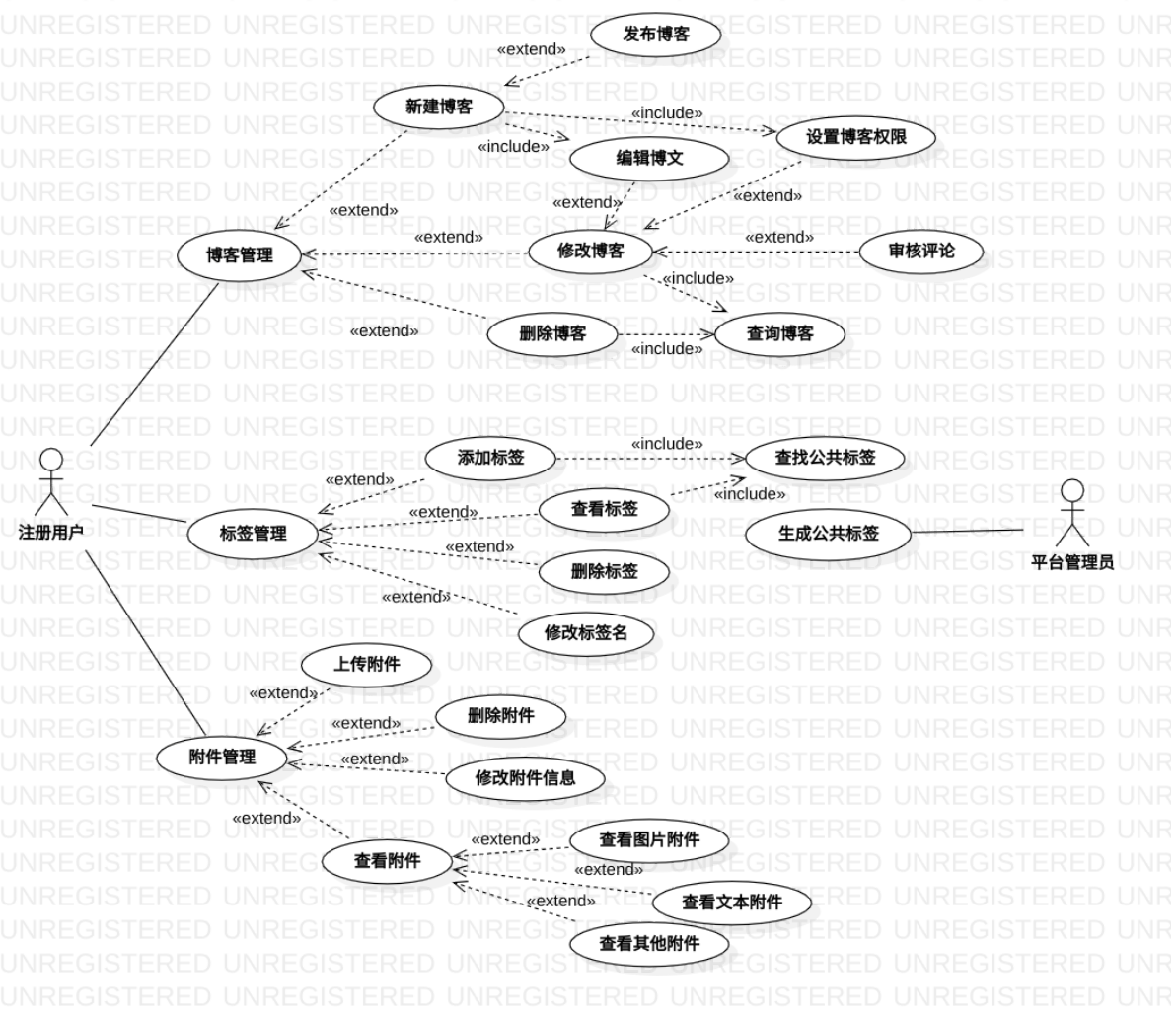


1. 用例图

下图为博客管理部分的用例图（后期水印可去除）：



