学院生活平台系统

可行性分析研究报告

班级：　　学号：　　姓名：

本报告说明拟议中的学院生活平台系统在技术、经济和社会条件方面的可行性，对各类可行方案进行评估，并给出倾向性意见。

本报告基于仓库 `campus-hub-next` 已实现的代码与目录结构（Next.js 16、React 19、Supabase、Drizzle ORM），结合学校现状进行分析。报告力求保持格式、章节与编号与模板一致，以便在后续迭代中直接替换与扩展。

本报告由以下表格组成：

可行性分析研究报告目录	文档序号
1. 引言	A1.
2. 可行性研究的前提	A2.
2.1. 项目基本要求	A2.1.
2.2. 项目基本目标	A2.2.
2.3. 项目开发条件、假定和限制	A2.3.
3. 对现有系统的分析	A3.
3.1. 现有系统的处理流程和数据流程	A3.1.
3.2. 现有系统简明情况	A3.2.
4. 对所建议系统的分析	A4.
4.1. 所建议系统的概述	A4.1.
4.2. 所建议系统的处理流程和数据流程	A4.2.
4.3. 改进、影响和局限性	A4.3.
5. 可行性分析	A5.
5.1. 技术条件可行性分析	A5.1.
5.2. 经济可行性分析	A5.2.
5.3. 社会因素方面的可行性分析	A5.3.
6. 可行性分析结论	A6.

本报告由受托单位提出，作为委托单位对学院生活平台系统开发的决策依据。

系统分析员:_____　　_____

(签名)

日 期: 2025 年 12 月 22 日

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	1. 引言	子文档序号	A1.
编者		编写日期	2025 年 12 月 22 日

（1）编写目的

学院生活平台系统面向在校学生与管理人员，覆盖课程资源分享、功能房预约、通知公告、数字图书馆、失物招领等业务场景。仓库 `campus-hub-next` 已基于 Next.js 16 + React 19 + Supabase + Drizzle ORM 完成基础框架和公告（notice）MVP，实现 Portal/Console 双端页面与 BFF API。本文档用于：

评估当前代码与架构能否支撑后续各模块上线，并识别需要优先补齐的能力。

为学校信息化部门和业务管理者提供立项与迭代决策依据。

明确进一步设计、开发、部署的约束、风险与资源需求，形成可落地的实施计划。

记录本阶段调研结论，便于后续评审、交接与审计。

（2）项目背景

学生生活信息分散在群聊、纸质登记、零散网页，缺乏统一入口、审计与权限控制。

学校已有 Supabase Postgres 与 Storage 资源，可利用免费层完成 MVP，降低初期成本。

代码已提供 `app/(portal)` 与 `app/(console)` 双端布局，Route Handler 通过 `requireUser()`、`listPortalNotices()` 等实现鉴权与分页

查询，说明技术路径可落地。

数据访问通过 `packages/db` 的 Drizzle schema 与迁移脚本管理，`lib/modules/notices/notices.service.ts` 展示了内容可见性、附件签名下载、审计日志等能力，为其他模块复用提供示例。

AGENTS 文档明确了模块化工作流、权限前缀、Adapter 抽象层，使系统具备可持续扩展和替换后端云厂商的能力。

（3）符号、缩略语和定义

CHN：Campus Hub Next（学院生活平台系统代号）。

Supabase：托管 Postgres、Auth、Storage 的后端服务。

Portal：面向学生的前台界面，如 `app/(portal)/notices`。

Console：面向管理员的后台界面，如 `/console/notices`。

BFF：Next.js Route Handler 作为后台接口层。

RLS：Row Level Security，Supabase 数据库行级安全策略。

（4）参考资料

仓库根目录 `AGENTS.md` 的技术栈、抽象层约定、API 设计规范与工作流。

`README.md` 中的启动步骤、环境变量说明及模块进度。

`packages/db/migrations/0001_baseline.sql` 与 `packages/db/src/schema` 提供的表结构与迁移。

`app/api/notices/route.ts`、`lib/modules/notices/notices.service.ts` 体

现的接口、鉴权、审计与数据权限实现。

`docs/requirements`、`docs/api` 目录下的需求与契约占位结构，用于后续补充。

相关课程教材：《软件工程导论》《实用软件工程》与 Next.js、Supabase 官方文档。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	2.1. 项目基本要求	子文档序号	A2.1.
编者		编写日期	2025 年 12 月 22 日

学院生活平台系统应便于统一管理校园生活场景，提高通知、预约、资源分享的工作效率，减轻管理人员压力。

（1）主要功能

通知公告：发布、编辑、撤回、置顶公告；按角色/部门/岗位可见范围推送；Portal 侧支持分页、搜索、已读/未读筛选，附件通过 Supabase Storage 签名链接下载（参见 listPortalNotices 与附件签名实现）。

Console 支持草稿、发布、撤回、删除、置顶等状态流转；notice_reads 记录已读，notice_attachments 管理附件。

内容校验禁止内联 HTML（见 assertNoInlineHtml），减少 XSS

风险。

课程资源与学习资料：材料、课程资源模块提供上传、分类、搜索与下载入口，前后台分离页面已预留（app/(portal)/materials、app/(console)/console/materials），可复用公告的附件签名与权限策略。

