



JS 论坛 APP

软件需求规格说明书



修改历史

日期	版本	作者	修改内容
2018/5/13	1.0	张胜利	新建
2018/6/12	2.0	张胜利	功能需求
2018/7/25	2.1	张胜利	用户特点

一、引言	3
1.1 编写目的	3
1.2 背景	3
1.3 定义	4
1.4 参考文献	5
二、项目概述	6
2.1 产品描述	6
2.2 产品功能	6
2.3 用户特点	7
2.4 一般约束	7
2.5 假设和依据	8
三、具体需求	8

3.1 功能需求	8
3.2 外部接口设计	17
3.3 性能需求	18
3.4 验收标准	19

一、引言

1.1 编写目的

本文详细描述 JS 论坛 APP 系统的需求，表述的需求信息要求明确、无二义性。用于后续软件开发，是设计的依据。

1.2 背景

当下使用 Node.js 技术的开发者们越来越多，却苦于缺少一个专业的交流和信息资源汇总的互动平台，导致初学者们的自学之路较为艰辛挫折，不利于 Node.js 的传播与发展。

鉴于此现状，我们决定开发一个 **Node.js 论坛 React APP**，采用 **B/S**，向广大技术爱好者们开放。此论坛允许游客浏览帖子、注册，允许已注册的用户登录、发布、回复帖子、点赞回复，便于实时互动。同时，还对帖子进行了分类，分为“全部”、“精华”、“分享”、“问答”、“招聘”等多类。

1.3 定义

1. 游客

尚未注册的用户，还没有用户名，在系统中没有记录，只拥有最基本的注册账户、登录账户和浏览帖子的权限。

2. 已注册用户

已经在系统中注册了的用户，在数据库中有记录，可以在对应的分类区发布帖子，对感兴趣的帖子进行回复，对认为有用的回复进行点赞互动。

3. 管理员

管理员具有最高权限，可以管理项目的基本信息，包括添加用户为达人、主题加精、主题置顶、锁定话题（不能进行回复操作）、禁言用户、删除所有用户的帖子等操作。

4. 所有

所有发布的帖子，不分类，展示在首页。

5. 精华

干货满满，适合大家学习的精品帖子。

6. 分享

“分享”类帖子具有丰富的干货，适合刚入门的新手阅读，通过前辈们的经验总结，在技术探索的道路上可以少走“弯路”。

7. 问答

“问答”类帖子是开发者们日常遇到技术上的问题交流之处，不仅可以帮助自己解决困难，还可以将自己曾经踩过的“坑”与他人分享，互帮互助，共同进步。

8 . 招聘

“招聘”类帖子集合了各地、各公司的开发岗招聘信息，使得满腔热血的技术人才有用武之地。

9 . 发布

发布相关主题的帖子，需要带标题（10 字以上）以及内容（不限字数）。

10. 消息

收到其他人的点赞、评论、回复的消息通知

11. 我

可看到昵称、加入论坛的时间，以及收藏的帖子，最近浏览的主题，最近回复的帖子，以及退出登录。

1.4 参考文献

[1] GB-T8567-2006, 《计算机软件文档编制规范》[S]

二、项目概述

2.1 产品描述

基于 js 的论坛 app，便于用户交流沟通。“全部”类帖子是在论坛中发布的所有的帖子，不分类型，可以直接在这个版块做统一的浏览；“分享”类帖子具有丰富的干货，适合刚入门的新手阅读，通过前辈们的经验总结，在技术探索的道路上可以少走“弯路”；“问答”类帖子是开发者们日常遇到技术上的问题交流之处，不仅可以帮助自己解决困难，还可以将自己曾经踩过的“坑”与他人分享，互帮互助，共同进步；“招聘”类帖子集合了各地、各公司的开发岗招聘信息，使得满腔热血的技术人才有用武之地；“精华”类帖子则是干货满满，适合大家学习的精品帖子。

2.2 产品功能

论坛分为游客、已注册用户两种角色。

游客即为未登录的用户，仅有注册账户、登录账户和浏览帖子的权限；

而已注册用户则在游客权限的基础上还可以在对应的分类区发布帖子，对感兴趣的帖子进行回复，对认为有用的回复进行点赞回复互动，收藏帖子并可以查看收藏的帖子，浏览最近看过的话题以及最近的回复，可以收到回复的消息。