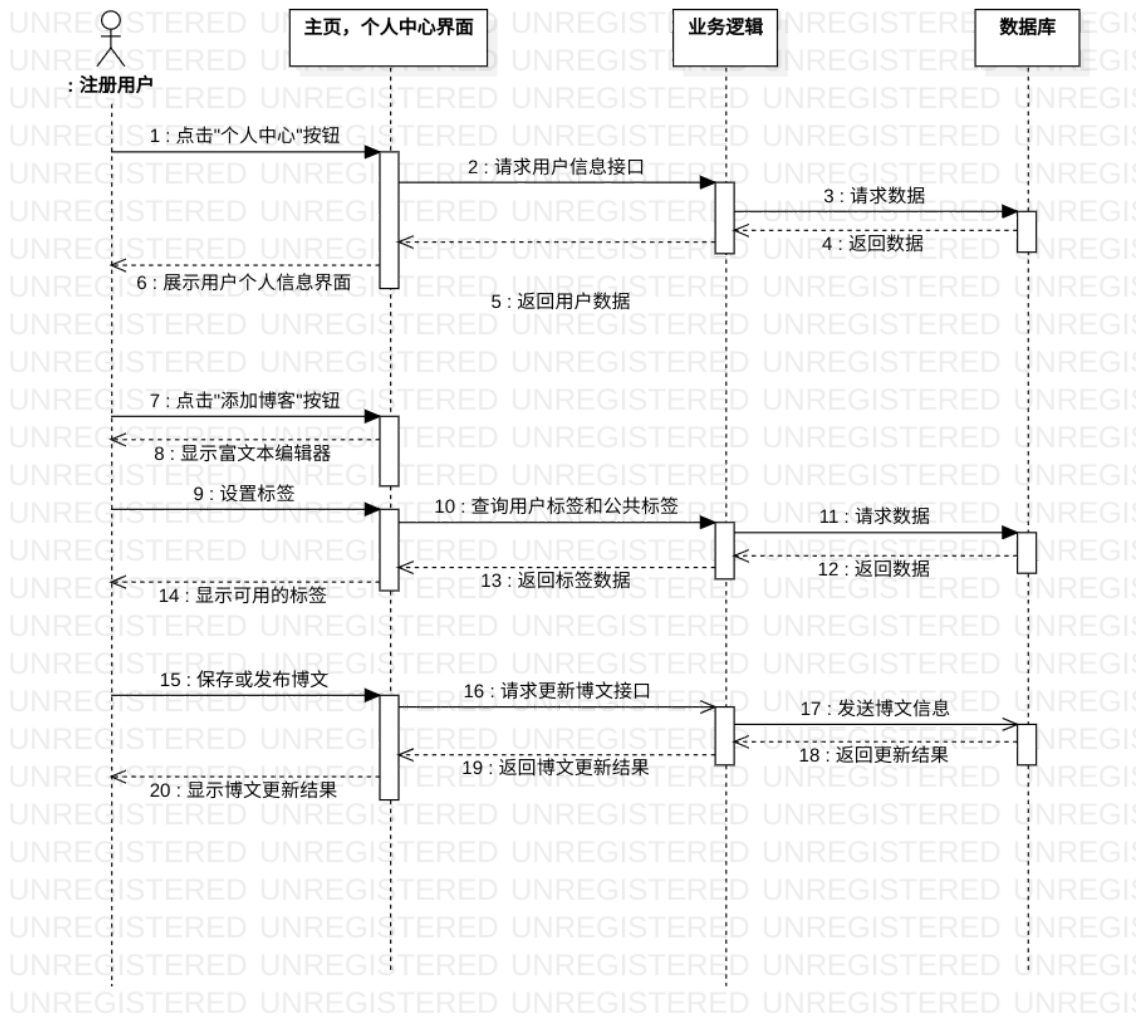
2. 功能列表

2.1 新建博客

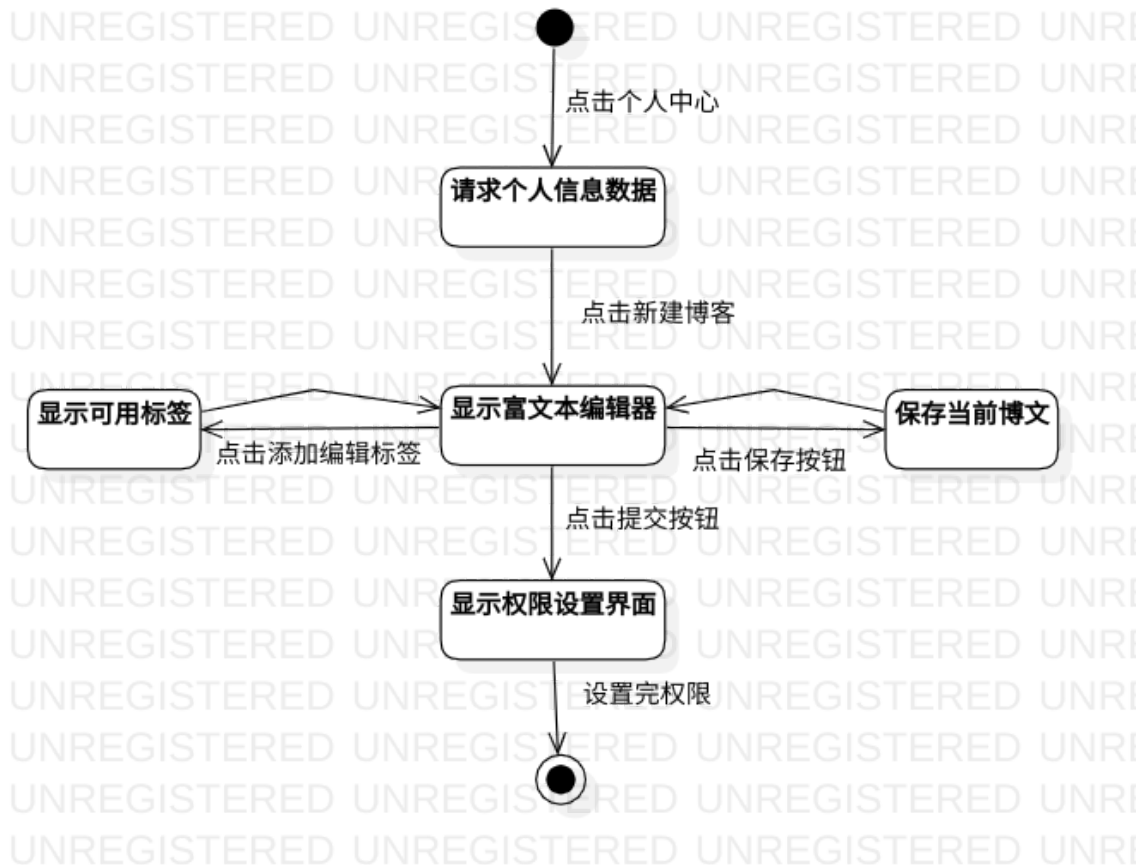
- 模块功能介绍:

