

**分析模型**

新闻分享app分析模型



**指导教师：**

**孙萍**

**小组成员：**

**1650350乔 宇**

**1853115李家瑞**

**1852343潘翔宇**

**1852726李逸舟**

目录

[成员贡献度 1](#_Toc40114595)

[修订历史 2](#_Toc40114596)

[1. 介绍 3](#_Toc40114597)

[1.1项目目标 3](#_Toc40114598)

[1.2进展、改进和当前状态 3](#_Toc40114599)

[2. 架构分析 4](#_Toc40114600)

[2.1总体架构设计 4](#_Toc40114601)

[3. 分析机制 5](#_Toc40114602)

[3.1 文字描述 5](#_Toc40114603)

[4. 分析模型 7](#_Toc40114604)

[5. 参考 7](#_Toc40114605)

# 成员贡献度

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **参考** | **介绍** | **构架分析** | **分析机制** | **分析模型** | | | |
| 李家瑞 | 乔 宇 | 乔 宇 | 潘翔宇  李逸舟 | 乔 宇 | 李家瑞 | 潘翔宇 | 李逸舟 |
| 4.1-4.2 | 4.7-4.8  4.11 | 4.3-4.6 | 4.9-4.10  4.12 |

# 修订历史

|  |  |  |  |
| --- | --- | --- | --- |
| **编写日期** | **版本** | **说明** | **作者** |
|  |  |  | 乔 宇 |
|  |  |  | 乔 宇 |
|  |  |  | 李家瑞，潘翔宇 |
|  |  |  | 李逸舟 |
|  |  |  | 潘翔宇 |
|  |  |  | 乔 宇 |

# 1. 介绍

## 1.1项目目标

在智能手机普及之后，各个新闻网站首先开始推出自己网站的 APP 服务（Application 第三方智能手机应用程序），随后其他的传统媒体也及时跟进，三联生活周刊等纸媒也迅速推出了自身的 APP 软件。我们正是参考这些软件，制作一款具有特色的全新的新闻APP。本项目是针对各个年龄段的受众，使他们通过本项目能够迅速查看到自己感兴趣的新闻，用户可以查看新闻、视频，对新闻发表一些自己的评论。同时，本项目也有发表、查看动态功能，找到意向相投的好友，共同商讨自己的看法与生活态度。

APP主要有四大模块：

a. 浏览模块：未登录和登录的用户都可以在此模块浏览新闻、所有人动态，并进行搜索。

b. 社交模块：登录的用户可以发表动态、发表评论、关注他人、被关注、查看关注和自己动态等等。

c. 个人收藏模块：登录的用户可以在这里对新闻进行收藏或取消收藏。

d. 账户管理：用户在这里注册登录，登录后可以查看修改个人信息，查看关注或粉丝列表，查看收藏列表等。

## 1.2进展、改进和当前状态

在之前的报告中我们已经分析了软件需求，完成了需求规约。我们使用用例在不同的系统参与者角度来规范要求，描述关键对象，操作流程及其相互关系。当我们进入分析阶段时，我们分析了系统架构，建立了类图时序图和通信图来捕获详细的参与者行为分析用例，阐明系统中不同组件和不同类中的沟通机制。

我们在讨论过程中发现了一些不足，并对他们做出了改进：

1. 我们将未登录用户的权限扩大，可以浏览新闻
2. 我们将用例分为四个字系统，将系统中的用例细化
3. 将动态页面细分为好友动态，所有动态和个人动态

# 2. 架构分析

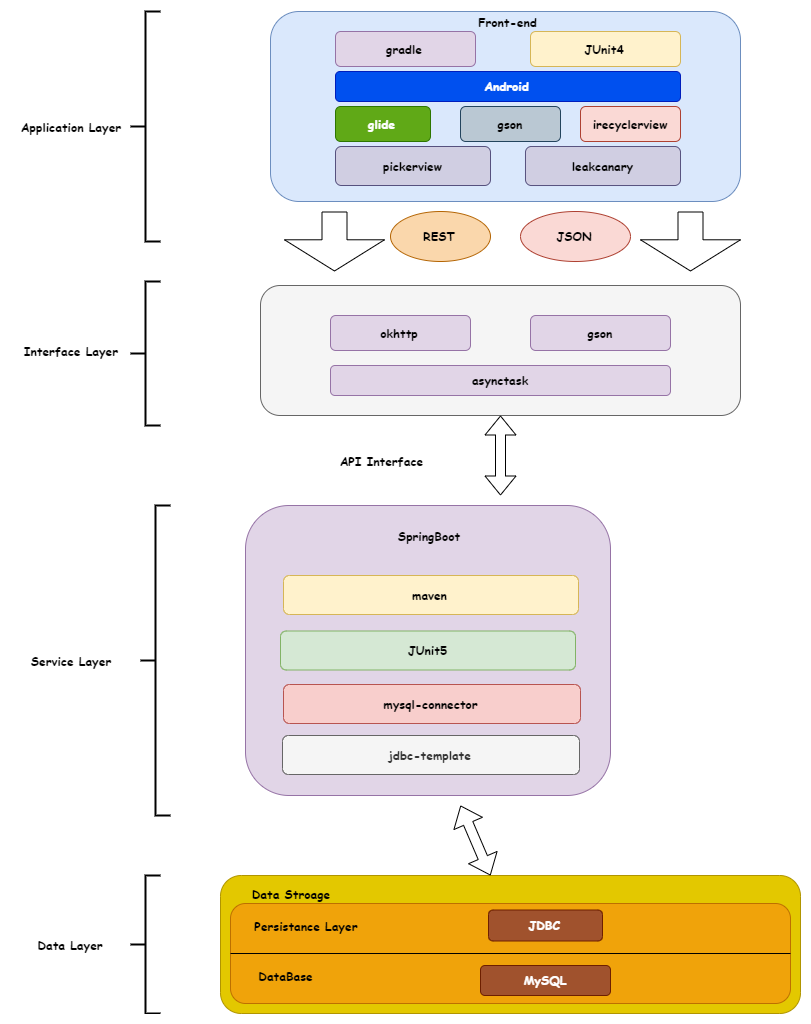
## 2.1总体架构设计

通过分析APP的系统环境，我们对他的上下文设计了ACD图。系统参与者、运行环境、运行要求等都包含在其中。



之后，我们对于系统的整体架构做架构设计，主要包括以下四层：

* 应用层(前端)由于本应用是新闻类应用，用户希望随时随地使用的原因而采用了移动端开发，这里我们选择了安卓平台开发。采用了JUnit4和leakcanary进行测试以及内存检测，同时我们使用了irecyclerview，pickerview等用于组件设计。
* 接口层采用了okhttp发送http请求，用gson进行json与类之间的转换，asynctask用来实现异步访问。前后端接口设计采用了RESTful API
* 服务层(后端)我们采用了SpringBoot框架，用maven安装了依赖，并进行依赖注入，采用Junit5进行单元测试并用mysql-connector和jdbc-template实现与MySQL数据库的连接。
* 数据层我们选择了MySQL数据库，这是一个关系型数据库。并实现了它与后端的连接。具体架构图如下:



# 3. 分析机制

分析机制是在分析阶段被用来降低分析复杂度和通过向设计者提供复杂行为的简单速记形式来提升分析的一致性。

## 3.1 具体分析机制

对本系统进行分析时，我们采用四种分析机制，即进程间通讯机制（Inter-process Communication mechanism）、持久性机制（Persistency mechanism）、传统接口机制（Legacy interface mechanism）和安全机制（Security mechanism）。对于四种机制，我们将分析中涉及的类划分如下：

### 3.1.1进程间通讯机制Inter-process Communication mechanism

User

Customer

ManagementController

NewsController

ChannelController

SearchNewsController

SearchTrendControlle

ViewTrendController

Follower\_CollectionController

UnsubscribeController

ShieldController

PublishController

CollectionController

Log\_RegisterController

NewsCatalog

UserInfoCatalog

UserIDCatalog

TrendCatalog

### 3.1.2持久性机制Persistency mechanism

UserIDCatalog

UserInfoCatalog

NewsCatalog

TrendCatalog

### 3.1.3传统接口机制Legacy interface mechanism

NewsSystem

TrendSystem

UserInformationSystem

UserIdenditySystem

ViewForNews

ChangeForChannel

SearchForNews

SearchForTrend

ViewForTrend

EnterForFollower\_Collection

EnterForUnsubscribe

EnterForShield

EnterForPublish

ViewForCollection

EnterForLog\_Register

EnterForManagement

### 3.1.4安全机制Security mechanism

UserIDCatalog

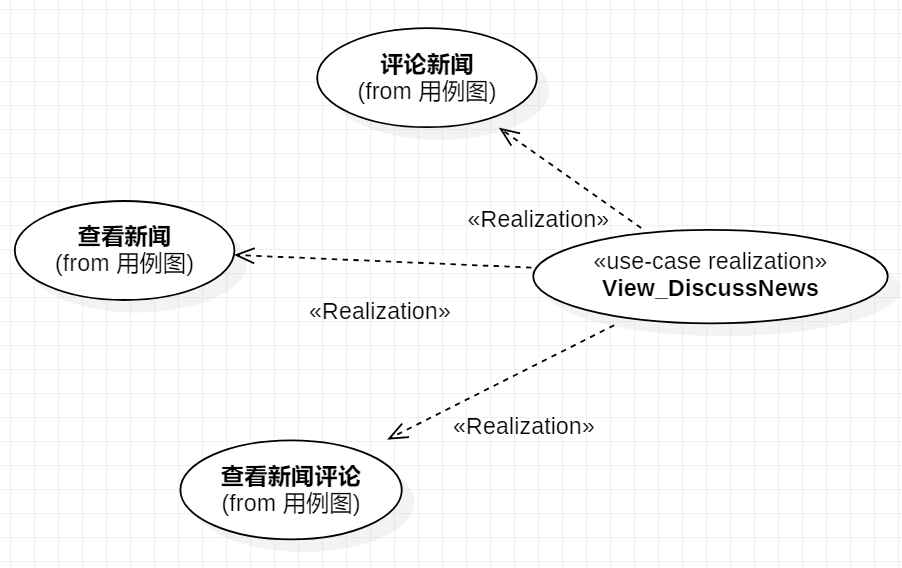
UserInfoCatalog

# 4. 分析模型

## 4.1查看、评论新闻

用户想要查看新闻、查看新闻评论或者评论新闻，则可以选择新闻模块。查看新闻边界类(ViewForNews)负责接受用户的请求和输入的内容，同时相应的展示给用户新闻列表、新闻详情或者一些提示信息，并根据信息向新闻控制类(NewsController)发送请求。此后新闻控制类(NewsController)向新闻系统边界类(NewsSystem)请求新闻列表、新闻详情，当用户评论新闻时，新闻控制类(NewsContoller)向新闻详情实体(NewsDetails)发送创建新的评论。

