

**《软件项目管理》项目计划书**



**Stream 游戏推荐平台**

|  |  |
| --- | --- |
| 学 院： | 软件学院 |
| 专 业： | 软件工程 |
| 组 别： | 第二组 |
| 组 长： | 张琛 |
| 学 号： |  |
|  |  |
| 指导老师： | 杨波 |

二零二五 年 三 月 五 日

目录

[1 引言 3](#_Toc192062511)

[1.1 编写目的 3](#_Toc192062512)

[1.2 背景 3](#_Toc192062513)

[1.3 定义 4](#_Toc192062514)

[1.4 项目内容 4](#_Toc192062515)

[1.5 项目功能 5](#_Toc192062516)

[2. 项目概述 7](#_Toc192062517)

[2.1 项目目标 7](#_Toc192062518)

[2.2 产品目标与范围 7](#_Toc192062519)

[2.2.1 产品目标 7](#_Toc192062520)

[2.2.2 产品范围 7](#_Toc192062521)

[2.3 项目开发环境 8](#_Toc192062522)

[2.3.1 开发工具与模式 8](#_Toc192062523)

[2.3.2 技术栈 8](#_Toc192062524)

[3. 项目团队组织 9](#_Toc192062525)

[3.1 人员分工 9](#_Toc192062526)

[4. 实施计划 9](#_Toc192062527)

[4.1 风险评估及对策 9](#_Toc192062528)

[4.1.1 技术风险 9](#_Toc192062529)

[4.1.2 外部依赖风险 9](#_Toc192062530)

[4.2 总体进度计划 9](#_Toc192062531)

[4.2.1 项目周期 9](#_Toc192062532)

[4.2.2 阶段划分与任务分配 10](#_Toc192062533)

[5. 预算 10](#_Toc192062534)

[参考文献 10](#_Toc192062535)

# 1 引言

## 编写目的

本软件项目计划书的编写旨在详细定义和规范Stream游戏推荐系统的开发需求，明确软件开发的具体细节，确保所有参与该项目的用户、软件开发人员和分析人员对系统的核心目标、功能和技术要求达成一致理解。本文件通过对系统各项功能需求、性能需求和数据需求的细致描述，进一步明确了系统实现的步骤、范围和实用背景，为系统的开发提供了清晰的方向和基准。

此外，此文档还将为客户提供一个清晰的参考框架，使他们能够理解该系统如何帮助其达成目标，并解决游戏推荐领域的核心问题。文档中对各项需求的定义将为开发人员提供一个评估和遵循的标准，确保系统的开发和交付符合既定的质量和效率要求。这不仅为软件开发的各个环节提供了重要的指导，也为后续的维护和升级工作奠定了坚实的基础。

## 背景

2024年夏季，国产3A游戏大作《黑神话：悟空》凭借其精良的制作水准、引人入胜的故事情节和浓厚的中国文化元素，在全球范围内掀起了一股热潮，成为游戏行业和大众关注的焦点。该游戏的成功不仅获得了国内外玩家的高度评价，还彰显了国产游戏在国际市场上的竞争力，使人们重新审视了游戏的文化价值和情感共鸣作用。

然而，在当今复杂多样的游戏市场中，玩家在面对众多游戏作品时往往感到信息过载，难以在质量参差不齐的众多选择中找到真正适合自己的优质游戏。市场中不仅有像《黑神话：悟空》这样的精品佳作，还充斥着大量依靠心理诱导机制牟利的低质量“快餐”游戏，虽然这些游戏可以吸引用户，但却未能真正满足用户的兴趣和需求，反而消耗了大量的时间和金钱。

在这样的市场环境下，如何根据玩家的个性化需求和兴趣，为其提供精准的游戏推荐，成为一个亟待解决的问题。传统的推荐方法主要基于简单的兴趣标签或购买历史，往往忽略了用户行为的深层次模式，导致推荐效果欠佳。

为此，本项目应运而生，旨在构建一个智能化的游戏推荐平台。该平台不仅采用先进的数据分析和推荐算法，深度挖掘和理解用户的兴趣偏好和行为数据，提供个性化的高质量推荐服务，同时还具备丰富的功能模块。平台为用户提供游戏浏览、游戏排行榜查看、游戏搜索、个性化推荐、收藏和评论等互动功能，帮助用户快速定位符合其需求的游戏。同时，用户还可以通过反馈和客服功能，提升用户体验。管理员则可以管理游戏库内容、优化推荐算法、管理广告内容和审核评论等，确保平台运行的质量和安全性。