注册用户打开网站主页时，会在右上角显示用户的头像，点击头像进入用户个人中心。在个人中心界面中点击"新建博客"按钮(处于"个人中心"界面左侧导航栏)，则在界面主部分显示一富文本编辑器，可显示markdown源码或发布后的效果。编辑完成后，用户可以点击按钮以发布或保存；选择发布时，会要求用户填写当前博客的权限。

优先级：高
- 时序图:



- 活动图:



- 用例文档:

名称	内容
用例名称	新建博客
用例编号	Blog-M1
行为角色	注册用户
简要说明	1. 显示富文本编辑器 2. 允许用户编辑文本,添加标签 3. 点击发布则将发布博文.
触发器	用户点击"新建博客"按钮
前置条件	用户已经进入个人中心界面
后置条件	如果用户保存了此次编辑,退出界面时记录已经编辑的博文
流程	1. 点击"新建博客"按钮 2. 编辑博文. 编辑过程中包括: 编辑标题,编辑正文,添加标签,插入图片等功能. 3. 点击保存,以生成当前博文的记录,但不公开. 4. 点击发布
异常处理	1. 网络故障: 提示"网络故障"
优先级	高
备注	-

- restful标准JSON交互数据

一些规定:

1. 数据结构我是按照typescript来写的. 某个类型后面如果加 "?", 说明该字段是可选的 (可能不包含该字段, 或该字段的值为null)
2. 返回给前端的数据总是这样的类型:

```

1  type Result<T> = {
2      code: int = 0    // 表示此次访问成功了
3      content: T       // 带上此次访问的结果.
4  } | {
5      code: int        // 错误码
6      reason: string   // 描述此次失败的原因
7  }

```

如果访问成功时不需要数据, 则类型标记为 `Result<>`.

3. RESTful接口示例如下:

```

1  [GET] "/blog/add/" (blog: Blog) → Result<>

```

这个接口表示的含义为: 此次访问的方法为GET; URL为 `/blog/add/`; 携带的参数为一blog类型的JSON; 该接口的返回值为不带数据的 `Result<T>` 类型.

```

1  // 博客的结构:
2  type Blog = {

```

```

3      id: long
4      author: User    // 作者，结构这里按下不表。
5      title: string   // 标题
6      content: string // 内容
7      tags: Tag[]     // 博客的标签
8      avatar: string? // 博文缩略图，在博客列表中展示（可选功能）
9      permissions: {
10         isPublic: boolean           // 是否公开
11         needReviewComment: boolean // 评论是否需要审核（是否开启精选评论）
12     }?
13 };
14
15 // 标签的结构
16 type Tag = {
17     id: long
18     name: string
19     owner: User? // 如果是null，则说明该标签是公共的
20     description: string
21 }
22
23 // 添加博客。在第一次保存博客时应调用此接口。
24 [POST] "/blog/add/" (blog: Blog) → Result<>
25
26 // 修改博客。可以是修改权限或博文内容。发布博文的时候只要修改`permissions`中的
27 // `isPublic`字段即可`。
28 [POST] "/blog/modify/" (blog: Blog) → Result<>
29
30 // 查询可用标签
31 [GET] "/blog/tags/" (user: User) → Result<{
32     personal: Tag[]
33     public: Tag[]
34 }>

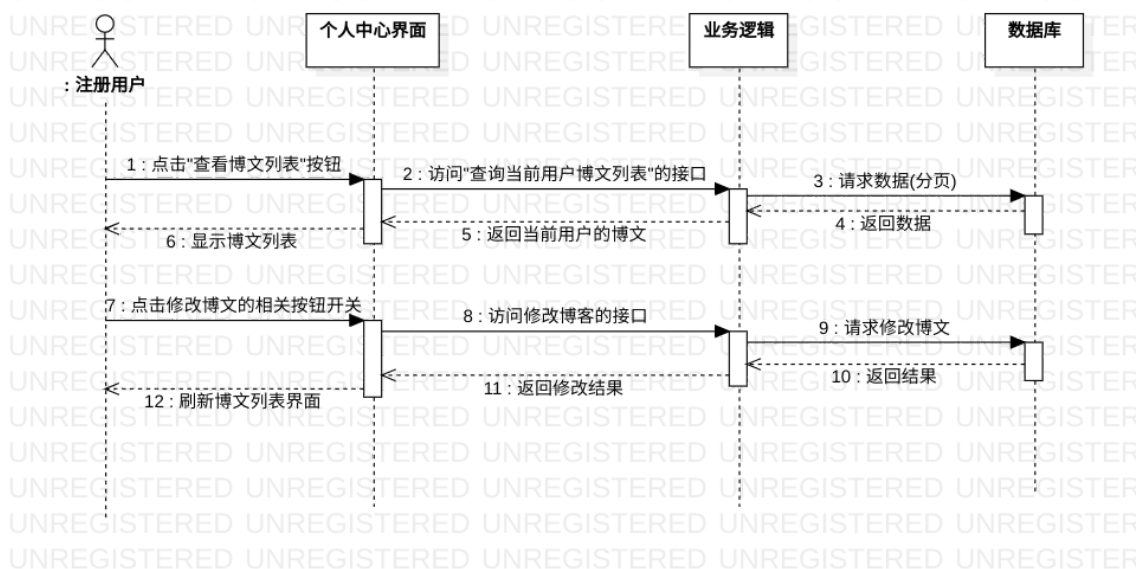
```

