

“福依旧”网站

需求分析文档

**迭代记录表格**

**当前版本**

|  |  |
| --- | --- |
| **项目** | 福依旧(Fuever) |
| **文档** | 《软件需求说明书》--SRS |
| **说明** | V3.0 |
| **作者** | 文档组 |
| **最后更新时间** | 2022-10-22 |

**更新概要**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **更新时间** | **更新人员** | **更新摘要** |
| V1.0 | 2022-10-19 | 陈家隆、黄骁羿 | 形成基本大纲 |
| V2.0 | 2022-10-20 | 打工人 | 撰写各部分内容 |
| V2.1 | 2022-10-21 | 陈家隆 | 更新思维导图 |
| V2.2 | 2022-10-21 | 焘 | 补充项目logo |
| V2.3 | 2022-10-21 | 余靖宇、王昕 | 更新项目功能 |
| V2.4 | 2022-10-21 | 黄骁羿 | 更新验收标准 |
| V2.5 | 2022-10-21 | 杜瑞学 | 更新3.1、3.2 |
| v3.0 | 2022-10-22 | 黄骁羿 | 审核校对文档 |
| V3.1 | 2022-11-1 | 黄骁羿、余靖宇 | 补充内容 |

# **一、介绍**

## 1.**1 引言**

#### **1.1.1 编写目的**

为明确软件需求、安排项目规划与进度、组织软件开发与测试，撰写本文档。

#### **1.1.2 软件需求分析 —— 理论、目标**

（1）理论依据

软件需求分析（Software Reguirement Analysis，以下简称SRA）是研究用户需求得到的说明文档，旨在完全理解用户对软件需求的完整功能后，确认对应的软件功能实现后，建立可确认的、可验证的一个基本依据。

SRA是一个项目的开端，也是项目实施最重要的关键点。据有关的机构分析结果表明，设计的软件产品存在不完整性、不正确性等问题80％以上是需求分析错误所导致的，而且由于需求分析错误造成根本性的功能问题尤为突出。因此，一个项目的成功软件需求分析是关键的一步。

（2）SRA的主要实现目标

① 对实现软件的功能做全面的描述，帮助用户判断实现功能的正确性、一致性 和完整性，促使用户在软件设计启动之前周密地、全面地思考软件需求；

② 了解和描述软件实现所需的全部信息，为软件设计、确认和验证提供一个基准；

③ 为软件管理人员进行软件成本计价和编制软件开发计划书提供依据；

(3)具体内容要求

① 软件的功能需求，

② 软件与硬件或其他外部系统接口，

③ 软件的非功能性需求，

④ 软件的反向需求，

⑤ 软件设计和实现上的限制，

⑥ 阅读支持信息。

SRA应尽量提供软件实现功能需求的全部信息，使得软件设计人员和软件测试人员不再需要需求方的接触。这就要求软件需求分析内容应正确、完整、一致和可验证。此外，为保证软件设计质量，便于软件功能的休整和验证，软件需求表达应具备无岔意性，可追踪性和可修改性。

#### **1.1.3 参考文献**

[1] 邹欣.构建之法-现代软件工程[M].第三版.北京:人民邮电出版社，2017

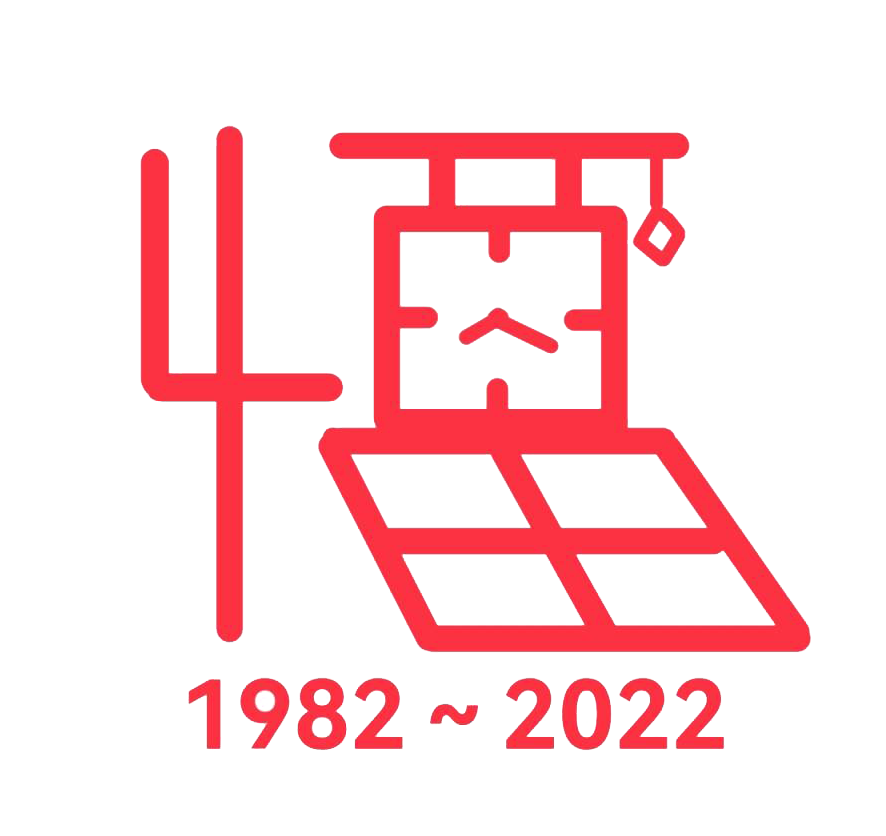
[2] [美]Roger S.Pressman著,郑人杰等译.软件工程[M].第八版.北京:机械工业出版社,2017

