**知吾校园相亲论坛系统软件设计规格说明书**

**目录**

[1、引言 4](#_Toc18346)

[1.1 编写目的 4](#_Toc31179)

[1.2 读者对象 4](#_Toc251)

[1.3 软件项目概述 4](#_Toc13860)

[1.4 文档概述 4](#_Toc3116)

[1.5 定义 4](#_Toc26748)

[1.6 参考资料 4](#_Toc15237)

[2、软件设计约束 4](#_Toc16618)

[2.1 软件设计目标和原则 5](#_Toc20973)

[2.2 软件设计的约束和限制 5](#_Toc9534)

[3、软件设计 5](#_Toc6764)

[3.1 软件体系结构设计 5](#_Toc9972)

[3.2 用户界面设计 6](#_Toc20695)

[3.3 用例设计 6](#_Toc27048)

[3.4 类设计 6](#_Toc3319)

[3.5 数据设计 6](#_Toc1592)

[3.6 部署设计 6](#_Toc11160)

# **1、引言**

## 1.1 编写目的

明确系统需求和功能：通过详细阐述系统的需求和功能，确保开发团队对系统目标和功能有清晰的认识，避免开发过程中的误解和不必要的返工。

提供开发标准和方法：规格说明书应明确系统应遵守的编程规范、接口标准、安全性要求等，以确保开发的软件具有一致性和可维护性。

指导开发过程：作为开发团队的参考手册，规格说明书应在整个开发过程中随时查阅，以确保开发团队遵循统一的设计思路和实现方法。

保证系统质量：通过提供详细的测试计划和标准，确保系统在投入使用前经过充分的测试和验证，从而提高系统的质量和可靠性。

方便后续维护和升级：规格说明书应包含系统的维护要求、升级方法和相关注意事项，以方便后续对系统的维护和升级工作。

统一团队沟通语言：规格说明书应使用简洁明了的语言描述系统需求、功能和设计，以确保团队成员之间使用统一的语言进行沟通和协作。

降低风险：通过在开发初期就明确系统的需求和设计，可以降低开发过程中出现的风险和不确定性，从而减少返工和延期的可能性。

提升用户体验：规格说明书应关注用户体验方面的要求，包括界面设计、操作流程等，以确保系统能够提供良好的用户体验。

指导用户使用：规格说明书应包含用户手册和操作指南，以指导用户正确地使用系统和处理常见问题。

符合相关法规标准：规格说明书应确保开发的系统符合相关的法规和标准要求，以确保系统能够顺利地投入市场并满足用户需求。

## 1.2 读者对象

用户、软件设计人员、程序员、软件需求分析人员、质量保证人员、软件测试工程师、配置管理工程师。

## 1.3 软件项目概述

* 项目名称: 知吾校园相亲论坛系统
* 用户单位: 大学生群体
* 开发单位: 软件工程知吾校园相亲论坛系统小组
* 软件项目的背景和大致功能：随着人们生活水平的提高，对精神生活的追求也越来越高，相亲作为一种传统的认识方式，在校园中仍然有着广泛的市场。电子相亲，识赛博情人，本软件由523宿舍全体单身成员提出，为广大学生提供一个安全、可靠、有趣的相亲平台，帮助他们更快地认识合适的人并建立稳定的感情。经过本项目开发人员讨论，最终确定软件名称为“知吾”。“知吾”，即知我者，亦为知己。红尘漫漫，争渡争渡，知我之者，为我辈红颜知己也。主体功能有普通用户的私聊、个人空间、信笺、帖子功能；管理员有系统管理、发布公告。

## 1.4 文档概述

1）软件设计的目标是，根据软件系统的需求（包括功能性需求和非功能性需求），综合考虑软件开发过程中的各种制约因素（如技术、资源、进度等），遵循软件工程的设计原则（如模块化、信息隐藏、问题分解等），给出软件系统的实现解决方案和蓝图，产生可指导编码实现的设计模型及文档。约束限制有：代码开发规范：阿里巴巴Java开发手册，API规范：RESTful,变量及函数命名规范：变量-小驼峰，函数-大驼峰，类名-大驼峰，数据库表名：XXX\_XXX\_XXX

1. 该软件系统大致可分为：用户帖子论坛子系统、用户私信交流子系统、用户信笺交流子系统以及管理员子系统。其中，用户帖子论坛子系统中，用户可以在不同的板块发布帖子，给帖子评论，点赞等，在此寻觅有好感的另一半，用户私信交流子系统中，用户在此与其他用户进行私信交流，探寻建立关系的奥秘，用户信笺子系统中，用户在此发布自己的信笺到系统，系统将用户的信笺随机发送给正在打捞的用户，以实现用户和陌生人之间的私密交流。管理员子系统中，由管理员在此管理用户的各种行为，包括帖子、评论、用户信息等，还能由管理员发布系统公告，并管理。

## 1.5 定义

1.登录（load）：用户填写用户名和密码，进入计算机操作系统。

2.读取（read）：将内存中的指令或数据载入到中央处理器（CPU）寄存器中的过程。

3.查询（find）：是用户通过设置某些查询条件，从表或其他查询中选取全部或者部分数据，以表的形式显示数据供用户浏览。

4.菜单（menu）：电子计算机程序进行中出现在显示屏上的选项列表。

5.操作系统（operating system）：管理计算机硬件与软件资源的计算机程序。

6.修改（change）：改动、删节或增添，改正。

7.删除（del）：将已经不需要了的文件从系统的目录清单中删掉。

8.排序（sort）：将杂乱无章的数据元素，通过一定的方法关键字顺序排列的过程叫做排序。

9.保存（save）：使事物、性质、意义、作风等继续存在，不受损失或不发生变化。

10.管理员（Administrator）：一般是指负责一定系统或者软件的维护或管理更新的实际个人或帐号，也有的是专门管理违反原则的。

11.效率（efficiency）：指在给定投入和技术等条件下、最有效地使用资源以满足设定的愿望和需要的评价方式。

12.经费（funds）：行政事业单位为完成工作任务和事业计划所需经常费用。

13.精度（Accuracy）：表示观测值与真值的接近程度。

14.浮点数（Floating Point）：属于有理数中某特定子集的数的数字表示，在计算机中用以近似表示任意某个实数。

15.实数（real number）：是有理数和无理数的总称。

16.字符串（Character string）：由数字、字母、下划线组成的一串字符。

17.响应时间（response time）：计算机显示器成像多个领域的概念。

18.数据（data）：事实或观察的结果，是对客观事物的逻辑归纳，是用于表示客观事物的未经加工的原始素材。

19.运行环境（Runtime environment）：指一种把半编译的运行码在目标机器上运行的环境。

20.运行系统（run-time system）：用来支持程序运行的基础软件。

21.设备(Capital Items) ：是指工业购买者用在生产经营过程中的工业产品，包括固定设备和辅助设备等。

22.软件（software）：一系列按照特定顺序组织的计算机数据和指令的集合。

23.JAVA语言：一门设计语言，广泛应用于开发。

24.文件（record）：国家机构、社会组织或个人在履行其法定职责或处理事务中形成的各种形式的信息记录。

25.SSM：Spring，SpringMVC，Mybatis。

26.Spring：Spring 是最受欢迎的企业级 Java 应用程序开发框架，数以百万的来自世界各地的开发人员使用，Spring 框架来创建性能好、易于测试、可重用的代码。

