
11. **安全性**：
    * 采用SSL加密传输数据，用户数据在服务器端加密存储。
12. **数据隐私**：
    * 遵守GDPR和其他相关数据保护法规，确保用户隐私。
13. **性能要求**：
    * 应用启动时间不超过3秒，鸟类识别响应时间不超过5秒。
14. **可靠性和容错性**：
    * 系统具备错误检测和异常处理机制，确保稳定运行。
15. **维护和支持**：
    * 提供在线帮助文档和客户服务支持，定期发布更新和补丁。
16. **法律和合规性**：
    * 符合所有适用的国际版权法、隐私法和行业标准。
17. **技术栈**：
    * 开发语言：Swift（iOS）、Kotlin（Android），后端：Node.js，数据库：MongoDB。
18. **版本控制**：
    * 使用Git进行版本控制，GitHub作为代码托管平台。
19. **依赖管理**：
    * 使用依赖管理工具如CocoaPods（iOS）、Gradle（Android）管理外部库。

### 开发背景

《翼界探秘系统》的开发背景源于对自然世界认知需求的增长以及人工智能技术的快速发展。随着公众环保意识的提升，准确识别和了解鸟类变得尤为重要。本项目旨在利用最新的AI技术，为用户提供一个便捷、高效的鸟类识别工具，同时普及鸟类知识，增强自然保护意识。市场需求分析表明，存在一个对综合性、教育性强的鸟类识别应用的明显需求。技术进步，尤其是在深度学习和图像识别领域，为本软件的开发提供了坚实的技术基础。项目得到了政策的支持，并与环保组织合作，以确保其教育价值和社会影响。目标用户群体广泛，包括自然爱好者、教育工作者和科研人员等。项目面临的风险和挑战，如技术实现和市场接受度，已通过详细的市场调研和技术开发计划得到评估和缓解。法律和伦理考量，包括数据隐私和用户安全，已被纳入开发过程。预期成果是创建一个既能教育用户又能促进生物多样性保护的工具，对个人用户和社会均产生积极影响。开发环境采用现代化的技术和敏捷开发方法论，确保软件质量和快速迭代。项目的关键里程碑包括原型设计、用户测试、正式发布等阶段，旨在实现一个用户友好、高准确度的鸟类识别应用。

### 软件功能

1. **用户账户管理**：
   * 用户可以创建和配置账户，包括登录、注册、密码管理和个人资料设置。
2. **鸟类识别**：
   * 用户可以通过上传鸟类照片，利用深度学习算法进行快速识别。
3. **社交分享**：
   * 用户可以将自己的观察成果分享到社交网络，与朋友和社区交流。
4. **教育资源库**：
   * 提供丰富的鸟类知识库，包括鸟类分类、习性、栖息地等信息。
5. **数据记录**：
   * 允许用户记录观察到的鸟类数据，形成个人观察日志。
6. **智能推荐**：
   * 根据用户的地理位置和季节，推荐可能观察到的鸟类种类。
7. **搜索功能**：
   * 提供一个搜索工具，使用户能够快速查找特定鸟类或相关信息。
8. **通知系统**：
   * 用户可以接收关于软件更新、鸟类观察提示和其他相关信息的通知。
9. **用户反馈**：
   * 用户可以通过内置的反馈机制报告问题或提出改进建议。
10. **帮助和支持**：
    * 提供帮助文档、FAQ和客户支持服务，帮助用户解决使用中的问题。

## 用户的特点

《翼界探秘系统》面向一个多样化的用户群体，具有广泛的特征和需求。用户可能包括从初学者到专家级的观鸟爱好者，年龄跨度从儿童到老年人，拥有不同的教育和技术背景。他们使用软件的目的可能是出于个人兴趣、教育需求、科研或参与环保活动。用户可能来自全球各地，有着不同的文化和语言，因此需要软件提供多语言支持。技术熟练度也不尽相同，从技术新手到数字产品的熟练用户，这要求软件界面直观易用。此外，用户可能存在视觉或操作障碍，需要软件具备相应的无障碍功能。考虑到用户的时间可用性和参与度不同，软件应提供灵活的使用方式和社交互动功能。用户对隐私和数据保护的关注也指导我们在设计中加入高标准的安全性。了解这些用户特点有助于我们设计一个全面、包容且用户友好的软件产品。限制与约束

