[一、项目背景 2](#_Toc15532)

[二、项目优势分析和可行性分析 2](#_Toc2162)

[三、系统需求分析 3](#_Toc9331)

[四、 UML用例图 4](#_Toc21538)

[Uml用例图 4](#_Toc29935)

[数据库设计 7](#_Toc4651)

[五、用例规约与流程图 10](#_Toc19968)

[1、用例规约 10](#_Toc26771)

[2、 时序图与流程图 13](#_Toc21655)

[时序图 13](#_Toc11138)

[流程图 16](#_Toc11084)

[六、系统类图 19](#_Toc24058)

[七、界面 20](#_Toc12477)

# 一、项目背景

木交流平台基于前身人人网。通过每个人真实的人际关系，满足各类用户对社交、资讯、娱乐等多方面的沟通需求。把社交这个行为绑定在校园里，校内交流平台的出现在一定程度上解决了因为贴吧过于开放在校园内的一些弊端，比如你在里面说话老师也能看到。更重要的是你在校内可以相对清楚的了解白己周围的同学在说什么在做什么:人们总是喜欢直到一些身边介于熟和不熟的人隐私。这种模式就是帮助人窥探这些隐私的同时不对被窥探人产生打扰。

校内社交平台这一领域，空缺了许多年。近些年，随着大学招生名额的不断扩大，每年的大学生数目也在不断增加。在以前，学生们可以在人入网上迅速地找到校友或者认识一-些其他大学的学生从而达到交流的目的，而这一切都随着校内网的落幕而逐渐消失。虽然说社交通讯软件还是有很多，比如QQ，微信和微博。但是这每一款软件都不是针对学生的社交半台，并且这几个社交半台对于校园群体的功能也乏善可陈。因此我们决定制作一个大学生交流平台，在这个平台上用户可以迅速地寻找自己的校友，发布新鲜有趣的事，同时根据后期的发展，可扩展的范围可以随着用户体验的提升而夫拓展，比如进行学生之间学习之间的交流与沟通，生活上习惯的养成，也可以在不同大学之问建立一座桥梁，达到彼此沟通，从而推动教育事业的发展。

# 二、项目优势分析和可行性分析

1.目前市面上并无特别富有竞争力的校园交流平台网站或者app，并且几大社交平台例如QQ，微信它们对于校园交流的支持性并不是很好。

2.大学年年扩招，并且随着中国的发展，人们逐渐认识到了教育的重要性，因此以后每年的大学生会越来越多。根据国家统计局和教育部发布的最新数据显示，全国2018共有在校大学生人数为2695.8万，应届大学毕业生795万，普通本专科招生748.6万人，全国共有具有大学教育程度人口为1.9593亿人。在校大学生人数:2695.8万人。大学生逐渐成为一个不可忽视的群体，而且日益增长。

3.校内网已经为我们抬明了方向，它作为第一个做校内交流平台的企业为我们累计了宝贵的经验;与此同时地，它的失败也给我们了很多警示，我们既要参考它成功的经验，也要吸取它失败的教训。

# 三、系统需求分析

大学生交流平台总体而言,分为首页和个人主页.其中首页可以看到许多新鲜事情,而个人首页主要针对个人而言,这样就有利于实现多种的功能的不同应用。

从功能来看,大学生交流平台不仅提供类似于facebook的交友功能,而且采取了灵活的页面代码自定义功能,使个人的页面丰富多彩.主导航的作用是提供用户一个快捷方式，方便用户的点击查找。

作为一个叫交流网站，大学生交流平台主要提供日志、相册、好友、即时聊天、群、社区、站内信、校友录、分享等等功能，基本囊括了时下热门和不热门的所有功能，通过这些功能的集合，用户可以实现：