2.2 编辑个人博客

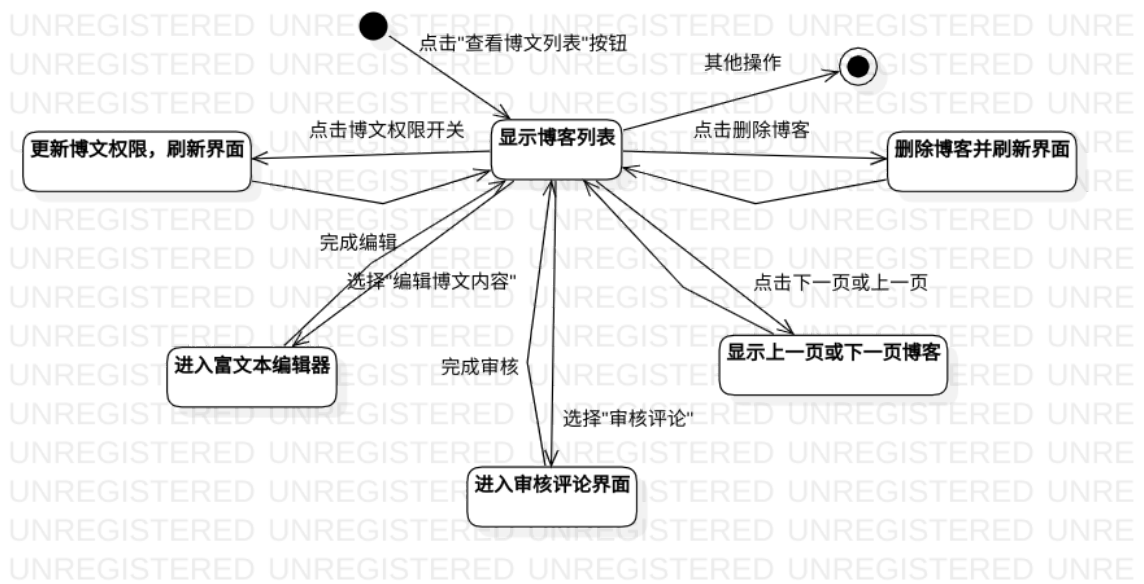
- 模块功能介绍

首先用户先进入一列表，显示当前用户的所有博客；列表上包括了设置每个博文权限的开关(公开，非公开；评论是否需要审核 两个开关)。也提供了"编辑博文"和"删除博文"的选项。点击"编辑博文"将进入和"新建个人博客"相同的界面，但内容将被填充。

- 时序图



- 活动图



- 用例文档

名称	内容
用例名称	编辑博客
用例编号	Blog-M2
行为角色	注册用户
简要说明	<ol style="list-style-type: none"> 1. 显示当前用户的博客列表。该列表是分页显示的。 2. 列表每项显示博文的标题和缩略图(如果有的话); 显示两个权限开关; 审核评论的按钮; 编辑内容的按钮; 删除博文的按钮。 3. 点击编辑内容则进入富文本编辑器; 点击审核评论则进入审核评论的界面。
触发器	用户点击"查看博文列表"按钮
前置条件	用户已经进入个人中心界面
后置条件	每次用户操作后将此次对博客的修改保存到数据库中。
流程	<ol style="list-style-type: none"> 1. 点击"查看博文列表"按钮; 2. 点击"上一页"或"下一页"浏览博文列表; 3. 点击开关修改博客的权限; 4. 点击删除按钮以删除博文; 点击审核评论按钮以审核评论; 点击编辑内容按钮则进入富文本编辑器。
异常处理	<ol style="list-style-type: none"> 1. 网络故障: 提示"网络故障"。
优先级	高
备注	<p>界面应该包含一"返回上一页"的按钮, 方便用户操作。</p> <p>如, 用户已经翻到博文列表的第5页了, 应该允许用户在审核完某博客评论后快速返回到第五页。</p> <p>审核评论按钮旁边应该显示当前有多少评论尚未被审核。</p>