# 具体需求

## 功能需求

#### 功能需求1：用户账户管理

**A. 输入：**

* 用户信息：包括用户名、密码、电子邮件地址、个人资料等。
* 输入源：用户通过注册和登录界面提供。
* 类型：文本、电子邮件格式。
* 长度：用户名6-20字符，密码8-64字符。
* 数值范围：无。
* 精度：无。
* 量纲：无。
* 数量：单一用户输入。
* 更新和处理频度：注册一次性，登录每次使用。

**B. 处理：**

* 验证：检查用户名和密码的格式及强度。
* 存储：将用户信息加密后存储在数据库。
* 会话管理：创建用户会话，处理登录和登出。
* 异常处理：捕获并响应无效输入或认证失败。

**C. 输出：**

* 认证结果：成功或失败的消息。
* 用户界面：登录后的个人主页或错误提示。
* 出错信息：如“用户名不存在”或“密码错误”。

#### 功能需求2：鸟类识别

**A. 输入：**

* 鸟类图片：用户上传的鸟类摄影。
* 输入源：应用内相机或图库上传。
* 类型：图像文件（JPEG、PNG）。
* 长度：文件大小不超过10MB。
* 数值范围：无。
* 精度：图像分辨率推荐1080p或更高。
* 量纲：无。
* 数量：每次处理一张图片。
* 更新和处理频度：按用户上传操作。

**B. 处理：**

* 图像预处理：调整大小、对比度增强。
* 识别算法：应用深度学习模型进行鸟类识别。
* 结果排序：根据置信度对识别结果进行排序。
* 异常处理：对无法识别或低置信度的结果提供反馈选项。

**C. 输出：**

* 识别结果：鸟类名称、常见特征、习性等信息。
* 用户界面：展示识别结果和相关信息。
* 出错信息：如“无法识别图片中的鸟类”。

#### 功能需求3：社交分享

**A. 输入：**

* 分享内容：用户选择的鸟类识别结果或个人观察日志。
* 输入源：用户在社交分享界面的操作。
* 类型：文本、图像链接。
* 长度：分享描述不超过500字符。
* 数值范围：无。
* 精度：无。
* 量纲：无。
* 数量：单一分享条目。
* 更新和处理频度：用户每次分享操作。

**B. 处理：**

* 内容格式化：确保分享内容符合目标社交平台的格式。
* 权限检查：验证用户对内容的分享权限。
* 发布：将内容发布到用户的社交网络。
* 异常处理：处理分享失败的情况并通知用户。

**C. 输出：**

* 分享结果：成功分享的确认或错误消息。
* 用户界面：更新的社交动态或分享失败提示。
* 出错信息：如“分享失败，请检查网络连接”。

## 性能需求

1. **响应时间**：
   * 系统应在用户执行操作后**2秒**内给予反馈或完成操作。
2. **并发用户数**：
   * 软件必须能够支持**1000**个以上用户同时在线使用。
3. **数据处理能力**：
   * 系统应能在**5分钟**内处理**1000条**用户数据记录。
4. **系统稳定性**：
   * 软件应保证在\*\*99.9%\*\*的正常运行时间内稳定运行，无意外中断。
5. **资源使用效率**：
   * 在正常使用情况下，软件应将CPU使用率控制在**70%以下，内存使用不超过512MB**。
6. **可扩展性**：
   * 系统应设计为可扩展架构，能够根据用户增长水平线性扩展资源。
7. **备份和恢复**：
   * 系统应能在**1小时**内完成数据的备份，并能在**2小时**内从备份中恢复数据。
8. **容错能力**：
   * 系统在遇到预期外的错误时，应能够保持关键功能运行，并提供错误日志以供分析。
9. **网络延迟**：
   * 软件应能在网络延迟**300毫秒**的情况下正常工作。
10. **兼容性**：
    * 软件在不同操作系统和设备上的性能表现应保持一致。

