

计科2203-需求规格说明

知识图谱构建子系统

本项目旨在从海外博物馆网站爬取中国文物信息，通过数据加工处理构建海外文物知识图谱。系统需实现数据爬取、建模、补充和存储四大核心功能，最终形成开放数据集供后续应用开发，知识图谱作为知识增强的输入。

核心功能需求

从海外博物馆爬取中国文物信息时，主要有以下四个步骤：

- 1. 使用 `urllib` / `request` / `selenium` 等库爬取网页数据；
- 2. 对爬取到的数据使用 `xpath` 或 `beautifulsoup` 进行数据解析；
- 3. 利用 Python 的 `csv` 库和 `pandas` 库将数据存储到 CSV 文件中；
- 4. 导入服务器端的MySQL数据库和Neo4j图数据库。

爬取目标数据范围

Inscribed	Title	Artist	CreditLine	Dynasty	
铭文/题字	标题	创作者	藏品来源说明	朝代	
Dimensions	Materials	Description	Museum	PlaceOri	
尺寸	材料	描述	博物馆	原产地	
Classifications	Img	Medium			
分类	图片	媒介			

数据爬取功能点与具体要求

多源适配

支持以下三种以上网站的数据抓取：

- 史密斯博物馆
- 旧金山亚洲艺术博物馆
- 芝加哥艺术博物馆

分页机制

- 自动识别“Next”按钮位置
- 支持动态加载分页

断点续爬

- 记录已爬取的 `object_id` 到 `existing_ids.txt`
- 支持从断点恢复

爬取方式说明

- 通过每一页文物数据的接口来获取数据；
- 通过改变接口中的时间戳值和页码，爬取新的文物信息；
- 使用 Selenium 模拟浏览器翻页行为，获取接口 JSON 文件；
- 转换成 Python 字典后，存入 CSV 文件中；
- 通过包装请求头和IP- Cookies代理池等方法反爬。

缺失数据处理

- 通过人工从百度百科或其他资源补充；
- 若翻译或提取失败，可人工修正或补充内容。

存储要求

- 使用 `csv` 将爬取信息存入 CSV 文件；
- 图片用ID自增命名，统一存储至特定文件夹；
- 分别对三个博物馆检索数据，汇总后统一清洗；
- 数据最终导入两个数据库：

数据库	存储内容	技术规范
MySQL	文物信息	-
Neo4j	节点关系图	<code>CREATE CONSTRAINT FOR (a:Artifact) REQUIRE a.object_id IS UNIQUE</code>

数据清洗

清洗内容

- 确保时间一致性、实体唯一性；
- 翻译国外数据；
- 使用 pandas 进行清洗；
- 未处理成功的部分，从百度百科补充；
- 对百度翻译不准确的部分进行人工修正。

检查项与处理方式

检查项	校验规则	处理方式
时间一致性	<code>creation_date ≤ discovery_date</code>	数据清洗标记
实体唯一性	URI 冲突检测	数据合并 / 人工干预

海外文物知识服务子系统

系统概述

本系统是一个基于Web的博物馆文物展示平台，旨在为用户提供便捷的文物浏览、搜索、收藏和评论功能。系统采用前后端分离架构，后端使用Spring Boot框架，前端使用Vue.js框架，通过RESTful API进行数据交互。

功能需求

1. 用户管理功能

系统提供完整的用户管理功能，包括用户注册、登录、密码找回等。用户注册时需要提供用户名、邮箱和密码，系统会通过邮箱验证码确保邮箱有效性。登录支持账号密码和邮箱两种方式，采用JWT进行身份认证。密码找回功能通过邮箱验证码实现，确保账号安全。

2. 文物展示功能

系统提供丰富的文物展示功能，包括文物列表展示、详情查看、多条件搜索等。文物列表支持分页显示，每页展示8条记录。搜索功能支持按文物类型、朝代、博物馆、艺术家等多个维度进行筛选，并支持按名称、年代等条件排序。文物详情页面展示文物的高清图片、基本信息、详细描述以及相关文物推荐。

3. 用户交互功能

系统提供完善的用户交互功能，包括文物收藏、评论发表、浏览历史记录等。用户可以对感兴趣的文物进行收藏，并在个人中心管理收藏列表。评论功能允许用户对文物发表评论，支持查看和删除自己的评论。系统自动记录用户的浏览历史，方便用户回顾查看过的文物。

非功能需求

1. 性能需求

系统需要保证良好的性能表现，页面加载时间控制在3秒以内，搜索响应时间不超过2秒，图片加载时间不超过5秒。系统需要支持1000用户同时在线，100用户同时操作。数据存储方面需要支持10000件文物信息、100000用户数据和1000000条评论数据。