管理员则具有最高权限，可以管理项目的基本信息，包括添加用户为达人、主题加精、主题置顶、锁定话题（不能进行回复操作）、禁言用户、删除所有用户的帖子等操作。

各功能模块的用例图如下：

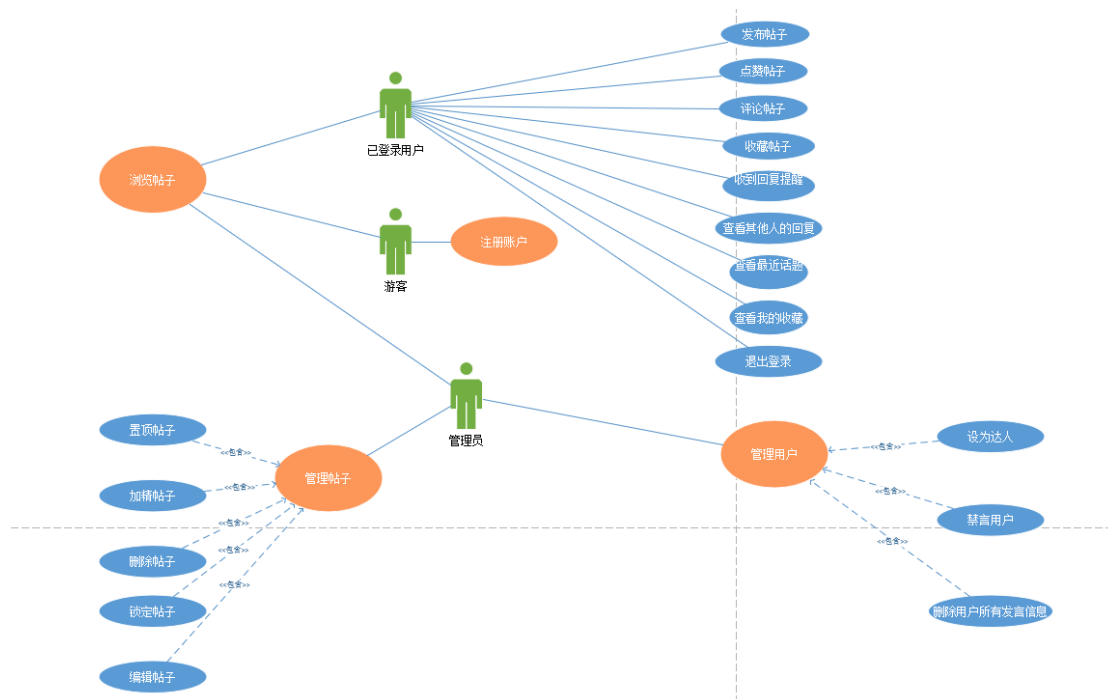


图 2-1：用例图

2.3 用户特点

用户基于在校大学生。用户主体包括团活动成员，组织者以及各种学科的讨论与解答的用户。管理员为活动组织者，问题发布人。

基于该论坛，用户能快速高效的发布活动，问题并能及时查看解答，收藏、评论、回复。

2.4 一般约束

- 1) 开发时间有限。小组成员的时间限制导致开发的功能完善程度受到一定限制。
- 2) 编程能力有限。小组成员人数 7 人，编写平台以及方法有限。

2.5 假设和依据

- 1) 根据先前的项目经验，明确分工合作
- 2) 基于 JS 完成设计
- 3) 采用 MVC 架构

三、具体需求

3.1 功能需求

3.1.1 管理员项目基本信息

简要描述	管理员管理项目的基本信息，包括添加用户为达人、主题加精、主题置顶、锁定话题（不能进行回复操作）、禁言用户、删除所有用户的帖子操作。
参与者	管理员
事件流	<p>用例起始于管理员使用管理员账号登录系统：</p> <ol style="list-style-type: none">1. 选择需要管理的用户或帖子2. 管理员选择对用户执行操作，跳转至用户的界面3. 管理员选择对主题帖执行操作，进入主题帖的详细界面 <p>置顶主题帖子流程：</p> <ol style="list-style-type: none">1. 进入帖子界面2. 点击置顶3. 完成置顶 <p>加精主题帖子流程：</p>