功能房预约：学生提交预约，管理员审核、批准或拒绝，避免撞档与重复占用。接口路径 app/api/facilities 预留，未来接入审批流与时间冲突校验。

数字图书馆与资料阅览：支持电子书元数据管理、借阅/下载计数、格式过滤，后续可对接外部 ISBN API。

失物招领：提交拾到/丢失信息，图片走 Storage 保存，状态可变更为“已认领”；Portal 提供搜索和筛选。

问卷与投票：app/(portal)/surveys、app/(portal)/votes 及对应后台，支持针对特定群体发起问卷/投票，统计结果导出。

权限与角色：campus:<module>:<op> 权限码体系，Console 按权限展示菜单；lib/auth/permissions.ts 与 user_roles 表实现细粒度控制，配合数据权限限定操作范围。

审计与数据权限：lib/modules/audit 记录操作轨迹；dataPermission 控制按创建人或管理范围过滤；contentVisibility 负责可见性计算，形成“谁能看到/谁能操作”的闭环。

个人中心与设置：app/(portal)/profile、/profiles、/settings 等页面预留，支持用户信息维护、偏好设置与通知订阅。

多端适配与主题：Tailwind 4 + shadcn/ui，默认亮/暗主题切换，

响应式布局适配桌面端与移动端。

平台整体采用模块化路由与 UI 组件复用策略：所有网络请求统一通过封装的 API 客户端，页面层仅负责状态与展示；Console/Portal 共用基础布局与权限校验，便于后续新增模块时快速复制粘贴骨架。

跨模块共享的通用能力包括：配置校验（lib/config）、AI 适配器占位、StorageAdapter 抽象、logger、外部 HTTP 客户端封装等，确保业务代码不直接依赖第三方。

UI 组件库与表单/表格封装将逐步抽出到 components 或 features，便于复用与主题切换。

接口遵循 /api/<resource> 与 /api/console/<resource> 统一前缀，命名与权限前缀一致，方便前端调用与调试。

（2）性能要求

前端采用 Next.js 16 App Router，支持 SSR/ISR 降低首屏时延；Tailwind CSS 4 无构建模式减轻客户端负担，减少 CSS 体积。

后端 BFF 通过 Drizzle SQL 直接落地 Postgres 查询，分页默认 20 条（pageSize 限制 50），排序字段限定，避免全表扫描；listConsolePublishedNoticesExpiringSoon 提供即将过期公告的轻量查询示例。

Storage 读取使用 60 秒签名 URL，减少重复鉴权开销且控制有效期；上传限制 20MB 以内，避免过大文件阻塞。

目标：Portal 列表接口在 200ms 内返回（不含网络延迟），并发 200 请求/分钟不降级；后台批量操作需事务化，确保数据一致；文件上传/下载成功率 99.9%。

构建脚本 pnpm build 采用 Webpack，要求在 Vercel/Supabase Edge 环境内 3 分钟内完成；本地开发 pnpm dev 冷启动不超过 15 秒。

监控指标：API 错误率 <1%；数据库慢查询（>300ms）需告警；存储签名失败率 <0.5%。

性能测试计划：使用 Postman/Apifox + k6 进行接口压测，覆盖公告列表/详情、附件上传下载、预约提交等高频链路。

缓存策略：列表接口可使用 Edge Cache 或浏览器缓存 ETag；元数据变更时通过时间戳或版本号失效。

队列与批处理：批量导入资源、问卷答卷统计可在后台任务中执行，避免阻塞请求。

资源占用控制：长内容采用 Markdown 文本存储，附件走对象存储，避免数据库膨胀；定期归档过期数据。

前端优化：按需加载组件，使用 react-markdown + remark-gfm 渲染 Markdown，减少 bundle 体积。

（3）系统的输入、输出

输入：用户注册/登录凭证（Supabase Auth），公告标题/Markdown 正文/附件文件流，预约信息（场地、时段、用途），资源元数据（课

程名、文件类型、标签），问卷题目与选项，权限分配指令，搜索条件与筛选参数等。

输出：分页 JSON（含 total/page/pageSize）、详情 JSON、签名下载链接、审核状态更新结果、导出数据（后续可扩展为 CSV/Excel）、统计数据（未读公告数、即将过期公告数、预约占用率）以及 Console 看板展示。

接口形式：全部以 /api/** Route Handler 提供 REST 风格 JSON，Portal 侧通过封装的 API 客户端调用，避免裸 fetch；错误统一使用 jsonError 结构，包含 code、message、detail。

表现层输出：前端组件使用 shadcn/ui 与 Radix，无障碍属性遵循组件默认规范；Markdown 正文通过 React Markdown 渲染，支持 GFM。