2. 安全需求

系统采用多重安全措施保护用户数据和系统安全。用户密码使用BCrypt算法加密存储，采用JWT进行身份认证，实现会话管理。数据库采用加密存储，定期进行数据备份。接口安全方面实现了CORS配置、请求验证和防SQL注入等安全措施。

3. 可靠性需求

系统需要保证724小时稳定运行，故障恢复时间控制在30分钟以内。系统实现了完整的异常处理机制，包括异常捕获、错误日志记录和用户友好的错误提示。数据备份采用每日增量备份和每周全量备份相结合的方式，确保数据安全。

系统接口

系统采用RESTful API设计，主要接口包括：

- 用户认证接口：处理用户注册、登录、密码找回等认证相关请求
- 文物管理接口：处理文物信息查询、搜索、详情获取等请求
- 用户交互接口：处理收藏、评论、浏览历史等用户交互相关请求

所有接口都采用JSON格式进行数据交换，并实现了统一的错误处理机制。

数据需求

系统采用关系型数据库存储数据，主要数据表包括：

- 用户相关表：存储用户基本信息、收藏记录、浏览历史等
- 文物相关表：存储文物基本信息、图片资源、评论数据等

数据备份采用每日增量备份和每周全量备份相结合的方式，确保数据安全。系统支持数据恢复功能，可以在数据丢失时快速恢复。

系统部署

系统采用Docker容器化部署，支持快速部署和扩展。服务器配置要求：

- CPU：4核以上
- 内存：8GB以上
- 存储：100GB以上
- 网络：100Mbps以上带宽

系统支持自动化部署和监控，包括日志收集、性能监控、告警通知等功能。

验收标准

系统验收需要满足以下标准：

- 功能验收：所有功能模块测试通过，接口响应符合要求，数据操作正确
- 性能验收：响应时间、并发处理能力、系统稳定性达到要求
- 安全验收：通过安全测试，数据保护和访问控制有效

系统上线前需要进行完整的测试，包括功能测试、性能测试、安全测试等，确保系统质量。

知识问答子系统

1. 导言

1.1 目的

本文档详细说明了我們组所开发的知识问答子系统的设计、功能及非功能需求。旨在为开发团队、质量保证人员、系统运维及最终用户提供清晰、详尽的指导和参考，确保项目各方达成一致的理解和预期。

1.2 适用人群和阅读建议

本文档适用于：

- 软件开发团队：用于实现系统功能和细节。
- 测试与质量保证团队：创建测试计划和测试用例。
- 系统运维与维护人员：确保后期维护和更新的便利性。
- 最终用户：提供基础使用说明和支持。

1.3 项目范围

本项目旨在开发一个基于 Neo4j 知识图谱和RAG架构的问答子系统，支持用户针对海外文物等特定领域的智能问答，提供友好、直观的界面。

2. 总体描述

2.1 产品愿景

打造一个智能、交互友好且高效的问答系统，基于丰富的文物知识图谱，准确回答用户提出的问题，并支持多轮问答和历史记录管理。

2.2 产品功能

1. 简单问答：支持单实体单属性、单实体多属性、多实体问题等基本问答，覆盖文物的收藏地、材质、时代、作者等信息。
2. 复杂问答：支持多实体多属性及复杂上下文对话。
3. 闲聊：支持与用户进行日常对话互动。
4. 用户身份管理：支持用户注册、登录及用户信息管理。
5. 历史记录管理：自动保存用户历史提问及回答内容。

2.3 用户类型和特征

- 对文物和历史文化有兴趣的学者、研究人员。
- 对海外文物有好奇心的普通用户。

2.4 操作环境

- 操作系统：Windows、MacOS、Linux。
- 需稳定的网络连接以保证数据交互。
- 推荐浏览器：Chrome、Edge、Firefox。

2.5 设计和实施约束

- 后端基于 Django 4.1.7 框架，Python 版本需 $\geq 3.8.0$ 。
- 前端基于 Vue3 框架。
- 使用 Docker 容器化和宝塔面板进行部署与维护。

