

**需求分析规约**

软件工程课程设计



**指导教师：**

**杜庆峰**

**小组成员：**

**1752026 李航**

**1652977 王星洲**

**1650350 乔宇**

**1753499 潘小逸**

**2019-11-9**

目录

[修订历史 2](#_Toc25074696)

[1. 引言 3](#_Toc25074697)

[1.1 项目背景 3](#_Toc25074698)

[1.1.1 背景介绍 3](#_Toc25074699)

[1.1.2 必要性 4](#_Toc25074700)

[1.2 参考资料 4](#_Toc25074701)

[1.3 假定和约束 4](#_Toc25074702)

[1.4 用户特点 4](#_Toc25074703)

[2. 系统概述 4](#_Toc25074704)

[2.1 设备 4](#_Toc25074705)

[2.2 与第三方平台的接口 4](#_Toc25074706)

[2.3 普通接口 4](#_Toc25074707)

[2.4 需求分析 4](#_Toc25074708)

[2.4.2数据建模 4](#_Toc25074709)

[3. 功能需求 7](#_Toc25074710)

[4. 非功能需求 7](#_Toc25074711)

[4.1 性能要求 7](#_Toc25074712)

[3.1.1 用户体验 7](#_Toc25074713)

[3.1.2 时间特性要求 7](#_Toc25074714)

[3.1.3 输入输出要求 7](#_Toc25074715)

[4.2 数据管理能力要求 7](#_Toc25074716)

[4.2.1 常量约定 7](#_Toc25074717)

[4.2.2 数据存储要求 7](#_Toc25074718)

[4.2.3 负载 7](#_Toc25074719)

[4.3 安全及保密性要求 7](#_Toc25074720)

[4.3.1 软件使用数据 8](#_Toc25074721)

[4.3.2 用户账号数据 8](#_Toc25074722)

[4.4 灵活性要求 8](#_Toc25074723)

[4.5 其他要求 8](#_Toc25074724)

[4.5.1 可维护性 8](#_Toc25074725)

[4.5.2 灾难恢复 8](#_Toc25074726)

[4.5.3 法律限制 8](#_Toc25074727)

# 修订历史

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 编写日期 | SEPG | 版本 | 说明 | 作者 | 评审时间 | 评审人员 | 批准日期 | 签字人员 |
| 2019/11/12 |  | 1.0 | 初稿 | 李航 |  |  |  |  |
| 2019/11/19 |  | 1.2 | 数据建模 | 乔宇 |  |  |  |  |

# 1. 引言

## 1.1 项目背景

### 1.1.1 背景介绍

互联网技术的不断发展，使得国内拥有的移动终端的用户数量已经远远超过了个人电脑的拥有量，并且目前移动互联网用户数已经超过了宽带上网用户数，据调查，目前有超过60%的人通过手机接入互联网，因此移动互联网这个信息平台也成为各个媒体抢占数据流量的重要入口，而对于新闻媒体来说，这种便捷的信息传输平台为保证新闻时效性提供了更好的技术保障，所以随着移动互联平台成为信息传输和接收的主流渠道平台，各个新闻媒体也开始争抢移动互联网这块蛋糕。所以，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。

### 1.1.2 必要性

市场上的新闻APP已经五花八门，他们都能看实时新闻、热点消息等。但是有些APP不免具有一些缺点：如有大量的广告，还有一些以“多刷赚钱”为噱头吸引顾客长时间停留在他们的APP中。我们这款APP不仅可以查看实时新闻、评论新闻、查看视频等，还可以就看过的系列新闻、经历的事情或此刻的心情发表专属于自己的动态。在动态圈中，用户可以寻找志同道合的好友，共同探讨新闻。