错误码约定：常用 BAD_REQUEST、FORBIDDEN、NOT_FOUND、INTERNAL_ERROR 等，前端按 code 显示提示与重试入口。

文件与媒体：图片、PDF、Office 文档均走 Storage；大文件上传前端应校验大小与类型，必要时分片或压缩。

分页/筛选参数：统一使用 page、pageSize、q、sortBy、sortOrder、includeExpired 等，保持接口一致性与可预测性。

导出/导入：后续扩展 CSV/Excel 导出与批量导入时，需要在 Service 层加入任务队列与审计记录，避免阻塞在线请求。

（4）安全和保密要求

Supabase Auth 提供会话管理；后端使用 requireUser()、hasPerm() 强制登录与授权；敏感接口附带数据范围校验（见 buildUserIdDataScopeCondition）。

附件与大文件仅通过 Storage 签名 URL 暴露，过期自动失效；文件名清洗防止路径注入；上传大小与类型可配置。

审计日志写入 audit_logs，记录操作人、动作、目标与差异快照；异常场景同样记录以便追责；Console 可扩展审计查看入口。

建议部署阶段开启 HTTPS，配置环境变量最小权限（单独 Service Role Key 仅在后台使用），并在 Supabase 开启 RLS，限制跨表访问；定期备份数据库与存储。

对内容安全进行预处理：公告正文禁止内联 HTML，后续可增加敏感词过滤；导出接口需校验权限与访问频率，防止批量爬取。

账号安全：密码策略需满足复杂度要求，建议开启 Magic Link 或 2FA；会话过期后需强制刷新。

日志安全：生产环境避免输出敏感字段，必要时对日志进行脱敏；审计日志需至少保留 12 个月。

备份与恢复：数据库每日自动快照，Storage 周期性归档；演练恢复流程，确保 RPO/RTO 满足学校要求。

漏洞防护：定期升级依赖（pnpm outdated），使用 ESLint/TypeScript 捕获潜在漏洞；对上传文件执行 MIME 与扩展名双重校验。

业务约束与兼容性：

所有模块遵循中文界面与说明，后续可按需增加英文版；默认使用 UTF-8（无 BOM）编码。

前端禁止在业务组件中直接调用外部 HTTP，必须经 Service/API 客户端封装。

数据库操作统一使用 Drizzle，禁止在 Route Handler 中拼接 SQL；所有关键约束（唯一索引、检查约束）需在 schema 中显式声明。

文件存储仅限 Supabase Storage 或实现 StorageAdapter 的后端，禁止直接写本地磁盘，以便云端部署。

权限命名必须遵循 campus:<module>:<op>，并在前后端保持一致；页面按钮显隐与接口校验需双保险。

路由命名遵循 /api/<resource>、/api/me/<resource>、/api/console/<resource> 规范，保持可预测性。

审计日志需在成功与失败两种分支均记录，确保可追溯；敏感字段（密码、密钥）不得写入日志。

任何导出接口需控制输出条数与频率，防止批量爬取或资源过载。

AI/外部接口调用需通过 Adapter，默认关闭，后续根据需求启用并添加计费与额度控制。

代码提交遵循 Conventional Commits，合并前需通过最小

lint/typecheck/测试。

（5）项目与其他软件系统的关系

依赖 Supabase Postgres/Storage/Auth 作为后端基础设施，可平滑迁移到自建 Postgres + S3 兼容存储（遵循 Adapter 设计）。

前端可托管在 Vercel，利用其 Edge CDN；若需内网部署，可改用自建 Next.js 服务器或容器化部署。

未来可接入统一身份（SSO）、邮件/短信网关、AI 模型（封装在 aiClient），通过 Adapter 扩展，不破坏现有业务层。

可与校园一卡通、门禁或教务系统进行数据对接（如场地状态、课程信息），通过独立 HTTP 客户端模块实现。

若需与旧系统并行运行，可通过数据同步脚本定期导入/导出，确保切换期间数据一致；同步过程需带审计与去重规则。

与学校信息门户或微信公众号可建立跳转与轻量嵌入，提供统一入口体验。

（6）完成期限要求

第一阶段：公告模块 MVP，完成基础权限、审计与附件签名，Route Handler/Service/Repository 分层可用。

第二阶段：功能房预约、失物招领、课程资源列表与上传，实现核心流程可用；补充 API 文档与最小测试。

第三阶段：问卷/投票、数字图书馆、统计报表与运维监控，完

成上线前性能与安全加固；准备导出/导入与故障应急方案。

验收前需完成：Supabase 数据迁移执行验证、Portal/Console 端到端演练、最小用户培训与操作手册。

每阶段结束需提交：功能演示、API 契约更新、测试结果、已知问题清单与下一步计划。

上线前进行灰度发布与回滚演练，确保遇到异常可在 15 分钟内恢复。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	2.2. 项目基本目标	子文档序号	A2.2.
编者		编写日期	2025 年 12 月 22 日

随着校园规模扩张、信息流量增长，每年公告、预约、资源、失物等数据呈指数上升。通过统一的学院生活平台，可实现规范化管理、科学统计与快速查询，减少管理工作量，为学生与管理者提供高效体验。

系统的基本目标如下：

(1) 时间经济性：利用 Next.js SSR 与分页/排序查询，保障查询与发布在秒级完成；DevOps 管线采用 pnpm 快速安装、增量构建，减少迭代时间。