2.6 用户文档

提供完整的用户使用手册及系统部署文档，便于用户快速上手和问题排查。

3. 系统特性

3.**1** *问答处理

- 系统可智能解析用户问题，生成 Cypher 查询并返回基于 Neo4j 知识图谱的准确答案。
- 支持多轮上下文管理及复杂问题处理。

3.**2** *用户交互

- 提供基于 Web 的可视化界面，界面美观简洁、交互便捷。

4. 外部接口需求

4.1 用户接口

- 提供基于 Web 的交互界面，兼容主流浏览器。

4.2 硬件接口

- 无特殊硬件接口需求。

4.3 软件接口

- 后端需与 Neo4j 数据库集成。
- 前后端通过 API 进行数据交互。

5. 其他非功能需求

5.**1** *软件质量属性

- 可维护性：代码应模块化、易于扩展。
- 可扩展性：支持未来功能的拓展与升级。

6. 其他需求

- 提供完整的系统部署文档和用户手册。
- 提供系统使用演示视频或 PPT，便于后期推广与培训。

掌上博物馆子系统

1. 引言

1.1 编写目的

本文档是软件工程课程设计的关键组成部分，旨在详细描述掌上博物馆系统的功能和非功能需求，为系统的设计和开发提供明确的指导和参考。通过本文档的编写，学生将实践需求分析和规格说明的软件工程方法，体验软件开发全生命周期中的需求工程阶段。本文档的主要读者包括课程设计小组成员、指导教师以及参与评审的同学。

1.2 项目背景

掌上博物馆是一个基于HarmonyOS开发的移动端应用，专注于展示和管理海外藏中国文物的知识服务平台。该应用旨在为用户提供便捷的文物浏览、搜索和交流功能，帮助用户了解和欣赏分散在海外博物馆的中国文化遗产。

本项目作为软件工程课程设计的实践内容，旨在让学生通过实际项目开发，掌握软件工程的核心概念和方法，包括需求分析、系统设计、编码实现、测试验证和项目管理等环节。项目选择"掌上博物馆"这一主题，既有实际应用价值，也具有较高的技术挑战性，适合作为综合性软件工程实践项目。

许多珍贵的中国文物分散在世界各地的博物馆中，如波士顿美术馆、纳尔逊-阿特金斯艺术博物馆和明尼阿波利斯艺术博物馆等。这些文物代表了中国丰富的历史文化遗产，但由于地理位置分散，公众难以全面了解和欣赏。掌上博物馆项目通过数字化方式，将这些分散的文物信息整合在一个平台上，提供便捷的访问和互动方式。

1.3 项目范围

作为软件工程课程设计项目，本系统将完整实践软件开发流程，涵盖以下主要功能：

- 文物信息数据库建设与管理
- 文物浏览和详情展示
- 多维度文物搜索，包括文字搜索和图像搜索
- 用户互动系统，包括评论、点赞、收藏
- 社区交流平台
- 用户账户管理
- 基于华为云的后端服务支持

1.4 软件工程方法论

本项目将采用迭代增量开发模型，结合敏捷开发的部分实践。开发过程将包括以下阶段：

1. 需求分析与规格说明（本文档）
2. 系统架构设计
3. 详细设计
4. 迭代开发与实现
5. 测试与验证
6. 部署与维护

在每个阶段，团队将应用相应的软件工程工具和技术，并进行阶段性评审和反馈，确保项目质量和进度。

1.5 术语定义

术语	定义
HarmonyOS	华为开发的分布式操作系统
掌上博物馆	本项目的移动应用程序名称
HMS ML Kit	华为移动服务机器学习套件
AGC	华为AppGallery Connect服务
SRS	软件需求规格说明书(Software Requirements Specification)
UML	统一建模语言(Unified Modeling Language)

2. 总体描述

2.1 产品功能概述

掌上博物馆系统提供以下核心功能：

- 海外藏中国文物的数字化展示
- 多种方式的文物搜索和浏览
- 基于图像识别的文物查找
- 用户社区互动与分享
- 个人收藏与兴趣管理
- 基于云服务的数据同步和存储

2.2 用户特征

本系统的主要用户包括：

1. 普通用户

- 文化爱好者：对中国传统文物感兴趣的用户
- 学生与教师：用于学习和教学的用户
- 游客：计划参观博物馆的用户

2. 专业用户

- 文物研究人员：需要详细文物资料的专业人士
- 博物馆工作人员：关注文物信息准确性的专业人员

3. 系统管理员

- 负责系统维护和内容审核的管理人员

4. 课程设计评审人员

- 指导教师：评估项目完成质量和符合教学要求的程度
- 同学评审员：参与项目评审和反馈的其他学生

2.3 运行环境

- 客户端：搭载HarmonyOS 2.0及以上版本的华为设备
- 服务器端：华为云服务器
- 数据库：MySQL 8.0
- 开发环境：DevEco Studio 3.0及以上
- 版本控制：Git
- 项目管理：GitHub项目管理工具
- 文档工具：Markdown、UML工具