### 4.1.1用例实现



### 4.1.2类设计

User

ViewForNews

NewsController

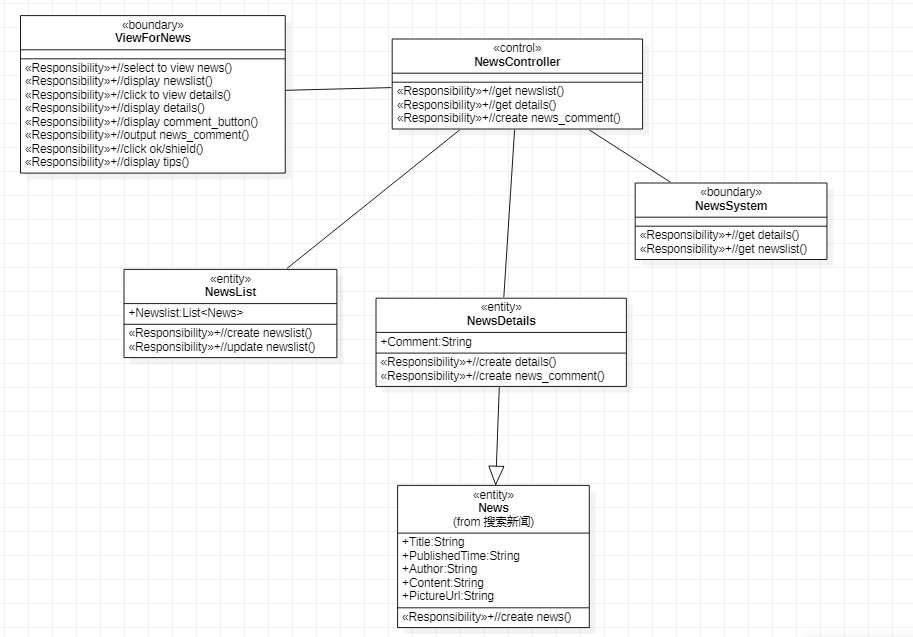
NewsList

NewsDetails

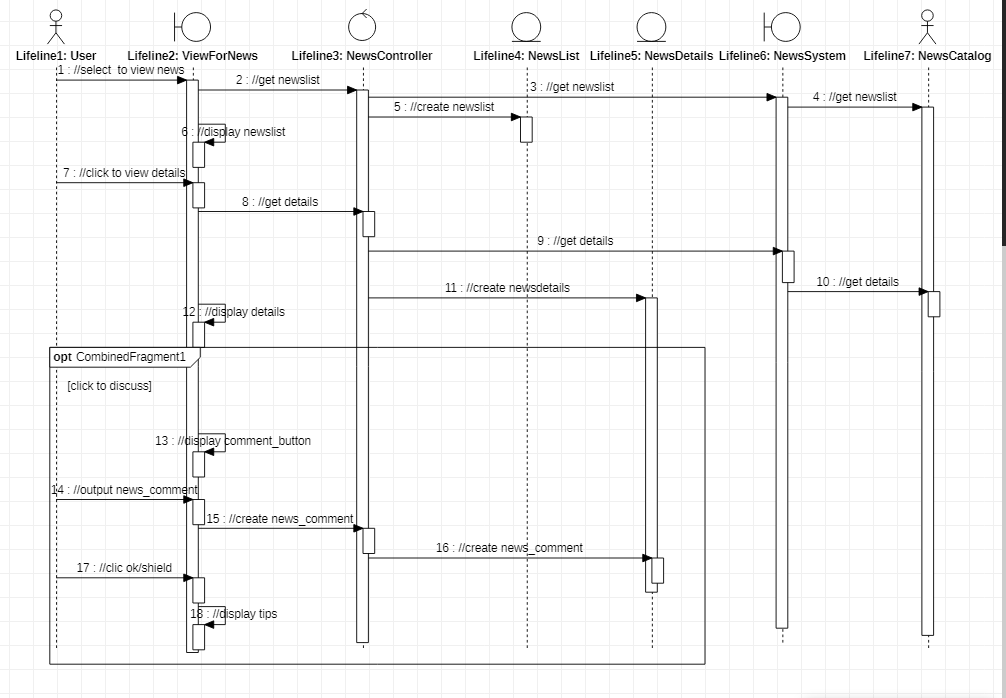
NewsSystem

NewsCatalog

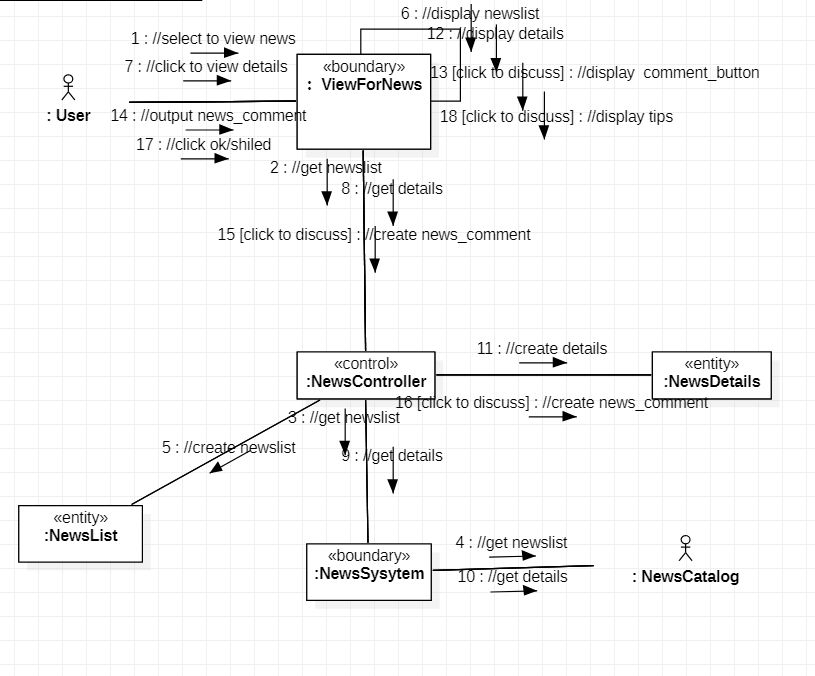
### 4.1.3 参与类图



### 4.1.4 序列图



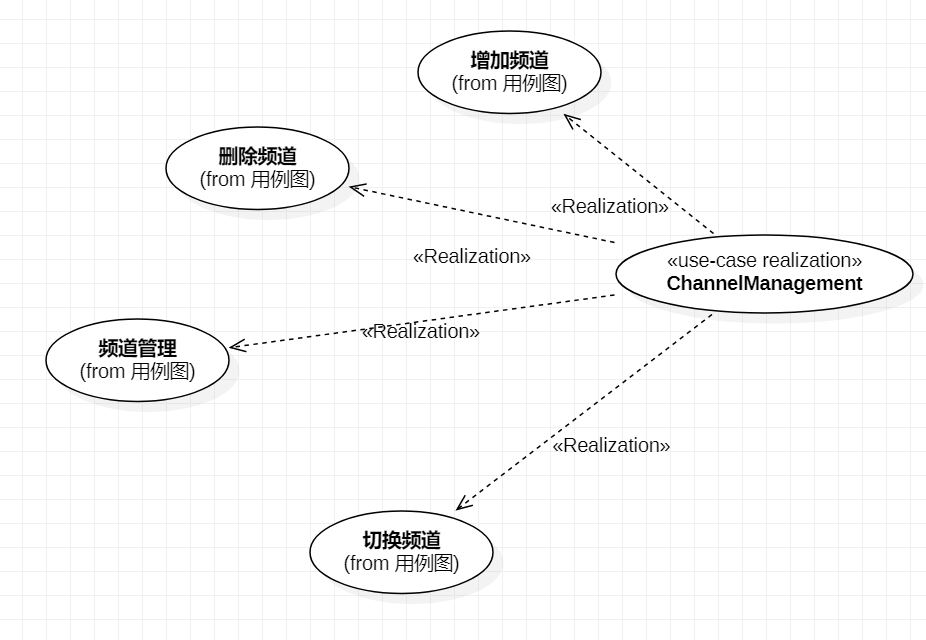
### 4.1.5 通信图



## 4.2频道管理

用户在查看新闻时可以根据自己的喜好去调整显示的新闻的频道类别。用户可以切换、增加或者删除频道，频道改变边界类(ChangeForChannel)接受到相关信息后，显示出相应的按钮以及展示新的新闻列表，同时根据信息向频道控制类(ChannelController)发送请求，频道控制类(ChannelController)向新闻系统边界类(NewsSystem)获取新的新闻列表，并向新闻列表实体类(NewsList)发送更新列表或者创建列表的信息。