(2) 可靠性：Service 层通过事务与并发控制（如公告已读计数的事务处理）保证数据一致；异常统一 jsonError 返回；提供健康检查与日志输出。

(3) 可理解性：前后端按模块分层（Route Handler → Service → Repository/Adapter），命名与权限前缀统一，目录结构清晰，便于培训、评审与交接。

(4) 可维护性与适应性：Drizzle schema 可迁移，Adapter 隔离 Auth/Storage/AI，Tailwind 主题统一，支持快速迭代与功能扩展；采用 TypeScript 提供类型约束。

(5) 可用性：Portal/Console 双端覆盖主要角色，列表、搜索、筛选、编辑、审核流程可闭环；离线导出与附件查看由 Storage 保证；前端提供加载态、错误提示。

(6) 安全保密性：权限矩阵、审计日志、签名 URL、防内联 HTML 校验共同降低安全风险；可进一步加密敏感字段与启用双因素认证。

(7) 可观测性：预留日志、审计、慢查询与指标采集接口，方便后续接入 APM 或集中日志，提升运维效率。

以上目标将转化为验收指标：核心接口成功率、响应时延、公告到达率、预约冲突率、用户满意度（问卷）等，作为阶段性验收与持续改进依据。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	2.3. 项目开发条件、假定和限制	子文档序号	A2.3.
编者		编写日期	2025 年 12 月 22 日

（1）所建议系统运行的寿命的最小值

系统设计寿命不少于 5 年，要求数据模式与依赖可演进；通过 Adapter 隔离云厂商，降低替换成本，数据库迁移脚本保持向后兼容。

（2）进行系统方案选择比较的时间

方案比选与 PoC 验证时间为 3 周，涵盖 Supabase 免费层容量测试、Vercel 部署验证、性能压测与安全基线检查。

（3）经费、投资方面来源

以学校信息化专项与教学改革经费为主，必要时申请创新基金。

云资源优先使用 Supabase 与 Vercel 免费配额，必要时按量升级低档付费；测试环境与生产环境分离。

开发与测试使用校内服务器或开发机，不额外采购专用硬件；线下培训与文档编写计入人力成本。

（4）硬件环境

开发机：8 核 CPU、16GB 内存以上，SSD 20GB 以上可用空间；建议开启硬件虚拟化便于本地容器化测试。

服务器/托管：Vercel Hobby（支持 Node 20）、Supabase 免费实例（500MB Postgres、1GB Storage）；如需校内部署，需 4 核 8GB 服

务器一台（含备份盘）。

终端：现代浏览器的 PC 或移动端，需支持 ES2020 与 Fetch；移动端适配以响应式为主，无需原生 App。

（5）软件环境

Node.js 20+、pnpm、Next.js 16、React 19、TypeScript 5、Tailwind CSS 4、shadcn/ui、Radix UI。