⑴、日志，分享自己的心情和观点，除了具备普通的博客功能外，还有自动通知好友、推荐以及及时反馈别人的点评的功能；

⑵、相册，提供一般的相册服务，上传照片与人分享，并对其尽行管理；

⑶、好友，通过人人网，可以轻松找到许久不曾联系的昔日好友，可以根据自己的偏好主动去搭讪自己想认识的朋友；

⑷、即时聊天；

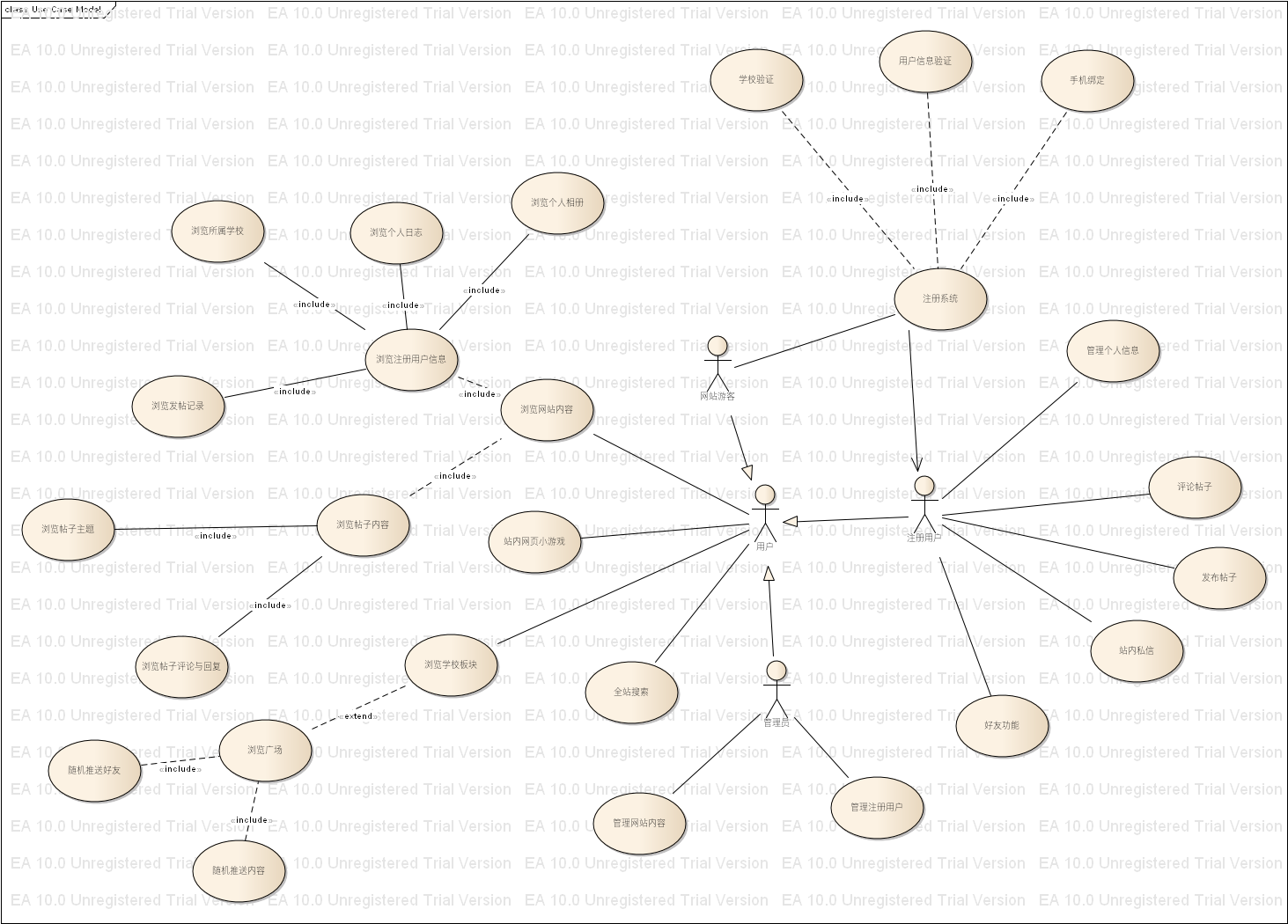
⑸、群、校友录都是常用的功能；

(6)、大学生交流平台与其它网络服务的最大差异就在于其弥补了传统网络人际传播可信度低的弊端，用户实名注册，上传的个人资料也比较真实。通过网站提供的搜索、推荐和根据用户登记的资料挑选可能相识、同兴趣的人认识，就可以让用户认识更多人。按六度分割理论，按此方法，任何人都可以与世界上其他人联系上。