2.4 设计和实现约束

- 系统必须符合HarmonyOS设计规范
- 系统必须遵守相关数据安全法规
- 文物图像必须获得相关博物馆授权或使用开放许可的资源
- 系统架构必须支持未来功能扩展
- 应用大小应控制在合理范围内
- 项目必须按照软件工程课程设计的时间节点完成相应阶段成果
- 设计和实现过程必须体现软件工程的思想和方法

2.5 软件工程教学目标

通过本项目的设计与实现，学生将获得以下软件工程能力：

1. 需求分析与规格说明能力
2. 软件架构设计能力
3. 模块化与面向对象设计能力
4. 团队协作与项目管理能力
5. 软件测试与质量保证能力
6. 技术文档编写能力
7. 软件维护与版本控制经验

3. 具体需求

3.1 功能需求

3.1.1 文物浏览模块

1. 文物列表展示

- 描述：系统应提供多种布局方式展示文物列表
- 输入：用户选择的筛选条件和排序方式
- 处理：根据用户选择加载相应文物数据
- 输出：格式化的文物列表
- 优先级：高

具体功能点：

- 支持瀑布流、网格等多种布局方式
- 按分类（青铜器、陶瓷器、玉器、书画、货币、石器、雕塑等）展示文物
- 支持按时代、材质等属性筛选
- 支持按热度、时间等方式排序

2. 文物详情页

- 描述：展示单个文物的详细信息
- 输入：用户选择的文物ID
- 处理：从数据库获取文物详细信息
- 输出：文物详情页面
- 优先级：高

具体功能点：

- 展示文物高清图片（支持多角度查看）
- 显示文物基本信息（名称、年代、材质、尺寸等）
- 提供详细的历史背景和文化价值介绍
- 显示文物所在博物馆信息
- 相关文物推荐功能

3. 文物搜索功能

- 描述：允许用户通过关键词搜索文物
- 输入：用户输入的搜索关键词
- 处理：在数据库中匹配相关文物
- 输出：搜索结果列表
- 优先级：高

具体功能点：

- 关键词搜索（支持文物名称、时代、博物馆等多维度搜索）
- 分类筛选搜索
- 搜索历史记录
- 搜索结果排序和筛选

3.1.2 图像搜索模块

1. 以图搜图功能

- 描述：用户通过上传图片查找相似文物
- 输入：用户上传的图片
- 处理：基于HMS ML Kit进行图像识别与匹配
- 输出：相似度排序的文物列表
- 优先级：中