## 1.2 参考资料

## 1.3 假定和约束

**发布时间：**

**androud版本要求：**

**开发条件：**成员自备电脑与移动应用，配置开发环境

**经费支持：**

**设计成本：**自行承担设计成本

**开发效率：**使用SCRUM敏捷开发模型

## 1.4 用户特点

1. 经常喜欢看新闻的用户

2. 喜欢分享见闻的用户

3. 乐于交友的用户

# 2. 系统概述

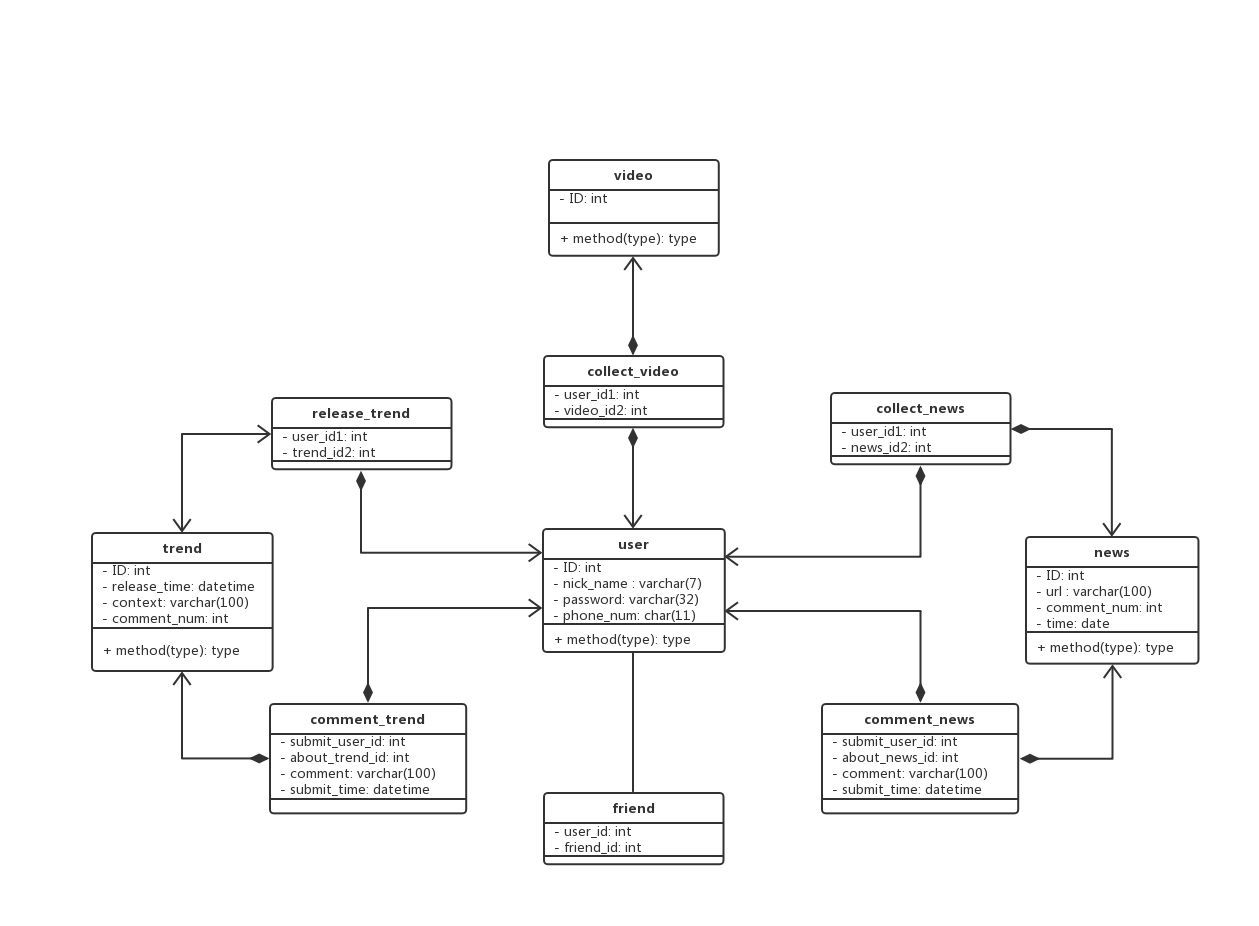
## 2.1 设备

## 2.2 与第三方平台的接口

## 2.3 普通接口

## 2.4 需求分析

### 2.4.2数据建模



#### 2.4.2.1 用户类（user）

编号：CD1

职责：存储和管理用户的基本信息。

属性：用户ID，昵称，密码，电话号码。

说明：用户类和好友类双向关联，一个用户可以有多个朋友，也可以是多用户的朋友。用户类的所有属性都是持久的，它存储了注册用户的基本信息。

#### 2.4.2.2 好友（friend）

编号：CD2

职责：存储用户之间的好友关系。

属性：用户ID，好友ID。

说明：好友类与用户类双向关联，一个用户有多个好友，也可以是多个用户的好友。在类的实例化中，查询时注意避免重复。

#### 2.4.2.3 新闻类（news）

编号：CD3

职责：存储新闻的网址以及其他基本信息。

属性：新闻ID，新闻URL，新闻评论数量，新闻发布时间。

说明：新闻类中所有属性都是持久化的，它负责存储新闻的基本信息。

#### 2.4.2.4 新闻评论类（comment\_news）

编号：CD4

职责：负责存储用户对于某一新闻的评论。

属性：提交评论的用户ID，相关新闻ID，评论内容，评论提交时间。

说明：新闻评论类和用户之间存在关联关系，一个用户可以提交多条评论。新闻评论类和新闻之间存在关联关系，一个新闻可以有多条评论。

#### 2.4.2.5 新闻收藏类（collect\_news）

编号：CD5

职责：负责存储用户对于某一新闻的收藏。

属性：用户ID，新闻ID。

说明：新闻收藏类和用户之间存在关联关系，一个用户可以收藏多条新闻。新闻收藏类和新闻之间存在关联关系，一个新闻可以有多条收藏信息。

#### 2.4.2.6 视频类（video）

编号：CD6

职责：负责存储视频的基本信息。

属性：视频ID。

说明：视频类中所有属性都是持久化的，它负责存储视频的基本信息。

#### 2.4.2.7 视频收藏类（collect\_video）

编号：CD7

职责：负责存储用户对某一视频的收藏。

属性：用户ID，视频ID。

说明：视频收藏类和用户之间存在关联关系，一个用户可以收藏多个视频。视频收藏类和视频之间存在关联关系，一个视频可以有多条收藏信息。