27.SpringMVC是Spring的一个后续产品，是Spring的一个子项目

SpringMVC 是 Spring 为表述层开发提供的一整套完备的解决方案。在表述层框架历经 Strust、WebWork、Strust2 等诸多产品的历代更迭之后，目前业界普遍选择了 SpringMVC 作为 Java EE 项目表述层开发的首选方案。

28.MyBatis最初是Apache的一个开源项目iBatis, 2010年6月这个项目由Apache Software Foundation迁移到了Google Code。随着开发团队转投Google Code旗下， iBatis3.x正式更名为MyBatis。代码于2013年11月迁移到Github。iBatis一词来源于“internet”和“abatis”的组合，是一个基于Java的持久层框架。 iBatis提供的持久层框架包括SQL Maps和Data Access Objects（DAO）。

29.Maven项目[对象模型](https://baike.baidu.com/item/%E5%AF%B9%E8%B1%A1%E6%A8%A1%E5%9E%8B/1772960?fromModule=lemma_inlink)(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的[项目管理工具](https://baike.baidu.com/item/%E9%A1%B9%E7%9B%AE%E7%AE%A1%E7%90%86%E5%B7%A5%E5%85%B7/6854630?fromModule=lemma_inlink)软件。

30.Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新[Spring](https://baike.baidu.com/item/Spring/85061?fromModule=lemma_inlink)应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的[快速应用开发](https://baike.baidu.com/item/%E5%BF%AB%E9%80%9F%E5%BA%94%E7%94%A8%E5%BC%80%E5%8F%91/206653?fromModule=lemma_inlink)领域(rapid application development)成为领导者。

31.设备(Capital Items)是指工业[购买者](https://baike.baidu.com/item/%E8%B4%AD%E4%B9%B0%E8%80%85/12734085?fromModule=lemma_inlink)用在[生产经营](https://baike.baidu.com/item/%E7%94%9F%E4%BA%A7%E7%BB%8F%E8%90%A5/11064952?fromModule=lemma_inlink)过程中的[工业产品](https://baike.baidu.com/item/%E5%B7%A5%E4%B8%9A%E4%BA%A7%E5%93%81/1113400?fromModule=lemma_inlink)，包括固定设备和辅助设备等。固定设备是主要设备，购买或建设过程要经过一个较长的[决策过程](https://baike.baidu.com/item/%E5%86%B3%E7%AD%96%E8%BF%87%E7%A8%8B/6714639?fromModule=lemma_inlink)。

## 1.6 参考资料

【尚硅谷】SSM框架全套教程，MyBatis+Spring+SpringMVC+SSM整合一套通关：

https://www.bilibili.com/video/BV1Ya411S7aT/?p=69&share\_source=copy\_web&vd\_source=14b67655ff21d6ea29fec9232edeefc1

尚硅谷Maven教程（maven入门+高深，全网无出其右！）：

https://www.bilibili.com/video/BV12q4y147e4/?p=7&share\_source=copy\_web&vd\_source=14b67655ff21d6ea29fec9232edeefc1

尚硅谷百度网盘学习资料：

https://pan.baidu.com/s/10apGpovwhIeWdkLWZXn8Tw#list/path=%2Fsharelink4035995002-345208975156835%2F%E5%B0%9A%E7%A1%85%E8%B0%B7Java%E6%96%87%E6%A1%A3%E6%8A%80%E6%9C%AF%E6%96%87%E6%A1%A3&parentPath=%2Fsharelink4035995002-345208975156835

# **2、软件设计约束**

## 2.1 软件设计目标和原则

（1）软件设计目标

本校园相亲论坛系统旨在提供一个安全、可靠、高效且易于使用的在线交流平台，旨在帮助学生们找到合适的伴侣，并促进校园文化的交流。通过本系统，用户可以轻松地浏览和搜索其他用户的个人信息、发表个人日记和评论、以及与其他用户进行实时交流。因此，本系统的软件设计目标包括以下几点：

易用性：系统应提供简单易用的用户界面，使得用户可以轻松地浏览和搜索其他用户的个人信息、发表个人日记和评论、以及与其他用户进行实时交流。

安全性：系统应采取必要的措施来保护用户隐私和信息安全，包括对用户身份进行验证、对敏感操作进行授权、以及加密传输数据等。

可扩展性：系统应具备良好的可扩展性，可以方便地添加新功能、新模块以及支持更多用户。

可维护性：系统应具备易于维护的特点，方便开发人员进行升级、维护和故障排除。

用户体验：系统应关注用户体验，提供个性化的推荐和服务，使用户可以更加便捷地找到合适的伴侣。

（2）软件设计原则

为了实现上述软件设计目标，本系统的软件设计将遵循以下原则：

面向对象原则：采用面向对象的设计方法，将问题域中的实体抽象为对象，并定义类和对象之间的关系，以提高代码的可重用性和可维护性。

单一职责原则：每个类应该只有一个职责，即完成一个特定的功能。这将有助于减少类之间的耦合，提高代码的可维护性和可重用性。

接口隔离原则：通过定义和使用专门的接口来隔离不同的模块和组件之间的依赖关系。这将有助于减少模块之间的耦合，提高代码的可维护性和可重用性。

## 2.2 软件设计的约束和限制

* 运行环境要求：Windows7以上系统、Linux活跃版本如：Ubuntu或CentOS
* 开发语言：Java、JavaScript、HTML、CSS
* 标准规范： 阿里巴巴Java开发手册
* 开发工具：IDEA、VSCode

# **3、软件设计**

## 3.1 软件体系结构设计

该软件系统大致可分为：用户帖子论坛子系统、用户私信交流子系统、用户漂流瓶交流子系统以及管理员子系统。其中，用户帖子论坛子系统中，用户可以在不同的板块发布帖子，给帖子评论，点赞等，在此寻觅有好感的另一半，用户私信交流子系统中，用户在此与其他用户进行私信交流，探寻建立关系的奥秘，用户漂流瓶子系统中，用户在此发布自己的漂流瓶到系统，系统将用户的漂流瓶随机发送给正在打捞的用户，以实现用户和陌生人之间的私密交流。管理员子系统中，由管理员在此管理用户的各种行为，包括帖子、评论、用户信息等，还能由管理员发布系统公告，并管理。

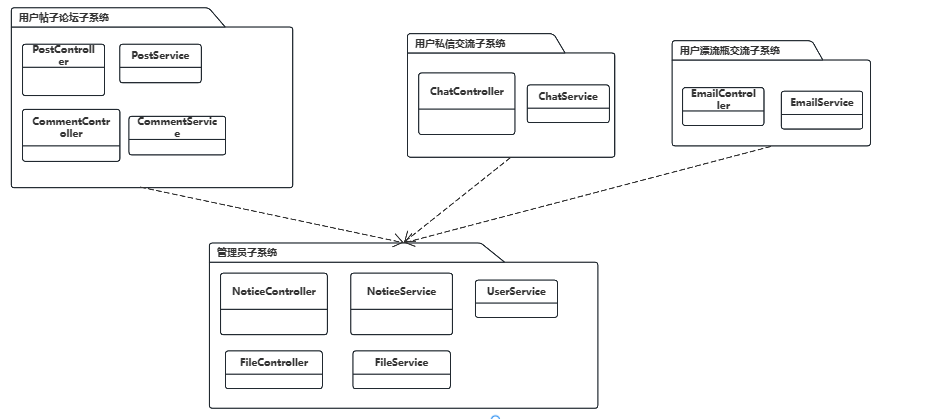


图3.1.1知吾校园相亲论坛系统体系结构图

用户帖子论坛子系统：

它负责给用户提供一个群体交流的平台，用户可以在此查看不同板块的帖子，并在不同的板块中发布自己的帖子，还可以在不同的帖子下发布自己的评论，点赞等。该子系统依赖于管理员子系统。它内部包含控制帖子行为的PostController以及控制帖子评论的CommentController。

