

需求分析报告

一、系统开发背景

1. 现行系统:

传统的学术会议和期刊都会采用邮寄和电子邮件两种。邮寄投稿方式需要很长时间的反馈,并有可能投递失误。近年来电子邮件投稿方式的弊端也逐渐凸显,一方面论文作者需要通过电话来咨询稿件的审稿情况,大量的稿件也需要编辑部逐一下载、登记、打字排版,这无疑增加编辑部的工作量;另一方面,接受的电子邮件稿件相关信息往往不是很完整,如果处理不及时,会延长稿件处理周期。

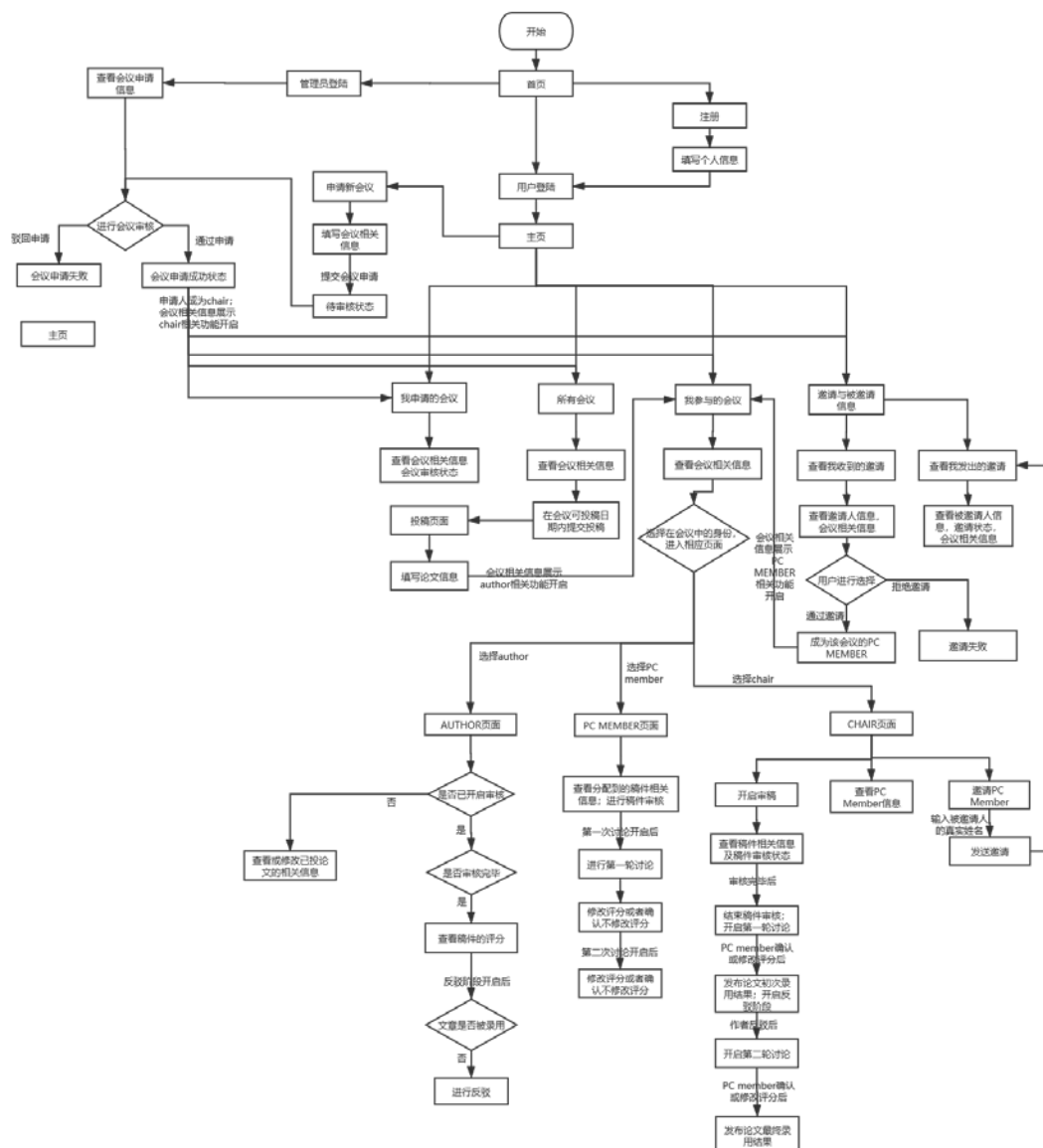
随着学术界的论文越来越多,论文种类越来越繁杂,学术界对效率要求越来越高,传统方法已经很难满足对大量稿件进行高效的处理;同时,在全球化的今天,世界各地的学术交流也越来越重要,传统方法无法实现即时沟通。