- restful标准JSON交互数据

```

1 // 查询当前用户的博客, 分页查询。page下标从0开始, perpage是每页的数量。
2 // 如, (page, perpage) = (0, 30), 则查询第0~29个博文(按时间从近期到过去的顺序),
3 // (page, perpage) = (2, 50) 则会查询第100~149个博文。
4 [GET] "/blog/list/" (user: User, page: int, perpage: int) → Result<{
  ..Blog, unReviewedCount: int }[]>
5 // 返回一个列表, 列表元素除了包含博客的信息, 还包含了未审核的评论的数量。
6
7 // 删除博客
8 [POST] "/blog/remove/" (blog: Blog) → Result<>

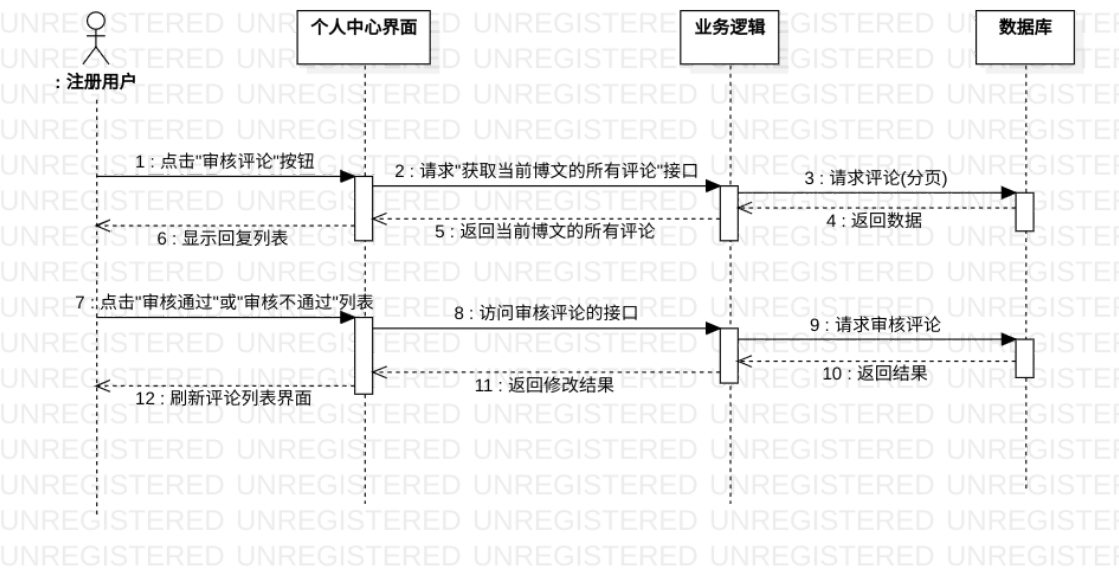
```

2.3 审核评论

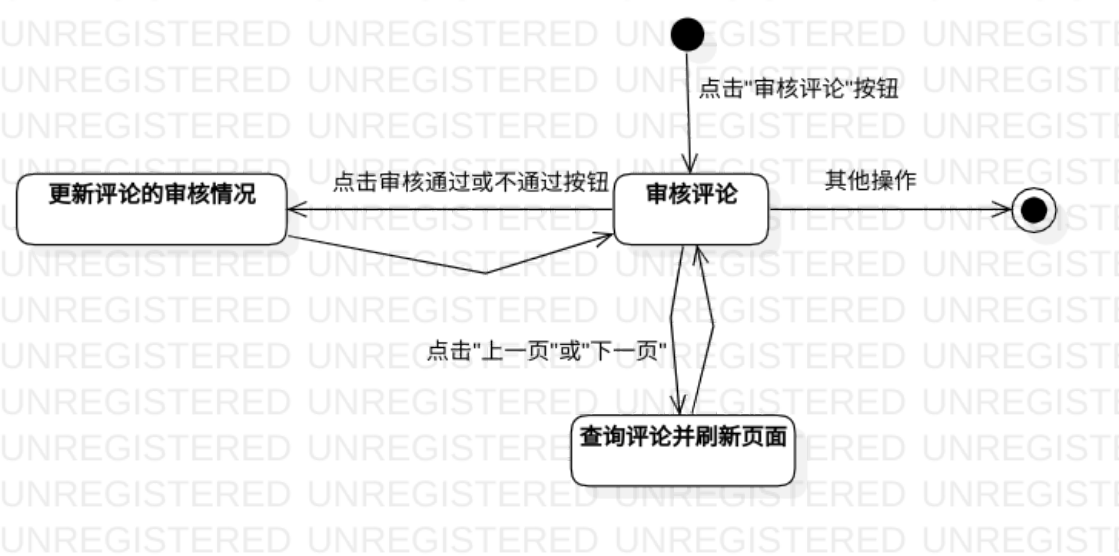
- 模块功能介绍

用户在博客列表中可以点击"审核评论"的按钮，就可以进入当前功能。此功能下将列出某博客下的所有评论形成列表，博主可以依次审核通过与否。

- 时序图



- 活动图



- 用例文档

名称	内容
用例名称	审核评论
用例编号	Blog-M3
行为角色	注册用户
简要说明	1. 显示博客的所有评论(不包括回复其他人的评论)，分页显示 2. 每个博客显示"通过审核"和"不通过审核"两个按钮，点击即可设置评论的审核情况。
触发器	点击"审核评论"按钮。
前置条件	用户已经进入博客列表界面。
后置条件	每次设置博客权限都保存此次操作。
流程	1. 点击"审核通过"则通过此评论，并使该评论在评论区可见； 2. 点击"审核不通过"则记录下，下次审核评论时该评论不列在列表中。
异常处理	1. 网络故障：提示"网络故障"
优先级	高
备注	设置了"允许自由评论"的博客也可以进入此功能： 有些博客可能在中途才设置允许自由评论，需要进入此界面审核该设置前的评论。