管理员子系统：

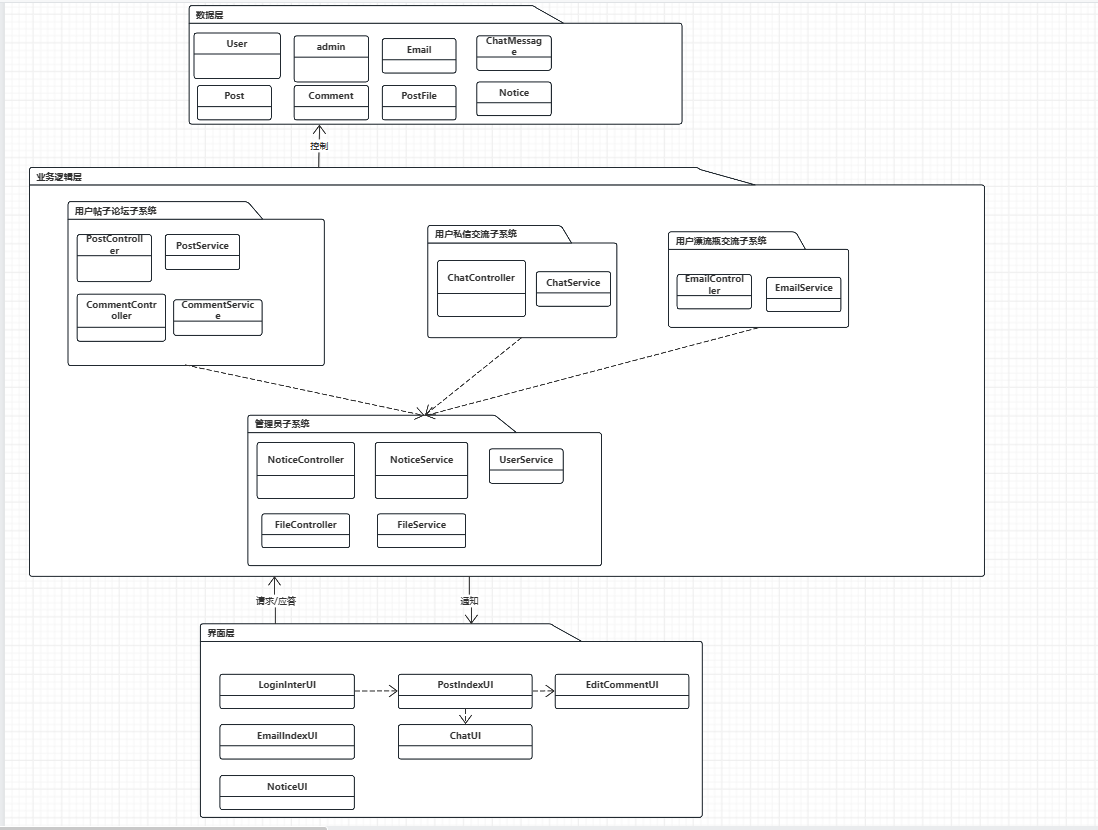
该系统中管理员拥有最高权限，管理员可在此对用户的帖子、评论、信息等进行增添、修改、删除等操作，同时管理员拥有发布和管理公告的权限，对于公告部分其内部包含管理公告的NoticeController。

用户信笺子系统：

它负责给用户提供一个随机交流的平台，用户可以在此查看不同用户的信笺，并可以发布自己的信笺。该子系统依赖于管理员子系统。它内部包含控制信笺行为的EmailController。

用户私信交流子系统：

它负责给用户提供一个聊天的平台，用户可以在此与对方进行聊天，进行行交流。该子系统依赖于管理员子系统。它内部包含控制聊天行为的ChatlController。



整体上，知吾校园相亲论坛系统分为三个层次，有数据层、业务逻辑层、界面层。其中，数据层用于接收数据库传来的数据，放到各个实体类中进行操作。业务逻辑层用于处理基本的逻辑行为，界面层用于和用户进行交互跳转。