### 4.2.1用例实现



### 4.2.2类设计

User

ChangeForChannel

ChannelController

ChannelList

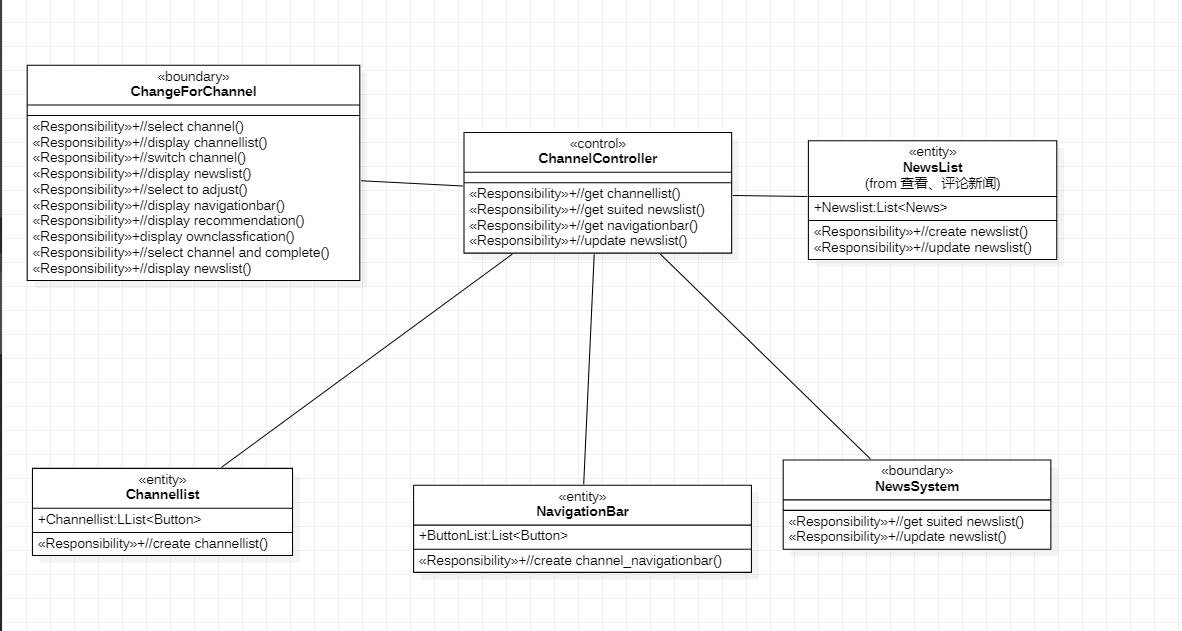
NavigationBar

NewsList

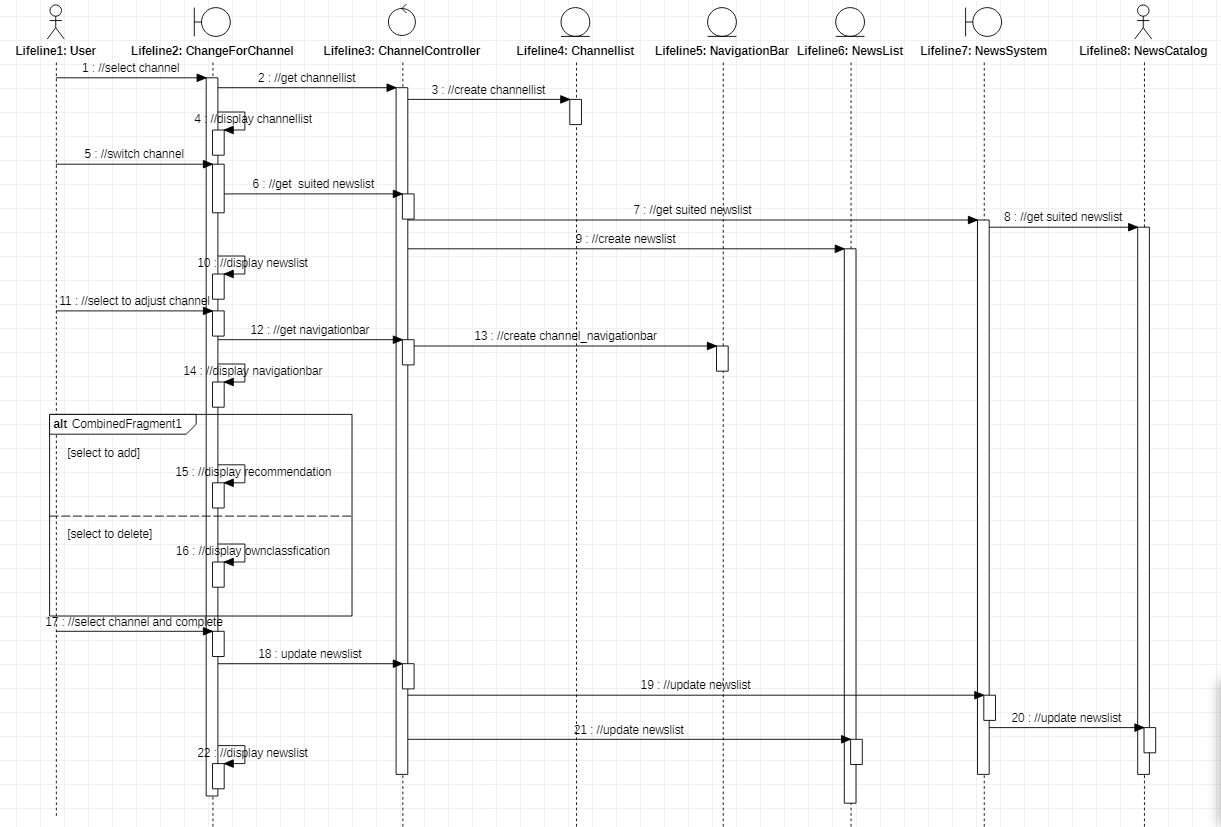
NewsSystem

NewsCatalog

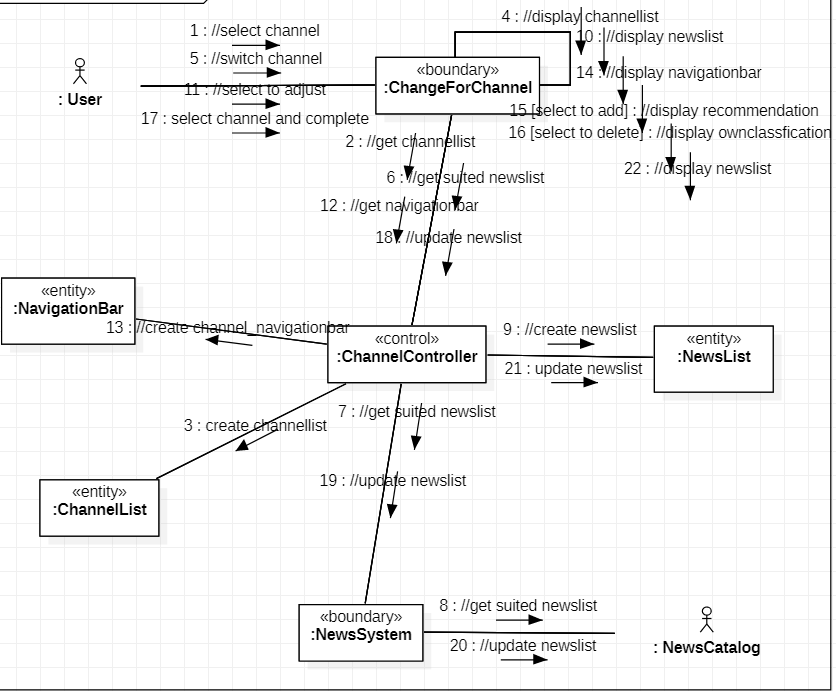
### 4.2.3 参与类图



### 4.2.4 序列图



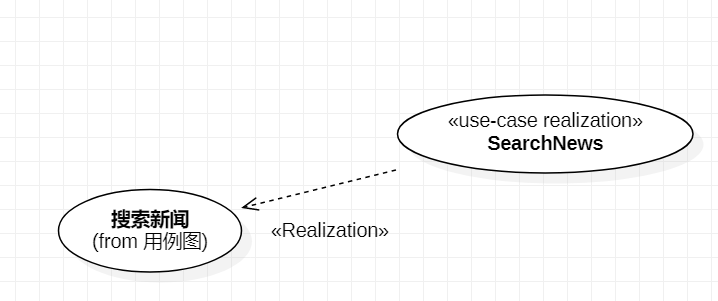
### 4.2.5 通信图



## 4.3搜索新闻

用户输入关键字、选择新闻发布的时间以及频道类别进行搜索新闻。搜索新闻边界类(SearchForNews)负责显示时间表、频道列表，接受用户的输入，并根据信息向搜索新闻控制类(SearchNewsController)发送获取新闻列表，搜索新闻控制类(SearchNewsController)向新闻系统边界类(NewsSystem)发送请求，获取新闻列表，最后由搜索新闻边界类(SearchNewsController)展示获取到的新闻列表。