# UML用例图

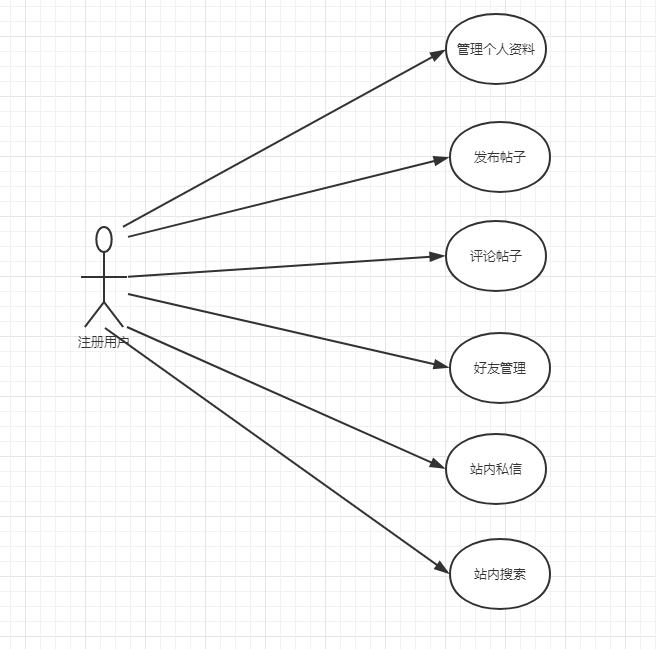
## Uml用例图

4.1-UML用例图



通过图4.1可以清楚看出各个用户所对应的用例及包含内容，和各个用户之间的关系。

4.2-注册用户个人用例



看图4.2知道注册用户有管理个人资料、发布帖子、评论帖子、好友管理、站内私信和站内搜索等用例。

4.3-管理员个人用例

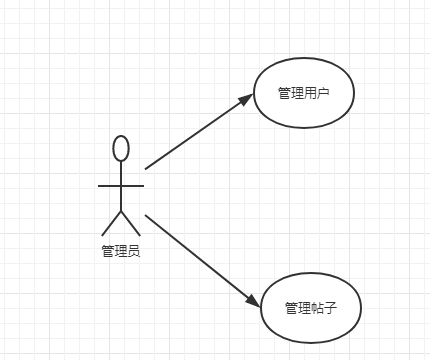
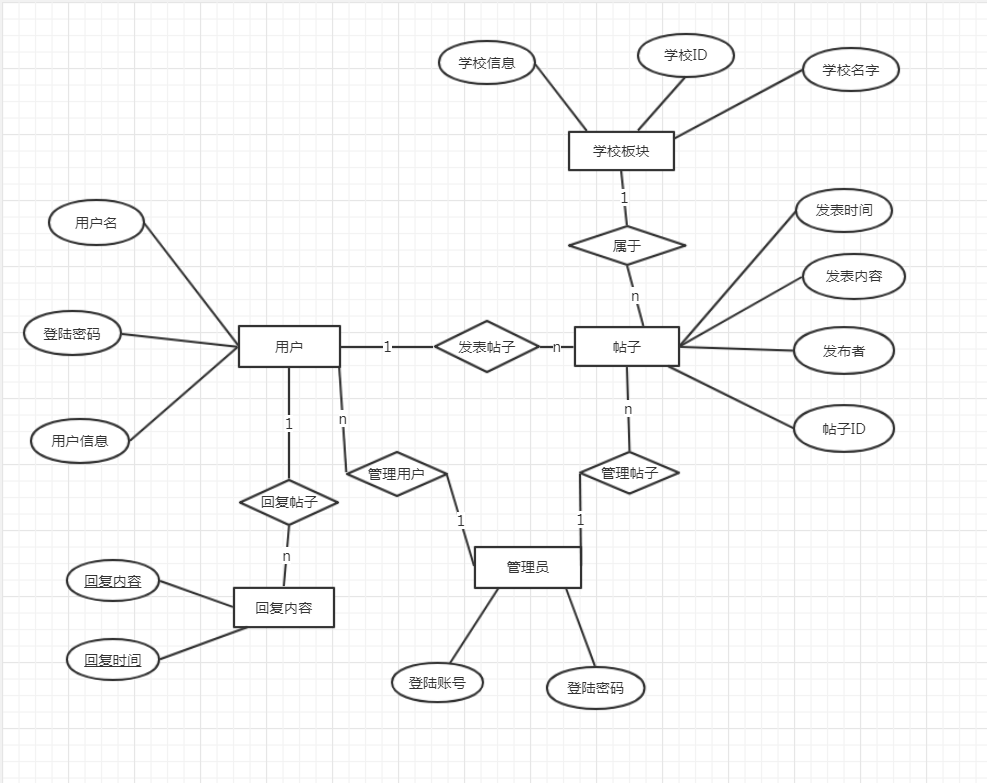