通过这些功能，本项目致力于为用户打造一个高效、个性化的游戏推荐环境，帮助他们在纷繁复杂的市场中快速找到优质游戏，从而提升用户体验，为游戏行业的可持续发展贡献力量。

## 定义

|  |  |
| --- | --- |
| **术语/缩写** | **定义** |
| **LLM** | 大语言模型（Large Language Model），用于实现自然语言交互与游戏资讯 |
| **Supabase** | BaaS 后端托管平台，为开发者提供了一个完整的后端解决方案，让开发者无需花费时间在后端开发上 |
| **Vercel AI SDK** | 云端 AI 工具包，提供与 LLM 的集成接口 |
| **Next.js** | React 框架，用于服务端渲染与 API 开发 |

## 项目内容

本项目核心内容为开发一个多功能的游戏推荐系统，具体包括用户注册、登录等基础用户功能，以及首页轮播推荐、排行榜、AI 游戏推荐助手等游戏推荐功能。

技术实现上，前端基于 React + Tailwind CSS 实现响应式界面；后端使用 Next.js 构建 API，集成 Supabase 数据库存储用户与游戏数据；AI 模块，通过 Vercel AI SDK 连接 LLM，实现智能对话与游戏推荐。

## 项目功能

|  |  |  |  |
| --- | --- | --- | --- |
| **功能名称** | **输入** | **功能详细描述** | **输出** |
| 注册 | 用户名、邮箱、密码、验证码 | 注册时，用户的角色类别默认为客户。用户需要输入系统所提供的验证码，这是为了防止机器暴力破解系统，保证系统的安全性。除此以外，用户还需要输入一些系统要求用户所输入的信息。同时，注册界面需要做的美观，增强用户的体验。 | 注册的结果 |
| 登录 | 用户名、  密码 | 用户如果想要进入网站主页访问系统，则必须先登录。登录时，用户需要输入自己的用户名和密码。 | 登录的结果 |
| 退出登录 | 无 | 用户在登录以后，如果想要切换另外一个账号登录系统，或者退出登录状态，用户需要选择系统中提供的退出登录的选项。 | 退出登录的结果 |
| 个人中心管理 | 用户ID、个人信息数据 | 管理用户个人资料，包括基本信息修改、头像更新、隐私设置等 | 更新后的个人信息 |
| 修改账号信息 | 用户ID、修改项、新数据 | 修改用户账号相关信息，如密码、邮箱、手机号等 | 修改状态、更新后的账号信息 |
| 搜索游戏 | 搜索关键词、筛选条件 | 根据关键词搜索游戏，支持标签筛选、价格区间筛选等 | 搜索结果列表、搜索统计信息 |
| 查看游戏推荐 | 用户ID、浏览历史、用户偏好 | 基于用户兴趣和行为特征推荐游戏 | 推荐游戏列表、推荐理由 |
| 收藏游戏 | 用户ID、游戏ID | 添加/取消游戏收藏，管理收藏列表 | 收藏操作状态、更新后的收藏列表 |
| 评分功能 | 用户ID、游戏ID、评分值 | 对游戏进行评分，更新游戏总体评分 | 评分结果、更新后的游戏评分信息 |
| 评论功能 | 用户ID、游戏ID、评论内容 | 发表游戏评论，查看其他用户评论 | 评论发布状态、评论列表 |
| 智能客服 | 用户问题、会话上下文 | 提供智能化的客服服务，解答用户问题 | 答复内容、相关推荐 |
| 更新游戏库 | 游戏信息、更新类型 | 新增、修改、下架游戏信息 | 更新状态、操作日志 |
| 管理广告内容 | 广告内容、投放设置 | 管理平台广告，包括添加、修改、下架广告 | 广告操作状态、广告统计数据 |
| 更新推荐算法 | 算法参数、训练数据 | 更新和优化游戏推荐算法 | 算法更新状态、性能评估报告 |
| 管理注册用户 | 用户ID、操作类型 | 管理用户账号，包括封禁、解封、权限管理等 | 操作结果、用户状态更新 |
| 审核用户评论 | 评论ID、审核结果 | 审核用户发布的评论内容 | 审核状态、处理结果 |
| 统计分析 | 分析维度、时间范围 | 生成平台运营数据分析报告 | 统计报表、数据可视化图表 |
| 收藏排序 | 排序规则、数据范围 | 对用户收藏的游戏进行智能排序 | 排序结果、排序依据说明 |

