



# 知你社交——基于 Springboot 的 AI 个性化社交平台

---

## 项目管理

课程：专业方向综合项目

指导老师：唐剑锋

小组成员：

2253551 李沅衡

2250763 李俊旻

2254272 赵子毅

2251760 黄志栋

2251225 王铭乾

完成日期：2025 年 3 月 27 日



## 一、项目总览

### 1.1 项目背景

### 1.2 项目目标

### 1.3 项目地址

## 二、项目范围管理

### 2.1 包含内容

### 2.2 不包含内容

### 2.3 项目约束与假设

### 2.4 验收标准

## 三、SWOT 分析

### 3.1 优势 Strengths

### 3.2 劣势 Weaknesses

### 3.3 机会 Opportunities

### 3.4 威胁 Threats

### 3.5 应对策略

## 四、风险管理

### 4.1 风险识别与评估表

### 4.2 风险应对策略

## 五、质量管理

### 5.1 质量保证 (QA)

### 5.2 质量控制 (QC)

## 六、资源管理

### 6.1 资源管理流程

### 6.2 项目团队构成

### 6.3 资源监控与优化机制

## 七、附录资料

项目成员

组长：2253551 李沅衡

学号	姓名
2253551	李沅衡
2251225	王铭乾
2250763	李俊旻
2254272	赵子毅
2251760	黄志栋

一、项目总览

1.1 项目背景

随着用户对社交平台个性化与智能化需求的快速增长，传统社交应用在内容推荐、用户连接与互动深度方面显得力不从心。本项目致力于开发一个基于 Spring Boot 构建的 AI 个性化社交平台，通过智能算法驱动推荐系统和用户匹配机制，打造一个理解用户需求、提升社交效率、增强使用体验的现代社交解决方案。平台还注重用户隐私保护与系统稳定性，以应对未来扩展及市场挑战。

1.2 项目目标

- 实现高并发环境下稳定运行的社交平台架构；
- 开发个性化推荐系统与聊天匹配机制；
- 构建用户友好的交互界面和高效的后台管理工具；
- 确保用户数据安全、平台合规性和高性能表现；
- 完成从系统分析到部署的闭环开发流程。

1.3 项目地址

[GitHub 项目链接](#)

二、项目范围管理

## 2.1 包含内容

- 用户注册与身份验证系统
- 个性化推荐算法与社交匹配服务
- 即时聊天与社群互动功能
- 活动发布与参与模块
- 内容展示与兴趣标签体系
- 后台管理与权限控制面板

## 2.2 不包含内容

- 第三方平台账号集成与数据同步
- 音视频直播功能
- 商业广告与电商模块
- 高级会员订阅体系
- 海外部署与多语言支持（初期）

## 2.3 项目约束与假设

### 约束条件：

- 系统架构需支持水平扩展与高并发处理；
- 所有数据传输与存储符合数据安全与隐私法律法规；
- 项目时间控制在指定开发周期内，预算不超标。

### 前提假设：

- 用户偏好能通过算法建模实现有效标签化；
- 开发团队具备必要的 AI 与后端系统能力；
- 测试环境能模拟真实高并发业务场景。

## 2.4 验收标准

- 平台系统在高峰时段无明显延迟或宕机；
- 所有关键功能点均通过回归测试与性能验证；
- 后台可有效执行内容与用户管理任务；
- 所有接口均符合预设 API 契约规范；

- 提交完整的开发文档、测试报告及部署说明。

## 三、SWOT 分析

### 3.1 优势 Strengths

- 引入 AI 算法增强用户社交推荐精度；
- 垂直化产品定位增强目标用户黏性；
- 结构清晰、组件模块化，利于快速迭代升级。

### 3.2 劣势 Weaknesses

- 初始阶段缺乏用户基础，面临冷启动问题；
- 市场认知度低，获客难度较高；
- 内容审核流程与机制有待完善。

### 3.3 机会 Opportunities

- 智能社交趋势日益增强，用户接受度上升；
- 多样化虚拟社交场景需求正在增长；
- 国家政策支持 AI 技术在应用层落地。

### 3.4 威胁 Threats

- 面临小红书、微博等头部平台激烈竞争；
- 内容违规风险或数据泄露可能损害品牌形象；
- 技术依赖或开源组件安全漏洞带来稳定性隐患。

### 3.5 应对策略

#### 短期策略：

- 建设优质初始内容池，吸引首批用户；
- 结合兴趣标签推送内容，提升首日留存；
- 建立合规审核机制，杜绝不当内容风险。

#### 中期策略：

- 引入激励机制（例如内容点赞、活跃奖励）；
- 建立用户分层管理，提升高质量社交匹配效率。

长期策略：

- 推进品牌合作及社群合作拓展平台影响力；
- 探索平台商业模式如广告植入、社交电商等；
- 逐步开放 API 接入能力，支持外部生态发展。

四、风险管理

4.1 风险识别与评估表

风险编号	风险描述	类型	可能性	影响程度	优先级
R1	技术栈整合复杂	技术	高	高	高
R2	用户数据泄露	安全	中	极高	高
R3	高并发导致系统瓶颈	技术	中	中	中
R4	团队配合不畅	管理	低	中	低
R5	项目延期或预算超支	管理	中	中	中
R6	依赖第三方服务不稳定	技术	中	高	中