图4.3可看出管理员可以对用户进行修改或者删除用户，管理员也可以对帖子还有评论等内容进行修改或者删除。

## 数据库设计

4.4-数据库ER图



4.5帖子类数据库设计

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 注释 |
| topic\_id | int | 主题ID |
| Topic\_Name | nvarchar(50) | 主题内容 |
| School\_ID | int | 主题所选学校ID |
| Topic\_form | nvarchar(50) | 发帖者用户ID |
| content | nvarchar(50) | 评论内容 |
| from\_uid | Int | 评论用户id |
| topic\_time | datetime | 发布时间 |

表4.5记录帖子的所属学校信息及帖子内容和评论内容

4.6-评论内容数据库设计

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 注释 |
| from\_uid | Int | 评论用户id |
| to\_uid | int | 评论目标用户id |
| Reply\_time | datetime | 评论时间 |
| Reply\_type | Nvarchar(50) | 评论内容 |

表4.6记录用户评论内容和评论时间以及评论某人的评论

4.7-学校类数据库设计

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 注释 |
| School\_ID | int | 学校ID |
| School\_Name | nvarchar(50) | 学校名字 |
| Topic\_id | int | 该模块主贴ID |

表4.7用于储存各个学校里主题帖

4.8-管理员类数据库设计

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 注释 |
| manage\_id | nvarchar(50) | 管理员账号 |
| School\_Name | nvarchar(50) | 管理员密码 |

表4.8记录管理员的登陆所需要的账号密码

4.9用户数据库类设计

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 注释 |
| user\_id | int | 用户ID |
| user\_name | nvarchar(50) | 用户昵称 |
| user-pwd | nvarchar(50) | 用户密码 |

表4.9用于储存用户的信息

# 五、用例规约与流程图

## 1、用例规约

注册用户

|  |  |
| --- | --- |
| 参与者： | 注册用户 |
| 前置条件： | 网站游客通过注册系统 |
| 后置条件 | 注册用户可以：  管理个人信息  评论帖子  发布帖子  站内私信  好友功能 |

管理个人信息

|  |  |
| --- | --- |
| 用例名称： | 管理个人信息 |
| 参与者： | 注册用户 |
| 用例说明： | 注册用户可以通过这个用例来管理自己的个人信息 |
| 前置条件： | 用户是注册用户 |
| 后置条件 | 管理个人信息资料  管理个人评论纪录和发帖纪录 |

评论帖子

|  |  |
| --- | --- |
| 用例名称： | 评论帖子 |
| 参与者： | 注册用户 |
| 用例说明： | 注册用户可以进行对帖子的评论 |
| 前置条件： | 用户是注册用户 |
| 后置条件 | 评论主贴的内容  评论主贴评论的内容 |