	<ol style="list-style-type: none"> 1. 进入帖子界面 2. 点击加精 3. 加精帖子 <p>删除主题帖子流程：</p> <ol style="list-style-type: none"> 1. 进入帖子界面 2. 点击删除 3. 删除帖子 <p>锁定主题帖子流程：</p> <ol style="list-style-type: none"> 1. 进入帖子界面 2. 点击锁定 3. 锁定帖子 <p>编辑用户主题帖子流程：</p> <ol style="list-style-type: none"> 1. 进入帖子界面 2. 点击编辑 3. 编辑帖子 <p>禁言用户子流程：</p> <ol style="list-style-type: none"> 1. 进入用户界面 2. 点击屏蔽用户 3. 禁言用户 <p>设用户为达人子流程：</p> <ol style="list-style-type: none"> 1. 进入用户界面 2. 点击设为达人 3. 完成设置用户为达人
--	---

	<p>删除用户发言子流程：</p> <ol style="list-style-type: none"> 1. 进入用户界面 2. 点击删除所有发言 3. 删除用户发言
前置条件	使用管理员账号登录系统
后置条件	管理员账号在对主题帖进行置顶、加精、删除、锁定，以及对用户进行禁言、删除、设置为达人时，系统会反馈信息。
活动图	<pre> graph TD Start(()) --> IsAdmin{是否是管理员} IsAdmin -- 否 --> End(()) IsAdmin -- 是 --> AdminSelect[管理员选择管理操作] AdminSelect --> TopicOp{对主题帖进行操作} AdminSelect --> UserOp{对用户进行操作} TopicOp --> Pin[置顶] TopicOp --> AddPin[加精] TopicOp --> Delete[删除] TopicOp --> Lock[锁定] TopicOp --> Edit[编辑] UserOp --> Ban[禁言] UserOp --> MarkDaRen[标记为达人] UserOp --> DeleteUser[删除用户所有发言] Pin --> Join(()) AddPin --> Join Delete --> Join Lock --> Join Edit --> Join Ban --> Join MarkDaRen --> Join DeleteUser --> Join Join --> End </pre> <p>The diagram illustrates the administrative workflow. It begins with a start node leading to a decision diamond '是否是管理员' (Is the user an administrator?). If '否' (No), it proceeds to an end node. If '是' (Yes), it leads to '管理员选择管理操作' (Administrator selects management operation). This then branches into two decision diamonds: '对主题帖进行操作' (Perform operation on the topic post) and '对用户进行操作' (Perform operation on the user). The first diamond branches into '置顶' (Pin), '加精' (Add to favorites), '删除' (Delete), '锁定' (Lock), and '编辑' (Edit). The second diamond branches into '禁言' (Mute), '标记为达人' (Mark as a celebrity), and '删除用户所有发言' (Delete all user posts). All these actions converge at a final join node before reaching the end node.</p>