# 2. 项目概述

## 2.1 项目目标

本项目旨在通过技术开发与用户需求结合，构建一个高效、智能的游戏推荐系统，具体目标如下：

* 完成用户系统开发，支持注册、登录、个人中心（收藏、评分、评论）功能，确保用户数据安全存储。
* 实现动态游戏推荐功能，包括首页轮播图与滑动交互、智能排行榜、AI 聊天交互界面等，从而实现个性化的游戏咨询。

## 2.2 产品目标与范围

### 2.2.1 产品目标

通过智能推荐与交互设计，解决用户在游戏选择中的信息过载问题，提供个性化、实时的游戏发现体验。主要面向游戏爱好者和新手玩家，通过融合 AI 对话与传统推荐算法，提升推荐的灵活性与用户参与度，并基于 Supabase 的实时数据同步，确保推荐结果动态更新。

### 2.2.2 产品范围

* 功能模块
  + 用户系统：
    - 注册/登录
    - 个人中心：收藏管理、评分记录、评分历史
  + 游戏展示与推荐：
    - 首页动态轮播
    - 多维度排行榜：按照评分、发行时间对游戏进行排行
    - 模糊搜索：支持游戏名称、标签、简介关键词检索
  + AI 交互：
    - 聊天界面：通过自然语言提问获取游戏推荐
    - 推荐解释：展示推荐逻辑
  + 技术实现：
    - 响应式前端：适配 PC 端主流分辨率，支持多种浏览器
    - 后端服务：基于 Next.js 的 API 接口和 Supabase 后端服务存储用户行为与游戏元数据。
* 非功能需求
  + 性能：主界面加载时间 <= 1.5s，API 响应延迟 <= 300ms
  + 可维护性：提供完整的日志系统与错误监控

## 2.3 项目开发环境

### 2.3.1 开发工具与模式

|  |  |  |
| --- | --- | --- |
| **类别** | **工具/技术** | **说明** |
| **开发工具** | * IDE: VSCode | 配合插件实现代码规范检查与版本控制 |
| **开发模式** | * 瀑布开发模式 | 项目整体按照确定需求、设计、编码、测试和维护来推进。每周一次例会同步进度 |

### 2.3.2 技术栈

|  |  |  |
| --- | --- | --- |
| **层级** | **技术选型** | 用途 |
| **前端** | * 语言：TypeScript * 框架：React + Next.js * UI库：Tailwind CSS | 构建响应式界面，支持服务端渲染（SSR）与静态生成（SSG），优化SEO与加载性能 |
| **后端** | * 后端服务：Supabase * API 开发：Next.js API Routes | 作为一个 BaaS 平台，Supabase 为本项目提供了一个完整的后端解决方案，让开发人员无需花费时间在后端开发上 |
| **AI 集成** | * 工具：Vercel AI SDK | 实现自然语言交互功能，处理用户咨询并生成个性化推荐结果 |
| **部署** | * 托管平台：Vercel | 自动化部署前端与API，实时监控系统异常与性能瓶颈 |

## 2.4 项目运行环境

为了确保项目在不同环境中的一致性，采用 Docker 将项目打包为容器，从而实现跨平台运行。

# 3. 项目团队组织

## 3.1 人员分工

**项目经理：张琛**

* 核心职责：
  + 指定项目计划与里程碑节点，监控13周开发周期进度
  + 主持每日站会（15分钟）与周例会，协调前后端接口联调
* 关键技术动作：
  + 主导风险评估会议，指定对应预案

**前端开发：杨骑骏、田启泽**

* 主责：杨骑骏
  + 开发首页推荐模块/排行榜动态交互
  + 对接后端API数据格式规范
* 协作：田启泽
  + 实现个人中心/搜索功能
  + 配合测试团队完成UI自动化测试