- restful标准JSON交互数据

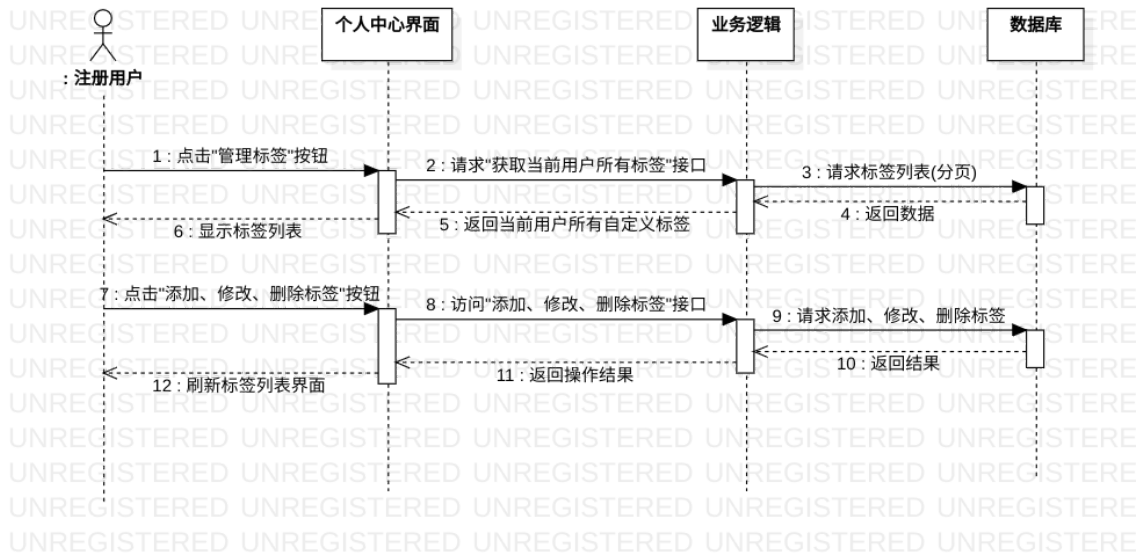
```

1  type Comment = {
2      id: long
3      author: User
4      content: string
5      fromBlog: Blog
6      replyTo: Comment?
7      reviewPassed: boolean? // 为空为未审核；真为审核通过
8  }
9
10 // 分页获取某博客的所有评论列表。如果all不设置为true，则只返回未审核的评论。
11 [GET] "/blog/comment/list" (blog: Blog, page: int, perpage: int, all:
boolean?) → Result<Comment[]>
12
13 // 审核某评论。也允许设置为"未审核"(比如手滑了，撤销用)
14 [POST] "/blog/comment/review" (comment: Comment, reviewPassed: boolean:
boolean?) → Result<>

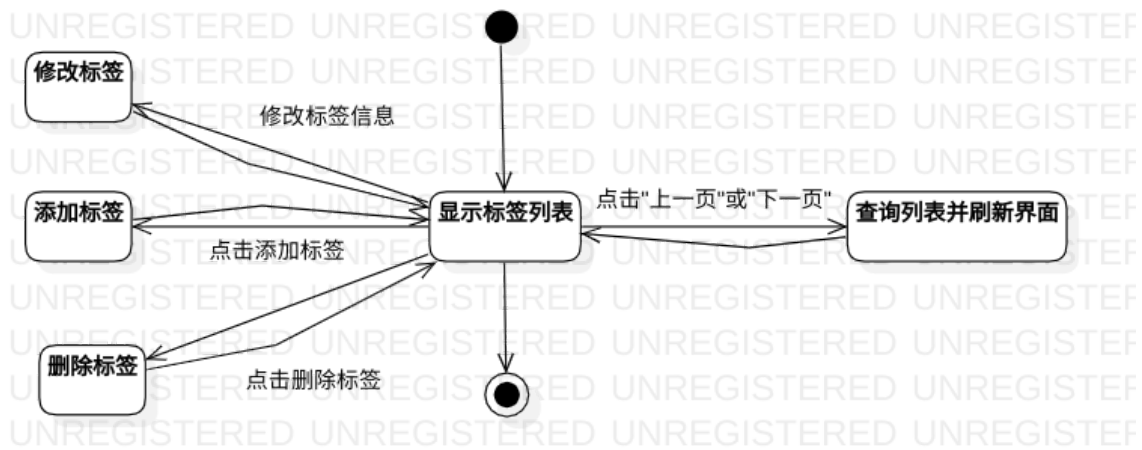
```


2.4 管理标签

- 模块功能介绍
包括添加,修改,删除标签。标签主要用在搜索中。
- 时序图



- 活动图



- 用例文档

名称	内容
用例名称	管理标签
用例编号	Blog-M4
行为角色	注册用户
简要说明	1. 这个用例包括了"添加", "修改", "删除" 三个功能; 2. 界面分为上下两部分; 上部分较小(一行), 用户输入想要添加的标签名, 点击"添加"即可添加标签; 3. 下半部分为标签列表, 一行一个标签; 用户可以直接编辑标签名和标签描述, 也可以删除特定标签.
触发器	用户点击"管理标签"按钮
前置条件	用户已经进入个人中心界面
后置条件	保存用户对标签的操作
流程	1. 点击"管理标签"按钮 2. 点击"添加标签"则添加标签. 网页会查询到相似的公共标签, 并询问用户是否要继续添加; 3. 修改标签名或描述后, 点击保存即可保存此次对标签的修改; 4. 点击删除按钮以删除标签.
异常处理	1. 网络故障: 提示"网络故障"
优先级	高
备注	-

