**1. 概述**

**1.1. 系统简述**

　　我们的系统名为：城见，是一个匿名网页聊天工具。使用我们的软件，用户可以选择单独匹配陌生人进行匿名聊天，也可以选择对某一话题进行群聊。

**1.2. 软件设计目标**

　 软件设计目标能够为大学城学生提供一个匿名聊天环境，主要功能包括以下方面：

①. 匹配任意陌生人聊天

可以选择作为男生还是女生进行匿名匹配，我们为匹配到的异性提供私密的一对一聊天页面。

②. 向聊天对象推送笑话

系统提供笑话推送功能。聊天时，聊天一方可以向另一方推送笑话。

③. 发送表情

在匿名聊天和聊天室聊天的过程中可以发送表情。

④. 浏览近期发生的热点

在聊天框的左侧推送由python爬虫提供的当天新闻头条链接，为聊天提供话题。

⑤. 在某一话题进行群聊

为对不同话题感兴趣的同学提供了一个群聊聊天室的平台。

**1.3. 参考资料**

需求上，我们参考了国外著名的匿名聊天网站 Omegle 设计，在此基础上提出了我们自己的创新。技术上，我们主要参考 python 的 tornado 框架文档以及 前端Bootstrap 框架文档

**1.4. 修订版本记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 修改内容 | 备注 |
| 1.0 | 2016/10/4 | 初始版本 |  |
| 1.1 | 2016/10/7 | 增加了笑话推送和话题聊天室的内容 |  |
| 1.2 | 2016/10/13 | 增加了发送表情的相关内容 |  |

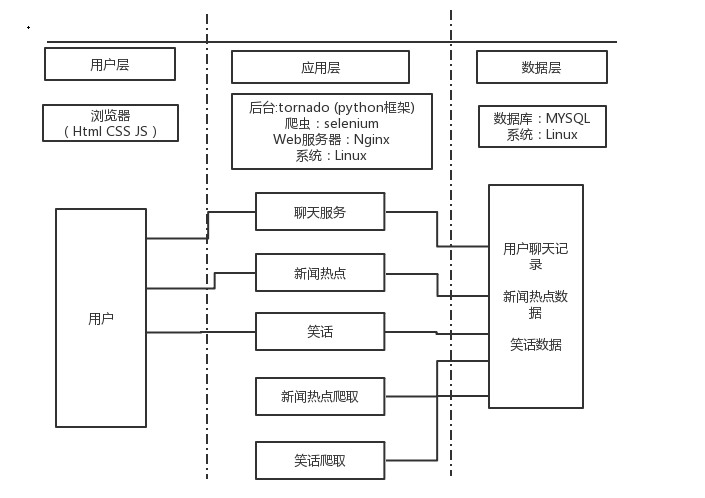
**2. 术语表**

|  |  |
| --- | --- |
| 术语 | 解释 |
| 笑话推送 | 系统提供笑话推送功能。聊天时，聊天一方可以向另一方推送笑话。 |
| 新闻热点滚动栏 | 在聊天框的左侧推送当天新闻头条链接，为聊天提供话题。 |
| 话题聊天室 | 为对不同话题感兴趣的同学提供了一个群聊聊天室的平台。 |
| 发送表情 | 在匿名聊天和聊天室聊天的过程中可以发送表情。 |
| 匿名匹配聊天 | 可以选择作为男生还是女生进行匿名匹配，我们为匹配到的异性提供私密的一对一聊天页面。 |
| 取名 | 我们为匿名用户提供我们准备好的昵称，用户可以选择喜欢的昵称开始聊天。 |