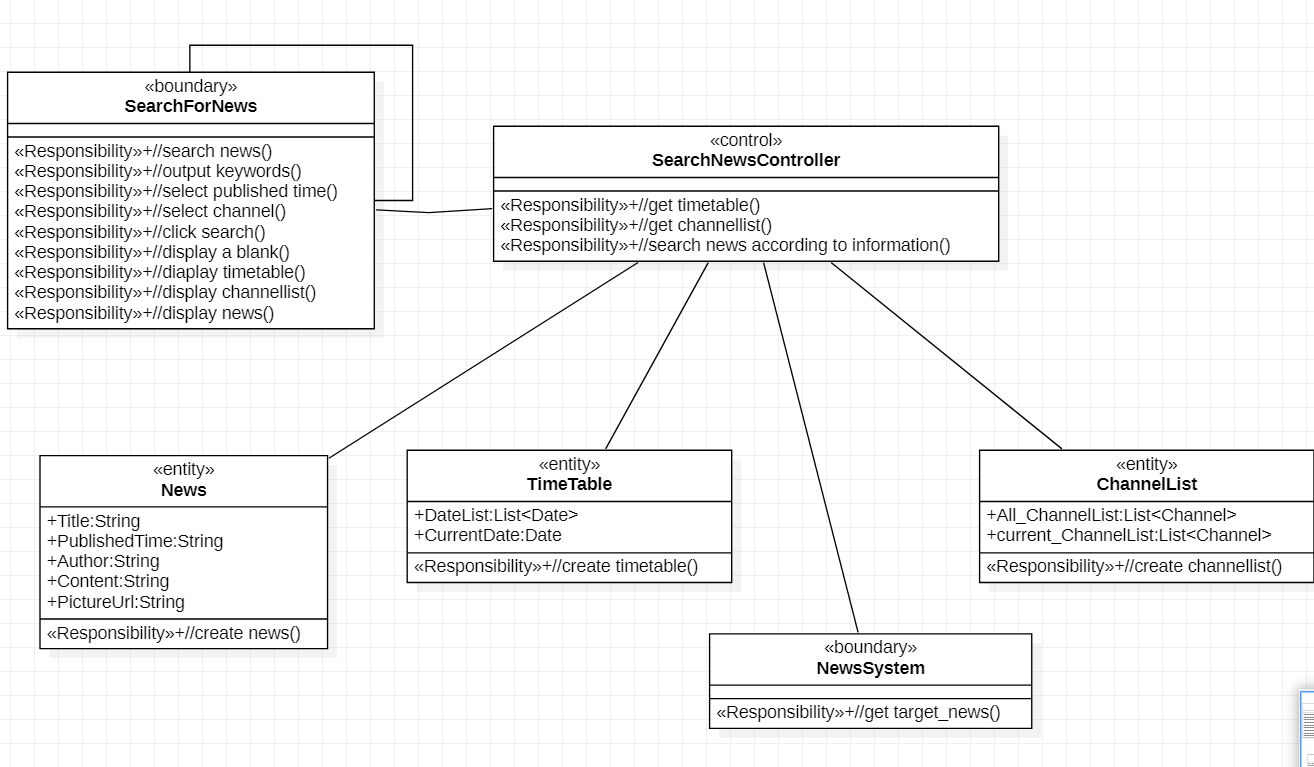
### 4.3.1用例实现



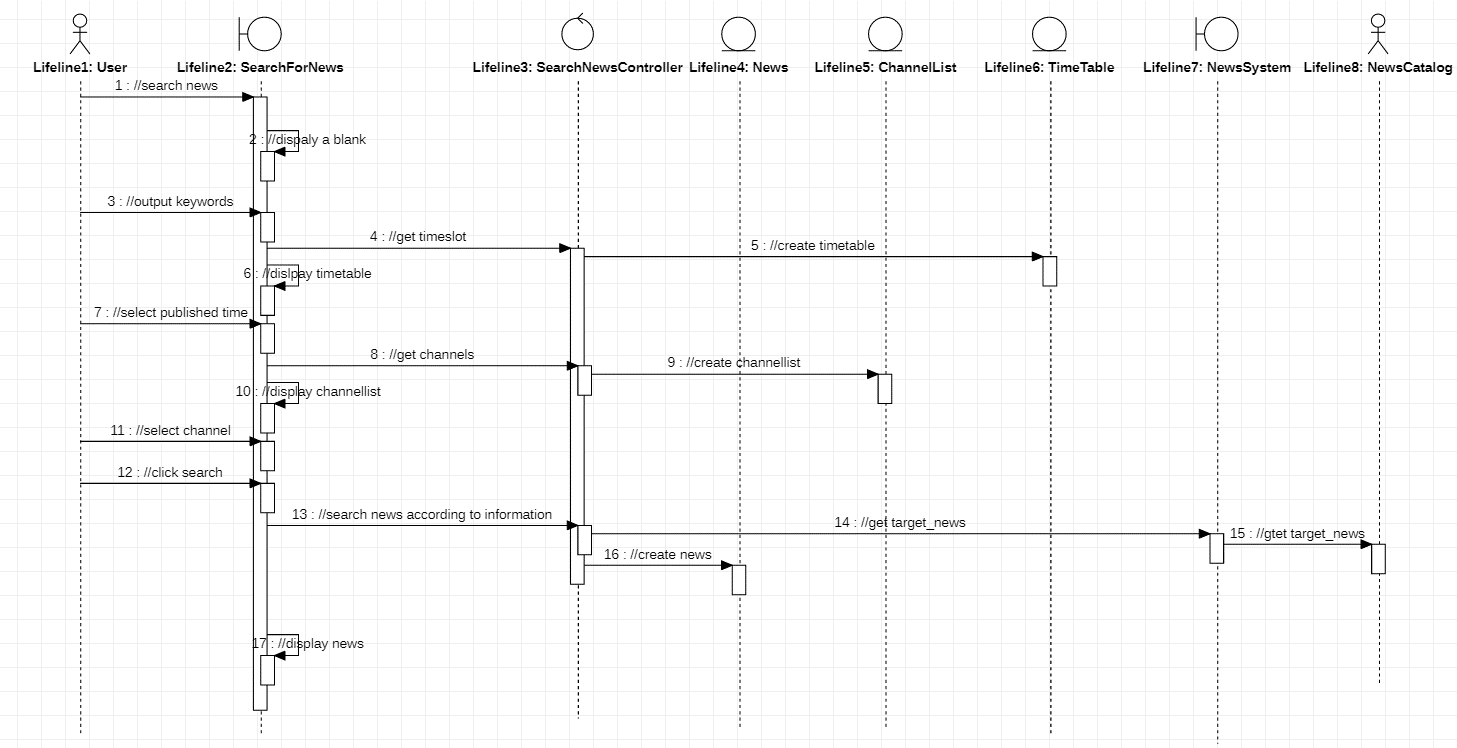
### 4.3.2类设计

User  
SearchForNews  
SearchNewsController  
TimeTable  
News  
ChannelList  
NewsSystem  
NewsCatalog

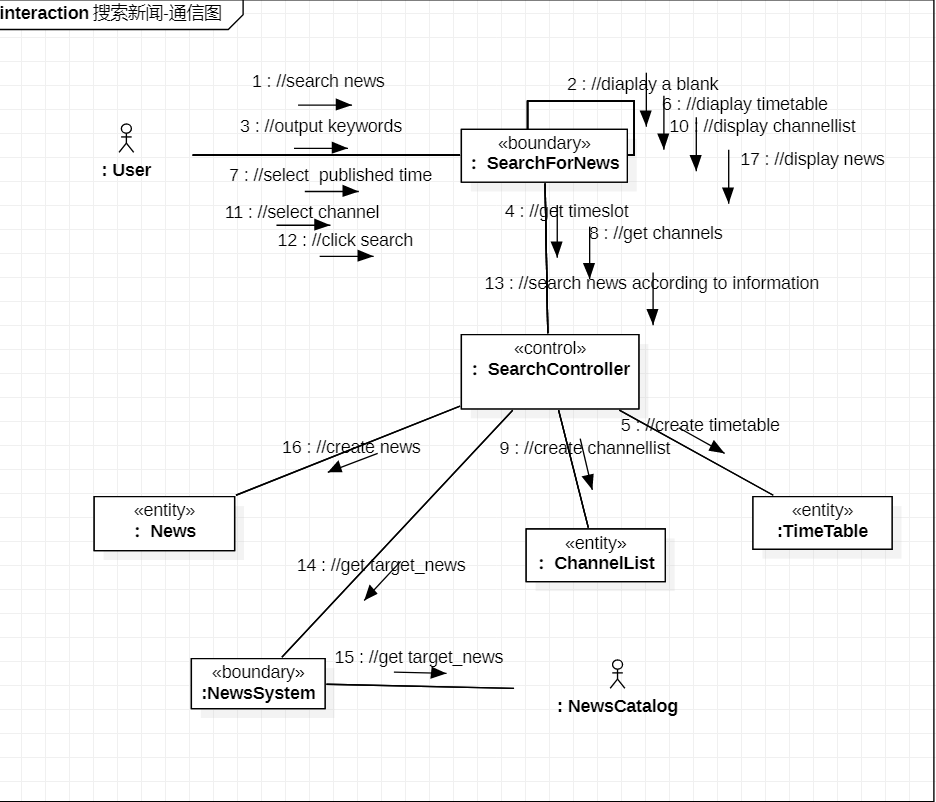
### 4.3.3 参与类图



### 4.3.3 序列图



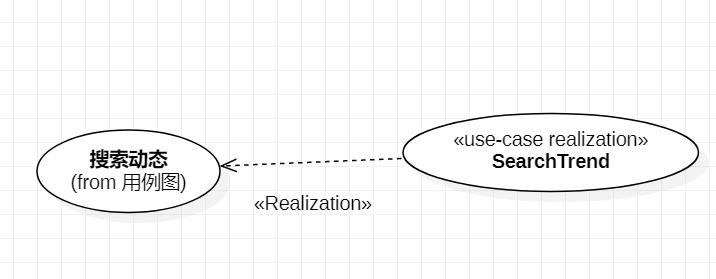
### 4.3.5 通信图



## 4.4搜索动态

用户根输入关键字，选择动态的发布时间，已登录的用户可选择关注的昵称进行搜索动态。搜索动态边界类(SearchForTrend)接受用户的输入，显示时间表在用户登录的情况下显示关注的昵称列表，并根据信息，向搜索动态控制类(SearchTrendController)发送获取动态列表。搜索动态控制类(SearchTrendController)向动态系统边界类(TrendSystem)发送获取新闻列表，由搜索动态边界类(SearchForTrend)显示的动态列表。

### 4.4.1用例实现

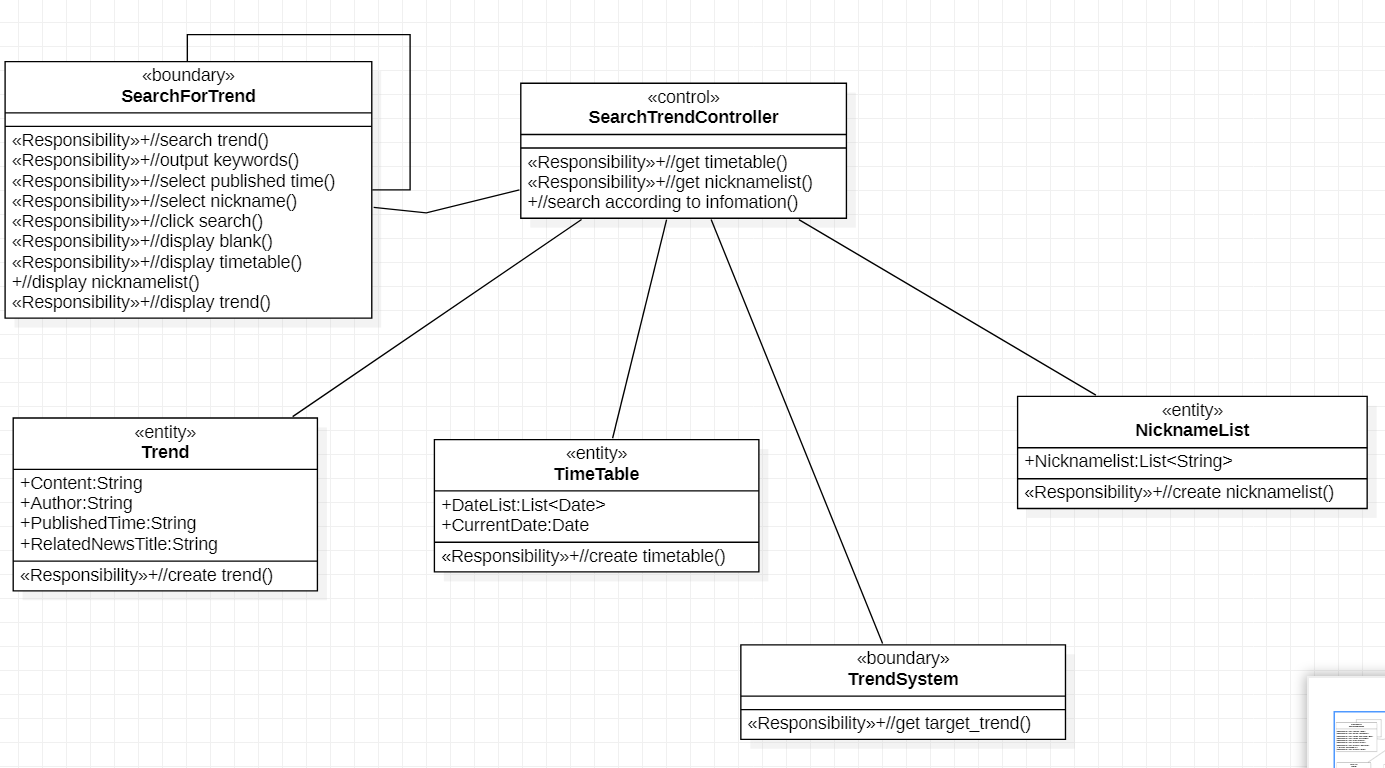


### 4.4.2用例实现

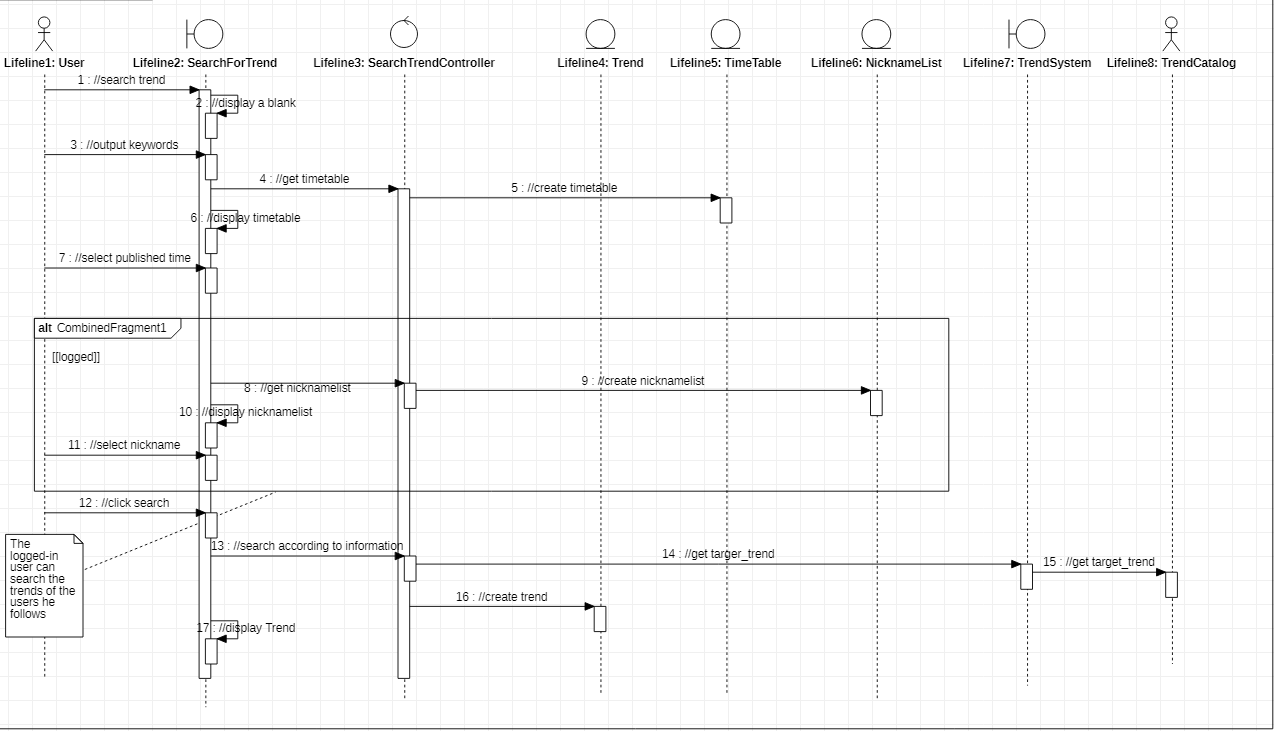
User

SearchForTrend  
SearchTrendController  
TimeTable  
NicknameList  
Trend  
TrendSystem  
TrendCatalog