[3] GB-T8567-2006，《计算机软件文档编制规范》[S]

[4] GB-T 9385-2008.《计算机软件需求规格说明规范》[S]

## **1.2 项目**

#### **1.2.1 项目标志**



#### **1.2.2 产品范围**

本产品在综合考虑既定要求以及可能存在的用户需求，拟为使用Web应用的用户提供以下服务：

•回望学院发展历史

•浏览学院新闻动态

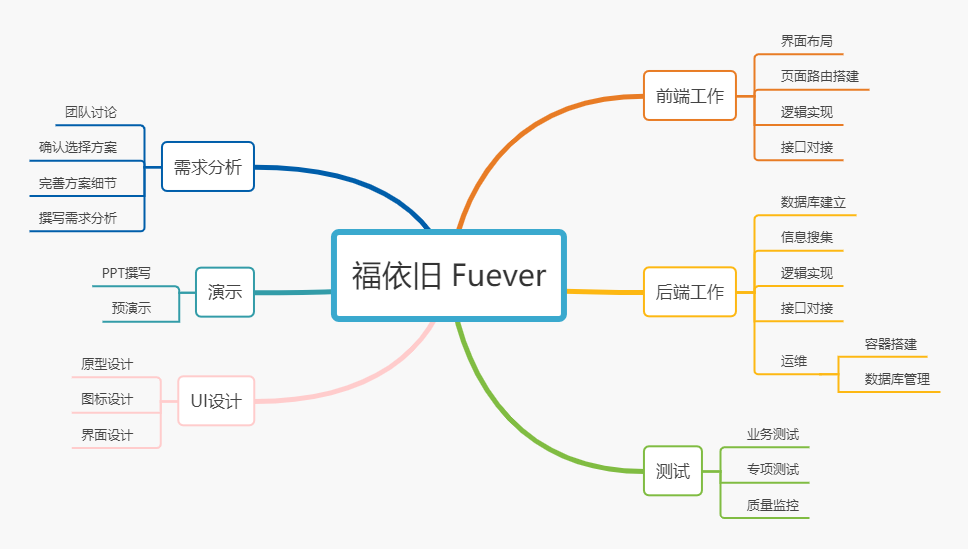
•院庆活动安排指南

•推荐可能认识的校友

•在交流区与校友交流互动

## **1.3 说明**

#### **1.3.1 思维导图**



#### **1.3.2 面向群体**

本文档面向多种读者对象：

（1）项目经理：项目经理可以根据该文档了解预期产品的功能，并据此进行系统设计。

（2）设计员：对需求进行分析，并设计出系统，包括数据库的设计。

（3）程序员：配合《设计报告》，了解应用功能。

（4）测试员：根据本文档对软件产品进行功能性测试和非功能性测试。

（5）销售人员：了解预期产品的功能和性能。

（6）用户：了解预期产品的功能和性能，并与分析人员一起对整个需求进行讨论和协商。

（7）其他人员：如老师、助教、其他同学等可以据此了解产品的功能和性能。

在阅读本文档时，首先要了解产品的功能概貌，然后可以根据自身的需要对每一功能进行适当的了解。

#### **1.3.3 文档约定**

（1）页面边距

页面的左边距为3.17cm，右边距为3.17cm，装订线靠左。

（2）标题最多分三级：

① 一级标题：一、 二、 三、…（宋体加粗二号）

② 二级标题：1.1 1.2 1.3 …(黑体加粗三号)

③ 三级标题：1.1.1 1.1.2 1.1.3 …（黑体加粗四号）

（3）正文

正文字体为宋体小四号，1.5倍行距。无特殊情况下，字体颜色均采用黑色。

（4） 人工编号

出现序号的段落不采用自动编号功能而采用人工编号，各级别的序号依次为（1）、①等，部分情况序号1为（一）。

（5）段落首行缩进2个字符

（6）本文所引用的图片均采用PNG格式进行引用。

（7）页码格式为"当前页/总页数"

# **二、综合描述**

## 2.**1 项目背景**

本产品是为福州大学计算机与大数据学院计算机系成立40周年暨软件学院成立20周年院庆所开发的院庆Web应用。

据调查，现有的各教育层次、各学科类别的学院院庆网站，普遍存在着同质化严重的问题。对于这些网站的内容，不外乎就是学院历史、历任领导之类的栏目，创新性与吸引力不足，更多程度上，它们更像是一个面向社会大众的学院宣传网站，对院庆的主体——老校友们的实用性不高。

着眼于一次院庆的过程，对于参与者而言，携带不便的日程安排手册、旅行团式的走马观花都是导致体验不佳的原因；对于主办方而言，院庆参与人员多通过电联学院负责人来进行活动报名，这一定程度上加重了这位负责人这一段时间内的工作量，同时存在人工登记错误的隐患。

本次待开发的产品将打造一个创新性的院庆网站以优化以上问题。这款应用以Web应用为载体，无需下载即可使用。它搭载了打卡讲解、活动指南、校友推荐、讨论区等功能，以期丰富用户的校庆活动参与体验。

## **2.2 产品功能--类图设计**

#### **2.2.1 用户场景分析**

|  |  |
| --- | --- |
| **典型用户** | 参加计算机院庆的校友 |
| **知识层级和能力** | 基本拥有使用Web应用的能力 |
| **动机** | 参加院庆时拥有更加美好的体验 |
| **困难** | 传统的校庆乏味枯燥，流于形式 |
| **典型场景** | 自主安排行程重游母校，身临其境回忆青春岁月 |