- restful标准JSON交互数据

```

1 // 获取所有用户自定义标签(分页)
2 [GET] "/tag/list/" (user: User, page: int, perpage: int) → Result<Tag[]>
3
4 // 修改标签
5 [POST] "/tag/modify/" (tag: Tag) → Result<>
6
7 // 删除标签
8 [POST] "/tag/remove/" (tag: Tag) → Result<>
9
10 // 添加标签. 后端会做操作, 以查询是否有相似的公开标签, 所以这个API返回的数据的code有
    挺大几率不是0的(0表示成功), 有可能就是遇到了相似的公开标签, 这种标签列表将会以JSON字
    符串形式出现在Result的错误原因中, 前端需要注意转换.
11 // 默认情况force为false. 如果为true, 则跳过公开标签检查, 直接添加标签.
12 [POST] "/tag/add/" (tag: Tag, force: boolean?) → Result<>

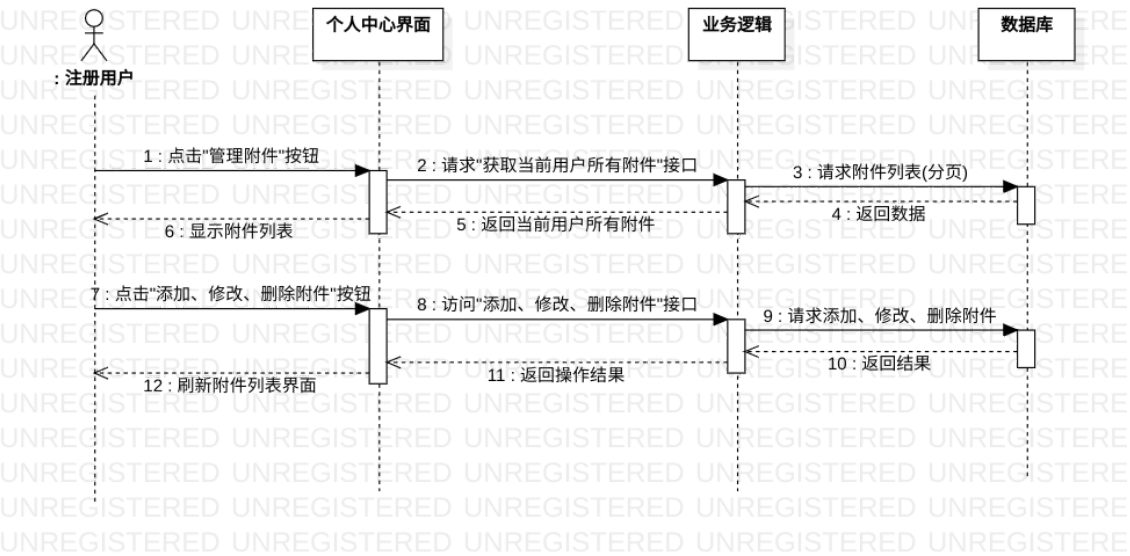
```

2.5 管理附件

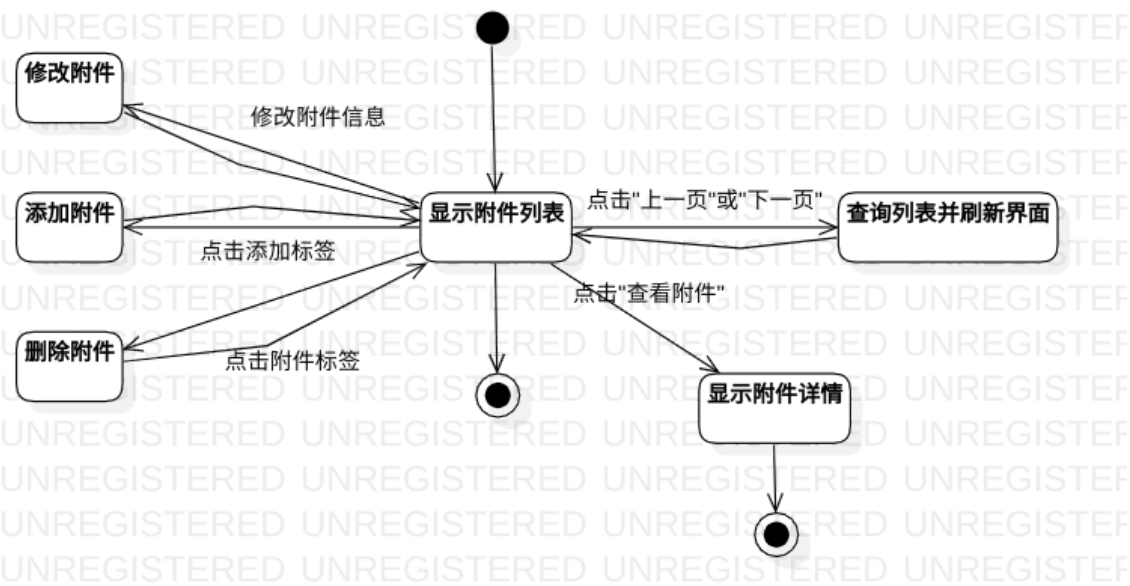
- 模块功能介绍

用户在编写博客时可能需要引用图片或提供附件。系统为注册用户提供文件存储的功能，以便用户在博客中使用。

- 时序图



- 活动图



- 用例文档

名称	内容
用例名称	管理附件
用例编号	Blog-M5
行为角色	注册用户
简要说明	1. 这个用例包括了"添加", "修改", "删除" 三个功能; 2. 界面分为上下两部分; 上部分较小(一行), 用户输入想要添加的附件名, 点击"添加"从本地选择文件, 即可添加附件; 3. 下半部分为附件列表, 一行一个标签; 用户可以编辑附件名并查看附件的URL, 也可以删除附件.
触发器	用户点击"管理附件"按钮
前置条件	用户已经进入个人中心界面
后置条件	保存用户对附件的操作
流程	1. 点击"管理附件"按钮 2. 点击"添加附件"则添加附件. 网站对用户总的附件大小有限制. 3. 修改附件名后, 点击保存即可保存此次对附件的修改; 4. 点击删除按钮以删除附件.
异常处理	1. 网络故障: 提示"网络故障"
优先级	高
备注	-