具体功能点：

- 允许用户上传本地图片
- 支持相机直接拍摄
- 基于HMS ML Kit的图像识别处理
- 按相似度排序展示搜索结果

- 搜索结果的缓存和历史记录

3.1.3 用户交互模块

1. 文物点赞功能

- 描述：允许用户对文物进行点赞
- 输入：用户点赞操作
- 处理：记录用户点赞状态并更新文物点赞数
- 输出：更新后的点赞状态和数量
- 优先级：中

具体功能点：

- 用户可对喜欢的文物进行点赞
- 显示文物的点赞数量
- 记录用户个人点赞历史

2. 评论系统

- 描述：提供文物评论功能
- 输入：用户提交的评论内容
- 处理：存储评论并关联到相应文物
- 输出：更新后的评论列表
- 优先级：中

具体功能点：

- 用户可对文物发表评论
- 支持文字和表情评论
- 评论列表展示和管理
- 评论点赞功能
- 评论举报功能

3. 收藏功能

- 描述：允许用户收藏感兴趣的文物
- 输入：用户收藏操作
- 处理：记录用户收藏状态
- 输出：更新后的收藏状态
- 优先级：高

具体功能点：

- 用户可收藏感兴趣的文物
- 收藏列表管理
- 收藏文物的分类组织
- 收藏夹创建和管理

4. 分享功能

- 描述：支持将文物信息分享到社交平台
- 输入：用户分享操作和选择的平台
- 处理：生成分享内容和链接

- 输出：分享结果
- 优先级：低

具体功能点：

- 支持将文物信息分享至社交平台（微信、微博等）
- 分享链接的优化展示
- 分享统计和分析

3.1.4 动态社区模块

1. 用户动态发布

- 描述：允许用户发布与文物相关的动态
- 输入：用户提交的动态内容
- 处理：存储动态内容并关联用户
- 输出：发布成功的动态
- 优先级：中

具体功能点：

- 支持发布文字、图片内容
- 与文物相关的心得、见解分享
- 动态审核机制

2. 动态流浏览

- 描述：展示用户关注的动态内容
- 输入：用户查看动态请求
- 处理：获取并组织相关动态
- 输出：动态列表
- 优先级：中

具体功能点：

- 时间线形式展示用户动态
- 关注好友的动态优先展示
- 动态内容的个性化推荐

3. 互动功能

- 描述：提供动态内容的互动功能
- 输入：用户互动操作
- 处理：记录互动状态并更新统计
- 输出：更新后的互动状态
- 优先级：低

具体功能点：

- 动态点赞
- 动态评论
- 热门动态推荐

3.1.5 用户管理模块

1. 用户账户管理

- 描述：提供用户账户相关功能
- 输入：用户提交的账户信息和操作
- 处理：账户信息验证和存储
- 输出：操作结果和状态
- 优先级：高

具体功能点：

- 注册与登录功能
- 个人信息维护
- 账号设置和安全管理
- 第三方账号绑定

2. 个人中心

- 描述：提供用户个人数据管理功能
- 输入：用户访问个人中心请求
- 处理：获取用户相关数据
- 输出：个人中心页面和数据
- 优先级：高

具体功能点：

- 个人动态管理
- 收藏文物管理
- 浏览历史记录
- 评论管理
- 通知消息中心

3.1.6 云服务功能

1. 云数据库同步

- 描述：实现数据的云端同步
- 输入：用户数据变更
- 处理：将数据同步至云端
- 输出：同步状态和结果
- 优先级：高

具体功能点：

- 用户数据云端存储和同步
- 文物信息的云端获取
- 离线数据缓存与同步

2. 云函数服务

- 描述：使用云端函数处理复杂业务逻辑
- 输入：业务请求和参数
- 处理：云端计算和处理

- 输出：处理结果
- 优先级：中

具体功能点：

- 复杂业务逻辑的云端处理
- 数据分析和推荐算法
- 资源密集型任务的云端执行

3. 云存储功能

- 描述：提供云端存储服务
- 输入：需要存储的数据
- 处理：数据加密和存储
- 输出：存储状态和访问地址
- 优先级：高

具体功能点：

- 用户上传图片的存储
- 文物高清图片的存储和访问
- 存储资源的权限管理

3.2 外部接口需求

3.2.1 用户界面

1. 设计风格

- 遵循HarmonyOS设计规范
- 采用中国传统文化元素
- 色彩搭配应体现古典与现代的融合

2. 主要界面

- 启动页：应用Logo和品牌展示
- 主页：导航栏、推荐文物、专题展览
- 文物详情页：图片展示区、信息区、互动区
- 搜索页：搜索框、筛选区、结果展示区
- 个人中心：用户信息、功能入口、数据统计