#### 2.4.2.8 动态类（trend）

编号：CD8

职责：负责存储用户发布的动态的信息。

属性：动态ID，发布时间，动态内容，评论数量。

说明：动态类中所有属性都是持久化的，它负责存储动态的基本信息。

#### 2.4.2.9 动态评论类（comment\_trend）

编号：CD9

职责：负责存储用户对某一动态的评论。

属性：提交评论的用户ID，相关动态ID，评论内容，评论提交时间。

说明：动态评论类和用户之间存在关联关系，一个用户可以提交多条评论。动态评论类和动态之间存在关联关系，一个动态可以有多条评论。

#### 2.4.2.10 动态发布类（release\_trend）

编号：CD10

职责：负责存储用户发布某一动态的信息。

属性：发布动态的用户ID，相关动态ID。

说明：动态发布类和用户之间存在关联关系，一个用户可以发布多条动态。动态发布类和动态之间存在单项关联关系，一条动态只有一条发布信息。

# 3. 功能需求

# 4. 非功能需求

## 4.1 性能要求

### 3.1.1 用户体验

本应用是新闻类APP，希望能够获取到最新的新闻，并可以读取新闻的具体内容。对实时性要求比较高，同时在获取新闻时要尽可能快。同时，此APP还要在朋友圈方面做到实时刷新，以便用户在第一时刻了解好友动态。

### 3.1.2 时间特性要求

1. 相应时间：

在网络状况良好的情况下，对于页面的跳转，响应时间应在1~2s，对于视频的加载，响应时间应在3s左右，对于朋友圈的动态加载，响应时间应小于1s。因APP大部分内容涉及与第三方平台的交互，以上数据均不准确，只有一个大概的范围，取决于第三方平台的数据传输速度。

2. 更新处理时间

3. 数据转换时间

4. 界面更新时间

### 3.1.3 输入输出要求

1. 新闻详情

2. 视频详情

3. 朋友圈详情

4. 个人详情

## 4.2 数据管理能力要求

### 4.2.1 常量约定

### 4.2.2 数据存储要求

### 4.2.3 负载

## 4.3 安全及保密性要求

### 4.3.1 软件使用数据

### 4.3.2 用户账号数据

## 4.4 灵活性要求

1.

## 4.5 其他要求

### 4.5.1 可维护性

可维护性是指在不影响系统其他部分的情况下修改现有系统功能中问题或缺陷的能力。开发人员创建和设计系统架构时，为了提高系统的可维护性，必须考虑以下几个方面的要素：低 耦合、高内聚合系统文档记录。本系统将采用严格的软件工程的规范进行开发，并采用良好的设计模式保证系统各模块之间的低耦合及模块之间的高内聚。

本系统的所有代码将会被详细注释，对于系统所有代码，我们会生成详尽的技术文档。对于系统开发过程可能出现的报错，我们将以文档的方式详细罗列报错码及对应的报错信息。

### 4.5.2 灾难恢复

### 4.5.3 法律限制