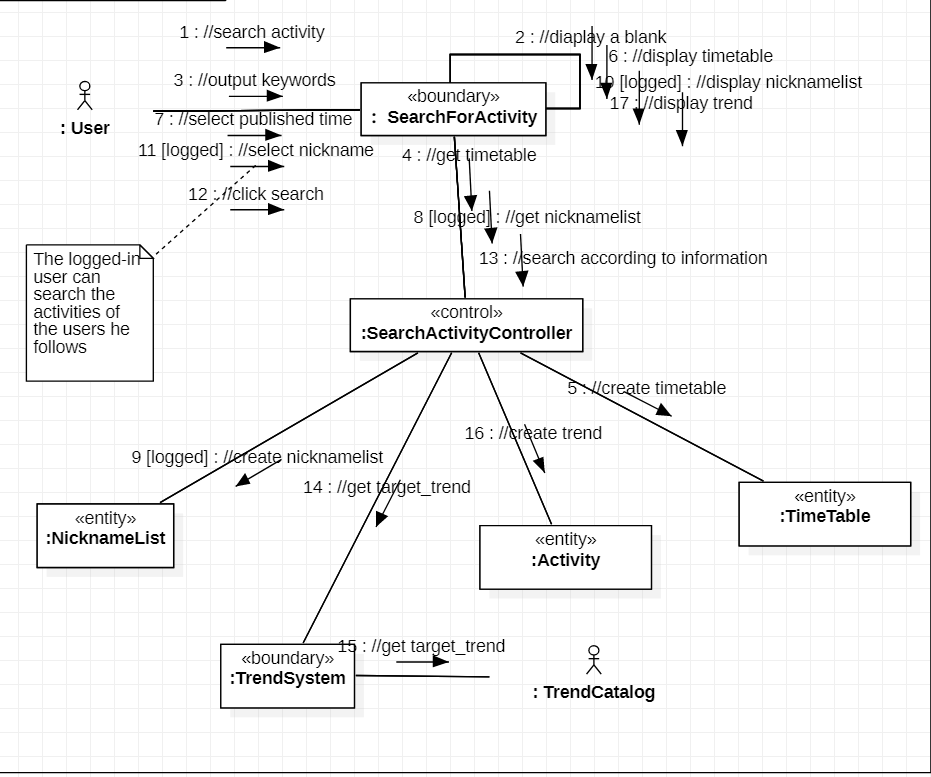
### 4.4.3 参与类图



### 4.4.4 序列图



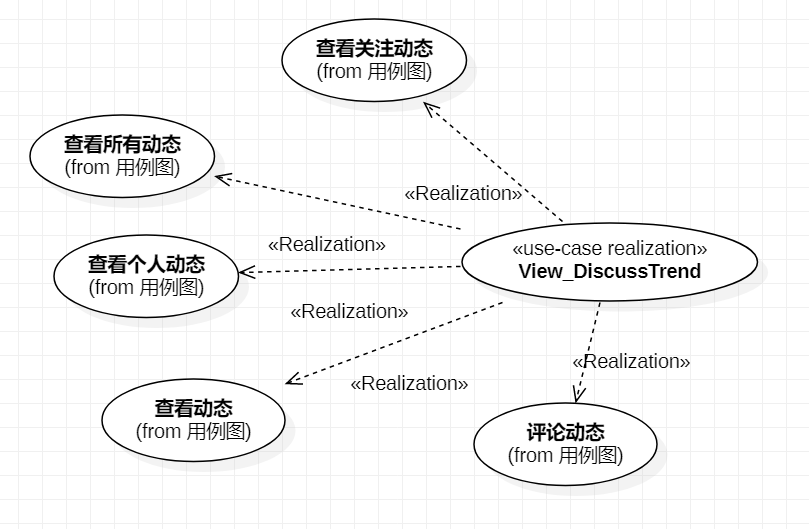
### 4.4.5 通信图



## 4.5查看、评论动态

用户可以常看所有动态，已登录的用户可以查看个人动态、关注的动态，进入动态详情页面查看详情，也可以评论动态。查看动态边界类(ViewForTrend)接受用户的请求，当已登录用户输入评论内容时，接受评论内容，向查看动态控制类(ViewTrendController)发送相应的请求信息，并显示动态列表、动态详情，相应的提示信息。查看动态控制类(ViewTrendController)向动态系统边界类(TrendSystem)发送获取动态列表、动态详情，当已登录用户键入评论内容时，当前动态里面插入新的评论。

### 4.5.1用例实现



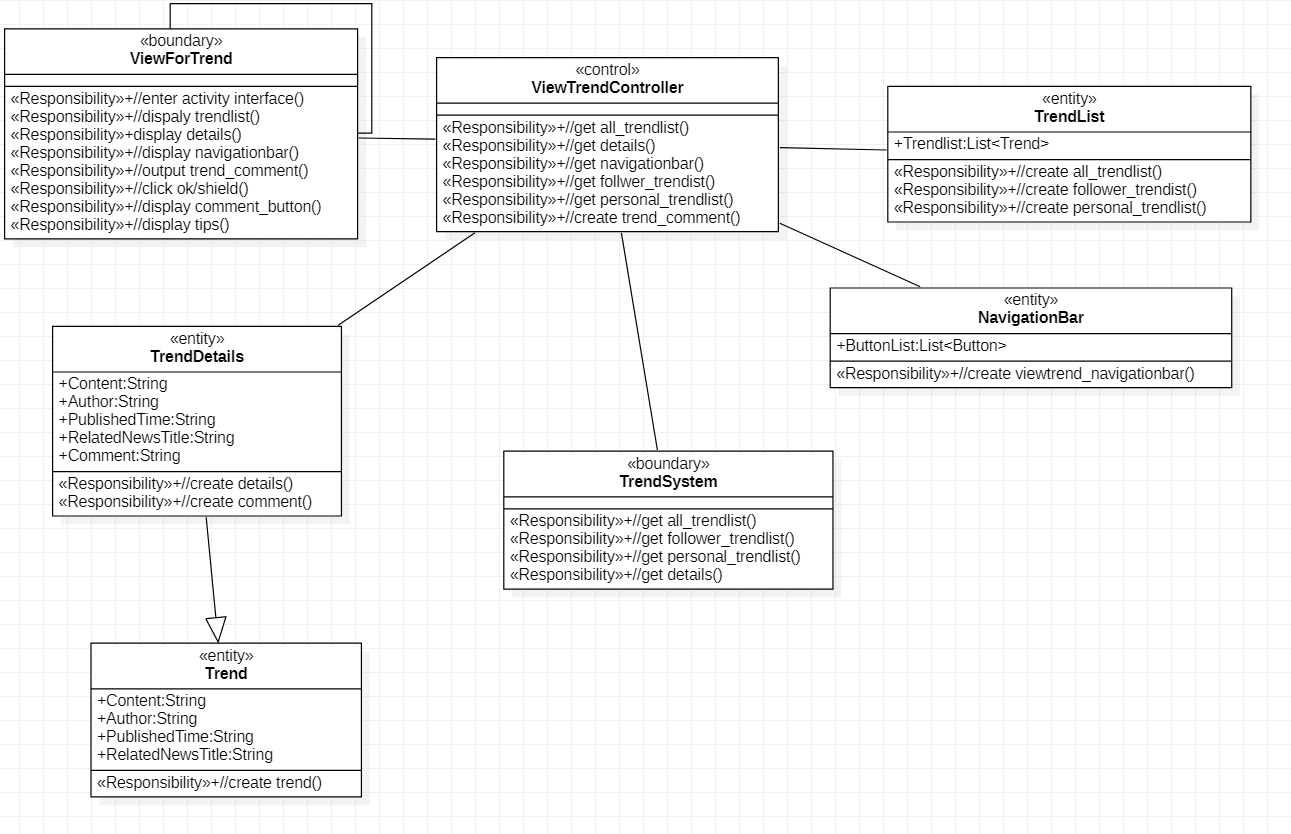
### 4.5.2类设计

User

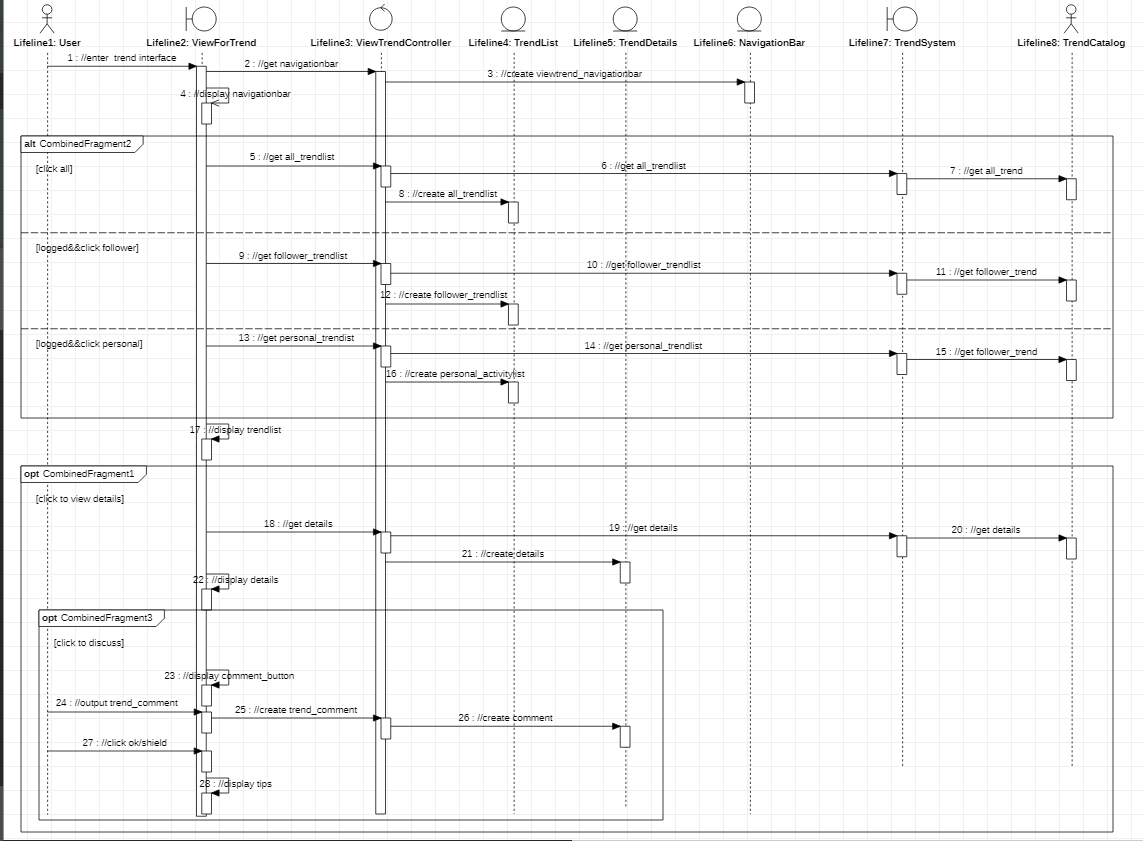
ViewForTrend  
ViewTrendController  
TrendList  
TrendDetails