## 设计约束

1. **技术栈限制**：
   * 软件必须使用开源技术栈进行开发，以降低成本并促进社区支持。
2. **平台依赖性**：
   * 应用必须兼容iOS和Android平台，且在两种平台上提供相同的功能和用户体验。
3. **编程语言**：
   * 后端服务必须使用Node.js开发，前端界面使用Swift（iOS）和Kotlin（Android）。
4. **数据库选择**：
   * 必须使用MongoDB作为主数据库，以支持灵活的数据模型和快速迭代。
5. **用户界面设计**：
   * 用户界面设计必须符合iOS的Human Interface Guidelines和Android的Material Design Guidelines。
6. **安全性标准**：
   * 软件必须符合OWASP安全标准，实施所有必要的安全措施以保护用户数据。

## 其它非功能性需求

1. **可用性**：
   * 软件应易于使用，用户能够快速学习和掌握基本操作。
2. **可靠性**：
   * 系统应具备高可靠性，确保在连续运行和高负载下稳定工作。
3. **效率**：
   * 软件应优化算法和数据结构，减少响应时间和提高数据处理速度。
4. **安全性**：
   * 必须实现多层安全措施，包括数据加密、安全认证和防止未授权访问。
5. **可维护性**：
   * 代码应遵循清晰的编码标准，确保易于维护和升级。
6. **可测试性**：
   * 软件设计应允许全面测试，包括单元测试、集成测试和系统测试。
7. **兼容性**：
   * 软件应兼容主流操作系统和设备，包括不同的屏幕大小和分辨率。

## 外部接口需求

### 用户接口

1. **屏幕布局**：
   * 界面应有清晰的布局，功能区域划分明确，重要信息突出显示。
2. **导航设计**：
   * 系统应提供一个直观的导航菜单，方便用户快速访问不同功能。
3. **输入方式**：
   * 支持触摸屏输入，以及键盘和鼠标操作，适应不同用户的操作习惯。
4. **报表和数据展示**：
   * 数据和报表应以易于阅读和理解的格式展示，包括图表、列表和网格视图。
5. **菜单格式**：
   * 菜单项应分类清晰，命名直观，避免使用行业术语或缩写。
6. **输入输出响应时间**：
   * 系统应在用户进行输入后迅速响应，反馈时间不超过2秒。

### 硬件接口

1. **移动设备兼容性**：
   * 软件应兼容主流的移动操作系统，如iOS和Android，并适应不同型号和尺寸的智能手机和平板电脑。
2. **触摸屏支持**：
   * 系统界面设计应充分利用触摸屏功能，包括多点触控和手势操作。
3. **相机访问**：
   * 软件必须能够访问设备的相机，以支持鸟类拍照和图像上传功能。
4. **存储访问**：
   * 用户应能够从设备的存储中选择图片进行上传，同时软件需要存储缓存和应用数据。
5. **传感器使用**：
   * 根据需要，软件可能需要访问设备的传感器，如GPS进行地理位置定位。

### 软件接口

1. **操作系统兼容性**：
   * 接口标识：OS-COMPAT
   * 功能描述：确保软件在iOS和Android操作系统上运行。
   * 数据流程：用户通过操作系统启动应用。
   * 数据格式：符合操作系统的应用程序打包标准。
   * 接口类型：自动，通过应用商店下载安装。
   * 异常处理：操作系统版本不兼容时提示用户。
2. **数据库管理系统**：
   * 接口标识：DB-MS
   * 功能描述：后端数据存储和查询。
   * 数据流程：应用服务器与数据库的读写操作。
   * 数据格式：SQL语句和数据库记录。
   * 接口类型：自动，通过应用程序代码实现。
   * 版本号：如MySQL 8.0或MongoDB 4.4。
3. **第三方地图服务**：
   * 接口标识：MAP-SERVICE
   * 功能描述：提供地理位置服务和地图显示。
   * 数据流程：应用请求地理位置数据，服务返回地图信息。
   * 数据格式：地理坐标和地图API响应。
4. **社交媒体平台**：
   * 接口标识：SOC-MEDIA
   * 功能描述：允许用户分享内容到社交媒体。
   * 数据流程：应用发送分享请求到社交媒体API。
   * 接口类型：手动，用户触发分享操作。
5. **支付网关**：
   * 接口标识：PAYMENT-GATEWAY
   * 功能描述：处理应用内购买和订阅支付。
   * 数据流程：用户支付信息通过应用发送到支付网关。
   * 接口类型：自动，集成到应用的购买流程。