## 3.2 用户界面设计

根据“知吾校园相亲论坛系统”的用例描述以及每个用例的交互图，可以发现该软件系统在用户端和管理员端需要有以下一组界面以支持用户的操作。

- 相亲论坛登录页面（LoginInterUl）：本界面为用户进行登录的界面，用户需要输入注册时所选用的用户名并输入设定的密码即可。



- 相亲论坛主页（PostIndexUI）：此页中显示不同帖子板块，用户点击一个板块后，进入板块，在此浏览该板块中的帖子。

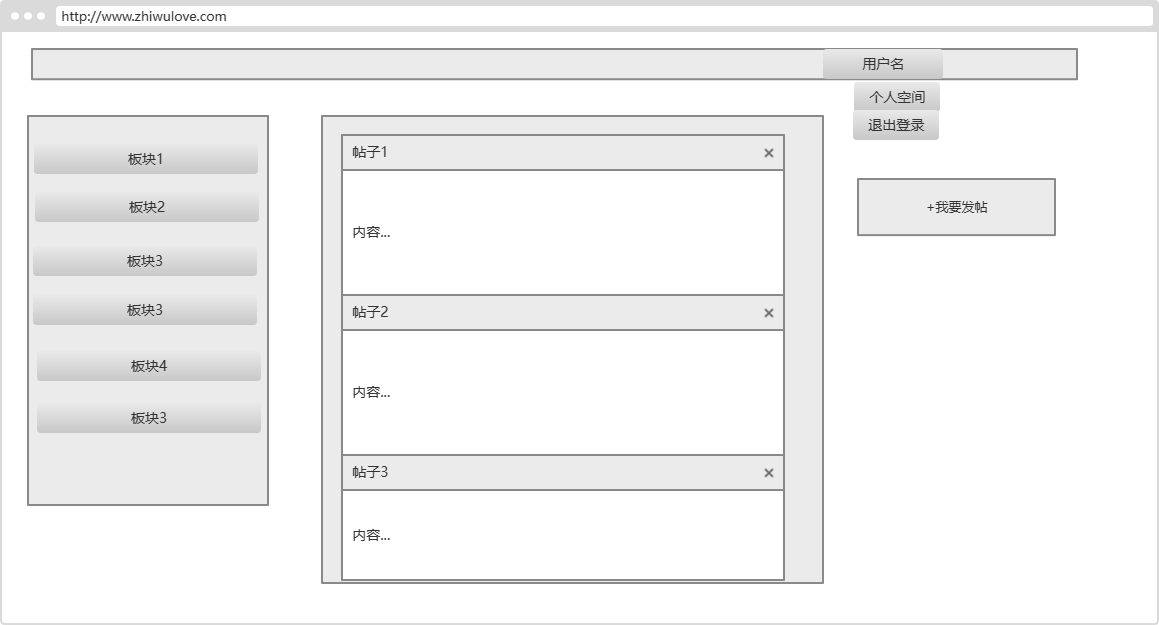


图3.2.1相亲论坛主页

- 帖子主页（PostUI）：此页显示帖子详情内容，包括其他用户发送的评论。

此页还包括评论编辑组件（EditCommentUI），用户可点击发布评论进行编辑评论并发布。



图3.2.2帖子主页

- 帖子编辑页(EditPostUI)：此页中用户进行编辑想要发布的帖子内容。

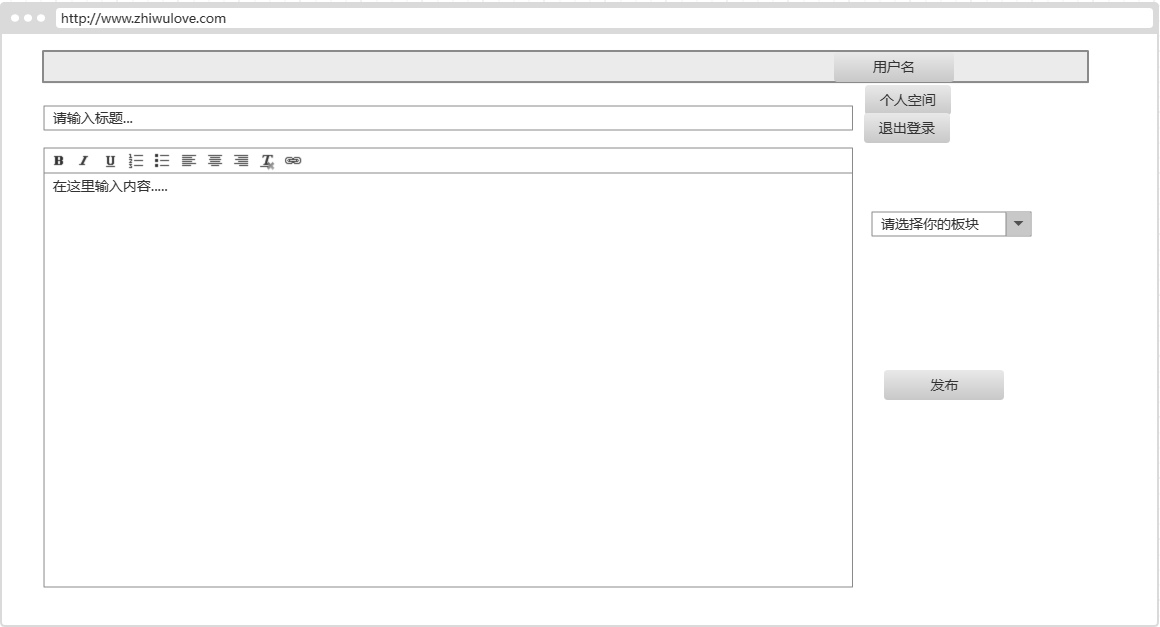


图3.2.3帖子编辑页

- 信笺编辑页(EditEmailUI)：此页中用户进行编辑想要发布的信笺内容。

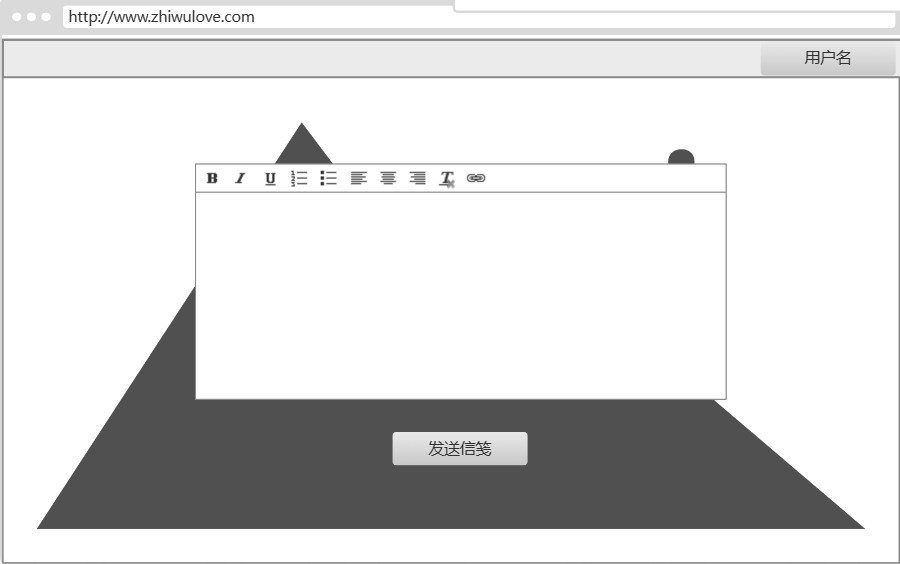


图3.2.4信笺编辑页

- 信笺主页（EmailUI）：此页显示获取信笺的详情内容，包括发送人，时间。

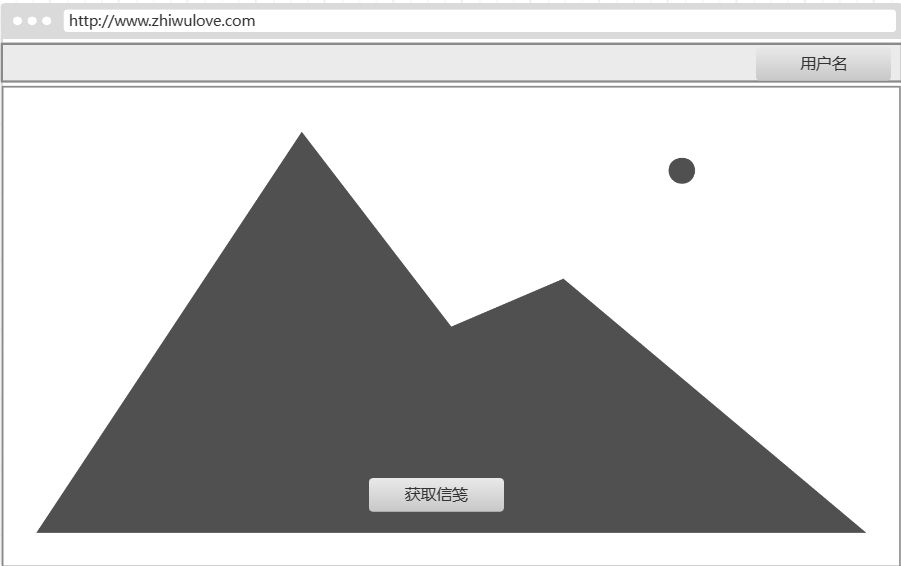


图3.2.5信笺获取页

私聊消息页(ChatUI)：此页中两名用户可以互相进行聊天。

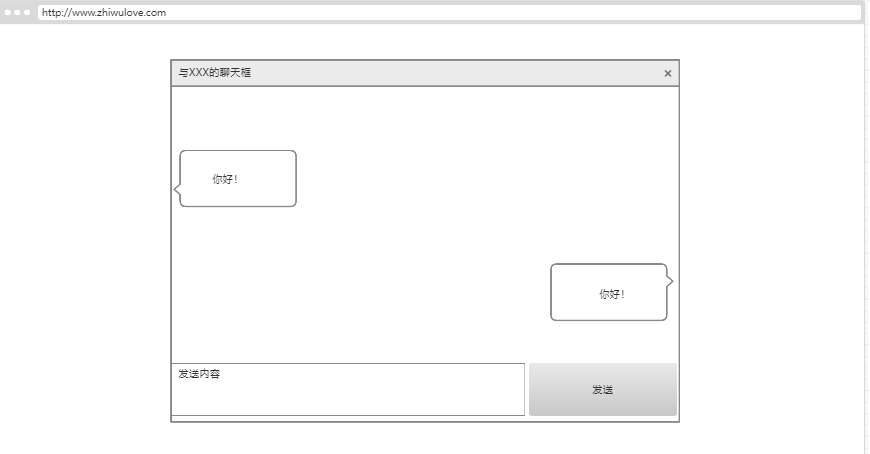


图3.2.6私聊消息页

- 公告管理页面（NoticeUI）：此页可实现公告的发布、查看、删除功能。



图3.2.7公告管理页

在此填写界面描述：

## 3.3 用例设计

（1）用户发帖实现方案：

用户点击在论坛主页的我要发帖按钮，跳转到帖子编辑页面，输入好内容之后，点击发布按钮，前台携带帖子内容，将请求发送到后端，由PostController接收，PostController将内容传递到PostService,经由SendPostContent(Content)函数，将帖子内容信息保存到数据库中进行持久化。同时，PostController调用文件传输流，将帖子内容里的文件对象经由FileService中的WriteFileStreamToMinio(fileStream)传输到对象存储服务器Minio中。之后将发帖结果逐层返回至PostController中，发帖成功后，跳转到论坛主页。