发布帖子

|  |  |
| --- | --- |
| 用例名称： | 发布帖子 |
| 参与者： | 注册用户 |
| 用例说明： | 注册用户可以发布帖子 |
| 前置条件： | 用户是注册用户 |
| 后置条件 | 帖子所属学校  帖子内容  帖子主题 |

站内私信

|  |  |
| --- | --- |
| 用例名称： | 站内私信 |
| 参与者： | 注册用户 |
| 用例说明： | 用户可以与发帖人或者评论人进行私信 |
| 前置条件： | 用户是注册用户 |
| 后置条件 | 进行私信  添加好友 |

好友功能

|  |  |
| --- | --- |
| 用例名称： | 好友功能 |
| 参与者： | 注册用户 |
| 用例说明： | 用户可以在网站上进行添加好友等功能 |
| 前置条件： | 用户是注册用户 |
| 后置条件 | 好友私聊  好友管理  添加好友 |

管理员

|  |  |
| --- | --- |
| 参与者： | 管理员 |
| 前置条件： | 管理员为内部人员 |
| 后置条件 | 管理员可以：  管理网站内容  管理注册用户 |

管理注册用户

|  |  |
| --- | --- |
| 用例名称： | 管理注册用户 |
| 参与者： | 管理员 |
| 用例说明： | 管理员可以对注册用户进行管理 |
| 前置条件： | 用户为管理员 |
| 后置条件 | 禁止注册用户的发帖、评论  禁封注册用户 |

管理网站内容

|  |  |
| --- | --- |
| 用例名称： | 管理网站内容 |
| 参与者： | 管理员 |
| 用例说明： | 管理员可以对网站内容进行管理 |
| 前置条件： | 用户为管理员 |
| 后置条件 | 对学校的管理  对帖子的管理  对评论的管理 |

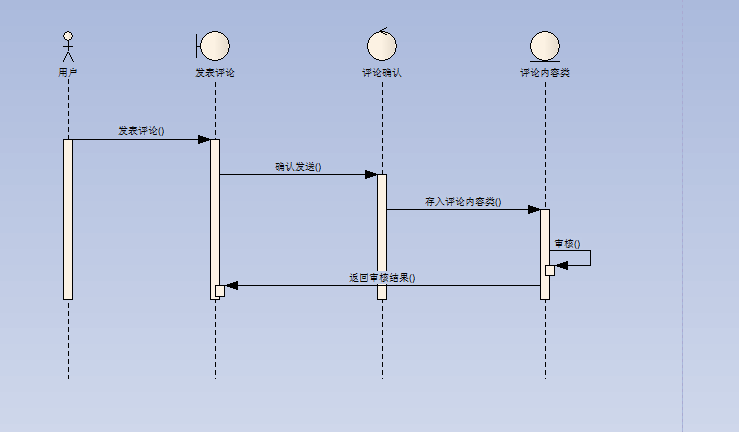
全站搜索

|  |  |
| --- | --- |
| 用例名称： | 全站搜索 |
| 参与者： | 全体用户 |
| 用例说明： | 用户可以通过全站搜索来进行搜索自己想要的信息 |
| 前置条件： | 任何进入网站的用户 |
| 后置条件 | 搜索学校  搜索帖子  搜索用户 |