#### **2.2.2 功能说明**

本软件主要有以下功能，随着Web应用的更新，功能会有所更改

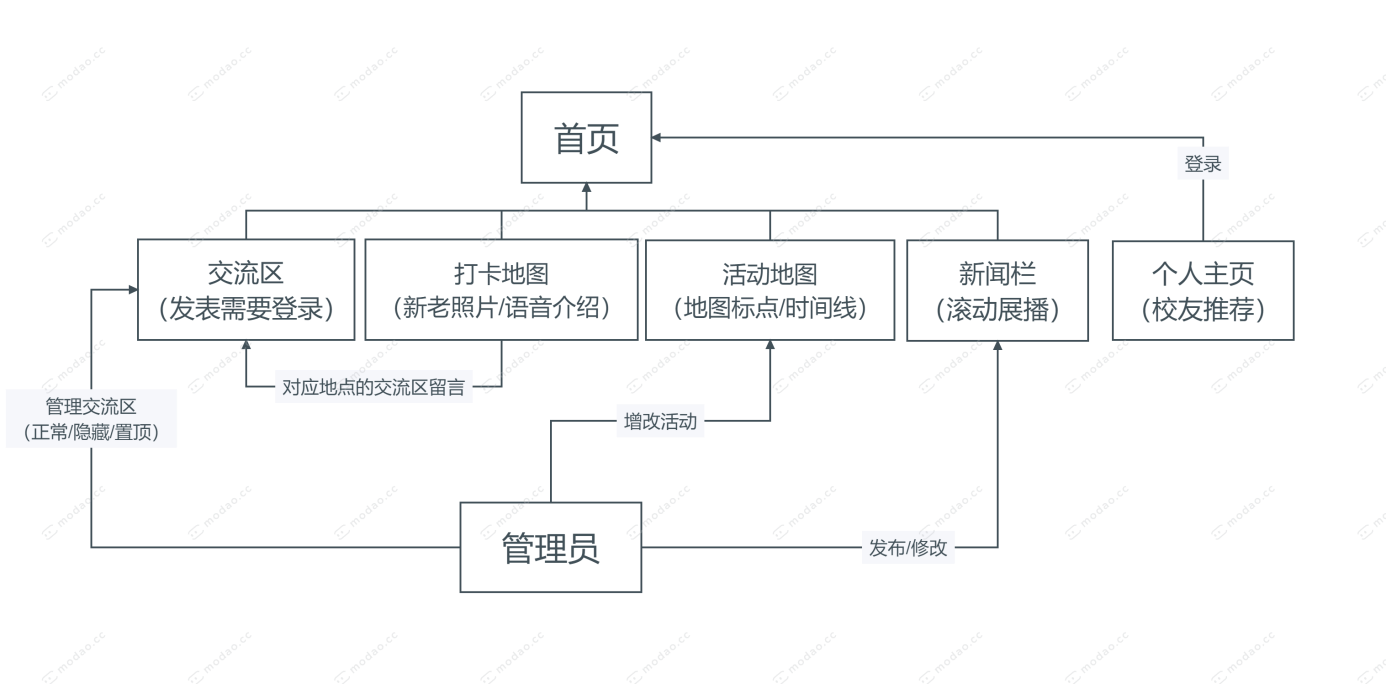
1. **登录功能**
2. 用户可以使用已有的初始账号密码进行登录
3. 用户可以在个人主页标注联系方式和个人信息，方便和校友进行联系
4. **校友交流区**
5. 用户可以在社区里发帖/回帖进行讨论，与校友交流感想，分享趣事
6. 用户可以在对应的板块发帖，高效检索心仪内容
7. **新闻墙**
8. 用户可以通过该模块浏览学院的大小新闻，快速了解学院历史与近况
9. **院庆活动地图**
10. 用户可以通过该模块了解院庆相关的信息和活动安排