数据库：Supabase Postgres，Drizzle ORM + drizzle-kit 生成迁移，配合 postgres 连接池。

监控与审计：自带审计模块，后续可接入 Sentry/Axiom；日志默认控制台输出，可经由 Vercel Log Drain 导出。

版本管理：Git，推荐按 feat/<module>/<desc> 分支模型合并，合并时使用 --no-ff 保留模块历史。

（6）可利用的信息与资源

AGENTS.md 的角色、权限、API 设计规范与 Definition of Done。

packages/db 的基线 schema 与迁移脚本，便于直接导入 Supabase。

已实现的公告模块代码，可作为模板快速复制到资源、预约等模块；notice.service.ts 中的审计与可见性逻辑可直接复用。

docs/requirements 与 docs/api 目录用于记录需求与契约，docs/report 用于报告与流程图。

ESLint/TypeScript 配置（eslint.config.mjs、tsconfig.json）确保代码质

量，Vitest 配置预留单元测试能力。

（7）系统投入使用的最迟时间

目标在 2026 年 3 月完成公告、资源、预约、失物四个核心模块的正式上线，确保迎新季可用；8 月前完成问卷/投票与图书馆模块。

（8）进行可行性研究的方法

代码走查：阅读 Route Handler、Service 与数据库 schema，确认现有实现的可扩展性。

原型演示：使用 pnpm dev 跑通公告列表/详情与后台创建流程，验证登录、分页、上传/下载链路。

利益相关者访谈：收集学生、辅导员、后勤管理员的使用场景、痛点与优先级，形成需求列表。

同类产品调研：参考校内旧系统与第三方 SaaS 的功能范围、交互、费用与合规要求。

风险评估：列出依赖云服务、数据迁移、培训成本等风险，并拟定缓解措施。

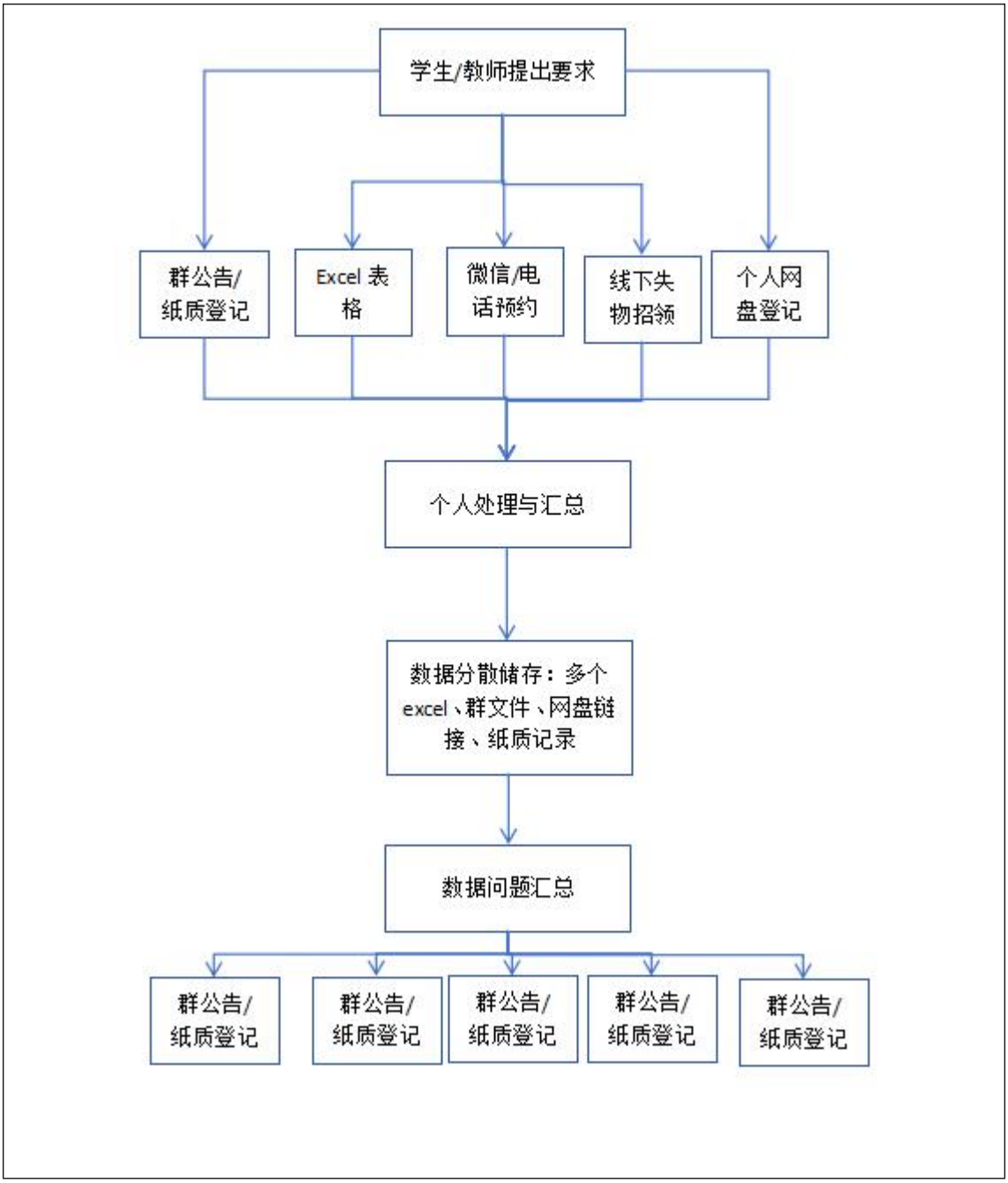
（9）假设与限制

假设学校可提供基础域名与邮件服务；假设学生均可访问互联网或校内网络。

免费层资源存在存储与并发限制，需在设计中加入分页、限流与文件

大小控制。 若需与校内 SSO 集成，可能需要额外审批与网络打通时间，需纳入计划。 若涉及敏感个人信息（如联系方式、证件号码），需遵循分级保护与最小化存储原则，并落实脱敏展示。 校园网络可能存在出口策略限制，需提前确认 Vercel/Supabase 的可访问性；必要时准备校内部署方案。 设备兼容性假设：主流 Chromium/Firefox/Safari 浏览器，IE 系列不再支持；移动端以现代 WebView 为主。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	3.1. 现有系统的处理流程和数据流程	子文档序号	A3.1.
编者		编写日期	2025 年 12 月 22 日


学生/教师提出要求
群公告/纸质登记
Excel 表格
微信/电话预约
线下失物招领
个人网盘登记
个人处理与汇总
数据分散储存：多个excel、群文件、网盘链接、纸质记录
数据问题汇总
群公告/纸质登记
群公告/纸质登记
群公告/纸质登记
群公告/纸质登记
群公告/纸质登记

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	3.2. 现有系统简明情况	子文档序号	A3.2.
编者		编写日期	2025 年 12 月 22 日

通知：QQ 群/微信群置顶消息，无法分角色发送，已读不可统计，历史难搜索，附件易失效。

预约：线下登记或电话沟通，重复预定、撞档与缺席无法留痕，统计工作量大。

资源分享：散落在网盘链接或个人云盘，缺少版本与权限管理，容易出现未授权传播。

失物招领：依赖公众号推文或海报，失物信息无法快速检索，已认领信息更新不及时。

问卷/投票：使用第三方工具，账号分散、数据导出困难，结果与身份无法关联。

数据安全：无统一备份策略，设备损坏或人员流动容易导致数据丢失，且无法追踪操作记录。

统计与决策：缺少统一报表，无法快速统计未读公告数、场地占用率、资源下载量等关键指标。

支撑运维：缺少告警与监控，出现故障或信息错误时无法及时发现；权限收回流程缺失。

培训与支持：新生与教师需反复学习多个工具；界面与操作分散，体验不一致。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	4.1. 所建议系统的概述	子文档序号	A4.1.
编者		编写日期	2025 年 12 月 22 日