### 通信接口

1. **网络通信协议**：
   * 接口标识：NET-PROTOCOL
   * 功能描述：使用HTTP/HTTPS协议进行网络数据传输。
   * 数据流程：客户端和服务器之间的请求和响应。
   * 数据格式：JSON或XML格式的网络请求和响应体。
2. **API接口规范**：
   * 接口标识：API-SPEC
   * 功能描述：定义RESTful API接口，供前端调用后端服务。
   * 控制流程方向：从客户端到服务器的API调用。
   * 接口类型：自动，通过HTTP方法（GET, POST, PUT, DELETE）。

# 附录 功能模型

一、数据流图

1、顶层数据流图

2、第0层

3、第1层

二、数据字典

1、数据流

1. **用户登录信息**：
   * 名称：UserLoginData
   * 简述：包含用户用于登录系统的信息。
   * 数据流组成：用户名（文本），密码（加密文本）
   * 数据流来源：用户界面
   * 数据流去向：身份验证服务
   * 注解：密码字段在传输前必须进行加密处理。
2. **鸟类识别请求**：
   * 名称：BirdRecognitionRequest
   * 简述：用户发出的鸟类识别服务请求。
   * 数据流组成：用户ID（数字），图片文件（二进制），拍摄时间（日期时间）
   * 数据流来源：用户界面
   * 数据流去向：图像处理服务
   * 注解：图片文件需符合系统支持的格式和大小限制。
3. **识别结果**：
   * 名称：RecognitionResult
   * 简述：系统返回给用户的鸟类识别结果。
   * 数据流组成：鸟类名称（文本），相似度评分（百分比），识别信心度（百分比）
   * 数据流来源：识别算法引擎
   * 数据流去向：用户界面
   * 注解：识别结果应包括至少前三个可能的鸟类匹配项。
4. **用户反馈**：
   * 名称：UserFeedback
   * 简述：用户对识别结果或其他功能的反馈信息。
   * 数据流组成：用户ID（数字），反馈内容（文本），评分（1-5分）
   * 数据流来源：用户界面
   * 数据流去向：反馈处理系统
   * 注解：反馈系统应能分类和优先级排序用户反馈。
5. **社交分享内容**：
   * 名称：SocialShareContent
   * 简述：用户选择分享到社交网络的鸟类识别结果。
   * 数据流组成：用户ID（数字），分享文本（文本），图片链接（URL）
   * 数据流来源：社交分享功能
   * 数据流去向：社交媒体API
   * 注解：分享内容需符合社交媒体的内容规范。
6. **地理位置数据**：
   * 名称：GeoLocationData
   * 简述：用户设备的地理位置信息，用于智能推荐鸟类种类。
   * 数据流组成：纬度（十进制度数），经度（十进制度数）
   * 数据流来源：设备传感器
   * 数据流去向：位置服务处理模块
   * 注解：地理位置数据用于提供个性化的用户体验。