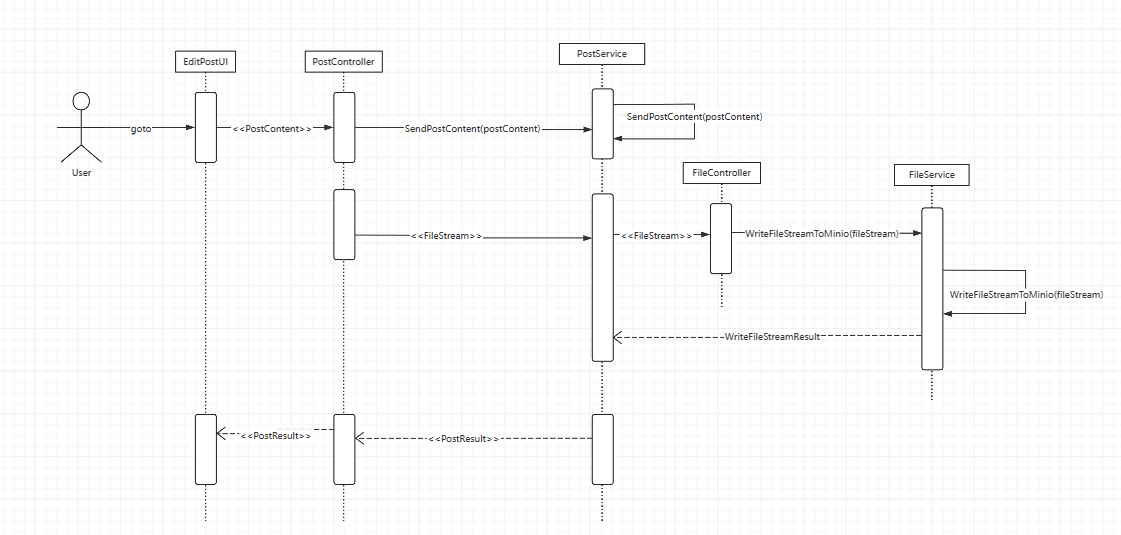


图3.3.1用户发帖时序图

1. 用户浏览帖子实现方案：

用户登录后，访问论坛主页，前端将请求发送至后端，PostController接收请求，再传递到PostService，进行SearchAllPost操作，将数据库中对应页的帖子信息查出，并返回，接下来，PostController经由SearchAllComment(PostList)函数，调用CommentController,再经由CommentService,SearchComment(PostID)，将帖子对应的评论查出，将评论数据CommentList返回，再在PostController封装成postList，返回前端，将帖子信息进行展示。

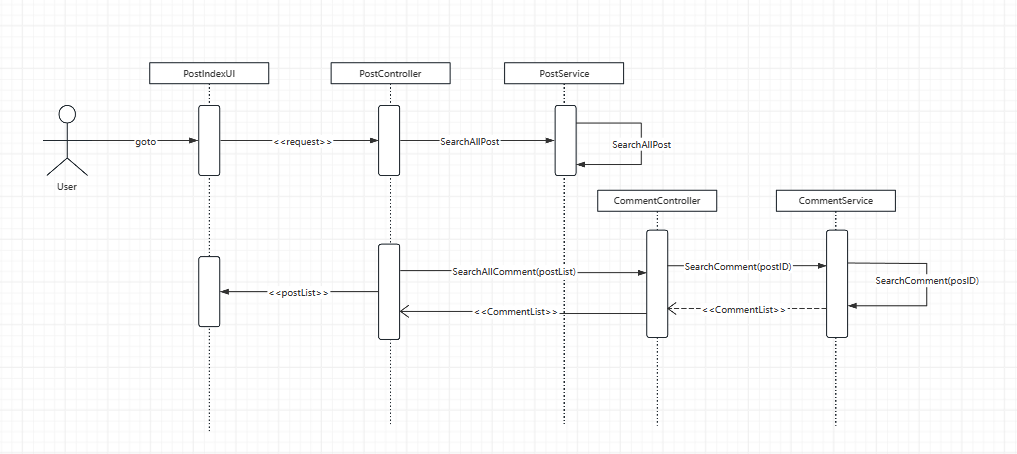


图3.3.2用户浏览帖子时序图

1. 用户发评论实现方案：

用户在帖子详情页，编辑好要发布的评论后，点击发布按钮，前端携带评论信息以及用户、帖子关联ID，将请求发送至后端，由CommentController接收，再传递到CommentService，经由addComment(userID，postID，commentContent)函数，将该条评论信息保存到数据库进行持久化处理，数据库返回结果后，再将发布结果逐层返回至CommentController，再将发布结果打包，发送到前端CommentUI进行显示。

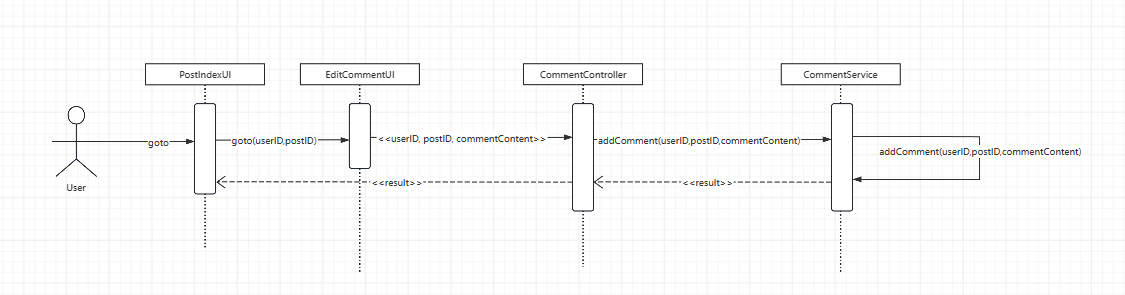


图3.3.3用户发评论时序图

在此填写用例设计：

（4）用户发送信笺实现方案：

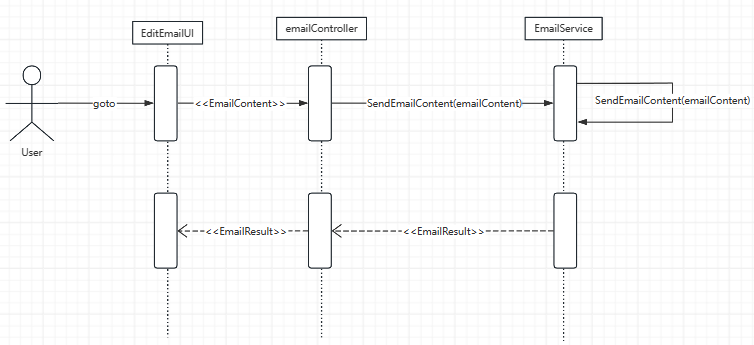
用户点击在论坛主页的我要发帖按钮，跳转到帖子编辑页面，输入好内容之后，点击发布按钮，前台携带信笺内容，将请求发送到后端，由EmailController接收，EmailController将内容传递到EmailService,经由SendEmailContent(Content)函数，将信笺内容信息保存到数据库中进行持久化。同时，PostController调用文件传输流，将信笺内容里的文件对象经由FileService中的WriteFileStreamToMinio(fileStream)传输到对象存储服务器Minio中。之后将发送结果逐层返回至EmailController中，发送成功后，跳转到论坛主页。