**（5）校友推荐**

1. 系统可以根据用户填写的相关信息，自动推荐可能想认识的校友

**（6）地图标注**

1. 接入地图功能，将院庆活动地点标注在地图上
2. 标注打卡地，开放对应交流区



#### **2.2.3 用例描述**

1. **用例：**登录

**主要参与者：**用户

**目标：**让用户通过已有的初始账号密码登录本Web应用

**前提条件：**通过浏览器进入Web应用

**触发器：**用户想要登录该Web应用并进行使用

**场景：**用户：进入该Web应用，输入账号密码进行登录

**异常：**无法打开Web应用：手机运行或网络异常，解决手机本身问题

**优先级：**必须实现

**使用频率：**首次登录或者重新登录

**使用方式：**通过控制面板的接口

1. **用例：**新闻、校庆信息浏览

**主要参与者：**用户

**目标：**让用户快速了解学院的历史与近况，以及校庆的相关活动安排

**前提条件：**用户打开web应用

**触发器：**用户想要了解学院新闻

**场景：**

（1）用户：点击新闻板块

（2）用户：选择感兴趣的新闻，进行浏览

**异常：**无法打开该板块：检查网络是否出现问题

**使用频率：**一天若干次 ~ 几天一次

**使用方式：**通过功能按钮的接口

1. **用例：**校友推荐

**主要参与者：**用户

**目标：**让用户结识可能认识的校友

**前提条件：**用户已登录Web应用并完成信息完善

**触发器：**用户想要认识新的校友

**场景：**

（1）用户：点击个人主页中的校友推荐模块

（2）用户：选择感兴趣的同学并通过其联系方式进行联络

**异常：**

（1）无法打开该板块：检查网络是否出现问题，用户是否登录成功

（2）显示无推荐好友：检查个人信息是否正确填写

**使用频率：**一天若干次 ~ 几天一次

**使用方式：**通过功能按钮的接口

1. **用例：**交流区发帖，评论回复

**主要参与者：**用户

**目标：**让用户间交流感想，分享

**前提条件：**用户已登录Web应用

**触发器：**

（1）用户想要分享自己的有关感想

（2）用户想要对他人的帖子进行回复

**场景：**

（1）用户：编辑文章标题与内容并进行发表

（2）用户：编辑评论内容参与相关帖子的讨论

**异常：**

（1）无法打开该板块：检查网络是否出现问题，用户是否登录成功

（2）帖子、评论内容出错：帖子、评论内容包含违法词汇信息等情况

（3）帖子、评论未及时发布成功：帖子、评论内容需经过审核及上传

**使用频率：**一天若干次至几天一次

**使用方式：**通过功能按钮的接口

1. **用例：**班级搜索

**主要参与者：**用户

**目标：**让用户找到当年的班级

**前提条件：**用户已登录Web应用

**触发器：**用户想要寻找当年班级和同学

**场景：**

（1）用户：点击班级搜索板块

（2）用户：输入班级名称进行搜索

**异常：**

（1）无法打开该板块：检查网络是否出现问题，用户是否登录成功

（2）搜索不到对应班级：检查班级是否输入正确