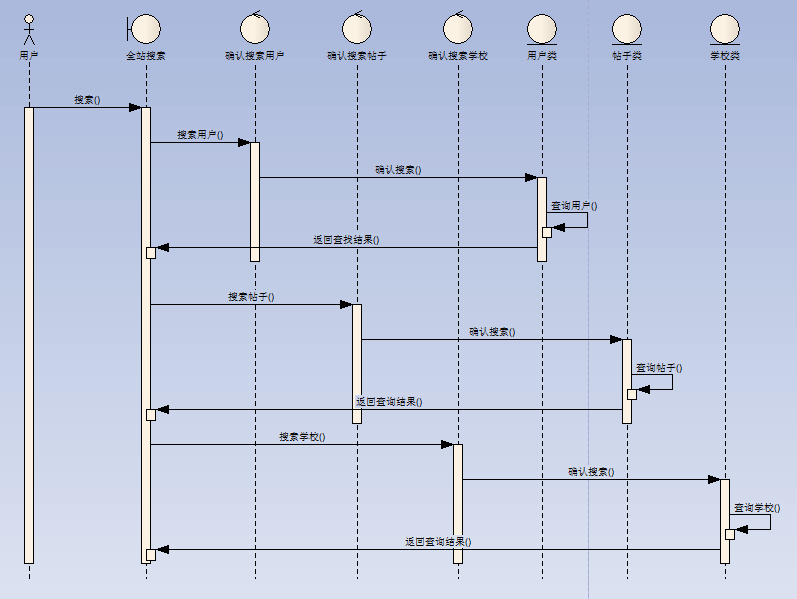
## 时序图与流程图

### 时序图

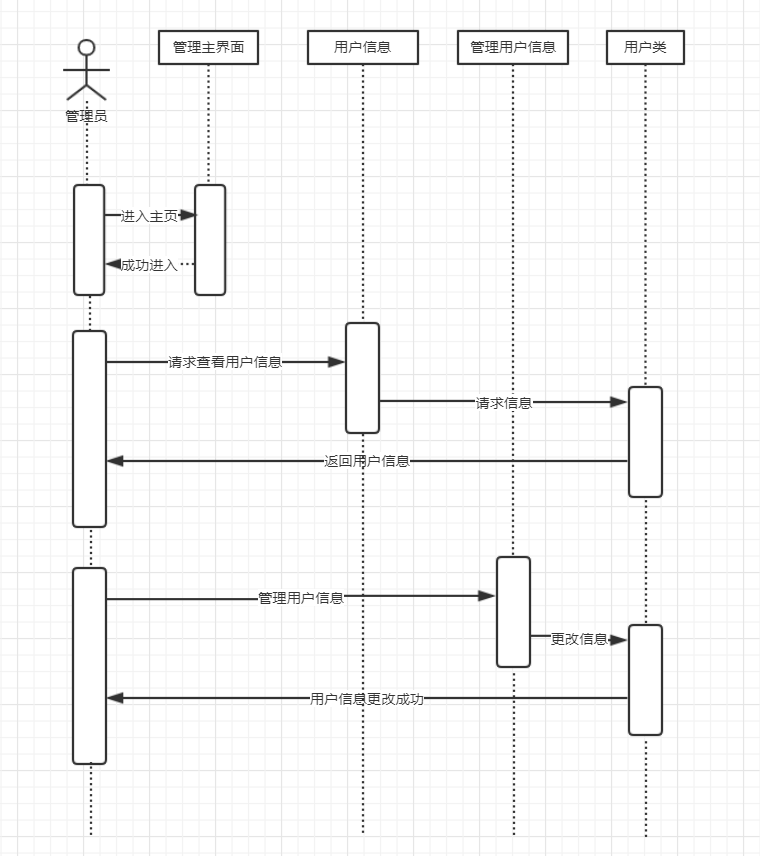
5.1-用户评论顺序图



5.2-全站搜索顺序图

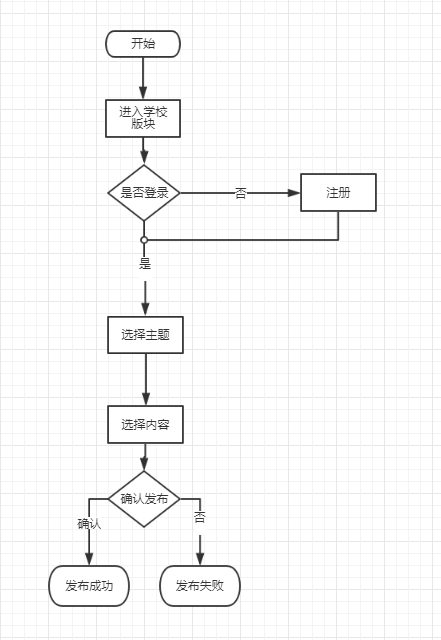


5.3-管理员管理用户顺序图

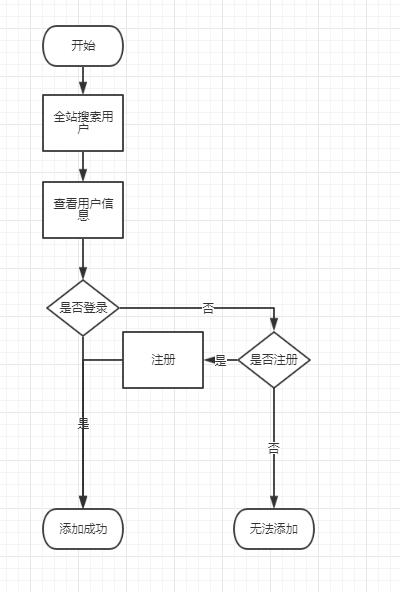


### 流程图

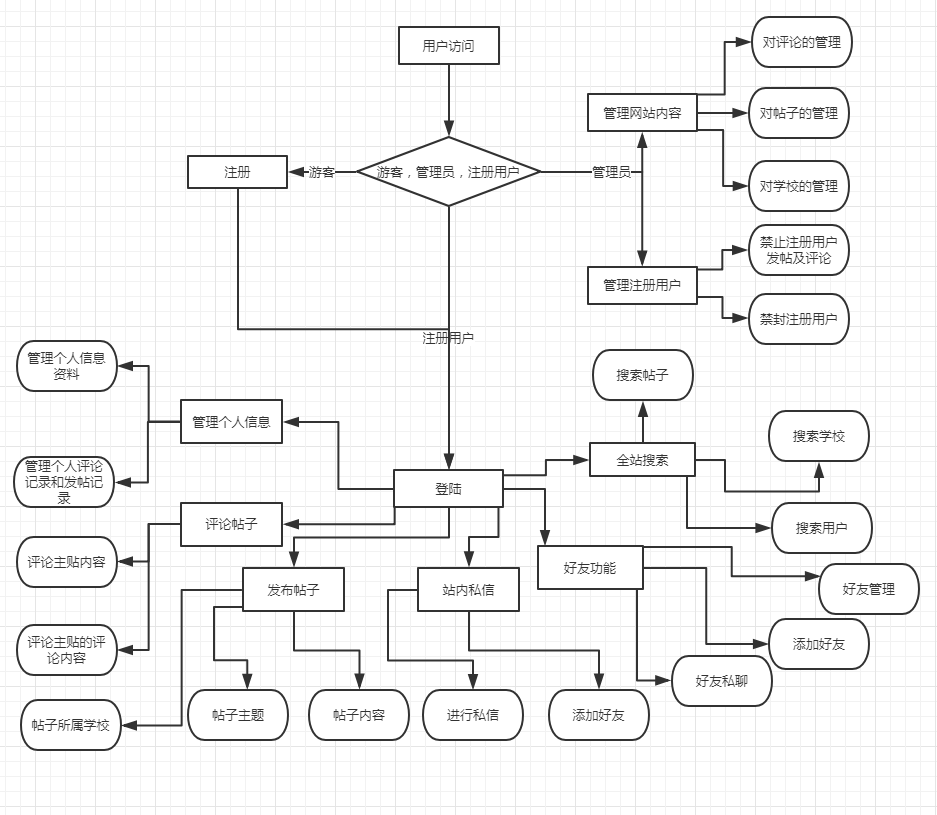
5.4-发布评论流程图



5.5-通过全站搜索添加好友流程图