2、加工

1. **用户身份验证**：
   * 名称：UserAuthenticationProcess
   * 加工编号：P01
   * 简述：验证用户输入的登录凭证。
   * 输入数据流：UserLoginData（用户名，密码）
   * 输出数据流：AuthenticationResult（验证成功或失败的结果）
   * 加工逻辑：比对数据库中存储的加密密码与用户输入的密码。
   * 注解：密码错误次数限制在3次以内，超过限制则锁定账户5分钟。
2. **图片上传处理**：
   * 名称：ImageUploadProcess
   * 加工编号：P02
   * 简述：处理用户上传的鸟类图片。
   * 输入数据流：BirdRecognitionRequest（用户ID，图片文件，拍摄时间）
   * 输出数据流：ImageUploadConfirmation（图片上传确认信息）
   * 加工逻辑：验证图片格式和大小，存储图片到服务器，并生成图片唯一标识。
   * 注解：不支持的格式或超过大小限制的图片将返回错误信息。
3. **鸟类识别处理**：
   * 名称：BirdIdentificationProcess
   * 加工编号：P03
   * 简述：识别上传图片中的鸟类并返回结果。
   * 输入数据流：ImageForRecognition（图片唯一标识，图片二进制数据）
   * 输出数据流：RecognitionResult（鸟类名称，相似度评分，信心度）
   * 加工逻辑：使用深度学习模型分析图片，匹配数据库中的鸟类信息。
   * 注解：如果识别信心度低于设定阈值，则提示用户图片质量可能不佳。
4. **用户反馈收集**：
   * 名称：UserFeedbackCollection
   * 加工编号：P04
   * 简述：收集用户对识别结果的反馈。
   * 输入数据流：UserFeedback（用户ID，反馈内容，评分）
   * 输出数据流：FeedbackAcknowledgement（反馈收到的确认信息）
   * 加工逻辑：存储用户反馈到数据库，进行分类和优先级排序。
   * 注解：所有反馈都将被审核，以改进服务和用户满意度。

3、文件（存储）

1. **用户账户信息文件**：
   * 名称：UserAccountFile
   * 简述：存储用户账户的详细信息。
   * 文件组成：用户ID，用户名，密码哈希，电子邮件，注册日期，账户状态
   * 写文件的加工：用户注册过程，用户资料更新过程
   * 读文件的加工：用户登录过程，用户资料查看过程
   * 加工逻辑：在用户注册时创建记录，在用户更新资料时修改记录。
   * 注解：密码以哈希形式存储，确保存储安全。
2. **鸟类识别记录数据库**：
   * 名称：BirdRecognitionLog
   * 简述：记录用户每次使用识别功能的结果。
   * 文件组成：记录ID，用户ID，鸟类名称，识别时间，图片文件路径
   * 写文件的加工：鸟类识别处理
   * 读文件的加工：识别记录查询，历史记录查看
   * 加工逻辑：每次识别操作后，将结果写入数据库。
   * 注解：图片文件存储在服务器的指定路径，数据库仅保存路径引用。
3. **用户反馈存储**：
   * 名称：UserFeedbackStorage
   * 简述：存储用户对系统的反馈信息。
   * 文件组成：反馈ID，用户ID，反馈内容，评分，反馈时间，处理状态
   * 写文件的加工：用户反馈收集
   * 读文件的加工：反馈审核过程，反馈查看
   * 加工逻辑：用户提交反馈后，存储至数据库并标记为未处理。
   * 注解：反馈内容需进行审核，不恰当的内容需进行处理。
4. **社交分享记录文件**：
   * 名称：SocialShareRecord
   * 简述：记录用户分享到社交网络的操作。
   * 文件组成：分享ID，用户ID，分享内容，分享时间，社交网络类型
   * 写文件的加工：社交分享处理
   * 读文件的加工：分享历史查看
   * 加工逻辑：在用户分享内容后，记录分享的详细信息。
   * 注解：分享记录用于分析用户分享行为，优化分享功能。
5. **地理位置数据缓存**：
   * 名称：GeoLocationCache
   * 简述：临时存储用户地理位置数据，用于快速处理。
   * 文件组成：缓存ID，用户ID，纬度，经度，时间戳
   * 写文件的加工：地理位置服务
   * 读文件的加工：智能推荐系统
   * 加工逻辑：在获取地理位置后，存储至缓存供后续处理使用。
   * 注解：缓存数据具有时效性，超过一定时间后自动清除。