**使用频率：**一次

**使用方式：**通过功能按钮的接口

1. **用例：**活动地图

**主要参与者：**用户

**目标：**让用户方便寻找院庆的活动地点

**前提条件：**用户打开Web应用

**触发器：**用户想要找到活动地点

**场景：**

（1）用户：点击地图板块

（2）用户：根据地图上的标记点，方便地导航到活动地点

**异常：**

（1）无法打开该板块：检查网络是否出现问题

（2）地图信息显示不正确或导航失败：检查是否授权定位权限

**使用频率：**一天若干次 ~ 几天一次

**使用方式：**通过功能按钮的接口

1. **用例：**打卡地图

**主要参与者：**用户

**目标：**让用户回顾旧地标地点与新地标地点的图片，并在留言区进行留言

**前提条件：**用户已授权登录

**触发器：**用户想要回顾学校地标建筑的变迁，并和校友进行讨论

**场景：**

（1）用户：点击打卡地图板块

（2）用户：点击地图上的具体标记点

（3）用户：在留言区进行留言

**异常：**

（1）无法打开该板块：检查网络是否出现问题，用户是否登录成功

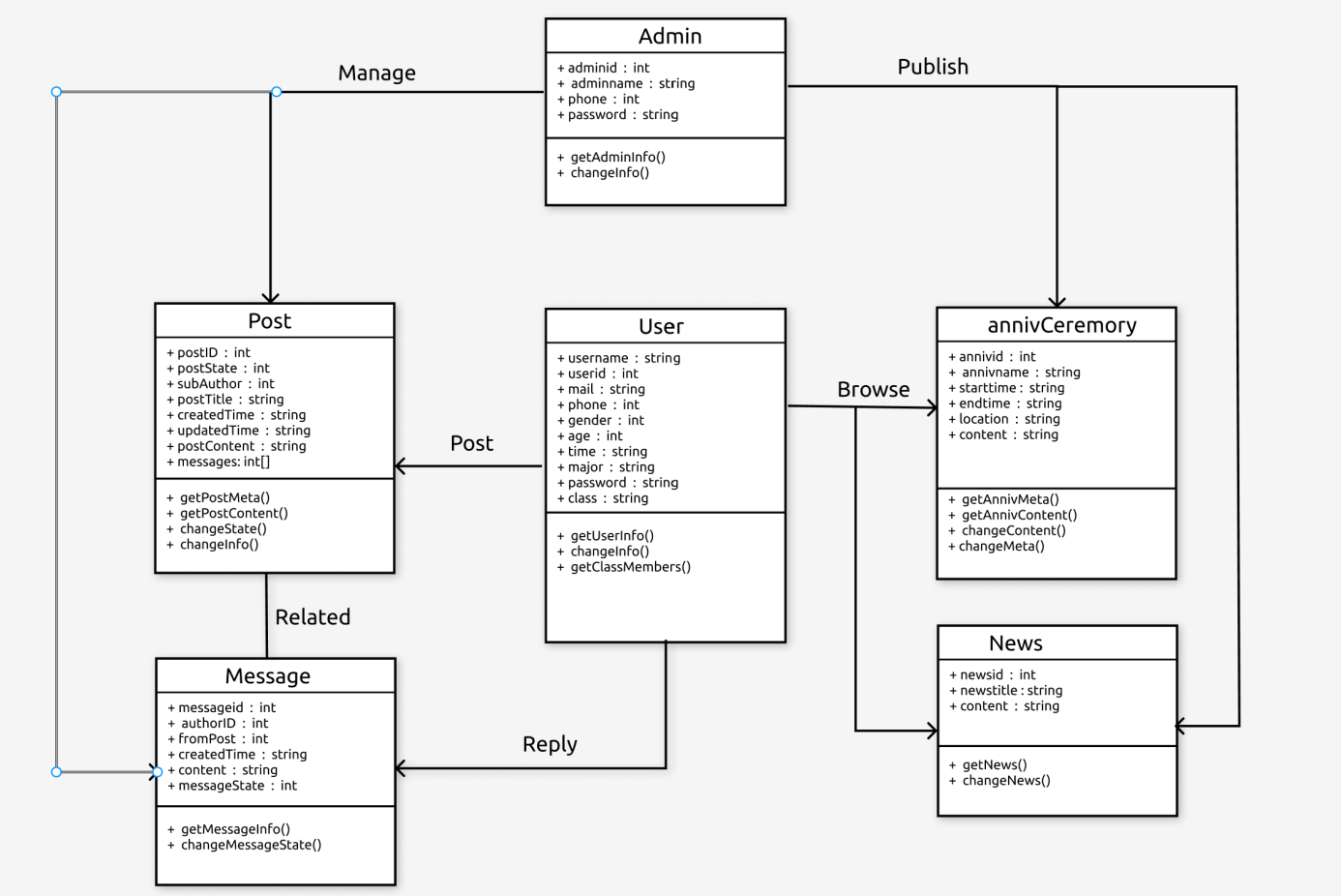
（2）留言内容出错：留言内容包含违法词汇信息等情况

（3）留言未及时发布成功：留言内容需经过一定审核及上传

**使用频率：**一天若干次 ~ 几天一次

**使用方式：**通过功能按钮的接口

#### **2.2.4 类图**



## 2.3 运行环境

#### 2.3.1 软、硬件要求

运行环境：

硬件设备：手机/PC

软件环境：Windows/Mac OS/Linux/Android

#### 2.3.2 设计与实现约束

（1）开发环境约束

网络环境限制导致团队开发时版本控制可能出现故障；web的移动端适配需要进行一定的调整

（2）开发周期较短

​ 鉴于总的开发时间较短，需要进行合理的时间规划，以保证产品的准时交付。

（3）所采用的方法与技术有限

​ 鉴于服务器价格昂贵，考虑到经济问题，本次作业使用的服务器性能较差，高并发场景下往往是带宽瓶颈，故没有必要做K8S集群和分布式微服务。

（4）采用一定的代码规范约束

​ 为了保证代码能够顺利整合，开发之前需要制定一定的代码规范约束。

#### 2.3.3 假设与依赖

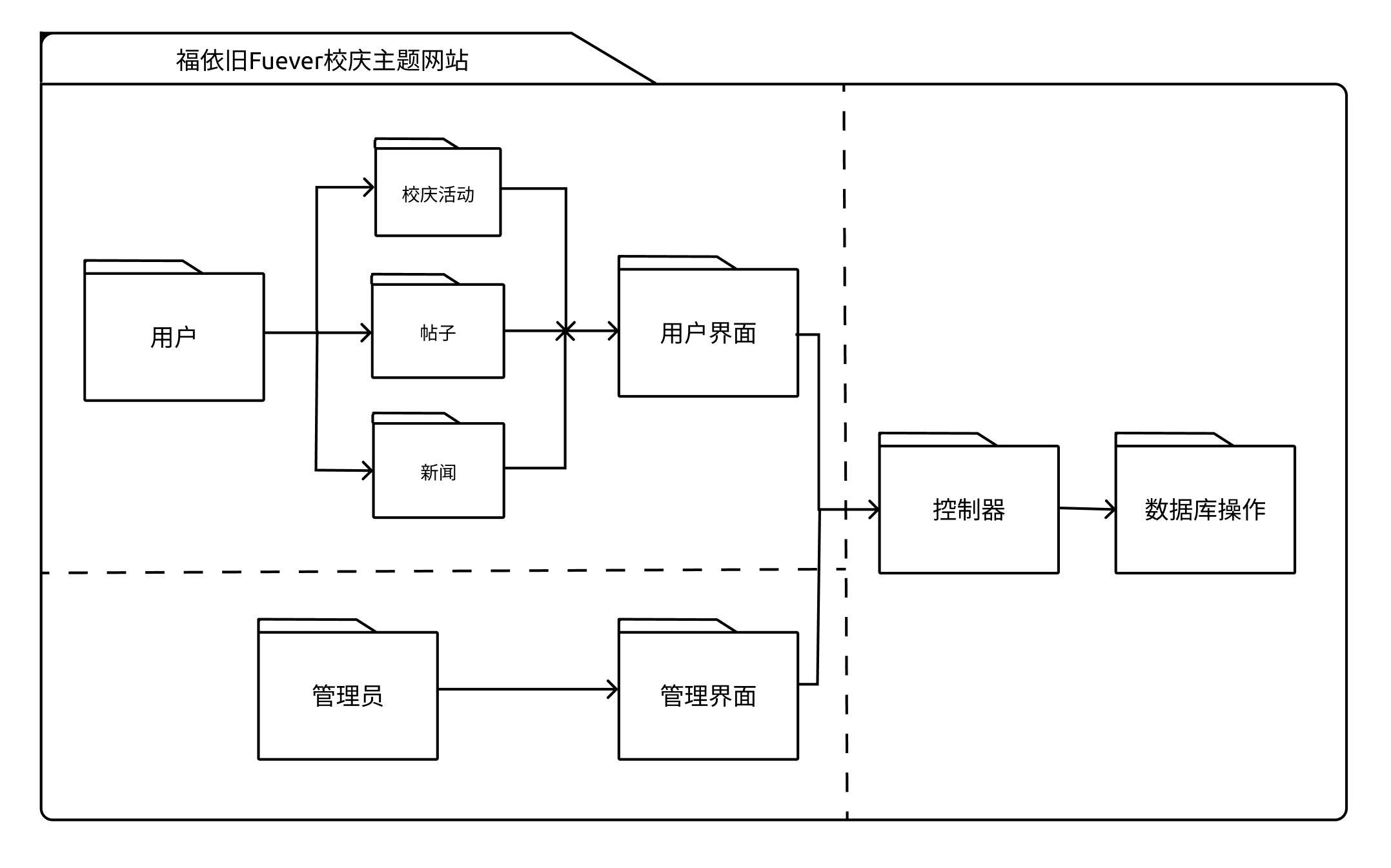
1. 假设
2. 可操作性：假定使用本Web应用的用户在经过一段时间熟悉之后，可以灵活地操作本Web应用来满足自己的需要。
3. 用户支持：假定在Web应用在开发的各个环节中得到用户的有效支持和配合。
4. 技术支持：假定开发初期，小组成员充分认识本系统的需求，认真学好相关知识。开发过程中遇到技术问题，可以及时得到其他同学或者老师的指导与帮助。
5. 人员配合：假定小组主要成员基本不会出现变动，并且在项目开发过程中不会因为突发情况的发生而导致项目成员无法正常参与开发工作。
6. 时间限定：假定项目的截止时间不会提前。
7. 需求限定：假定项目需求基本确定之后，不会有太大改变。