3.2.2 硬件接口

1. 设备要求

- 适配华为手机、平板等设备
- 支持不同分辨率和屏幕尺寸
- 适配触摸操作和手势控制

2. 传感器利用

- 摄像头：支持拍照搜索功能
- 陀螺仪：支持文物3D模型的交互浏览

3.2.3 软件接口

1. 系统集成

- HMS Core集成
- 华为账号服务集成
- 华为推送服务集成

2. 第三方服务接口

- 社交平台分享接口
- 支付接口（可选功能）
- 地图服务接口（展示博物馆位置）

3.2.4 通信接口

1. 网络通信

- HTTPS协议保障数据传输安全
- WebSocket用于实时通知
- RESTful API设计规范

2. 数据格式

- JSON格式数据交换
- 图片传输采用Base64或二进制流

3.3 性能需求

1. 响应时间

- 页面加载时间不超过3秒
- 搜索结果响应时间不超过2秒
- 图像搜索响应时间不超过5秒
- 文物图片加载时间不超过1.5秒

2. 并发处理

- 支持多用户同时访问和操作
- 单服务器支持不少于1000个并发连接
- 峰值访问应对策略

3. 资源占用

- 应用安装包大小控制在50MB以内
- 运行内存占用不超过设备可用内存的30%
- CPU使用率控制在合理范围内

3.4 属性

3.4.1 可靠性

1. 错误处理

- 网络异常时提供友好提示
- 数据加载失败时的重试机制
- 系统崩溃的自动恢复功能

2. 数据备份

- 用户数据的定期备份
- 数据恢复机制
- 关键操作的事务处理

3. 稳定性

- 系统7*24小时稳定运行
- 定期维护计划
- 故障转移机制

3.4.2 安全性

1. 用户认证

- 安全的登录认证机制
- 账号异常行为监控
- 多重身份验证选项

2. 数据保护

- 用户隐私数据加密存储
- 防止未授权访问
- 数据传输加密
- 敏感信息脱敏处理

3. 漏洞防护

- SQL注入防护
- XSS攻击防护
- CSRF防护
- 定期安全审计

3.4.3 可用性

1. 界面设计

- 符合HarmonyOS设计规范
- 具有中国传统文化特色的UI界面
- 良好的用户体验和交互设计
- 操作流程直观简明

2. 可访问性

- 支持不同屏幕尺寸的适配

- 考虑不同用户群体的操作习惯
- 支持弱视用户的界面放大
- 提供语音辅助功能

3. 国际化

- 支持多语言界面（中文、英文）
- 不同文化背景用户的适配
- 时间和日期格式本地化

3.4.4 可维护性

1. 模块化设计

- 系统按功能模块划分
- 模块之间接口清晰
- 代码重用性高

2. 扩展性

- 支持新功能的便捷添加
- 系统架构灵活可扩展
- API版本兼容性考虑

3. 文档完善

- 详细的系统设计文档
- 完整的API文档
- 代码注释规范

4. 系统架构

4.1 前端架构

1. 开发框架

- HarmonyOS开发套件
- ArkTS开发语言
- MVVM架构模式

2. 核心组件

- 页面导航组件
- 文物展示组件
- 搜索组件
- 用户交互组件
- 图片处理组件

3. 数据流管理

- 状态管理机制
- 内存优化策略
- 缓存机制

4.2 后端架构

1. 服务器端

- Node.js服务器
- Express框架
- MySQL数据库
- Sequelize ORM

2. 云服务

- 华为云AGC服务
- 云数据库
- 云函数
- 云存储

3. API设计

- RESTful API设计
- 版本控制策略
- 接口文档标准

4.3 数据架构

1. 核心数据模型

- 文物信息模型
- 用户模型
- 评论模型
- 动态模型
- 收藏模型

2. 数据关系

- 用户与文物的多对多关系
- 文物与博物馆的多对一关系
- 用户与评论的一对多关系

3. 数据流转

- 前后端数据交换流程
- 缓存与数据库同步策略
- 数据加载与更新机制

5. 数据需求

5.1 数据实体

1. 文物实体

- ID: 唯一标识符
- 名称: 文物名称
- 描述: 文物详细描述

- 年代：文物所属时代
- 材质：文物的制作材质
- 尺寸：文物的物理尺寸
- 博物馆ID：所在博物馆标识
- 分类：文物分类（青铜器、陶瓷器等）
- 图片资源：文物图片链接集合
- 浏览次数：文物被浏览的次数
- 点赞数：文物获得的点赞数量
- 创建时间：记录创建时间
- 更新时间：记录更新时间

2. 用户实体

- ID：唯一标识符
- 用户名：用户登录名
- 密码：加密存储的密码
- 昵称：用户展示名称
- 头像：用户头像链接
- 邮箱：用户邮箱地址
- 手机号：用户手机号码
- 注册时间：用户注册时间
- 最后登录时间：最近登录时间
- 状态：账号状态（正常、禁用等）

3. 评论实体

- ID：唯一标识符
- 文物ID：关联的文物ID
- 用户ID：发表评论的用户ID
- 内容：评论内容
- 时间：评论发布时间
- 点赞数：评论获得的点赞数量
- 状态：评论状态（正常、隐藏等）

4. 收藏实体

- ID：唯一标识符
- 用户ID：收藏用户的ID
- 文物ID：被收藏的文物ID
- 收藏夹ID：所属收藏夹ID
- 收藏时间：收藏操作的时间
- 备注：用户添加的收藏备注

5.2 数据量估计

- 文物信息：初始约10,000条，每年增加约1,000条
- 用户数据：初始约5,000用户，年增长率约30%
- 评论数据：每个文物平均10条评论
- 收藏数据：每个用户平均收藏15个文物

5.3 数据备份策略

- 数据库每日全量备份
- 关键数据实时同步备份
- 用户数据定期增量备份
- 备份数据保存期限不少于30天

6. 附录

6.1 术语表

术语	说明
掌上博物馆	本系统名称，指基于HarmonyOS的移动应用
HarmonyOS	华为开发的分布式操作系统，本系统的开发平台
HMS	华为移动服务，提供应用开发所需的云服务和能力
瀑布流布局	一种网页布局，其中内容按不同高度排列，形似瀑布

6.2 参考文献

1. HarmonyOS应用开发指南
2. 华为云服务开发文档
3. 博物馆数字化展示标准规范

6.3 相关文档

- 掌上博物馆系统设计文档
- 掌上博物馆测试计划
- 掌上博物馆部署文档

后台管理子系统

项目名称：海外文物后台管理系统

文档版本：v1.0

编写日期：2025年6月

编写团队：Team5海外文物后台管理系统开发组

1. 引言

1.1 编写目的

本需求规格说明书旨在全面、准确地描述海外文物后台管理系统的功能需求、性能需求、接口需求以及设计约束。本文档将作为系统开发、测试、验收和维护的重要依据，为项目相关人员提供统一的需求理解基础。