图3.3.4用户发送信笺时序图

（5）用户接收信笺实现方案：

用户登录后，访问论坛主页，前端将请求发送至后端，EmailController接收请求，再传递到EmailService，进行SearchAllPost操作，将数据库中对应页的信笺信息查出，如果系统中没有信笺则返回无信笺查看，并返回，如果系统中有信笺则接下来，EmailController经由SearchAllComment(PostList)函数，再在EmailController封装成EmailList，返回前端，将选中的信笺进行展示。

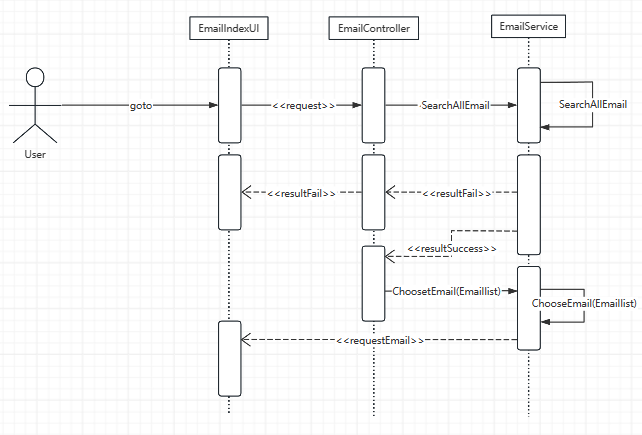


图3.3.5系统筛选信笺时序图

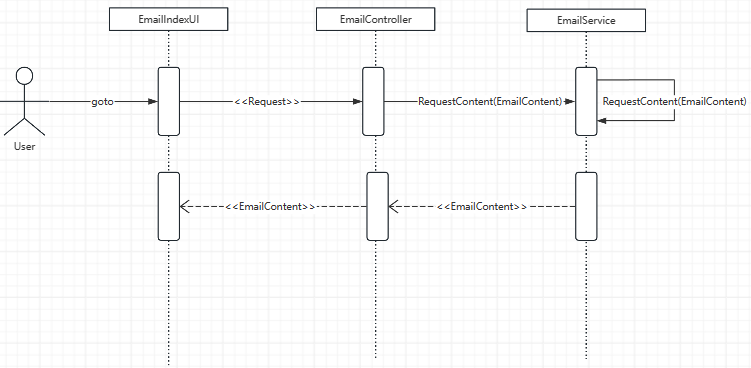


图3.3.6用户接收信笺时序图

1. 管理员发送公告实现方案：

管理员在发布界面，编辑好要发布的公告后，点击发布按钮，前端携带公告信息将请求发送至后端，由NoticeController接收，再传递到NoticeService，经由AddNotice(NoticeContent)函数，将该公告保存到数据库中，再返回到NoticeController，最后发送到前端NoticeUI进行显示。

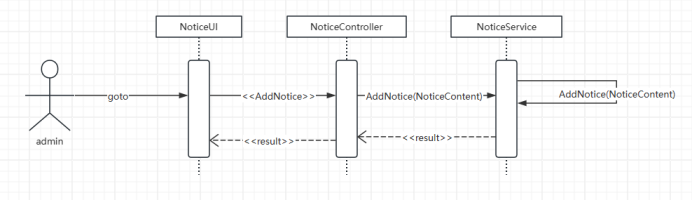


图3.3.7管理员发送公告时序图

1. 管理员查看、删除公告实现方案：

管理员在公告界面可以选择查看公告，前端携带查看公告信息将请求发送至后端，由NoticeController接收，再传递到NoticeService，通过SearchNotice(NoticeID)函数查询对应公告，查询到对应公告后将其封装为Notice返回到NoticeController，最后返回到前端通过NoticeUI显示公告。

管理员在公告界面选择删除公告，前端携带公告信息将删除请求发送至后端，由NoticeController接收，再传递到NoticeService，通过DelNotice(NoticeID)函数删除对应公告，将删除结果返回到NoticeController，最后返回到前端通过NoticeUI显示删除结果。

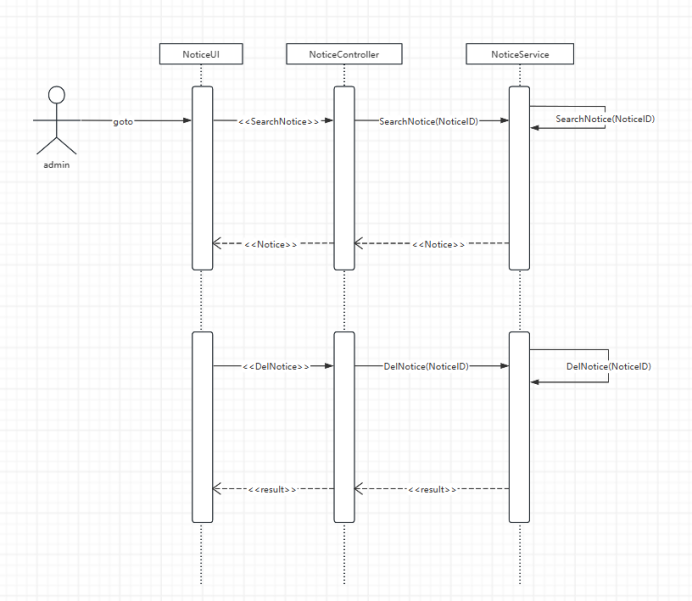


图3.3.8管理员查看、删除公告时序图

1. 用户登录实现方案：

用户在注册登录界面可以选择注册新用户，前端携带注册信息将请求发送至后端，由UserController接收，再传递到UserService，通过RegisterUser(UserInfo)函数注册新用户，将注册结果返回到UserController，最后返回到前端通过LoginInterUl显示注册结果。

用户在注册登录界面可以选择登录用户，前端携带登录信息将请求发送至后端，由UserController接收，再传递到UserService，通过LoginUser(UserInfo)函数验证用户信息，将验证结果返回到UserController，最后返回到前端通过LoginInterUl显示登录结果。