（2）依赖

本产品的使用率取决于于官方发布平台的可靠性、权威性、安全性，对于发布平台具有很强的依赖性。院或校官方平台对该软件的重视程度，即给予该软件的曝光率，很大程度影响了本产品的使用率。同时，院庆的时间局限性和举办时疫情政策的调整都会影响本产品的使用率。

# **三、系统体系结构设计**

### 3.1 系统结构分析(包图)

包图：一种维护和描述系统总体结构模型的重要建模工具。对复杂系统进行建模时，通常需要处理大量的类、接口、构件、节点和图（跟书包中的书和文具类似），这就有必要将这些元素进行分组，即把语义相近并倾向于同一变化的元素组织起来加入同一个包中，以方便理解和处理整个模型。

### 3.2 接口设计

#### 3.2.1 URL 设计

#### （1）动词 + 宾语

客户端发出的数据操作指令都是"动词 + 宾语"的结构。比如，GET /articles这个命令，GET是动词，/articles是宾语。动词通常就是五种 HTTP 方法，对应 CRUD 操作。根据 HTTP 规范，动词一律大写。

（2）动词的覆盖

有些客户端只能使用GET和POST这两种方法。服务器必须接受POST模拟其他三个方法（PUT、PATCH、DELETE）。这时，客户端发出的 HTTP 请求，要加上X-HTTP-Method-Override属性，告诉服务器应该使用哪一个动词，覆盖POST方法。

（3）宾语必须是名词

宾语就是 API 的 URL，是 HTTP 动词作用的对象。它应该是名词，不能是动词。

（4）复数 URL

为了统一起见，建议都使用复数 URL，比如GET /articles/2要好于GET /article/2

（5）避免多级URL

除了第一级，其他级别都用查询字符串表达。

#### 3.2.2 状态码

（1）状态码必须精确

客户端的每一次请求，服务器都必须给出回应。回应包括 HTTP 状态码和数据两部分。

HTTP 状态码就是一个三位数，分成五个类别

（2）2xx 状态码

200状态码表示操作成功。POST返回201状态码，表示生成了新的资源；DELETE返回204状态码，表示资源已经不存在。202 Accepted状态码表示服务器已经收到请求，但还未进行处理，会在未来再处理，用于异步操作。

（3）3xx 状态码

API 用到的3xx状态码，主要是303 See Other，表示参考另一个 URL。它与302和307的含义一样，也是"暂时重定向"，区别在于302和307用于GET请求，而303用于POST、PUT和DELETE请求。收到303以后，浏览器不会自动跳转，而会让用户自己决定下一步怎么办。

（4）4xx 状态码

4xx状态码表示客户端错误，主要有下面几种。

400 Bad Request：服务器不理解客户端的请求，未做任何处理。

401 Unauthorized：用户未提供身份验证凭据，或者没有通过身份验证。

403 Forbidden：用户通过了身份验证，但是不具有访问资源所需的权限。

404 Not Found：所请求的资源不存在，或不可用。

405 Method Not Allowed：用户已经通过身份验证，但是所用的 HTTP 方法不在其权限之内。

410 Gone：所请求的资源已从这个地址转移，不再可用。

415 Unsupported Media Type：客户端要求的返回格式不支持。比如，API 只能返回 JSON 格式，但是客户端要求返回 XML 格式。

422 Unprocessable Entity ：客户端上传的附件无法处理，导致请求失败。

429 Too Many Requests：客户端的请求次数超过限额。

（5）5xx状态码

5xx状态码表示服务端错误。一般来说，API 不会向用户透露服务器的详细信息，所以只要两个状态码就够了。

500 Internal Server Error：客户端请求有效，服务器处理时发生了意外。

503 Service Unavailable：服务器无法处理请求，一般用于网站维护状态。

#### 3.2.3 服务器回应

（1）不要返回纯本文

API 返回的数据格式，不应该是纯文本，而应该是一个 JSON 对象，因为这样才能返回标准的结构化数据。所以，服务器回应的 HTTP 头的Content-Type属性要设为application/json。

客户端请求时，也要明确告诉服务器，可以接受 JSON 格式，即请求的 HTTP 头的ACCEPT属性也要设成application/json

（2）发生错误时，不要返回 200 状态码

状态码反映发生的错误，具体的错误信息放在数据体里面返回。

（3）提供链接

在回应中，给出相关链接，便于下一步操作。这样的话，用户只要记住一个 URL，就可以发现其他的 URL。这种方法叫做 HATEOAS。对于用户来说，不需要记住 URL 设计，只要从 api.x.com 一步步查找就可以了。

