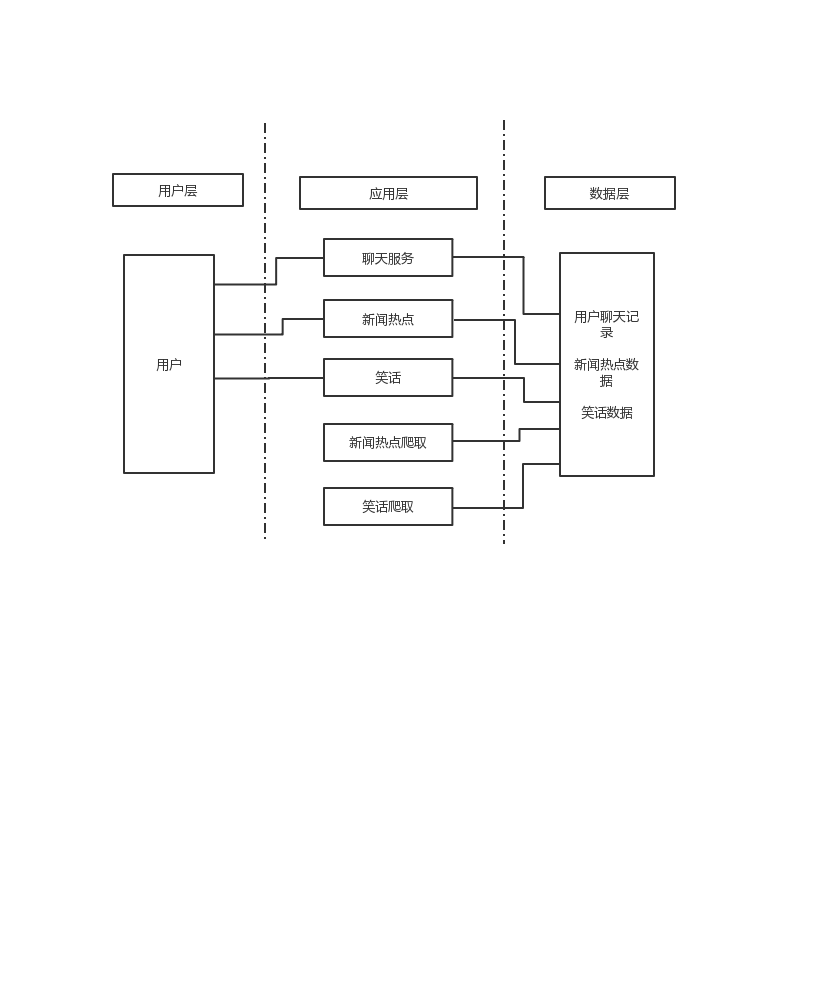
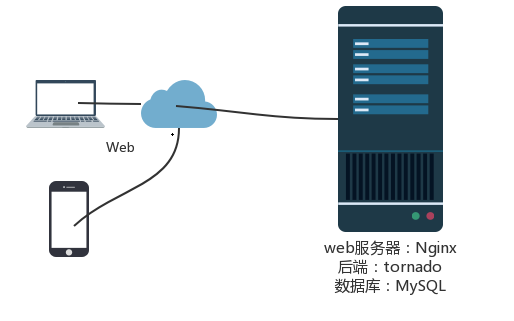
**3. 设计概述**

**3.1. 系统结构设计**

3.1.1.系统逻辑架构：



3.2.2.系统物理架构



**3.2. 系统接口设计**

3.2.1.用户界面：

3.2.2.内部接口：

**1. 聊天部分**

**1.1 返回用户ID**

url

119.29.161.184:8000?rand=xxx&sex=xxx&nameIndex=xxx

sex : 0为male 1为female

response

{

"status":1,

"id":"xsijfa-r21ja2j-asoj2aa-adh2saj"

}

**1.2 聊天**

url

119.29.161.184:8000/chat

params

id : string(用接口1.1返回的ID)

request

{

"type":"xxxx",

"text":"xxxxxxx",

"id":"xxxxxx-xxxx-xxxxx"

}

response

{

"res":{

"sender":"xxx",

"type":"xxx",

"content":"",

"time":"2016-10-16 13:45:20"

},

"status":1

}

该接口使用websocket协议

**1.3 返回昵称**

url

119.29.161.184:8000/name?sex=xxx

response

{

res: {

7: "郭芙",

11: "李莫愁",

16: "李沅芷",

24: "方怡",

28: "何红药",

31: "安小惠",

37: "杨不悔",

41: "袁紫衣"

},

status: 1

}

sex : 0为male 1为female

**2. 热点新闻部分**

**1.1 返回热点新闻**

url

119.29.161.184:8000?page=xxx

(page表示第几页，从 1 开始计数，每页返回十条热点新闻,按照 pv 进行降序排列)

response

success:

{

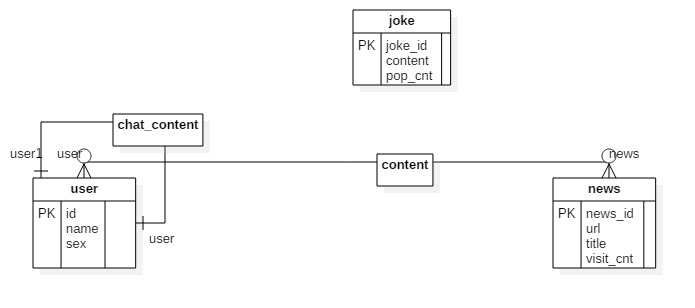
"status": 1,

"data": [{"id":xxx,"url":xxx,"title":xxx,"visit\_cnt":xxx},...]

fail:

{"status": 0, "data": "page exceed limits"}

3.2.3 系统数据模型



**3.3. 约束和假定**

3.3.1 时间约束：

由于项目需要在课程完成前结束时间有限制，而且由于项目组成员中有两名同学对python比较熟悉，因此我们采用python后台框架tornado和爬虫框架scrapy，并且最终部署在linux上。

3.3.2 设备约束:

团队只有一台腾讯云服务器，因此我们需要把前端页面、后台服务器、mysql数据库以及爬虫均部署这台服务器上面。

**3.4. 非功能性设计**

3.4.1. 匹配：

在用户匹配的过程中，我们设置异性匹配的优先级较高，如果此时并没有异性的用户，则进行同性聊天匹配

3.4.2. 热点新闻：

由于我们主要面向大学城的学生，因此我们的热点新闻主要从大学城各高校相关微信公众号爬取。