

需求分析报告

1. 项目文档简介

1.1 项目背景

当前许多社交媒体和内容平台充斥着碎片化信息和视觉干扰,缺乏沉静思考和深入探讨的氛围.Synapse项目旨在构建专注于纯粹文本交流的Web应用,通过剔除图片等视觉元素的干扰,为用户提供一个可以进行深度思考、理性讨论和知识分享的在线社区.

1.2 项目目标

本项目的主要目标是:

- 成功设计并实现一个功能完善、用户体验良好的基于文章的文本交流网站Synapse.
- 实现核心功能模块,包括用户管理、文章发布与管理、互动评论、内容发现及后台管理等.
- 提供简洁、美观、响应式的用户界面,确保流畅的阅读和交流体验.
- 深入理解并实践软件工程的完整开发流程,包括需求获取、系统设计、模块化开发、版本控制、测试与文档撰写.
- 最终产出一个可运行的Web应用(至少局域网可访问)、完整的项目源代码及符合课程要求的项目文档.

1.3 文档目的

本需求分析报告旨在明确Synapse项目的功能性与非功能性需求,为后续的系统设计、开发、测试和项目管理提供依据.本文档将详细阐述项目目标、用户角色、各项功能需求、性能需求、可维护性需求等关键方面.

1.4 项目范围

本项目范围包括Synapse网站的前端用户界面、后端业务逻辑、数据库设计与实现,以及必要的管理后台功能.具体功能点如下:

- 用户模块:** 注册、登录/登出、个人资料管理、个人主页、邮箱验证.
- 文章模块:** 文章创建(支持Markdown)、发布、保存草稿、编辑、删除、展示、阅读.
- 互动模块:** 文章评论、文章点赞、消息通知.
- 内容发现模块:** 关键词搜索(支持标题、内容、作者、标签搜索).
- 管理后台模块:** 用户管理、内容审核、评论管理、数据统计.
- 排除项:** 除用户头像外,网站不包含任何其他图片、视频等多媒体内容.

1.5 预期读者

本文档的预期读者包括项目开发团队成员以及其他相关干系人.

2. 用户角色与特征

用户角色	描述	主要诉求
访客	任何可以访问Synapse网站但未登录的用户.	浏览公开的文章列表和文章内容,了解平台氛围.
注册用户	在Synapse成功注册并登录的用户.	发布和管理自己的文章,评论他人文章,点赞感兴趣的内容,管理个人资料.
管理员	拥有特殊权限的用户,负责网站的日常管理和维护.	管理用户账户,审核和管理平台内容,查看网站运营数据,确保平台健康有序运行.

3. 功能需求

3.1 用户模块

- **FR1.1 用户注册**
 - **FR1.1.1** 系统提供用户注册功能,支持邮箱验证码验证.
 - **FR1.1.2** 用户注册时需提供: 用户名、邮箱(唯一)、密码、邮箱验证码.
 - **FR1.1.3** 系统对注册信息进行校验(邮箱格式、邮箱是否已存在、验证码有效性).
 - **FR1.1.4** 注册成功后,用户状态自动变为已登录.
- **FR1.2 用户登录/登出**
 - **FR1.2.1** 系统提供用户登录功能,支持用邮箱和密码登录.
 - **FR1.2.2** 系统验证用户凭据的有效性,登录失败时给出提示.
 - **FR1.2.3** 登录成功后,系统记录用户登录状态(Session管理).
 - **FR1.2.4** 系统提供用户登出功能,清除用户登录状态.
- **FR1.3 用户个人资料管理**
 - **FR1.3.1** 用户可以查看和编辑自己的个人资料,包括: 用户名、个人签名.
 - **FR1.3.2** 用户可以修改自己的登录密码,需要通过邮箱验证修改.
- **FR1.4 用户个人主页**
 - **FR1.4.1** 每个注册用户拥有一个个人主页,展示其个人资料.
 - **FR1.4.2** 个人主页展示该用户发布的文章列表.
 - **FR1.4.3** 支持查看他人主页.

3.2 文章模块

- **FR2.1 文章创建**
 - **FR2.1.1** 注册用户可以创建新文章.
 - **FR2.1.2** 文章包含标题、摘要、标签和正文内容.
 - **FR2.1.3** 系统提供Markdown编辑器,支持实时预览.