**后端开发：吴佳昊**

* 核心职责：
  + 设计Supabase数据库结构
  + 开发RESTful API接口
  + 实现推荐算法
  + 集成Vercel AI SDK，封装AI对话任务

**测试工程师：孙逍遥**

* 质量保障体系：
  + 测试策略设计：指定单元测试/E2E测试/压力测试方案
* 关键动作：
  + 编写20+测试用例（覆盖注册/登录/收藏/推荐主流程）
  + 输出《测试报告》
  + 主导UAT用户培训，收集并分析NPS满意度数据

**协作机制：**

* 代码管理：Github 仓库分支策略
* 文档同步：Confluence实时更新《接口文档》

# 4. 实施计划

## 4.1 风险评估及对策

### 4.1.1 技术风险

**描述：**AI 推荐算法模块会使用市面上已经开源的模型的 API，但是将会面临 API 免费试用时期有限、AI 响应速度不可控等问题。

**应对措施：**尽可能多地查阅资料，寻找多个基座模型作为替代方案。

### 4.1.2 外部依赖风险

**描述：**Supabase 数据库服务可能不稳定，且与前端交互的速度不可控，从而影响核心功能的正常运行。

**应对措施：**定期备份数据库至本地，确保数据安全。

## 4.2 总体进度计划

### 4.2.1 项目周期

**总时长：**13周

**团队规模：**5人

* 项目经理（1 人）：统筹协调、风险管理
* 前端开发（2 人）：界面开发、交互实现
* 后端开发（1人）：设计数据库、操纵 supabase
* 测试工程师（1 人）：质量保障、自动化测试

### 4.2.2 阶段划分与任务分配

**阶段一：**需求分析与设计（2周）

* 第 1 周：
  + 调研用户需求，初步完成项目计划书
  + 分析竞品 steam 平台
  + 初步技术选型
* 第 2 周：
  + 继续讨论改进，完成项目计划书

**阶段二：**核心功能开发（6周）

* 第 3 周：
  + 初始化 Git 仓库，配置 CI/CD 流程
  + 搭建 Next.js 项目框架
  + Supabase 后端服务搭建
* 第 4 周：
  + 用户系统开发（注册/登录）
  + 个人中心基础 UI
* 第 5-6 周：
  + 首页轮播图推荐功能
  + 游戏收藏/评分交互
  + 协同过滤算法开发
* 第 7 周：
  + AI 聊天模块开发
  + Vercel AI SDK 接入
* 第 8 周：
  + 模糊搜索功能
  + 响应式布局适配
  + 性能优化

**阶段三：**测试与优化（3周）

* 第 9 周：
  + 单元测试（用户模块/API）
  + E2E 测试（全流程覆盖）
  + 安全测试
* 第 10 周：
  + 压力测试
  + 性能调优
  + 多分辨率适配验证
* 第 11 周：
  + 用户验收测试
  + 收集反馈并优化 UI/UX
  + 编写用户手册/运维文档

**阶段四：**部署与上线（1周）

* 第 12-13 周：
  + 生产环境部署（Vercel + Supabase）
  + 项目总结报告
  + 结题答辩准备

# 5. 预算

## 5.1 预算构成与依据

本项目基于商业化场景设计，预算涵盖人力成本、基础设施租赁、技术工具采购及运维支持四大核心板块。

## 5.2 人力成本

**开发周期**：13周（约 3 个月）

**团队配置**：5人团队

* 项目经理：按市场月薪 15000 元/人月计算，3个月成本为 45000 元，负责需求管理、进度协调和风险控制
* 前端开发：2名工程师，12000元/人月，合计72000元，承担界面开发、交互逻辑实现几跨平台适配任务
* 后端开发：1名工程师，18000元/人月，负责数据库管理及推荐算法优化
* 测试工程师：10000元/人月，主导自动化测试、性能压测及用户验收流程

## 5.3 基础设施租赁

* 计算资源：部署前端与后端的 ECS 服务器，单价800元/月，3个月合计4800元
* AI 服务：DeepSeek API 调用费用，3个月约1000元

## 5.4 技术工具采购

* 协作平台：github，无费用
* 测试工具：Postman 企业版（API测试）+ LoadRunner（性能压测）

## 5.5 运维与不可预见费用

* 监控告警：阿里云ARMS应用监控服务，月费1000，3个月3000

# 参考文献

无