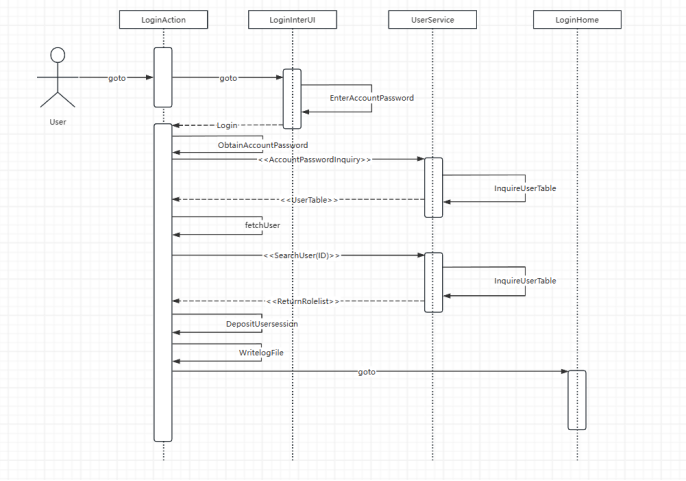
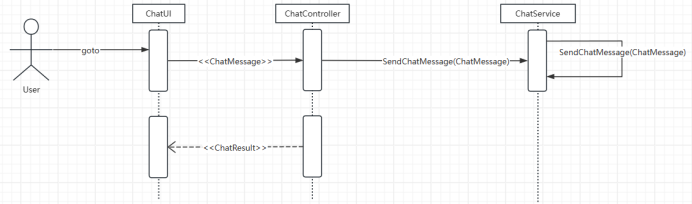


图3.3.9用户登录时序图

1. 用户发送私聊消息实现方案：

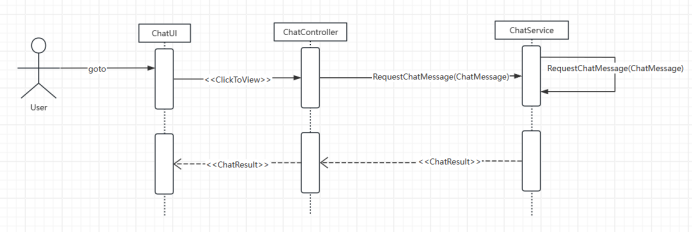
用户点击在他人主页的私聊按钮，跳转到聊天，输入好内容之后，点击发送按钮，前台携带私聊内容，将请求发送到后端，由ChatController接收，ChatController将内容传递到ChatService,经由SendChatContent(Content)函数，将私聊内容信息保存到数据库中进行持久化。同时，PostController调用文件传输流，将信笺内容里的文件对象经由FileService中的WriteFileStreamToMinio(fileStream)传输到对象存储服务器Minio中。之后将发送结果逐层返回至ChatController中，发送成功后，跳转到论坛主页。



3.3.10用户发送私聊消息时序图

1. 用户接收私聊消息实现方案：

用户登录后，访问个人主页，前端将请求发送至后端，ChatController接收请求，再传递到ChatService，将数据库中对应页的私聊信息查出，通过ChatResult返回到ChatController，再由ChatUI进行展示。