- FR2.1.4 编辑器不支持插入图片或其他多媒体元素.
- **FR2.2 文章发布、保存草稿**
 - FR2.2.1 用户可以将创建/编辑的文章直接发布.
 - FR2.2.2 用户可以将未完成的文章保存为草稿,草稿仅作者可见.
 - FR2.2.3 用户可以从草稿继续编辑并发布.
 - FR2.2.4 支持草稿列表管理和删除功能.
- **FR2.3 文章编辑、删除**
 - FR2.3.1 文章作者可以编辑自己已发布或存为草稿的文章.
 - FR2.3.2 文章作者可以删除自己的文章(已发布的或草稿).
 - FR2.3.3 删除操作前有确认提示.
- **FR2.4 文章列表展示**
 - FR2.4.1 系统提供文章列表页面,展示已发布的文章.
 - FR2.4.2 文章列表项显示文章标题、作者、发布时间、点赞/评论数等.
 - FR2.4.3 文章列表支持分页功能.
- **FR2.5 文章详情页阅读**
 - FR2.5.1 点击文章列表中的文章,进入文章详情页.
 - FR2.4.2 文章详情页完整显示文章标题、作者信息(可跳转至用户主页)、发布时间、正文内容.
 - FR2.5.3 文章详情页下方展示该文章的评论区和互动按钮(点赞).
 - FR2.5.4 文章内容排版简洁清晰,支持Markdown渲染.

3.3 互动模块

- **FR3.1 文章评论**
 - FR3.1.1 注册用户可以对已发布的文章发表评论.
 - FR3.1.2 评论内容为纯文本,支持长度限制(1000字符以内).
 - FR3.1.3 评论显示评论者信息(用户名)和评论时间.
 - FR3.1.4 新评论会触发消息通知给文章作者.
- **FR3.2 文章点赞**
 - FR3.3.1 注册用户可以对文章进行点赞操作.
 - FR3.3.2 同一用户对同一文章只能点赞一次,再次点击取消点赞.
 - FR3.3.3 系统显示文章的点赞数.
 - FR3.3.4 文章被点赞会触发消息通知给文章作者.
- **FR3.3 消息通知系统**
 - FR3.4.1 用户可以查看消息中心,查看所有通知.
 - FR3.4.2 支持标记消息为已读和批量清除已读消息.
 - FR3.4.3 通知类型包括: 文章被评论、文章被点赞.

3.4 内容发现模块

- **FR4.1 关键词搜索**
 - FR4.1.1 系统提供关键词搜索功能,支持模糊搜索.
 - FR4.1.2 搜索范围包括: 文章标题、文章内容、作者用户名、文章标签.

- **FR4.1.3** 支持按搜索类型过滤(全部、标题、内容、作者、标签).
- **FR4.1.4** 搜索结果以列表形式展示,支持分页.

3.5 管理后台模块

- **FR5.1 管理员认证**
 - **FR5.1.1** 管理员具有独立的权限系统,与普通用户完全分离.
 - **FR5.1.2** 管理员可以通过专门的管理后台登录界面访问.
 - **FR5.1.3** 支持超级管理员和普通管理员的权限分级.
- **FR4.2 用户管理**
 - **FR4.2.1** 管理员可以查看所有注册用户列表(包括用户名、邮箱、注册时间等).
 - **FR4.2.2** 管理员可以搜索用户.
 - **FR4.2.3** 超级管理员可以对用户进行管理操作,如添加/删除管理员权限.
- **FR5.3 内容审核**
 - **FR5.3.1** 管理员可以查看平台上的所有文章和评论.
 - **FR5.3.2** 管理员可以对不符合社区规范的文章或评论进行删除操作.
 - **FR5.3.3** 支持文章管理界面和评论管理界面.
- **FR5.4 数据统计**
 - **FR5.4.1** 管理员可以查看网站的关键数据统计,包括:
 - 总用户数、总文章数、总评论数
 - 今日新增数据统计
 - **FR5.4.2** 数据以简洁卡片和统计数字形式展示.