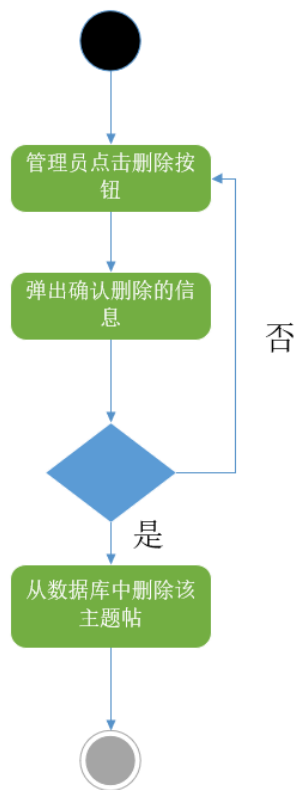


图 1-2：管理员删除帖子

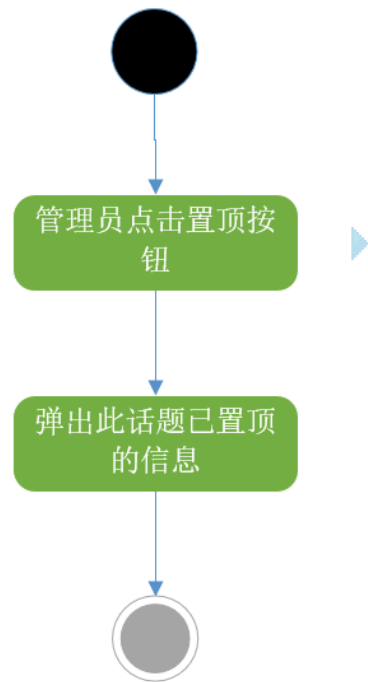


图 1-3：管理员置顶帖子

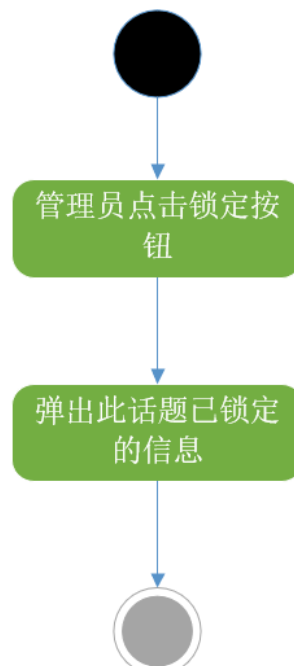
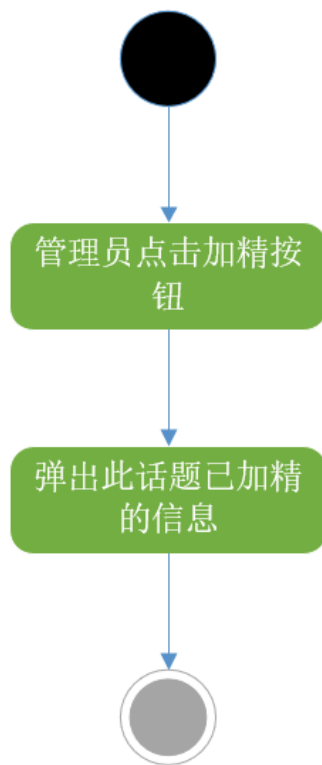