NavigationBar  
TrendSystem  
TrendCatalog

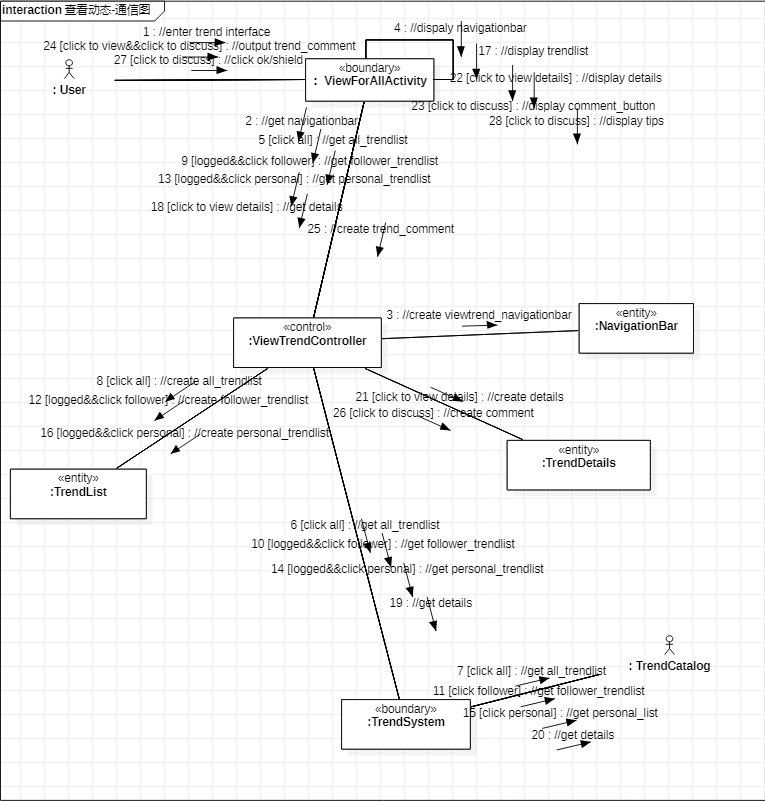
### 4.5.3 参与类图



### 4.5.4 序列图



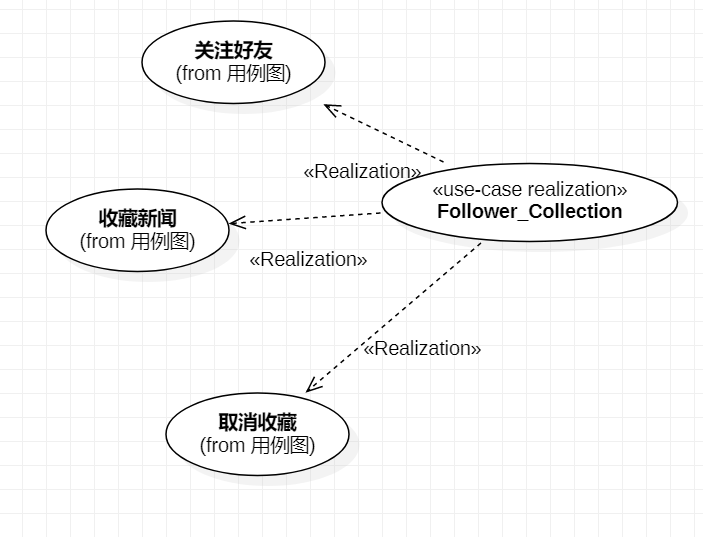
### 4.5.5 通信图



## 4.6关注好友、收藏新闻、取消收藏

已登录的用户进入新闻详情页面，可以收藏新闻，也可以取消收藏已收藏的新闻，同时在评论区可以关注他人；进入动态详情页面也可以关注他人。关注收藏边界类(EnterForFollower\_Collecion)接受已登录用户的请求，同时向关注收藏控制类(Follower\_Collection)发送请求。关注收藏控制类(Follower\_Collection)向用户信息系统边界类(UserInformationSystem)发送插入或者更新已登录用户的收藏列表或者关注列表。关注收藏边界类(EnterForFollower\_Collecion)给出相应的提示。