3.3.10用户查看私聊消息时序图

## 3.4 类设计

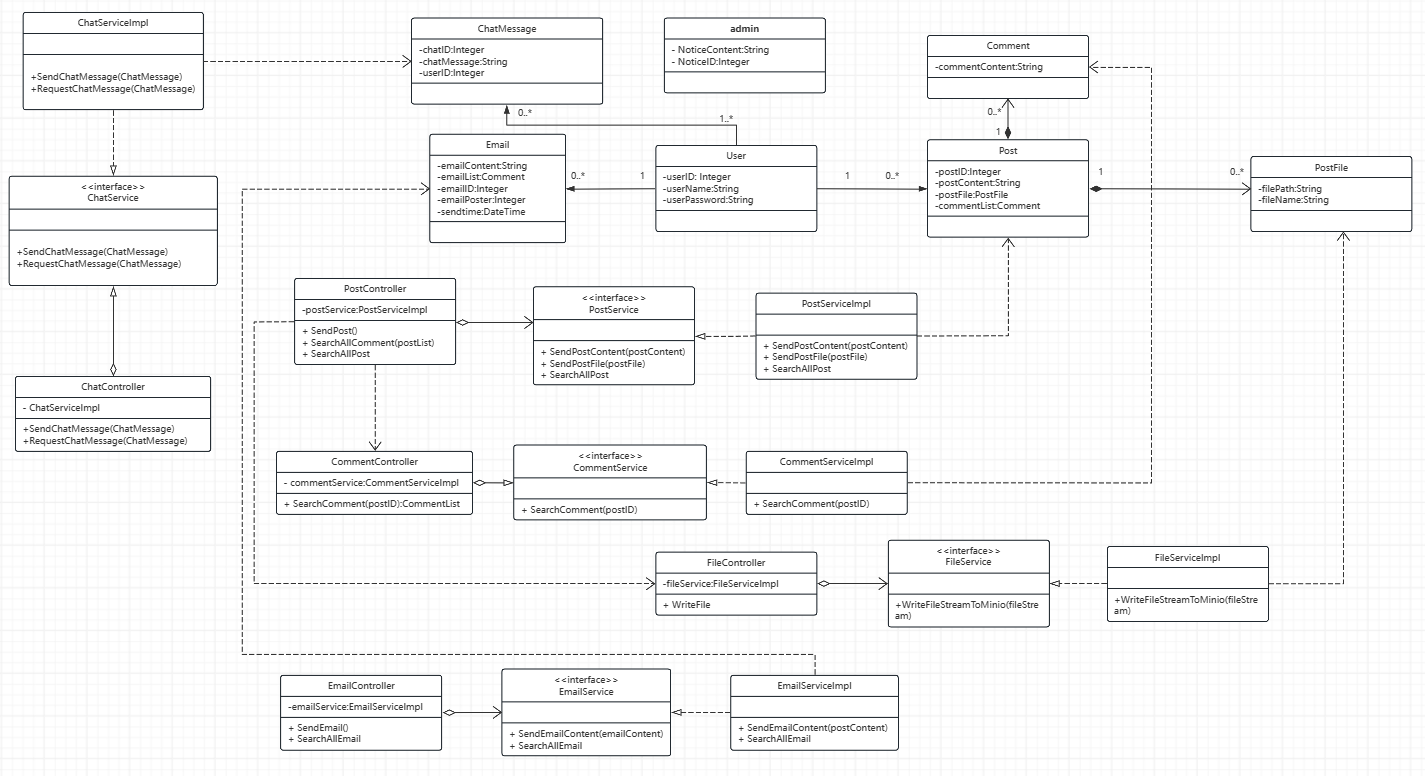


图3.4.1知吾校园相亲论坛系统类图

(1)精化帖子类中的属性：

postID：帖子ID，作为唯一可以索引到的主键，属性可见范围为私有，即“private”，类型为Integer。

postComment:帖子文本内容，属于帖子的私有信息，对其他外部类不可见，属性可见范围为私有，即“private”，类型为String。

postFile:帖子中附带的文件对象。

commentList:帖子中的评论列表，帖子与评论是一对多的关系，在进行封装时，将评论列表封装为一个List在帖子类中，属性可见范围同样是私有。

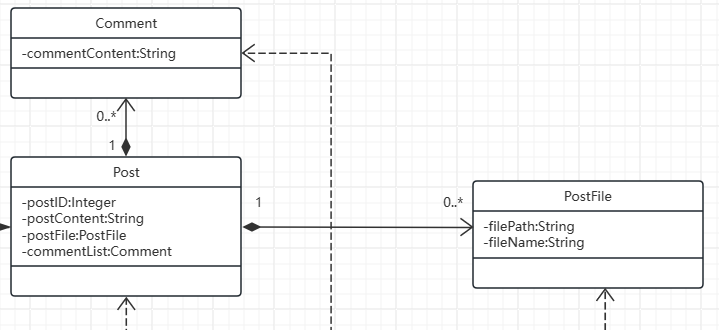


图3.4.2帖子相关类图

(2)精化信笺类中的属性：

emailID：信笺ID，作为唯一可以索引到的主键，属性可见范围为私有，即“private”，类型为Integer。

emailContent:信笺文本内容，属于信笺的私有信息，对其他外部类不可见，属性可见范围为私有，即“private”，类型为String。

emailList:系统存放信笺的列表，类型为Comment。

emailPoster:信笺的发送者，类型为Integer。

sendtime:信笺的发送时间，类型为DateTime。

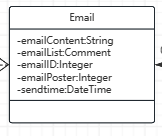


图3.4.3信笺相关类图

（3）精化管理员类中的属性：

NoticeContent：公告内容，属于公告的私有信息，对其他外部类不可见，属性可见范围为私有，即“private”，类型为String。

NoticeID：公告ID，作为唯一可以索引到的主键，属性可见范围为私有，即“private”，类型为Integer。

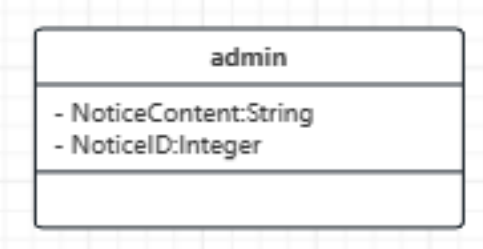


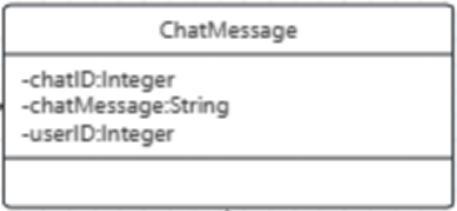
图3.4.4管理员相关类图

（4）精化私聊消息类中的属性：

chatID:聊天消息的ID，属于聊天的私有信息，对其他外部类不可见，属性可见范围为私有，即“private”，类型为String。

chatMessage:聊天内容，该消息的聊天过程中包含的内容，类型为String。

userID:用户的ID，发送者的ID。



在此填写类相关描述：

## 3.5 数据设计

（1）设计永久保存数据的数据库表及字段

针对“知吾校园相亲论坛系统”中的帖子类，为其设计持久保存的数据库表“post”，该表有n个字段：“post\_id”：自增id字段，作为该表唯一主键，类型为int auto\_increment。“post\_content”:帖子内容，类型为text,用于存放帖子的文本内容。“post\_title”：帖子标题，类型为varchar（60），用于存放帖子标题。“post\_file”: 帖子附件，类型为text,用于存放帖子附件所在URL。“user\_id”：帖子发布者id，用于记录本贴是何人发布，类型为int。

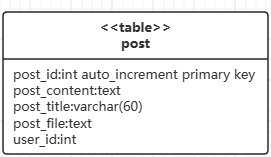


图3.5.1保存帖子数据的表post

针对“知吾校园相亲论坛系统”中的信笺类，为其设计持久保存的数据库表“email”，该表有n个字段：“email\_id”：自增id字段，作为该表唯一主键，类型为int auto\_increment。“email\_content”:信笺内容，类型为text,用于存放信笺的文本内容。“user\_id”：帖子发布者id，用于记录本贴是何人发布，类型为int。

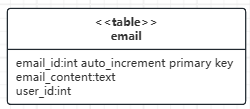


图3.5.2保存信笺数据的表email

1. 设计永久数据的操作

为了支持对“post”数据表的操作，设计模型中有一个关键类:PostService,它提供了一组方法，以实现将Post类对象的数据插入到post表中，或者删除、修改、查询相关信息等。具体接口描述如下：

boolean insertPost(Post)

boolean deletePost(postID)

booean updatePost(Post)

Post getPostByID(postID)

为了支持对“email”数据表的操作，设计模型中有一个关键类:EmailService,它提供了一组方法，以实现将Email类对象的数据插入到email表中，或者删除、修改、查询相关信息等。具体接口描述如下：

boolean insertEmail(Email)

boolean deleteEmail(emailID)

booean updateEmail(Email)

Email getEmailByID(emailID)

## 3.6 部署设计

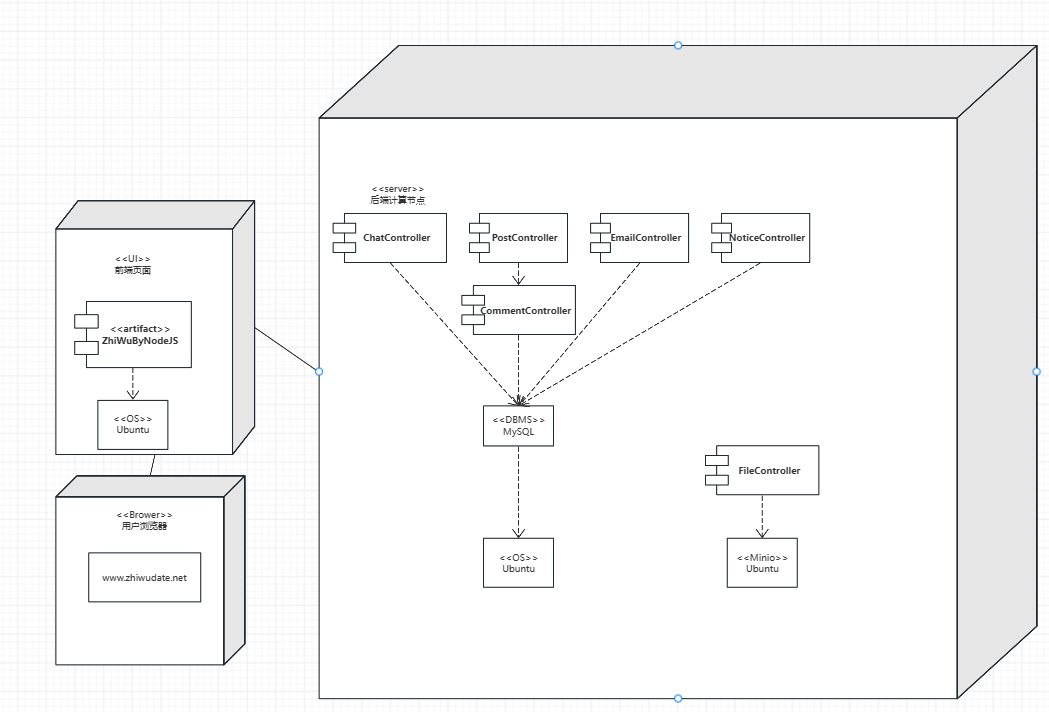


图3.6.1知吾校园相亲论坛系统部署图

“知吾校园相亲论坛系统”为B/S架构，故而不需要在用户端进行部署，用户只需要访问相应网址即可使用。服务端采用分布式部署模式进行部署，具体分为前端服务器，后端服务器，对象存储服务器。其中，前端服务器通过node.js部署在Ubuntu系统的一个端口，后端服务器部署在Ubuntu系统的另外一个端口，或者另外一个服务器上，对象存储服务器同理。后端计算节点还部署了MySQL数据库管理系统，以保存系统中的用户、帖子等信息。各个端之间通过网络进行连接，从而实现交互和通讯。