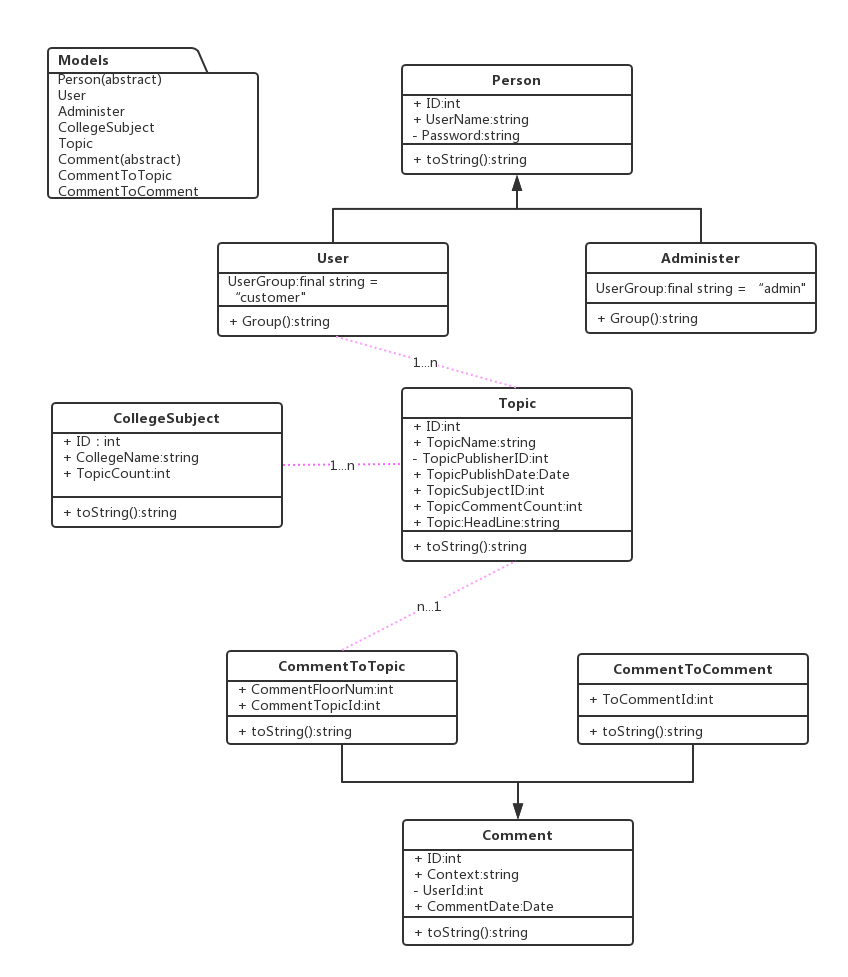


5.6-整体流程图



# 六、系统类图

6.1-系统类图



系统类图注释

|  |  |
| --- | --- |
| 类名 | 注解 |
| Person | 抽象类，作为注册用户和管理员的基类。 |
| User | 注册用户类，用于验证用户信息和保存用户状态。 |
| Administer | 管理员类，用于验证管理员登陆信息。 |
| CollegeSubject | 大学版块类，所有的内容以所在大学模块分类。 |
| Topic | 话题类，注册用户可以发布一个话题，并选择所在大学版块。 |
| Comment | 抽象类，作为评论的基类。 |
| CommentToTopic | 评论类，这个评论是对话题的评论，字段CommentTopicId用来唯一标识所评论的话题。 |
| CommentToComment | 评论类，但是这个评论是对评论的回复，字段ToCommentId用来唯一标识所回复的评论。 |

# 七、界面

7.1-个人信息界面



个人信息界面可以显示用户的基本资料和用户的发帖记录以及用户的评论记录。

7.2-个人资料管理界面



用户个人资料管理界面可以对用户的基本资料进行修改，可以对自己发布的帖子进行管理和回复的评论进行管理。

7.3-好友管理界面



好友管理界面可以对用户的好友进行私信或者添加好友、删除好友。

7.4-添加好友界面



用户可根据图7.4搜索好友添加或者通过推荐好友中推荐的好友添加