系统采用 B/S 架构，Next.js App Router 负责页面与 API，Supabase 提供身份、数据库与文件存储。Portal 与 Console 分区，前者供学生浏览与提交，后者供管理员审核、发布和统计。模块按业务拆分，所有外部能力通过 Adapter 封装，避免直接耦合云厂商。

代码目录划分清晰：app/(portal)、app/(console)、app/api、lib/modules、packages/db、components 等，每个模块具备独立的页面、路由与 Service/Repository，便于分工与协作。UI 采用 shadcn/ui 与 Radix，遵循一致的设计语言；样式通过 Tailwind 4 实现，减少样式冲突。

部署方案灵活：开发环境使用 pnpm dev；测试/生产可在 Vercel 自动构建，或使用 Docker 容器部署到校内服务器；Supabase 负责数据库与存储，减少自建运维压力。未来可将通用配置、lint/test 配置抽取到 packages/config，为其他子系统复用奠定基础。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	4.2. 所建议系统的处理流程和数据流程	子文档序号	A4.2.
编者		编写日期	20025 年 12 月 22 日

登录与权限：用户通过 Supabase Auth 登录，requireUser() 拦截未登录请求；Console 页面根据 user_roles 与权限码渲染菜单，业务接口使用 hasPerm 与数据权限条件控制操作范围。

公告发布：Console 创建公告（标题、Markdown 正文、可见范围、附件），createNotice 写入数据库与附件表，审计记录操作；发布后 Portal 读取 GET /api/notices，根据 read、includeExpired、sortBy 筛选。

阅读与已读回执：Portal 详情调用 getPortalNoticeDetail，并通过 markNoticeRead 事务更新 notice_reads 与 readCount，避免并发重复计数。

附件下载：Storage 生成签名 URL，前端直接下载，避免后端中转；下载链接有效期可控，减少暴露面。

预约流程：学生提交预约 → Route Handler 校验场地/时间 → Service 校验冲突并入库 → 管理端审核（通过/驳回/取消） → 通知申请人；状态变化写入审计。

资源/失物流程：提交元数据与附件 → 审核或自动发布 → Portal 展示并支持搜索/筛选 → 管理端可下架或标记已认领。

问卷/投票流程：管理员创建题目与选项，设置可见范围 → Portal 填答或投票 → 数据统计与导出；可按角色、部门限定范围。

数据可见性：contentVisibility 与数据权限模块控制用户能看到的

公告与数据范围，保证跨部门共享同时保护私密信息；可见范围存储在 notice_scopes 等表中。

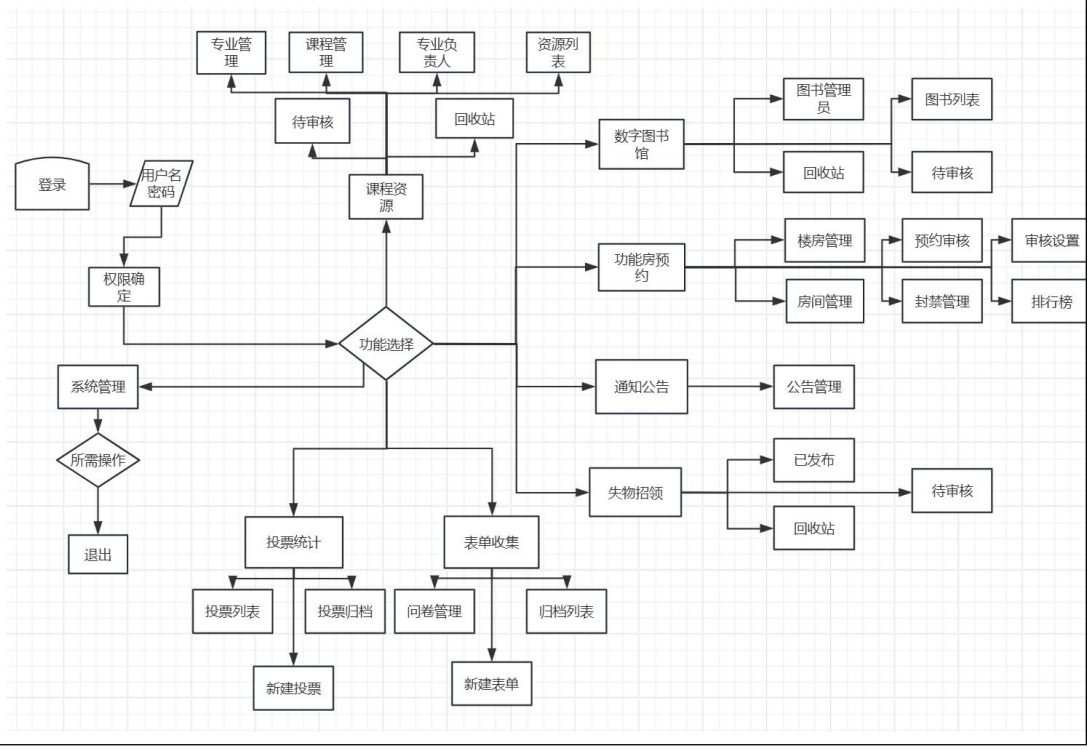
日志与监控：Service 层写审计日志，后续可将 console 日志接入 Sentry/Axiom；错误统一 JSON 结构，便于前端展示与监控告警。