图 1-4：管理员加精帖子

图 1-5：管理员锁定帖子

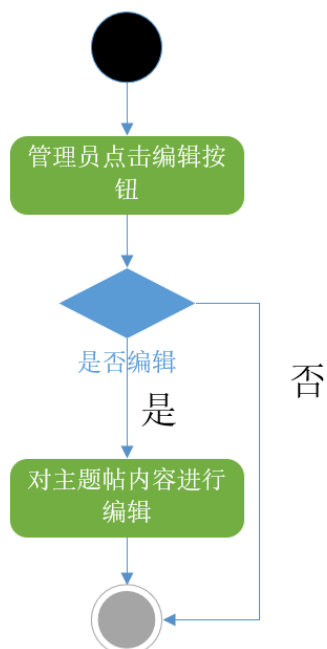
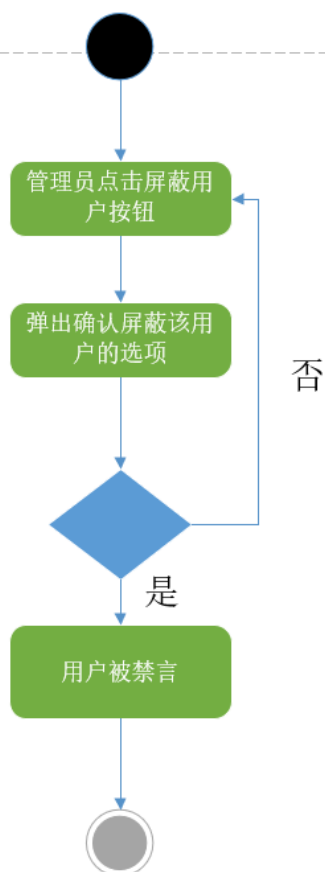
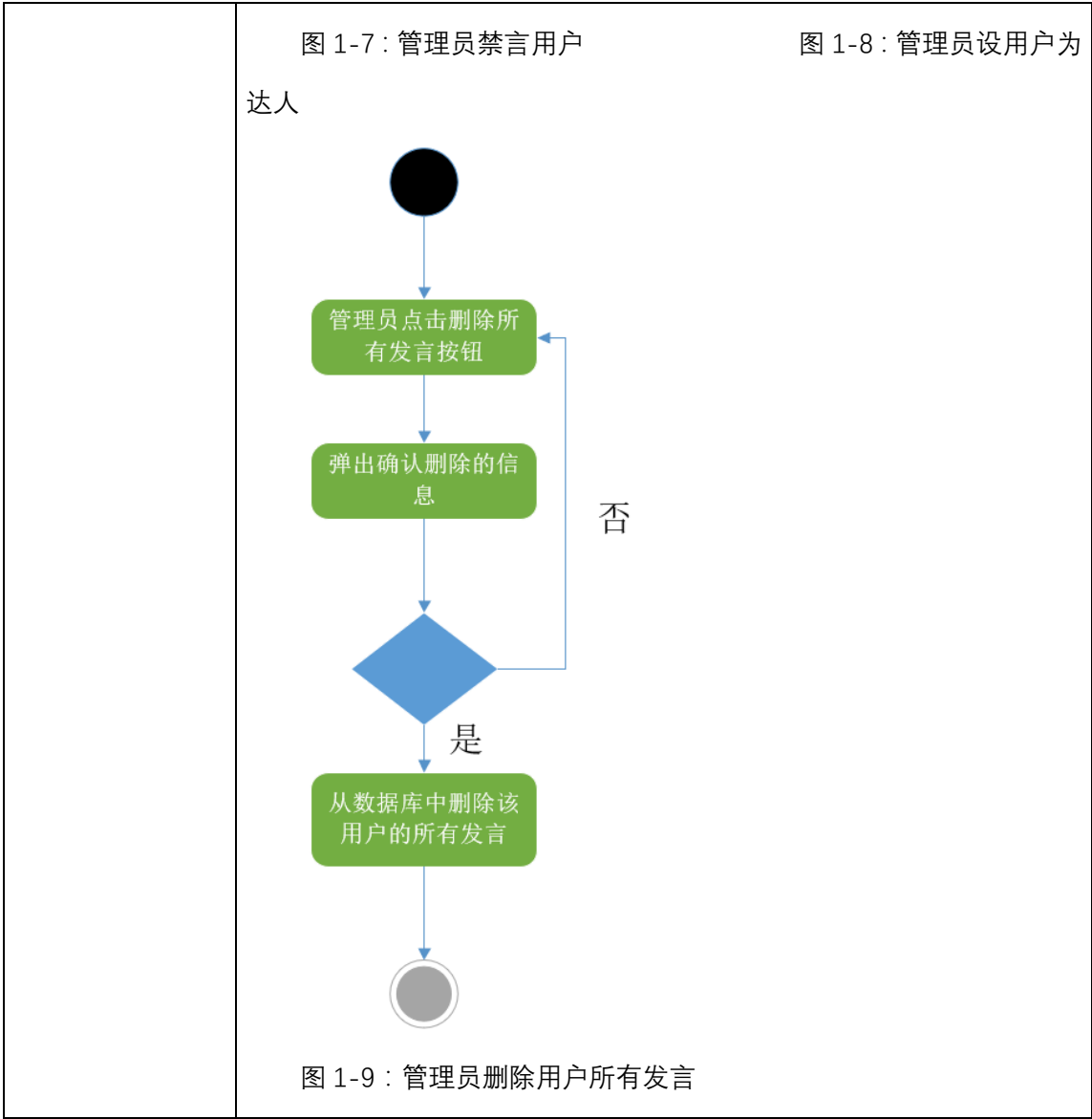


图 1-6：管理员编辑帖子





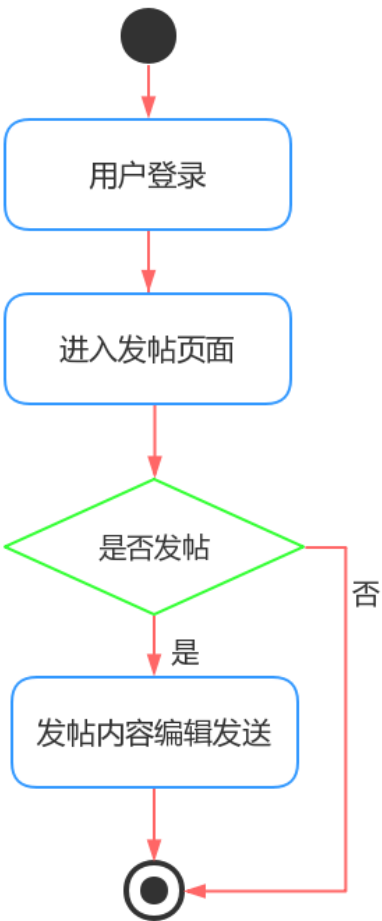
3.1.2 游客注册

简要描述	游客首次使用该 app 时，只拥有帖子的浏览权限；在注册时，输入需要的 access-token 后，注册成功，系统分配一个用户 ID 标识用户，加入用户数据库，获得用户权限。
参与者	游客