4. 数据需求

Synapse项目使用MongoDB作为数据库,主要数据实体(集合)包括:

4.1 核心数据集合

- **users:** 存储用户信息
- **articles:** 存储文章信息
- **comments:** 存储评论信息
- **messages:** 存储消息通知信息
- **admins:** 存储管理员信息
- **sessions:** 存储用户会话信息
- **RegisterCode:** 存储邮箱验证码信息

4.2 详细字段说明

4.2.1 users 集合

```
{
  _id: ObjectId;
  title: string;
  summary: string;
  tags: string[];
  authorId: ObjectId;
  authorName: string;
  body: string;
  createdAt: Date;
  updatedAt: Date;
  comments: ObjectId[];
  status: ArticleStatus;
  likes: number;
}
```

4.2.2 articles 集合

```
{
  _id: ObjectId,
  title: string,           // 文章标题
  summary: string,        // 文章摘要
  tags: string[],         // 文章标签
  body: string,           // 文章内容(Markdown)
  authorId: ObjectId,     // 作者ID
  status: 'draft' | 'published', // 文章状态
  likesCount: number,     // 点赞数
  commentsCount: number,  // 评论数
  createdAt: Date,        // 创建时间
  updatedAt: Date         // 更新时间
}
```

4.2.3 comments 集合

```
{
  _id: ObjectId;
  articleId: ObjectId;
  authorId: ObjectId;
  authorName: string;
  content: string;
  createdAt: Date;
}
```

4.2.4 messages 集合

```
{
  _id: ObjectId;
  userId: ObjectId;
  type: MessageType;
  articleId: ObjectId;
  articleTitle: string;
  commentId?: ObjectId;
  commentContent?: string;
  fromUserId: ObjectId;
  fromUserName: string;
  createdAt: Date;
  isRead: boolean;
}
```

4.2.5 admins 集合

```
{
  _id: ObjectId;
  userId: ObjectId;
  priority: number;
  createdAt: Date;
  updatedAt: Date;
}
```

4.2.6 sessions 集合

```
{
  _id: string;
  userId: ObjectId;
  userData: {
    _id: ObjectId;
    name: string;
    email: string;
    articles: ObjectId[];
    likes: ObjectId[];
    signature?: string;
  };
  expiresAt: Date;
}
```

4.2.7 RegisterCode 集合

```
{
  _id: ObjectId,
  email: string,
  code: string,
  expiresAt: Date,
}
```

6. 技术架构

6.1 技术栈

- **前端框架:** SvelteKit(支持SSR和SPA模式)
- **后端语言:** TypeScript/JavaScript (Node.js)
- **数据库:** MongoDB(支持本地和云端部署)
- **样式:** CSS
- **构建工具:** Vite(SvelteKit内置)
- **测试框架:** Vitest
- **文档生成:** TypeDoc
- **版本控制:** Git

6.2 项目特色

- 纯文本交流,专注内容质量
- Markdown编辑器支持,优化写作体验
- 响应式设计,支持多端访问
- 完善的用户管理和内容审核系统
- 全文搜索功能和标签分类
- 点赞评论等互动功能
- 管理后台和数据统计
- 消息通知系统

7. 约束与假设

- **CS1. 技术栈约束:** 项目使用SvelteKit作为前端和后端框架,MongoDB作为数据库.
- **CS2. 内容约束:** 网站严格限制图片的使用,仅允许用户头像.所有用户生成内容(文章、评论)均为纯文本.
- **CS3. 开发资源:** 开发团队规模较小,需合理分配时间和任务.
- **CS4. 部署环境:** 初期目标为局域网内可访问,后续可考虑公网部署.
- **AS1. 用户具备基本网络操作能力:** 假设用户熟悉常规Web应用的注册、登录、浏览、内容发布等操作.

- **AS2. 开发工具可用性:** 假设开发所需的软件(IDE, Node.js, npm/pnpm/yarn, Git, MongoDB)均可正常获取和使用.

8. 未来展望 (可选功能)

在核心功能实现并稳定运行后,可考虑以下增强功能:

- 文章收藏功能
- 更高级的搜索过滤器(按时间、热度等排序)
- 用户之间的私信功能
- 实时通知推送
- 草稿自动保存功能
- 文章版本历史
- 个性化内容推荐
- 用户头像上传功能