### 3.3 技术选型

#### 3.3.1 前端

##### 分析

本次待开发的应用主要面向移动端，其主要以webkit内核为主，对于HTML5等新技术支持的更好。同时，移动端存在网络质量变化大的问题，不适合大型的框架，故选择较为轻量级的Vue框架。由于JavaScript有动态类型和弱类型的特点，很难进行团队开发、功能扩展和后期维护，故语言选择Vue框架同样支持的TypeScript。结合渐进式web应用--PWA（Progressive Web App），能在保留Web应用优点的同时，给予用户贴近原生应用的体验。

###### 技术选型

* Vue
* HTML
* CSS
* TypeScript

#### 3.3.2 后端

##### 分析

本次项目构建的应用为院庆应用，在院庆当天需要面对高并发场景。固然小组成员大都具有一定C++编程基础，但考虑到没有工程经验，冒然选择C++会导致开发效率低下，内存泄露等严重问题，故选择带有GC，语法风格与C++相近的Go语言作为后端开发语言。同时选择Golang的Web框架中性能性能较高的Gin框架作为后端框架。使用PostgreSQL作为关系型数据库存储相关数据，Redis作为缓存减少高并发场景下服务器的压力。压测下表现良好。

##### 技术选型

* Gin
* Gorm
* PostgreSQL
* Redis

#### 3.3.3 推荐系统

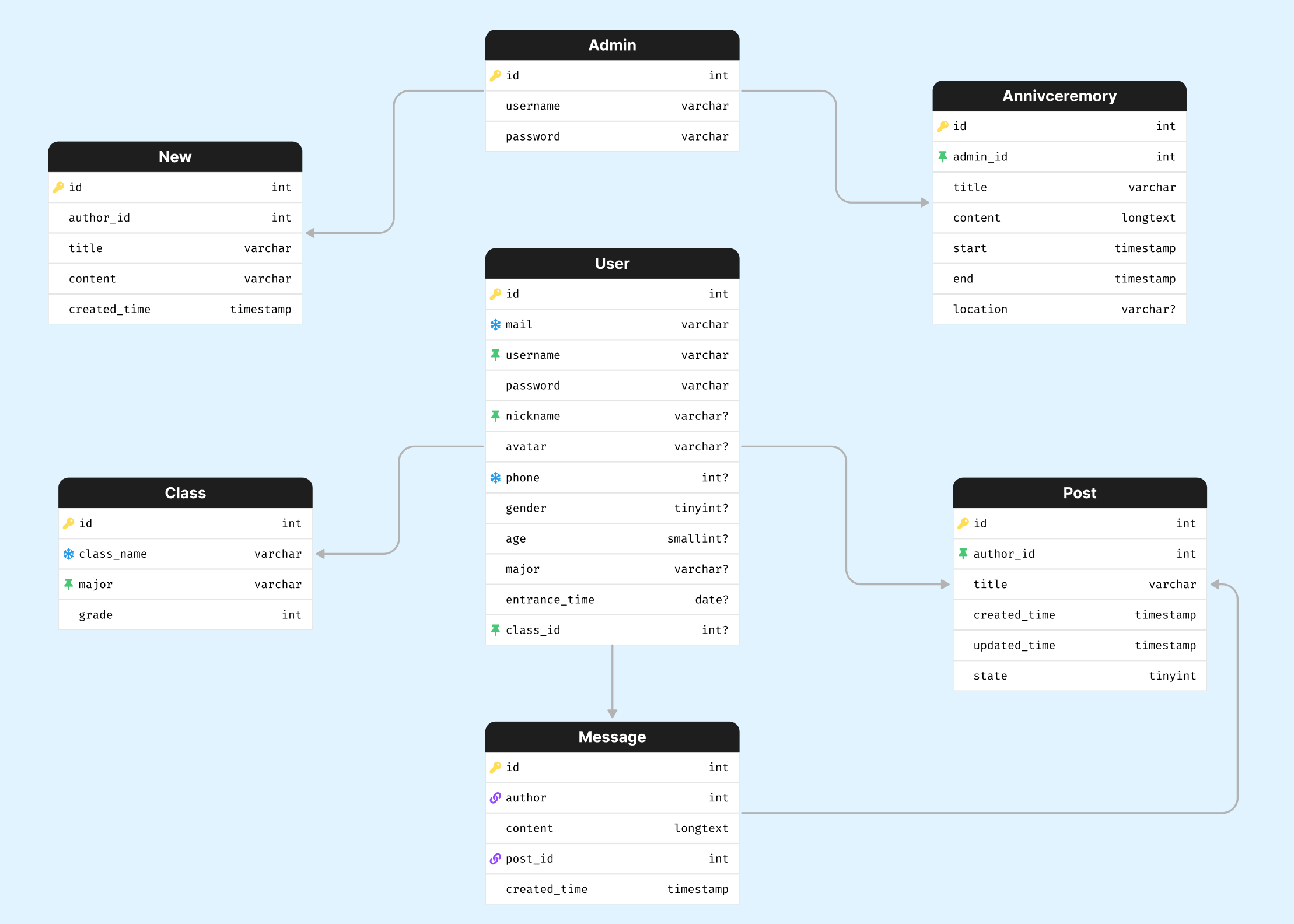
##### 分析

对用户的各属性值预处理以及特征工程后利用KNN算法选取推荐的校友。Python有成熟的机器学习库，基于C++的底层实现使得推荐系统也可以有较好的运行效率。基于RPC技术将结果跨语言进行传递，实现不同语言模块间的通信。