本文档的预期读者包括：

- 项目管理人员和决策者
- 系统架构师和开发工程师
- 测试工程师和质量保证人员
- 用户代表和业务分析师
- 系统运维和技术支持人员

1.2 项目背景

随着文化遗产数字化保护意识的不断提升，海外文物的数字化管理和展示成为文博机构的重要需求。传统的文物管理方式存在信息分散、管理效率低下、缺乏统一标准等问题。为解决这些痛点，需要构建一个专业的海外文物后台管理系统，实现文物信息的数字化管理、用户权限的精细化控制、内容审核的规范化流程以及数据分析的可视化展示。

1.3 项目范围

本系统覆盖海外文物展示平台的完整后台管理需求，包括但不限于：

核心业务范围：

- 文物数字化信息管理（录入、编辑、分类、展示）
- 多端用户统一管理（Web端管理员、移动端用户）
- 内容审核流程管理（评论审核、帖子管理）
- 数据统计分析和可视化展示
- 系统权限安全控制和日志审计

技术实现范围：

- 基于Plotly Dash框架的Web管理界面
- 支持MySQL和SQLite的数据存储方案
- 基于RBAC模型的权限管理机制
- 任务调度和自动化运维功能
- 响应式设计支持多设备访问

用户服务范围：

- 博物馆管理员的日常运营管理
- 文物研究人员的专业工具支持
- 系统运维人员的技术管理平台
- 移动端用户的间接服务支撑

1.4 定义、首字母缩写词和缩略语

术语	定义
RBAC	Role-Based Access Control, 基于角色的访问控制
ORM	Object-Relational Mapping, 对象关系映射
MVC	Model-View-Controller, 模型-视图-控制器架构模式
WSGI	Web Server Gateway Interface, Web服务器网关接口
DAO	Data Access Object, 数据访问对象
UI	User Interface, 用户界面
API	Application Programming Interface, 应用程序编程接口
JSON	JavaScript Object Notation, JavaScript对象表示法
HTTP/HTTPS	超文本传输协议/安全超文本传输协议
SQL	Structured Query Language, 结构化查询语言

1.5 参考资料

- 《软件需求工程》第2版 - 毋国庆等著
- 《软件工程：实践者的研究方法》第8版 - Roger S. Pressman
- 《Plotly Dash官方文档》 - <https://dash.plotly.com/>
- 《MySQL 8.0官方文档》 - <https://dev.mysql.com/doc/>
- 《Python 3.8官方文档》 - <https://docs.python.org/3.8/>
- 海外文物后台管理系统设计报告 v2.0
- 海外文物后台管理系统用户使用手册 v1.5

2. 产品概述

2.1 产品定位

海外文物后台管理系统是一个专业的文博机构管理平台，致力于为海外文物的数字化保护、展示和管理提供全方位的技术支撑。系统以用户体验为中心，以数据安全为保障，以业务流程为导向，构建现代化的文物管理生态体系。

核心价值主张：

- **专业化**：针对文博行业特点定制的管理功能
- **标准化**：建立统一的文物信息管理规范
- **智能化**：数据可视化分析和智能审核机制
- **安全化**：多层次权限控制和数据保护措施

2.2 产品功能概述

系统围绕文物管理的核心业务流程，提供六大主要功能模块：

2.2.1 用户管理中心

- **多端用户统一管理**：支持Web端管理员和移动端用户的集中管理
- **角色权限精细控制**：基于RBAC模型的多层级权限分配机制
- **用户生命周期管理**：从注册到注销的完整用户管理流程
- **用户行为分析**：用户活跃度监控和行为数据统计

2.2.2 文物信息管理

- **数字化信息录入**：支持文物基本信息、图片、描述等多媒体内容
- **分类标签管理**：按朝代、材质、类型等多维度分类体系
- **库存状态管理**：实时跟踪文物的在库、展出、借出等状态
- **搜索检索功能**：多条件组合搜索和智能推荐

2.2.3 内容审核系统

- **评论审核流程**：支持人工审核和自动过滤的混合审核机制
- **帖子内容管理**：用户发布内容的审核、编辑和撤销功能
- **敏感词过滤**：基于关键词库的自动内容过滤
- **审核记录追溯**：完整的审核日志和决策记录

2.2.4 数据可视化看板

- **实时数据监控**：访问量、评论量、用户活跃度等关键指标
- **多维度统计分析**：支持时间、分类、地域等多维度数据分析
- **图表可视化展示**：饼图、柱状图、折线图等多种图表类型
- **报表导出功能**：支持PDF、Excel等格式的数据导出