数字图书馆：管理员录入图书/电子书元数据与标签，Portal 支持搜索、下载或外链阅读；下载计数可用于采购与维护决策。

工作台与看板：app/(console)/console/workbench 可汇总待办、即将过期公告、热门资源、预约冲突提示，帮助管理员快速处理。

配置与权限管理：后台 permissions、roles、users、departments、positions 等模块维护组织架构与授权，所有业务模块复用该信息。

数据流图：程序流程图



主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	4.3. 改进、影响和局限性	子文档序号	A4.3.
编者		编写日期	2025 年 12 月 22 日

（1）与现有系统比较的优越性

统一入口与权限控制，公告、预约、资源、问卷集中管理，可按角色/部门精准投递。

过程可追溯：操作审计、已读统计、状态流转留痕，支持责任界定与合规检查。

模块化与可扩展：Adapter 抽象 Auth/Storage/AI，数据库 schema 迁移可复用；代码分层清晰，便于二次开发和多人协作。

用户体验提升：SSR 首屏快、分页搜索、附件签名下载，减少重复沟通；移动端适配便于学生随时查看。

数据安全：集中存储与备份，避免个人设备或网盘丢失；可开启 RLS 与细粒度权限，降低泄露风险。

决策支持：可统计未读数、过期公告、场地占用率、热门资源下载量，为教学与后勤决策提供数据支撑。

（2）采用建议系统可能带来的影响

对云服务可用性有依赖，需制定离线兜底（导出 CSV、本地缓存）与备份策略；需要监控配额使用。

管理员与教师需要短期培训以适应新流程；需编写操作手册与 FAQ。

旧数据迁移需要整理历史 Excel/网盘文件并清洗格式；部分数据可能存在缺失需人工补充。

免费配额可能不足，需要评估并预留升级预算；如校内部署，需网络与安全审批。

变更管理：上线初期需并行运行一段时间，与旧流程保持兼容，逐步切换。

局限性与改进计划：当前仅公告模块完成代码验证，其他模块需要按模板补齐；AI、推送、报表等高级能力暂未实现，需在后续迭代中规划；需要补充自动化测试与监控告警以支撑长期运行。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	5.1. 技术条件可行性分析	子文档序号	A5.1.
编者		编写日期	2025 年 12 月 22 日

仓库已用 Next.js 16、React 19、Supabase、Drizzle 建立可运行骨架，公告模块实现证明技术路径可行：

Route Handler 示例 app/api/notices/route.ts 可直接拉起并返回 JSON，已实现分页、排序、过滤与错误处理。

Service 层包含事务、审计、数据权限、内容校验，满足生产要求；uploadNoticeAttachment 展示了 Storage 上传与审计结合的写法。

Drizzle schema、迁移与 Supabase 兼容，数据库初始化脚本已提

供；packages/db 提供跨项目可复用的 schema。

Tailwind 4 + shadcn/ui 组件库简化 UI 落地，支持无构建样式与主题切换；globals.css 已定义基础样式。

代码使用 TypeScript，全局别名与 ESLint 已配置，降低类型错误与风格不一致风险。

潜在技术风险：Supabase 免费层性能受限、Vercel Hobby 并发限制；可通过分页、缓存、延迟队列与本地/校内部署规避。

综合考虑成熟度、开源生态与现有实现，技术可行性成立。

补充技术验证与测试计划：

单元/集成测试：使用 Vitest 覆盖 Service 层成功/错误/权限场景；对公告、预约、文件上传等关键路径编写用例。

E2E 测试：可使用 Playwright 模拟 Portal/Console 登录、发布公告、阅读、置顶、搜索、上传附件等流程。

安全测试：进行 XSS/CSRF/越权访问的渗透测试；验证内容校验、签名 URL 过期与 RLS 行为。

性能测试：针对高并发阅读、批量导出、附件下载场景使用 k6/Artillery 压测，记录指标并优化 SQL/索引。

主要技术风险与缓解措施：

Supabase 免费层 QPS/存储不足 → 提前设置限流、分页、附件

大小限制，并准备升级或校内部署预案。

Vercel Hobby 并发限制 → 可在高峰期切换到 Pro 或自建 Node 服务器；开启 ISR/缓存减轻压力。

Storage 签名链接滥用 → 控制有效期、下载频率；必要时增加一次性下载令牌。

数据迁移失败或脏数据 → 在测试环境演练迁移，编写幂等脚本并加入校验与回滚。

依赖包安全漏洞 → 定期 pnpm audit，使用 Dependabot 或手工升级，避免高危漏洞滞留。

浏览器兼容性问题 → 以现代浏览器为准，使用 lint/CI 检查不支持的 API；提供 Polyfill 方案。

审计日志膨胀 → 定期归档与清理，或迁移到单独的低频存储。

时间/时区问题 → 统一使用 UTC 存储、界面按本地时区显示；数据库字段使用 timestamp with time zone。

富文本/Markdown 渲染安全 → 保持内联 HTML 禁用，必要时使用 allowlist 过滤。

角色/权限配置错误 → 提供权限检查工具与脚本，上线前执行权限审计，防止过宽授权。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	5.2. 经济可行性分析	子文档序号	A5.2.
编者		编写日期	20025 年 12 月 22 日