##### 技术选型

* Python
* Sklearn
* gRPC

### 3.4 数据库设计



# **四、验收标准**

## 4.**1 功能模块验收标准**

（1）用户身份操作

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检验点** | **测试功能** |
| **用户身份认证** | 账号密码输入 | 文本输入 | 输入限制 | 输入合法 |
| 登录跳转 | 点击 | 身份确认与页面呈现 | 页面跳转正确 |
| **用户信息完善** | 联系方式 | 11位阿拉伯数字 | 输入限制 | 输入合法 |
| 个人标签 | 点击 | 弹窗 | 标签选择窗口弹出 |
| 标签选择 | 点击 | 选择效果 | 对用户选取的标签区别显示 |

（2）校友交流区

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检验点** | **测试功能** |
| **用户发帖** | 编写帖子 | 文字输入 | 发布帖子 | 帖子发布成功 |
| 板块选择 | 点击 | 板块选择 | 将贴子添加到某一板块下 |
| **阅读帖子** | 进入帖子 | 点击 | 跳转 | 跳转页面正确 |
| 阅读内容 | 滑动 | 内容显示 | 页面内容显示正确 |
| **用户回帖** | 编写回复 | 文字输入 | 发表回复 | 回复发表成功 |
| **搜索** | 主题帖搜索 | 检索词 | 检索词和检索出来的内容是否匹配 | 显示合法 |
| **管理员功能** | 发布帖子 | 文字输入 | 发布帖子 | 帖子发布成功 |
| 隐藏帖子 | 点击 | 隐藏 | 帖子及其回复被隐藏 |
| 置顶帖子 | 点击 | 置顶 | 帖子及其回复被置顶 |

（3）新闻墙

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检验点** | **测试功能** |
| **新闻阅读** | 新闻页面显示 | 进入首页 | 自动轮播 | 自动轮播功能良好 |
| 新闻页面跳转 | 点击 | 页面跳转 | 页面跳转正确 |
| 新闻内容浏览 | 滑动 | 文章内容可阅读性 | 内容显示正常 |
| 语音朗读 | 点击 | 文章内容朗读 | 对文章正文部分进行朗读 |
| **管理员功能** | 发布新闻 | 文字输入 | 发布新闻 | 新闻发布成功 |
| 修改新闻 | 文字输入 | 修改 | 旧文章被覆盖 |

（4）活动地图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检验点** | **测试功能** |
| **时间轴** | 绘制 | 滑动 | 绘制 | 时间轴信息展示无误 |
| 节点跳转 | 点击节点 | 跳转 | 页面跳转成功 |
| 时间动态效果 | 系统时间 | 时间节点自动前进 | 时间节点前后的事件显示有区别 |
| **标点地图** | 地点标注 | 点击 | 跳转 | 跳转至正确的页面 |

（5）打卡地图

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检验点** | **测试功能** |
| **照片展示** | 锚点跳转 | 点击锚点 | 跳转 | 页面跳转成功 |
| 照片切换 | 点击照片 | 切换 | 由旧的照片切换为新的 |
| **地点介绍** | 介绍内容浏览 | 滑动 | 介绍内容可阅读性 | 内容显示正常 |
| 语音朗读介绍文本 | 点击 | 文本内容朗读 | 对介绍的文本进行朗读 |

（6）校友推荐

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检验点** | **测试功能** |
| **校友推荐** | 校友推荐 | 点击 | 推荐相关性 | 推荐校友的相关性符合预期 |

## **4.2 系统结构设计验收标准**

满足MVC设计模式。

MVC设计模式是将整个系统划分为以下三个部分：

（1）表现层（Presentation layer）：

包含表示代码、用户交互GUI、数据验证。 该层用于向客户端用户提供GUI交互，它允许用户在显示系统中输入和编辑数据，同时 系统提供数据验证功能。

（2）业务逻辑层（Business layer）：

包含业务规则处理代码，即程序中与业务 相关专业算法、业务政策等等。该层用于执行业务流程和制订数据的业务规则。业务逻 辑层主要面向业务应用，为表示层提供业务服务。

（3）数据持久层（Persistence layer）：

包含数据处理代码和数据存储代码。数 据持久层主要包括数据存取服务，负责与数据库管理系统（如数据库）之间的通信。

三个层次的每一层在处理程序上有各自明确的任务，在功能实现上有清晰的区分， 各层与其余层分离，但各层之间存有通信接口。

## **4.3 数据库设计验收标准**

#### 4.3.1 数据库设计

数据库设计应该至少要满足第三范式：

第一范式（1NF）：强调的是列的原子性，即列不能够再分成其他几列。

第二范式（2NF）：首先是 1NF，另外包含两部分内容，一是表必须有一个主键；二是没有包含在主键中的列必须完全依赖于主键，而不能只依赖于主键的一部分。

第三范式（3NF）：首先是 2NF，另外非主键列必须直接依赖于主键，不能存在传递依赖。即不能存在：非主键列 A 依赖于非主键列 B，非主键列 B 依赖于主键的情况。

#### 4.3.2 业务算法与数据流

业务算法应该充分考虑到不同用户、不同应用场景下的用户需求差异。例如，对于不同用户等级，应该给予差异化功能实现。

（1）业务算法应该满足场景要求；

（2）业务算法应该很好的体现出设计模式的数据流向。