2.2.5 任务调度中心

- **定时任务管理**：支持周期性任务和一次性任务的配置
- **数据同步备份**：自动化数据备份和恢复机制
- **消息通知推送**：集成多种通知渠道的消息推送服务
- **脚本执行引擎**：支持Python、Shell等多种脚本类型

2.2.6 系统管理维护

- **权限角色配置**：系统角色和权限的灵活配置管理
- **操作日志审计**：完整的用户操作记录和系统日志
- **数据库维护**：一键备份恢复和数据库性能优化
- **系统配置管理**：运行参数和环境配置的集中管理

2.3 用户特征

2.3.1 主要用户群体

博物馆管理员

- **角色特征：**负责文物日常管理和展览策划的专业人员
- **技术水平：**具备基本的计算机操作能力，熟悉文博业务流程
- **使用频率：**每日高频使用，主要集中在工作时间
- **核心需求：**高效的文物管理工具，直观的数据展示界面

文物研究人员

- **角色特征：**从事文物研究和学术工作的专业学者
- **技术水平：**具备较强的信息检索和分析能力
- **使用频率：**根据研究需要灵活使用，注重数据的准确性和完整性
- **核心需求：**强大的搜索功能，详细的文物信息，数据导出能力

系统运维人员

- **角色特征：**负责系统技术维护和运行保障的IT专业人员
- **技术水平：**具备专业的技术背景和系统管理经验
- **使用频率：**按需使用，主要在系统维护和故障处理时
- **核心需求：**完善的监控工具，便捷的维护功能，详细的日志信息

2.3.2 用户使用环境

办公环境：

- 主要在办公室固定工作站使用
- 网络环境稳定，带宽充足
- 使用现代化的浏览器和操作系统

移动办公：

- 支持在平板电脑和移动设备上访问
- 可能在网络环境不稳定的情况下使用
- 需要响应式设计适配不同屏幕尺寸

远程访问：

- 支持通过VPN等方式远程访问系统
- 需要考虑网络延迟和安全性问题
- 提供离线功能和数据同步机制

2.4 运行环境

2.4.1 硬件环境

服务器端最低配置：

- CPU：4核心2.0GHz以上
- 内存：8GB RAM
- 存储：100GB可用空间（SSD推荐）
- 网络：千兆网卡

服务器端推荐配置：

- CPU：8核心3.0GHz以上
- 内存：16GB RAM或以上
- 存储：500GB可用空间（SSD）
- 网络：千兆网卡，双网卡冗余

客户端最低配置：

- CPU：双核心1.5GHz以上
- 内存：4GB RAM
- 存储：5GB可用空间
- 网络：宽带接入

2.4.2 软件环境

服务器端软件要求：

- 操作系统：Linux (Ubuntu 18.04+/CentOS 7+) 或Windows Server 2016+
- Python运行环境：Python 3.8+
- 数据库：MySQL 5.7+或SQLite 3.0+
- Web服务器：Nginx（推荐）或Apache
- 进程管理：Supervisor（Linux）或Windows服务

客户端软件要求：

- 操作系统：Windows 10+、macOS 10.14+、Linux（主流发行版）
- 浏览器：Chrome 80+、Firefox 75+、Safari 13+、Edge 80+
- 分辨率：最低1024×768，推荐1920×1080或以上

2.4.3 网络环境

带宽要求：

- 单用户最低：2Mbps下行，1Mbps上行
- 单用户推荐：10Mbps下行，5Mbps上行
- 服务器带宽：根据并发用户数量，建议100Mbps+

网络协议：

- 支持HTTP/HTTPS协议

- 使用标准80/443端口或自定义端口8090
- 支持WebSocket长连接（实时功能）

安全要求：

- 生产环境必须启用HTTPS加密
- 支持防火墙和VPN接入
- 建议配置CDN加速和DDoS防护

2.5 设计和实现约束

2.5.1 技术约束

开发框架约束：

- 必须基于Plotly Dash框架构建Web界面
- 使用Feffery生态组件库实现UI组件
- 采用Peewee ORM进行数据库操作
- 遵循Python PEP 8编码规范

数据库约束：

- 支持MySQL和SQLite两种数据库类型
- 数据库字符集必须为UTF-8
- 单表数据量建议不超过1000万条记录
- 必须实现数据库连接池管理

浏览器兼容性约束：

- 必须支持主流现代浏览器最近3个版本
- 不支持IE浏览器
- 必须实现响应式设计，支持移动设备访问