**后台管理子系统  
项目规划书**

**一、项目背景**

以国家海外文化遗产保护战略为核心，通过国际合作与技术融合，构建覆盖数据采集、管理、审核、展示的全链条后台支撑体系，服务于学术研究、文物保护与公众文化传播的多维目标。其技术实现需平衡高性能、安全性与用户友好性，同时适配多子系统的协同需求（如知识图谱、移动端应用）。

后台管理系统能支持海外藏中国文物知识管理与服务平台的全系统数据管理与运营，聚焦五大核心功能（用户管理、信息审核、数据管理、数据备份恢复、日志管理）。

**二、项目范围**

1.本项目主要技术路线

前端框架：Vue + Element UI

后端框架：Spring Boot

数据库：MySQL

2.主要功能

1）用户管理：管理后台管理子系统用户、掌上博物馆用户、知识服务子系统用户的基础信息。如：对用户信息的增删查改功能；用户权限的管理，如登录、点赞、评论等用户功能的管理。例如，如果掌上博物馆用户发表不良评论，则停止其发表评论的权限。

2）信息审核：审核用户发表的留言、图片、音视频等内容的功能。对于审核不通过的内容会被屏蔽，同时对该用户进行一定程度惩罚，如禁止发评论等。自动审核功能：对于留言，可以设置敏感词，当用户提交留言出现敏感词时，进行自动屏蔽；对于图片，需要判断是否为不良图片，如果审核不通过，则不显示。人工审核功能：后台管理员或审核人员人工检查用户提交的文本、图片、音视频。

3）数据管理：管理1-3中涉及的所有数据，对所有数据可以进行增删改查等操作，支持单个操作和批量操作。

4）数据备份和恢复：支持数据库的备份和恢复。实现手动备份恢复或定时备份功能。支持查看所有的备份和恢复记录，显示记录时间等相关功能。可通过点击备份记录来将数据库恢复到该备份记录点上。

5）日志管理：查看和检索后台管理子系统的操作日志。记录包括管理员等用户对后台数据的操作记录，数据库的备份还原等记录。

6）相关联系统数据交互：需与知识服务子系统（Web）、掌上博物馆（App）、问答子系统对接用户权限与内容审核接口。

3.交付物清单

项目管理计划、需求规格说明书、设计报告、测试报告、用户手册、每周会议纲要、源代码、可执行系统

**三、周计划**

第8周

- 确定团队分工

- 创建GitHub仓库，初始化项目结构

- 完成程序demo设计

- 发布《项目管理计划》初稿（含技术路线）

第9周

- 完成《需求规格说明书》初稿

- 明确跨子系统接口

- 启动核心模块开发（用户管理、审核模块基础框架）

- 启动开发数据备份模块、前端管理界面原型

第10周

- 完成《设计报告》初稿

- 完善模块开发

第11-12周

- 完成用户权限控制

- 实现敏感词过滤与图片审核

- 启动开发日志管理模块

- 每周增加1次进度同步会

第13周

- 系统进一步测试优化

- 每人演示负责模块

- 提交可执行系统

第14周

- 进行总结汇报演讲准备

- 制作PPT

- 录制演示视频

第15周

- 与掌上博物馆、知识服务子系统联调

- 提交最终交付物（代码、文档、APK协同文件）

**四、团队分工**

项目经理

* 制定并更新《项目管理计划》，跟踪周进度
* 协调跨团队沟通
* 统筹进度，主持关键会议

数据架构师

* 数据库设计
* 数据管理API开发
* 数据备份与恢复模块开发

前端开发工程师

* 后台管理系统前端开发

后端开发工程师

* 用户权限模块开发
* 内容审核模块开发
* 日志管理模块开发
* 跨子系统接口开发

软件测试工程师

* 编写自动化测试脚本测试
* 制作演示视频和汇报PPT

**五、协作与开发规范**

代码管理

* Git分支策略：

分支保护：main分支仅允许通过Pull Request合并

开发分支：feature/management-system（按模块命名）

文档更新机制

* 用户手册：需求变更时更新（如新增“文物数据批量导入”操作说明）
* 接口文档：每次接口变更后同步注释测试与交付物
* 测试报告：每次测试时提交

跨团队协作

* 接口对齐：每周与其他团队核对审核状态同步协议
* 数据一致性：其他系统数据变更时，触发后台管理数据备份任务

**六、风险管理**

1.风险分类

1）技术风险

* 技术依赖风险：第三方内容审核API（如阿里云）出现高延迟、服务中断或调用成本超支
* 数据安全风险：数据库备份文件损坏或恢复失败，导致文物元数据丢失
* 性能风险：高并发场景下（如文物数据批量导入）引发数据库锁表或服务崩溃
* 系统集成风险：跨子系统接口（如知识图谱数据同步、App端权限状态同步）协议不一致或响应超时

2）管理风险

* 项目进度延误：项目出现技术难题，或者团队成员间缺乏交流，问题解决时间过长导致项目延误

2.风险评估

|  |  |
| --- | --- |
| 风险类型 | 风险等级 |
| 技术依赖 | 高 |
| 数据安全 | 极高 |
| 性能风险 | 中 |
| 系统集成 | 高 |
| 管理风险 | 中 |

3、风险应对预案

案例1：内容审核API中断

应对预案：使用两家以上的内容审核API，如API的提供的服务都不可用，则启用本地内容审核库+人工审核。

案例2：数据库备份文件损坏或恢复失败

应对预案：多进行备份，建立从数据库，备份文件上。传至OSS对象存储，定期检查主从数据库一致性，发生故障及时进行从数据库切换。

**七、会议与沟通调整**

每日站会（线上，10分钟）：同步当日开发目标

周会（线上/线下，30分钟）：评审文档初稿

协作工具：Github

**八、项目质量保证与评估**

1. 代码质量管控

制定代码规范，在合并前进行代码审查。

2. 管理系统测试

进行单元测试、集成测试、端到端测试、性能测试和安全测试等测试，确保系统的各部分组件能正常运行，保证系统在高并发环境下的稳定性，并优化系统性能，减少资源占用。

3. 文档质量管理

需求规格说明书，设计报告，用户手册，接口文档等文档建立审核机制，关键文档需团队会确认（如接口文档变更需全体后端成员确认）。

4. 项目评估

在每周会议上评估系统开发和运行情况，评估系统出现的问题和解决方法以及预计剩余工作所需的时间和人力资源。