### 4.6.1用例实现



### 4.6.2类设计

Customer

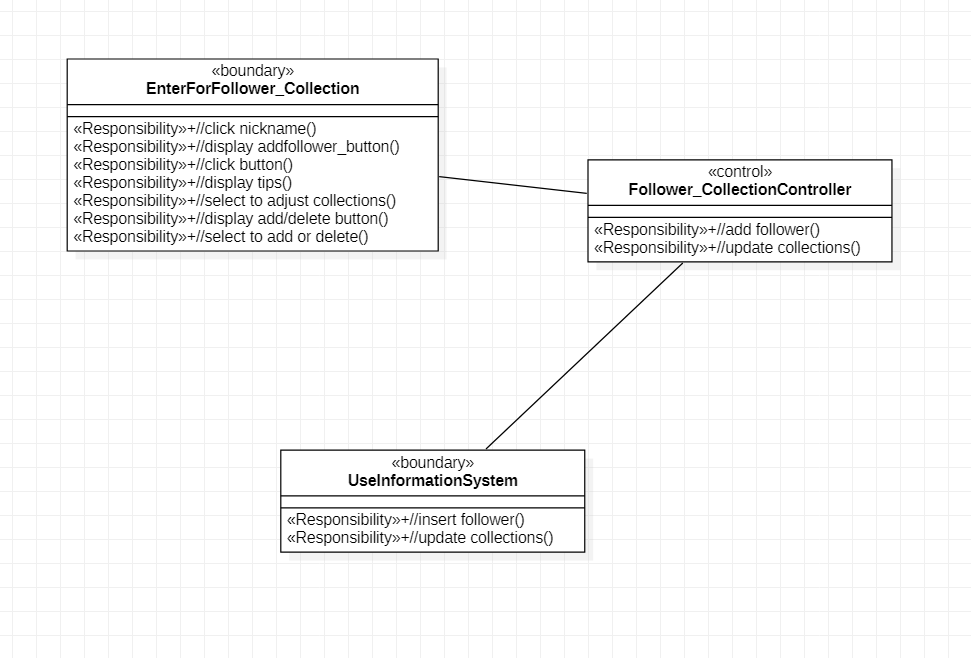
EnterForFollower\_Collection

Follower\_CollectionController

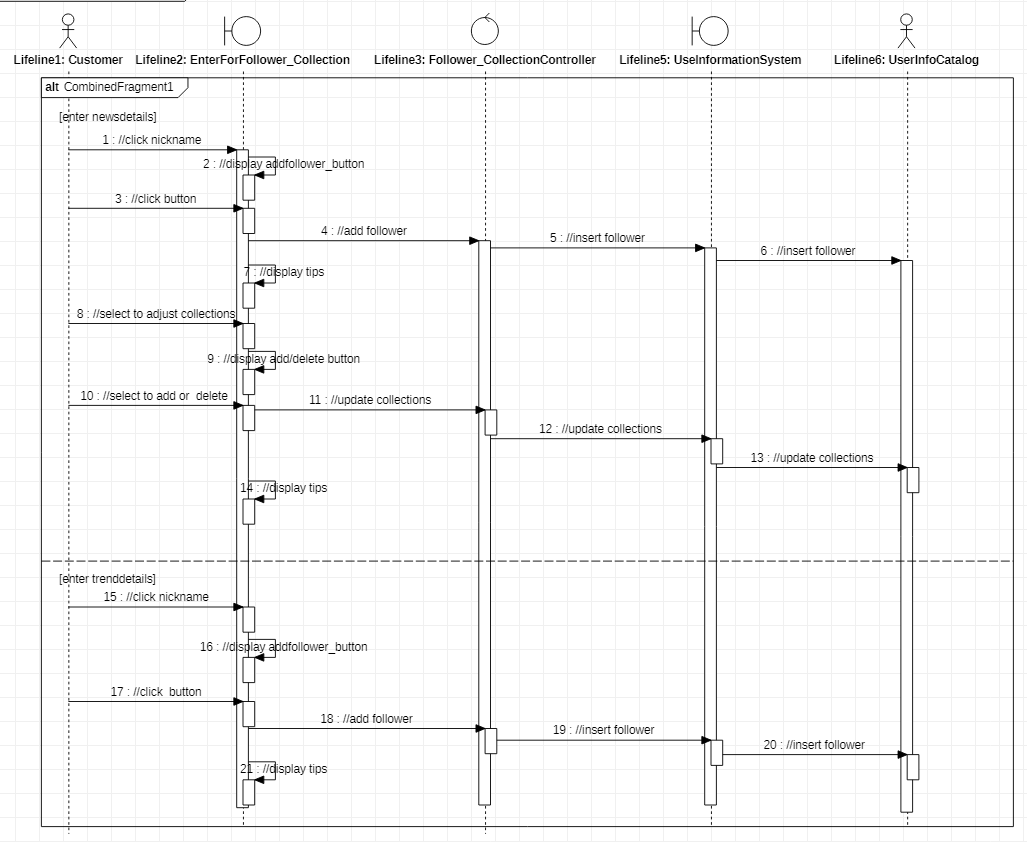
UseInformationSystem

UseInfoCatalog

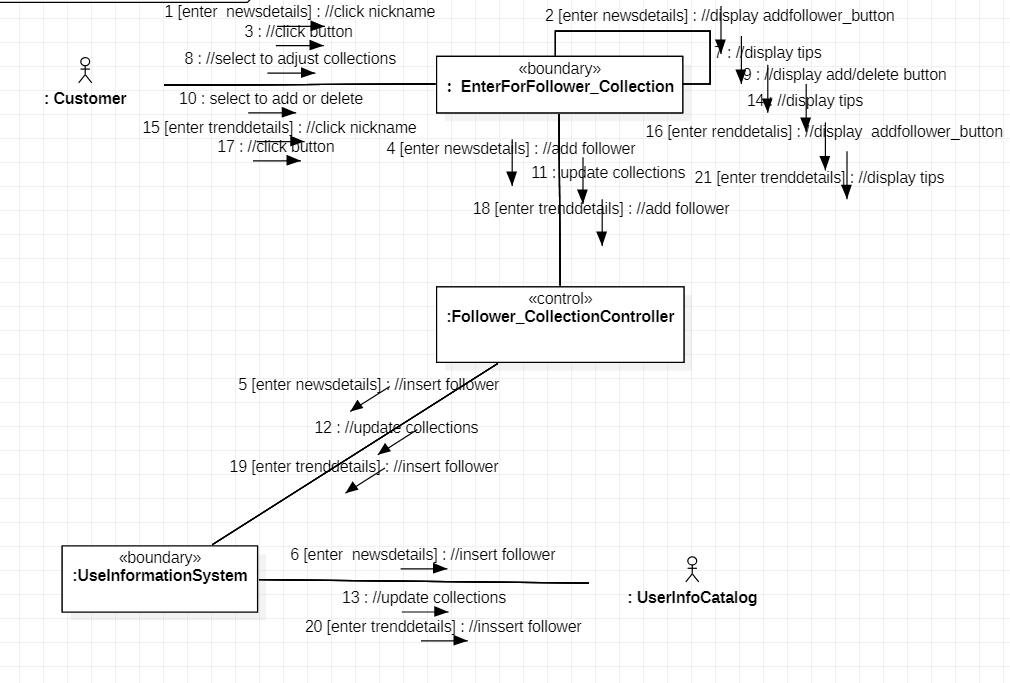
### 4.6.3 参与类图



### 4.6.4 序列图



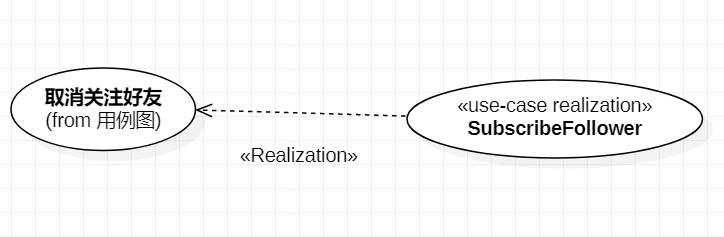
### 4.6.5 通信图



## 4.7取消关注

已登录的用户进入取消关注界面，可以取消关注已关注的人。取消关注边界类（EnterForUnsubscribe）接受已登录用户的请求，同时向取消关注控制类（UnsubscribeController）发送请求。取消关注控制类(UnsubscribeController)向用户信息系统边界类(UserInformationSystem)发送获取已登录用户的关注列表请求，系统从UserInfoCatalog中获取到实体类（Nicknamelist)并作为参数返回到取消关注边界类(EnterForUnsubscribe)。取消关注边界类(EnterForUnsubscribe)给出相应的提示，此时用户向取消关注边界类（EnterForUnsubscribe）发送取消关注某一或多好友的请求，这个请求被逐步传递到实体类（Nicknamelist)并对其进行修改，然后其再作为参数返回到取消关注边界类(EnterForUnsubscribe)，此边界类在界面上做出相应的更新。

### 4.7.1用例实现



### 4.7.2类设计

Customer

EnterForUnsubscribe

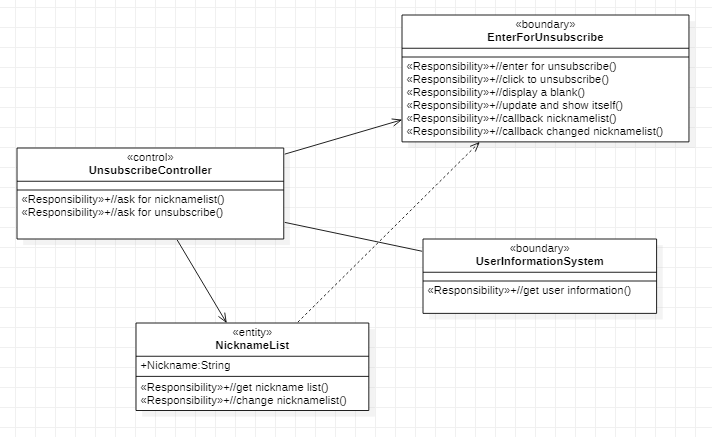
UnsubscribeController

UserInformationSystem

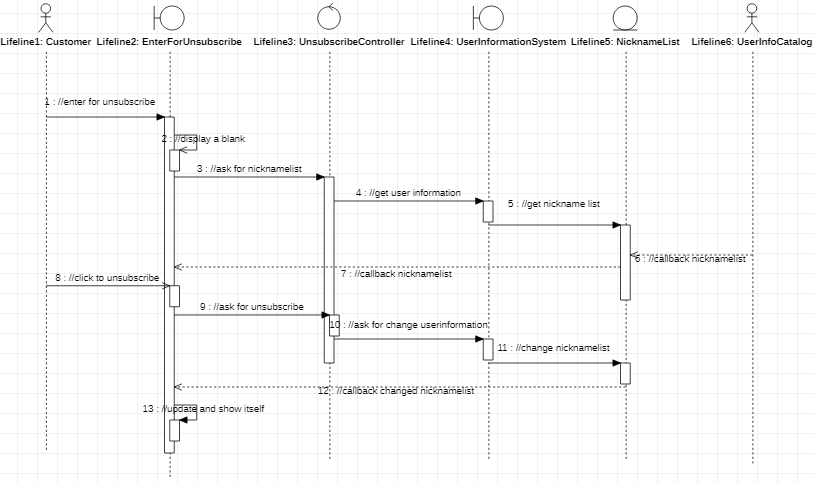
NicknameList

UserInfoCatalog

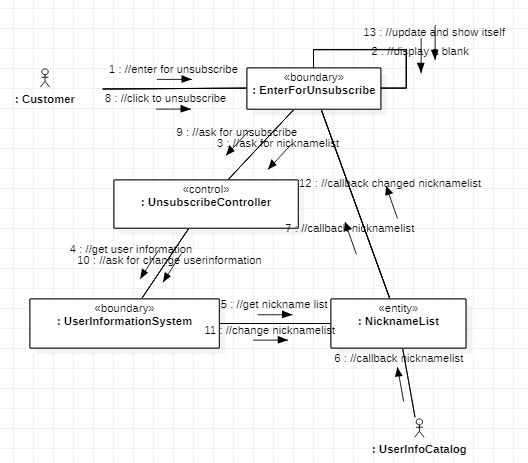
### 4.7.3 参与类图



### 4.7.4 序列图



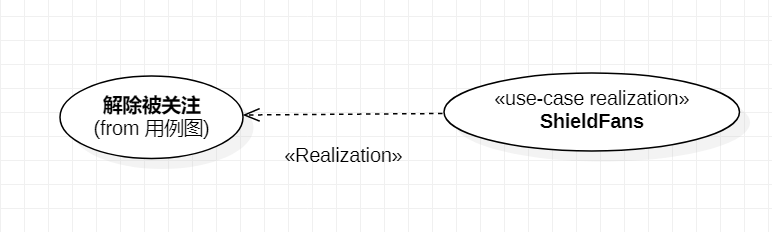
### 4.7.5 通信图



## 4.8解除被关注

已登录的用户进入解除被关注界面，可以对特定对象解除其关注自己的状态。解除被关注边界类（EnterForShield）接受已登录用户的请求，同时向解除被关注控制类（ShieldController）发送请求。解除被关注控制类(ShieldController)向用户信息系统边界类(UserInformationSystem)发送获取已登录用户的关注列表请求，系统从UserInfoCatalog中获取到实体类（Nicknamelist)并作为参数返回到解除被关注边界类(EnterForShield)。解除被关注边界类(EnterForShield)给出相应的提示，此时用户向解除被关注边界类（EnterForShield）发送解除被关注某一或多好友的请求，这个请求被逐步传递到实体类（Nicknamelist)并对其进行修改，然后其再作为参数返回到解除被关注边界类(EnterForShield)，此边界类在界面上做出相应的更新。