- restful标准JSON交互数据

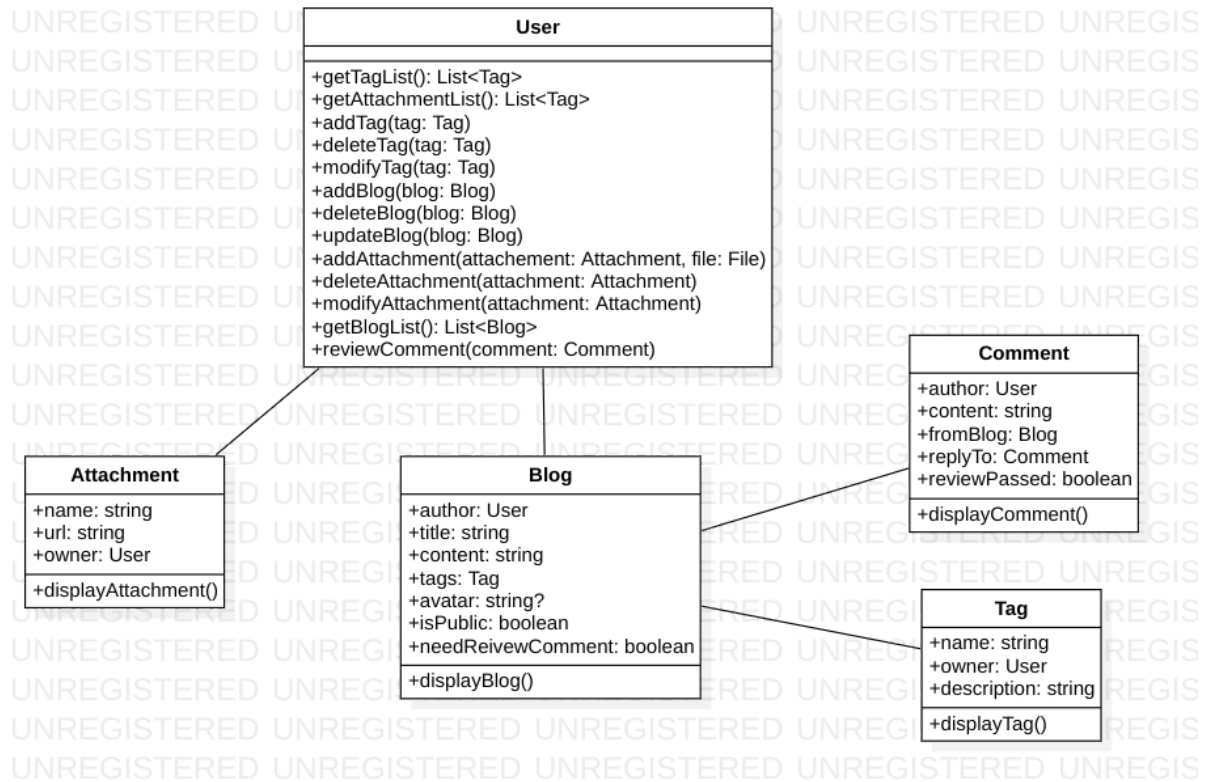
```

1  type Attachment = {
2      id: long
3      name: string
4      url: string
5      owner: User
6  }
7  // 添加附件 (File即以二进制上传文件)
8  [POST] "/attachment/add/" (tag: Attachment, file: File) →
   Result<Attachment>
9
10 // 获取所有用户上传的附件(分页)
11 [GET] "/attachment/list/" (user: User, page: int, perpage: int) →
   Result<Tag[]>
12
13 // 修改附件信息
14 [POST] "/attachment/modify/" (attachment: Attachment) → Result<

```

```
15
16 // 删除标签
17 [POST] "/attachment/remove/" (attachment: Attachment) → Result◇
```

3. 类图与CRC模型



类名：User	
编号：待定	
描述：使用这个系统的注册用户。	
功能	合作类
获取标签列表	
获取附件列表	
获取博客列表	
添加标签	Tag
删除标签	Tag
修改标签	Tag
添加附件	Attachment
删除附件	Attachment
修改附件	Attachment
添加博客	Blog
删除博客	Blog
修改博客	Blog
审核评论	Comment

类名：Blog	
编号：待定	
描述：博客。包含了标题,内容和标签	
功能	合作类
展示博客	

类名：Comment	
编号：待定	
描述：博客的评论。可以是对博客的评论，也可以是对其他评论的回复。	
功能	合作类
展示评论	

类名：Tag	
编号：待定	
描述：博客的标签。在注册用户创建博客时添加，方便浏览者查询博客。	
功能	合作类
展示标签	

类名：Attachment	
编号：待定	
描述：附件，可以是图片或其他文件。注册用户可以上传文件，上传的图片可以通过URL嵌入到博客中；上传的文件也可以以URL的形式分享。	
功能	合作类
展示附件	

4. 数据流图

顶层数据流图



一层数据流图