事件流	<p>用例起始于游客首次注册使用；</p> <ol style="list-style-type: none"> 1.输入用户名以及密码 1.加入用户数据库，记录用户相关信息 2.完成注册反馈
前置条件	未注册的游客
后置条件	成功注册后有注册成功信息反馈，获得发帖，评论，点赞权限
活动图	<pre> graph TD Start(()) --> Decision{是否是游客} Decision -- 是 --> Register[进入用户注册界面 (输入用户名及密码)] Register --> End((())) Decision -- 否 --> End </pre> <p>The diagram illustrates the user registration process. It begins with a start node (black circle) leading to a decision diamond labeled '是否是游客' (Is the user a visitor?). If the answer is '是' (Yes), the flow proceeds to a rounded rectangle labeled '进入用户注册界面 (输入用户名及密码)' (Enter user registration interface (input username and password)). If the answer is '否' (No), the flow bypasses the registration step and goes directly to the end node (bullseye). Both paths converge at the end node.</p>

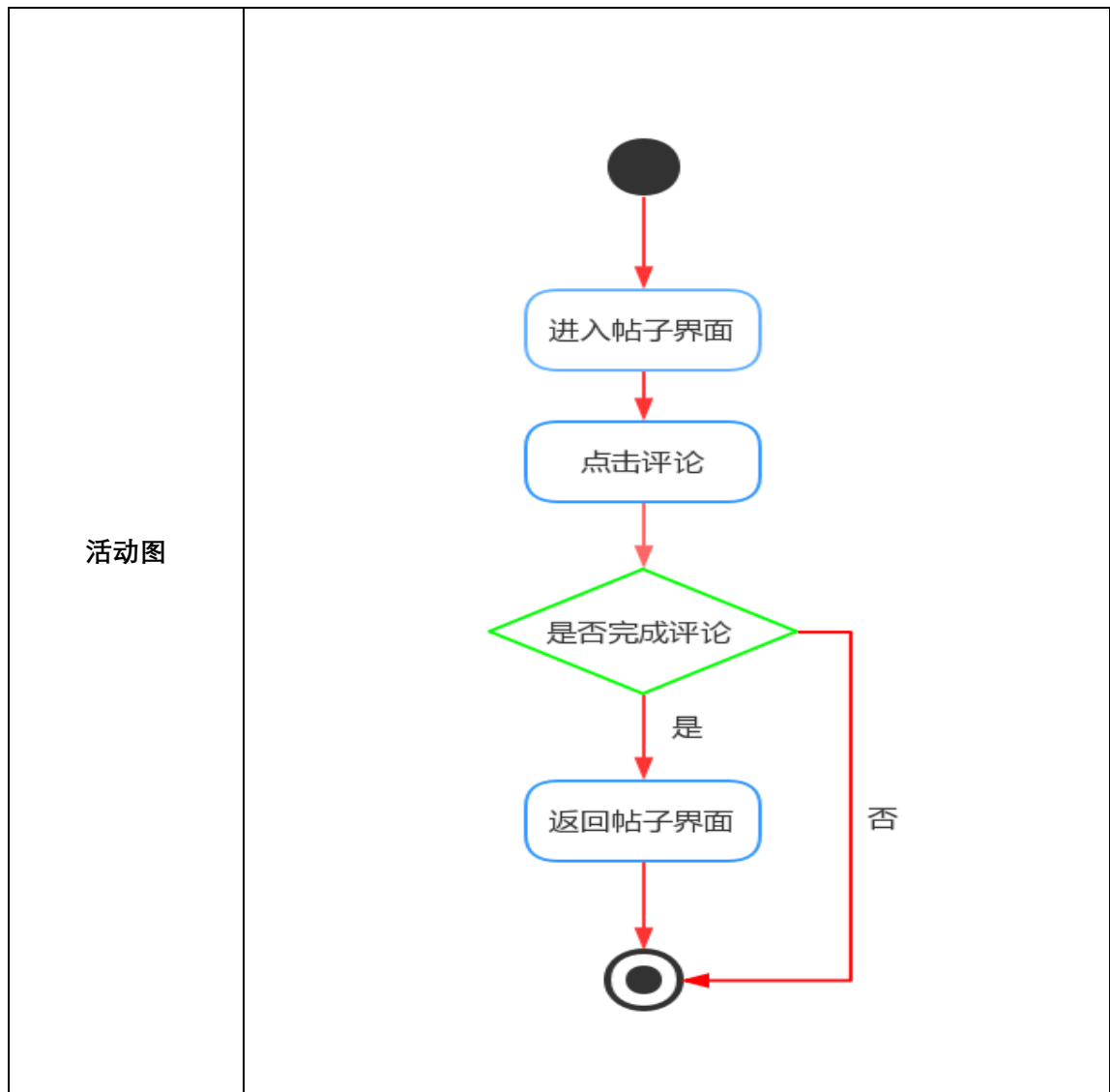
3.1.3 用户发帖

简要描述	<p>经过登录的用户可以发帖，进入发布的界面在相关分类话题社区发帖，发布的有标题，日期，发帖人等信息并且有浏览数，点赞数，回复以及分享的数量信息反馈。</p>
------	---

参与者	注册用户
事件流	用户已经注册； 1.完成用户登录 2.进入发帖页面 3.编辑发送帖子
前置条件	注册用户需登录以后。
后置条件	发帖成功后反馈成功信息。
活动图	 <pre> graph TD Start(()) --> Login[用户登录] Login --> Enter[进入发帖页面] Enter --> Decision{是否发帖} Decision -- 是 --> Post[发帖内容编辑发送] Post --> End((())) Decision -- 否 --> End </pre> <p>The diagram is a UML Activity Diagram. It begins with a solid black circle (start node). A red arrow