（1）支出

① 硬件与托管：Supabase/Vercel 免费层起步；若升级，预计每月 20~50 美元可覆盖数据库与存储，带宽超额按量计费。

② 研发成本：主要为开发与测试人力，使用开源技术栈无许可证费用；需要预算少量培训与文档编写时间。

③ 运营成本：域名、日志/监控（可先用免费版），偶发存储与流量超额费用；备份与灾备可使用对象存储低频层。

④ 数据迁移与培训：一次性成本，涵盖历史数据导入、流程演练与用户培训。

⑤ 设备与工具：如需移动端调试设备、设计工具（Figma/Draw.io）、压测工具，可使用免费版或学校已购授权。

⑥ 备用方案：若需校内部署，需购买或调拨 4 核 8GB 服务器及备份存储，估计一次性 1~2 万元人民币。

（2）效益

① 降低人工统计、重复沟通与线下登记成本，节省老师与学生时间；预计迎新高峰期可减少 30% 以上人工查询量。

② 提高公告到达率与可追踪性，减少遗漏与误传；已读统计便于检查信息覆盖度。

③ 预约冲突减少，场地利用率提高；资源共享效率提升，热门

资料可追踪下载趋势。

④ 数据可沉淀用于后续决策与改进（如统计热门场地、常见问题），提升学校管理水平与学生满意度。

⑤ 采用开源与云免费层，前期投入低，后续可按需弹性扩容，避免一次性大投入。

⑥ 缩短发布与反馈时间：公告从发布到确认已读的链路缩短，紧急通知更易到达；问卷投票可在小时级完成统计。

⑦ 透明度与责任落实：审计日志与可见范围让各角色责任清晰，减少推诿；数据留痕便于事后复盘。

⑧ 品牌与服务提升：统一门户提高学生体验，展示学校数字化水平，有助于招生与对外合作宣传。

（3）投资回收周期

预计在投入运营 1 学期内（约 6 个月）即可通过时间成本节省抵消云资源与培训开销，后续主要为低额运维成本；若扩展为全院统一门户，可进一步摊薄成本。

（4）敏感性分析与成本控制

若免费层容量不足，可优先优化数据存储（归档过期公告、压缩附件）与开启分页/限流；其次再评估升级付费。

如需校内部署，硬件投入一次性增加，但带宽与数据合规更易掌控，适合长期运行。

通过开源组件与自研 Adapter，避免供应商锁定，降低迁移成本；持续跟踪云资源用量，设定告警阈值。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	5.3. 社会因素方面的可行性分析	子文档序号	A5.3.
编者		编写日期	20025 年 12 月 22 日

（1）法律方面可行性

所用框架与库（Next.js、React、Drizzle 等）均为开源许可，遵守相关协议即可。

数据存储遵循学校信息化与个人信息保护要求，支持按需部署在校内机房；敏感数据可加密或限制查询范围。

附件与个人资料均受权限控制与日志记录，便于合规审计；可配置数据保留策略与删除流程。

（2）使用方面可行性

界面遵循 shadcn/ui 组件规范，交互简洁；Portal 面向学生，Console 面向管理员，角色分明，学习成本低。

登录使用邮箱/密码或 Magic Link，符合大多数用户习惯；可扩展 SSO 或短信登录以适配校内统一认证。

培训成本低：代码分层清晰，操作以表单和列表为主，提供占位

文档与手册即可上手；为老师与学生分别提供快速入门指南。

无障碍与多语言：组件支持基础无障碍属性，可按需增加英文或少数民族语言翻译；移动端自适应满足日常使用。

推广策略：先在一个学院/年级试点，收集反馈后逐步扩大；通过迎新培训、助教课堂、公告渠道推广使用；设置反馈入口与工单机制，快速响应问题。

支持与运维：建立 FAQ 与工单流程，约定响应时间；关键时间段（迎新、考试）安排值班或监控，确保系统稳定。

培训内容覆盖：账号注册与权限分配、公告发布与撤回、预约创建与审批、附件上传与安全注意事项。

主文档名称	可行性分析研究报告	主文档编号	A
子文档名称	6. 可行性分析结论	子文档序号	A6.
编者		编写日期	2025 年 12 月 22 日

结论意见： 通过对系统的技术、经济与社会因素的全面分析，学院生活平台系统具备立即开展开发与迭代的条件。建议在保持 Supabase/Vercel 免费层控制成本的前提下，优先完善公告、预约、资源与失物模块，按阶段上线并持续补充测试与文档。