### 4.8.1用例实现



### 4.8.2类设计

Customer

EnterForShield

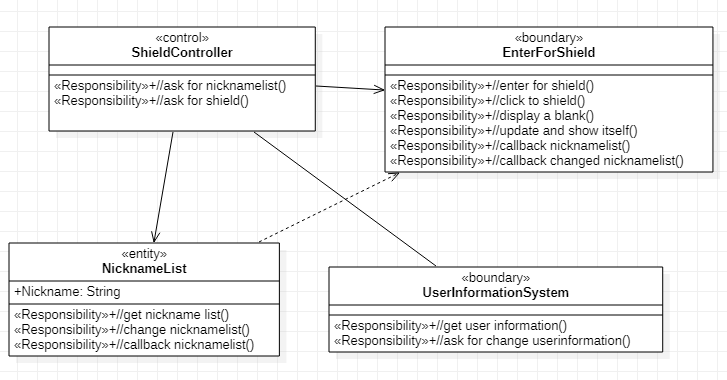
ShieldController

UserInformationSystem

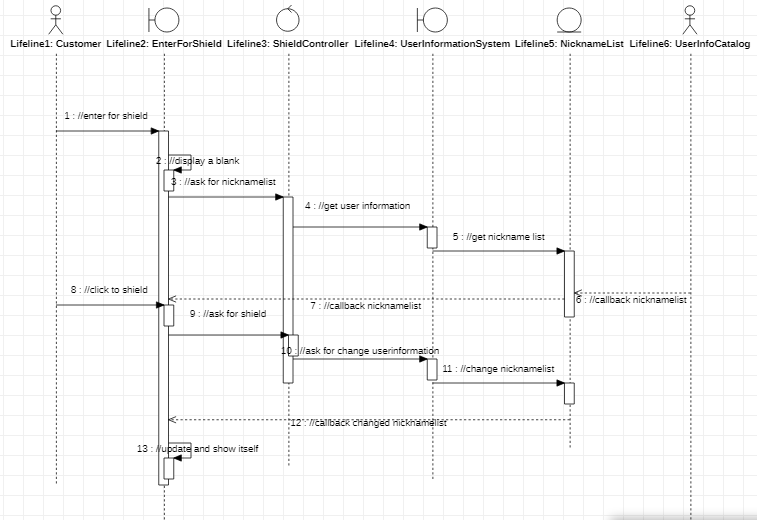
NicknameList

UserInfoCatalog

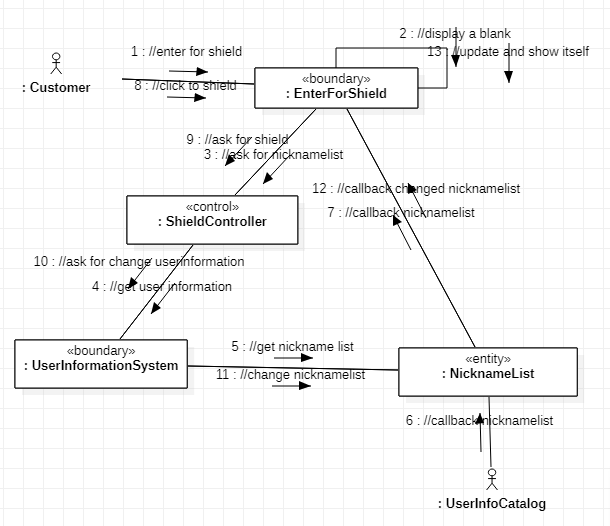
### 4.8.3 参与类图



### 4.8.4 序列图

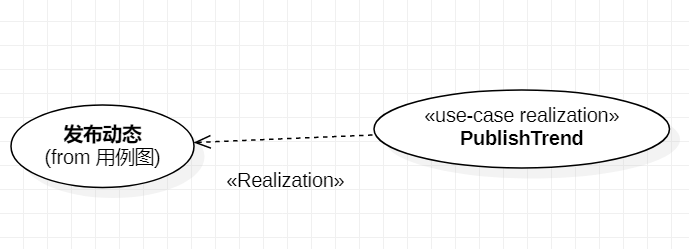


### 4.8.5 通信图



## 4.9发布动态

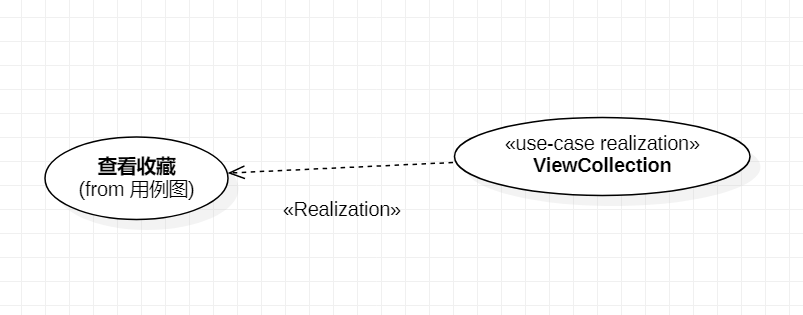
### 4.9.1用例实现



### 4.9.2类设计

## 4.10查看收藏

### 4.10.1用例实现

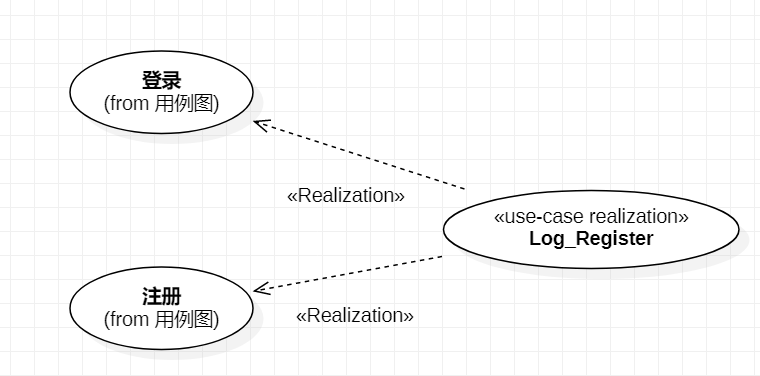


### 4.10.2类设计

## 4.11注册、登录

对于登录、注册系统，用户在使用系统前，输入用户名、密码，点击登录/注册按钮后，用户的登录/注册请求由登记用户边界类（EnterForLog\_Register），之后该边界类向登录/注册控制类（Log\_RegisterController） 发送登录/注册请求，由登录/注册控制类携带用户名（ID）和密码（Password）作为参数请求用户身份系统边界类（UserIdentitySystem），再由该边界类向用户身份数据库请求查询，将查询的结果经由用户身份系统边界类（UserIdentitySystem）、登录/注册控制类（Log\_RegisterController）回传登记用户身份类（EnterForLog\_Register），由该边界类将登陆/注册结果（登陆/注册成功或失败）反馈给用户。

### 4.11.1用例实现



### 4.11.2类设计

User

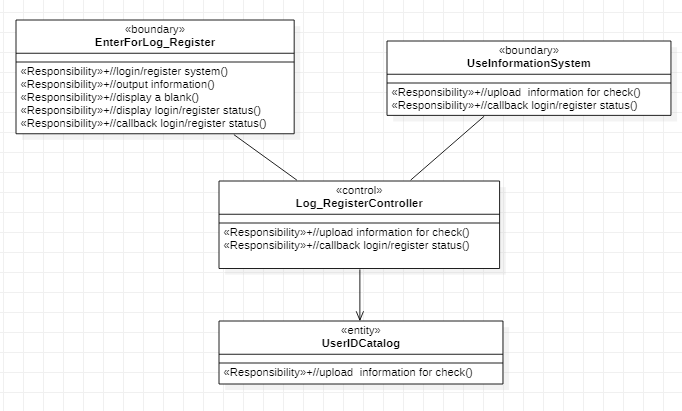
EnterForLog\_Register

Log\_RegisterController

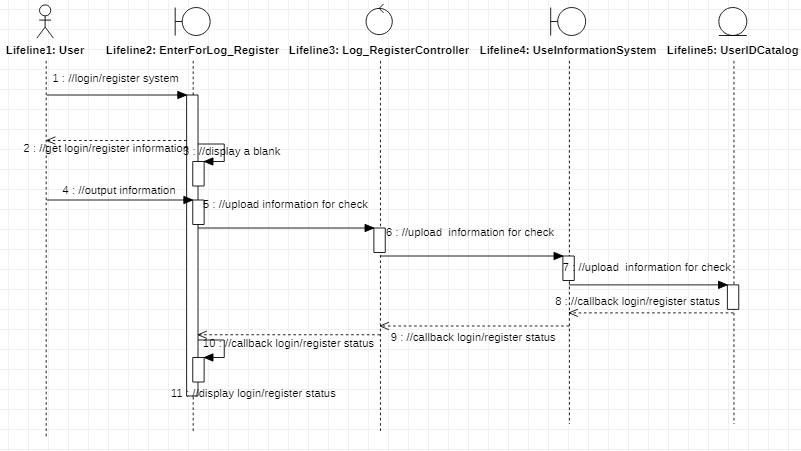
UserIdenditySystem

UserIDCatalog

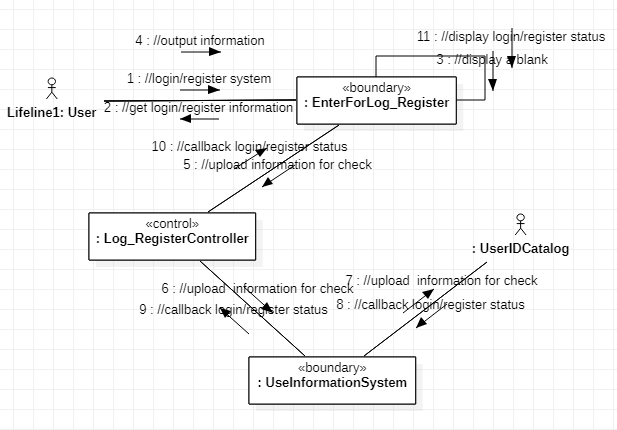
### 4.11.3参与类图



### 4.11.4序列图

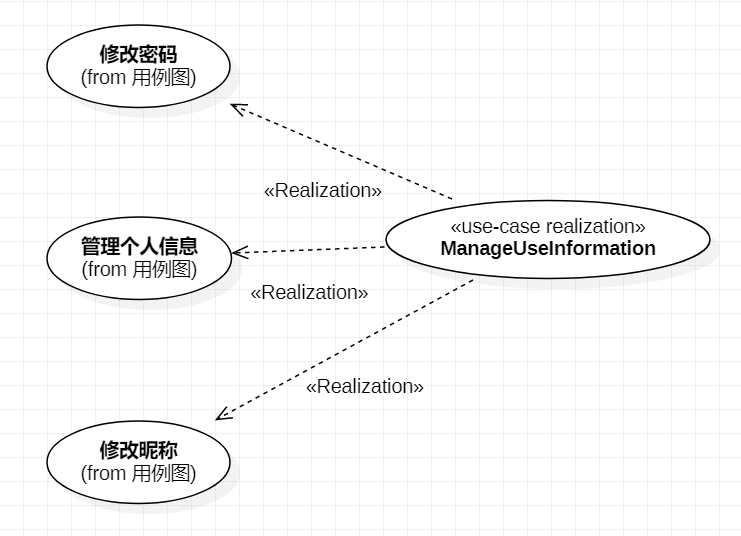


### 4.11.5通信图



## 4.12管理个人信息

### 4.12.1用例实现



### 4.12.2类设计

# 5. 参考

【1】Li H , Zhao A , Zhang D , et al. Research on building software usage model based on UML model[J]. International journal of systems assurance engineering and management, 2018, 9(3):675-683.

该篇论文提出了一种新的基于用户的软件使用模型构建方法。该方法根据UML活动图和UML用例图构建使用模型，该方法构建的模型是基于用户的：该模型描述了用户与系统之间的具体交互过程。对于本项目而言，新闻APP的分析模型是基于用户的软件使用，根据这个论文提出的基于用户的软件模型构建方法，对自身UML图的建立能更好的描述用户与系统的交互过程，实现了对UML画图的改善。

【2】张希. 以UML用例图为基础的系统需求分析[J]. 现代计算机:上下旬, 2002.

该文主要探讨以UML用例图为中心及出发点,类图、交互图为实现手段,捕获高层次的系统功能需求的策略。对于本项目而言，我们的系统功能需求主要以安全和实时两个大方向进行实现。在以使用UML用例图来更好的分析设计系统模型时，根据此篇论文对使用类图、交互图，来捕获高层次功能需求的策略，来对自身UML用例图进行改善。

【3】林子禹, 邓万涛, 彭德纯, et al. 基于面向对象的软件需求分析规范及实施方法研究[J]. 小型微型计算机系统, 1998(07):60-64.  
 该篇文采用面向对象的技术，对软件需求分析的任务、实施步骤和方法进行了研究，并给出了相应的规范说明。对于本项目而言，我们对新闻APP的需求规约完善过程中，利用了此篇论文中的面向对象思想。主要根据用户的使用功能需求进行分析，并基于文章给出的需求分析方法，完成了共27个需求规约的创建，并且根据文章对需求分析的规范化描述进一步完善了用户需求规约。

[4] Roberta Crumrine, Massachusetts Institute of Technology. Curricular Information System Software Requirements Specification Document. http://web.mit.edu/ssit/cis/CISRequirements.html, [Accessed March 25, 2017].

该文以麻省理工学院举办新的课程信息系统（CIS）为例，提供了一个软件需求说明文档的例子。该文档遵循 IEEE 推荐的良好软件需求规范文档的规范。 对于本项目中需求规约的文档撰写，基于此文章中描述的IEEE推荐的描述需求的文档规范，再根据自己的需求，将文档的用例需求进行规范化描述。

【5】Lodderstedt T , Basin D , Jürgen Doser. SecureUML: A UML-Based Modeling Language for Model-Driven Security[C]// International Conference on the Unified Modeling Language. Springer-Verlag, 2002.

这篇文章展示如何在应用程序的总体设计中使用UMLcan来指定与访问控制相关的信息，以及如何使用这些信息自动生成完整的访问控制基础结构。方法可以用于提高安全分布式系统开发过程中的生产力和由此产生的系统的质量。对于本项目而言，新闻APP的安全性实现部分的实现需要在设计过程中考虑到许多方面，利用本文提到的提高安全分布式系统开发过程，我们对本项目的安全性这一部分的实现进行了详细而系统的分析，对于信息的访问有了更多的控制基础结构。