points down to a rounded rectangle labeled '用户登录' (User Login). Another red arrow points down to a second rounded rectangle labeled '进入发帖页面' (Enter Posting Page). From there, a red arrow points down to a green diamond decision node labeled '是否发帖' (Whether to post). A red arrow labeled '是' (Yes) points down from the diamond to a rounded rectangle labeled '发帖内容编辑发送' (Post content edit and send). A red arrow labeled '否' (No) points right and then down from the diamond to the final node. The final node is a bullseye symbol (solid black circle inside a larger black circle). All activity nodes are connected by red arrows.</p>

3.1.4 用户评论

简要描述	在发帖下，用户可以进行评论。进入选择的帖子下，点击评论进入评论界面，完成后点击确认完成评论。评论完成后可以进行点赞，回复，置顶操作。
参与者	用户
事件流	用例起始于已经发布的帖子； 1) 进入帖子界面 2) 点击评论 3) 编辑发送评论
前置条件	已发布帖子
后置条件	评论下有点赞，回复，置顶功能；评论成功后返回帖子界面



3.2 外部接口设计

3.2.1 用户接口

采用 MVC 架构，用户通过图形界面前端的 button 来与后台交互。

3.2.2 硬件接口

无

3.2.3 软件接口

无

3.2.4 通信接口

无

3.3 性能需求

1. 功能

- 1) 该网站满足在同一时间内被多个用户同时访问。
- 2) 如果某个用户的帖子被评论或被回复，应该通知该用户。

2 . 可行性

本系统为 APP 类应用，可在 Android 或 ios 系统使用。

3 . 可靠性

该 APP 需要在每周七天，每天二十四小时内都应是可以使用的。宕机的时间应少于 1%。

4 . 性能

- 1) 服务端要能够在 3~5 秒钟内，给予用户响应并有所显示。
- 2) 允许大量的并发用户访问

5 . 可支持性

无

6 . 安全性

- 1) 防止登录成员篡改他人发布的帖子信息。
- 2) 只有注册登录的用户才可以发布帖子，对帖子进行回复互动。
- 3) 无论是游客还是登录用户都不可以删除自己或他人的帖子。

3.4 验收标准

各功能设计要求满足功能需求以及性能需求。