4.2 风险应对策略

预防型措施：

- 为高风险技术点制定专项攻关计划；
- 建立监控体系（Prometheus + Grafana）实时报警；

- 每周开展团队同步会审查关键风险点；
- 形成知识库记录历史问题与经验。

#### 响应型措施：

- UAT 阶段引入实际用户测试进行预警检验；
- 发生系统或数据安全事件时立即启动应急响应机制；
- 制定快速修复流程并设置内部通报窗口；
- 风险处理结果形成文档化闭环。

## 五、质量管理

### 5.1 质量保证（QA）

- 代码质量：GitHub Pull Request 审查 + SonarQube 静态分析；
- 测试机制：PyTest 编写单元测试，Selenium 完成端到端集成流程验证；
- 数据质量：数据集清洗去重率  $\geq 99\%$ ，误杀率  $\leq 2\%$ ，Kappa 系数  $\geq 0.8$ ；
- 接口测试：OpenAPI + Postman 验证参数与返回规范一致性；
- 安全机制：定期执行 OWASP Top 10 安全扫描，无高危漏洞（CVE  $\geq 7$ ）。

### 5.2 质量控制（QC）

- 缺陷分级处理：
  - P0 阻断型缺陷：4 小时内修复上线；
  - P1 关键功能缺陷：24 小时内修复并回归测试；
  - P2 一般缺陷：48 小时内处理完毕；
- 所有关键路径测试用例通过率需达 100%；
- 完成安全、性能、兼容性三类测试报告归档。

---

## 六、资源管理

### 6.1 资源管理流程

- 制定项目阶段性资源投入计划；
- 识别关键角色（开发、设计、AI、测试、运营）及对应工时需求；

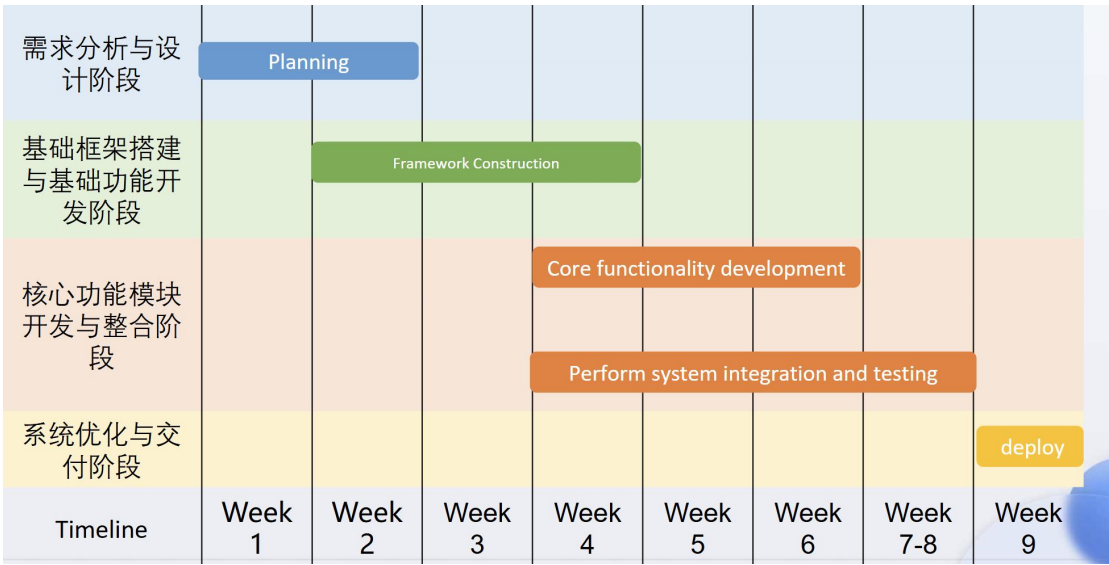
- 引入项目协作工具如 JIRA 管理任务进度；
- 每周审查人员工作量和资源匹配状态；
- 项目结束后回收资源，输出总结复盘报告。
- 运维支持人员 1 名

## 6.2 资源监控与优化机制

- 实时监控各类资源使用情况，预警短缺与浪费；
- 借助 JIRA、Redmine 等工具分析任务瓶颈与人力使用率；
- 建立问题反馈机制，快速响应资源变更需求；
- 通过工时报表与任务完成率衡量投入产出比。

## 七、附录资料

- 项目开发阶段甘特图（任务排期）



- 活动资源管理——开发团队



活动 人员	李沅衡	李俊旻	赵子毅	黄志栋	王铭乾
分析设计	R	R	A/R	C	C
质量管理	A/R	I	I	C	I
前端开发	R	A/R	R	I	I
后端开发	R	I	R	R	A/R
AI 模块开发	C	R	I	A/R	R
接口测试	C	A/R	R	I	R
系统测试	R	I	R	A/R	R
部署	A/R	I	I	I	I

R: 参与执行; A: 对任务负责; C: 提供信息辅助; I: 得到及时通知

- 资源需求分析图