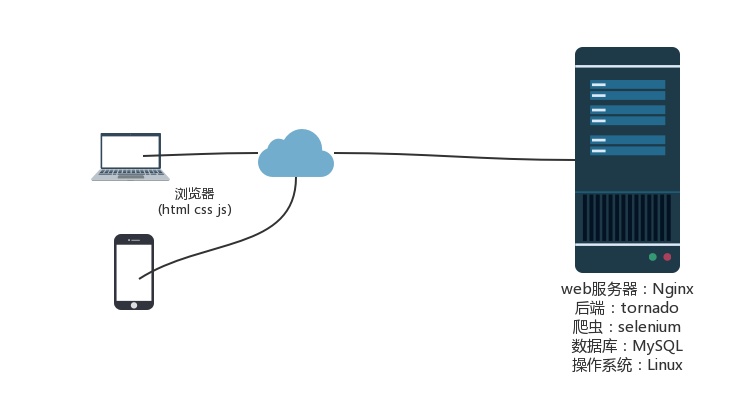
**3. 设计概述**

**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：

3.2.2.1 聊天部分

3.2.2.1.1 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  
success:

{

"status":1,

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

}

3.2.2.1.2 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*

3.2.2.1.3 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*

3.2.2.1.4 返回对方昵称

url

119.29.161.184:8000/othername?id=xxx

response

{

res:{'mine': "郭芙",'other':"对面那个"}

status: 1

}

3.2.2.1.5 群聊

url

119.29.161.184:8000/groupchat?topic=泡馆&rand=xxxx

params

rand : 随机数

request

{

"type":"xxxx",

"text":"xxxxxxx",

"id":"xxxxxx-xxxx-xxxxx"

}

response

websocket建立连接后：

{

"id":"xxxxxx-xxxx-xxxxx",

"status" : 1,

"name" : "就是个名字",

}

聊天进行时：

{

"res":{

"sender":"xxx",

"type":"xxx",

"content":"",

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

},

"status":1

}

*该接口使用websocket协议*

3.2.2.2. 热点新闻和笑话部分

3.2.2.2.1 返回热点新闻

url

119.29.161.184:8000/news?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"}

3.2.2.2.2 热点新闻访问量更新

url

119.29.161.184:8000/news (post)

数据格式：

{

"news\_id":"以整数表示",

"cnt":"以整数表示"

}

如

{

"news\_id":"2",

"cnt":"1"

}

response success:

{

"status":1

}

fail:

{

"status":0,

"data":"XXX" (给出失败原因)

}

3.2.2.2.3 返回笑话

url

119.29.161.184:8000/joke

(随机返回一个笑话)

response success:

{

"status":1,

"data":"笑话内容"

}

fail:

{

"status":0,

"data":"XXX"

}

3.2.2.2.4 返回群聊话题

url

119.29.161.184:8000/topic?type=[sport movie music library game travel]

(随机返回十条数据，其中 sport movie game 数据是新闻 | travel 数据来自蝉游记 | music library 数据来自豆瓣250排行榜)

response success:

{

"status":1,

"data":[

{'url':'...',

'title':'...'},

{'url':'...',

'title':'...'},

...

](长度为10)

}

fail:

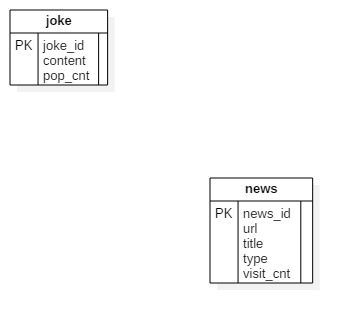
{

"status":0,

"data":"XXX"

}

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鲁棒性：

系统应该尽可能的提供可靠服务，控制服务器宕机次数

3.4.2易用性：

系统应该尽可能简单易用。主要要求包括下面：

1. 首页提供单人匹配功能，减少用户点击次数

2. 热点展示应该控制展现数量，减少信息拥塞

3. 单聊或者群聊要保证消息准确无误的传送，对于无法正确传送的消息，应该可以友好的提示消息。

3.4.3安全性：

系统应该保证用户消息不泄露，如果用户选择不下载消息记录，那么应该从系统中及时删除

3.4.4可扩展性：

系统应该合理设计接口，使得能很好的修改或者添加功能