2. 开发目标:

在线投稿系统立足于网络投稿,旨在为用户提供一个友好的网络论文投稿平台。

- 1) 保证信息的准确性和时效性。随时通过计算机对系统进行管理,使得论文、作者等信息处于最新状态,保证统计数据和分析的准确。
- 2) 减轻劳动强度、提高工作效率。网络在线投稿系统可以发挥计算机的强大功能,让管理人员从大量繁琐的手工劳动中解放出来,全方位整合资源,提高工作效率。
- 3) 吸引稿源。本系统旨在为作者提供一个方便的平台,作者可以及时获得稿件处理意见及相关进程,操作过程非常友好,加上友好的用户界面,因此作者也会更愿意在此平台投稿。
- 4) 增加论文管理的透明度,规范化管理。相较于邮寄和电子邮件,从投稿,审稿,编辑,发行等全过程的网络化管理更加规范,同时能即时发现学术不端等问题,论文管理更加透明,可以使学术更加公正。

二、 业务流程



1. 会议申请

普通用户输入用户名密码登录后，进入会议申请页面，输入会议相关信息并提交，会议进入待审核状态。若管理员通过会议申请，则会议申请成功，该用户自动成为会议的 chair 和 pc，否则会议申请失败。

2. 管理员审核会议申请

管理员用管理员账号登录，进入会议审核页面，若通过申请，则会议申请成功，申请人自动成为该会议的 chair 和 pc，否则拒绝申请，会议申请失败。

3. 开启投稿

用户进入我参与的会议页面，选择 chair 身份进入会议，点击开启投稿按钮，会议进入投稿状态。

4. 邀请 pc

用户进入我参与的会议页面，选择 chair 身份进入会议，在搜索框中输入指定用户的 fullname，进行搜索，勾选结果，发送邀请，等待用户通过或拒绝邀请。可进入邀请界面查看我发出的邀请具体信息及其状态。

5. 用户查收邀请

用户登录后进入邀请界面，可查看我收到的邀请具体信息。若通过邀请，选择相关 topic，则成为该会议的 pc，否则拒绝邀请。

6. 论文投稿

用户登录后进入所有会议页面，可查看所有已开启投稿的会议。选择会议后，填写相关投稿信息，选择与内容相关的 topic，提交投稿，成为该稿件的投稿人。可进入我参与的会议，以 author 身份进入该会议，查看稿件详情。

7. 开启审稿

用户登录后进入我参与的会议页面，选择指定会议以 chair 身份进入。可查看投稿详情，pc 详情。点击开启审核按钮，选择按 topic 相关度分配或按平均负担分配。若某篇稿件的 pc 数不足 3 人，则稿件分配失败，需继续邀请 pc，否则稿件分配成功，会议进入审稿状态。

8. 审稿

用户登录后进入我参与的会议，选择指定已开启审稿的会议以 pc 身份进入。可查看稿件详情，预览或下载稿件，填写评分后提交。

9. 结束审稿

用户登录后进入我参与的会议，以 chair 身份进入指定会议。若每篇稿件的三个 pc 都已提交评分，则可点击结束审稿按钮结束审稿，同时开启第一轮讨论。

10. 讨论

用户登录后进入我参与的会议，以 pc 或 chair 的身份进入指定会议。若会议处于一轮讨论或二轮讨论状态，可进入文章的讨论页面，发布回贴、回复。

Pc 进行讨论后，可修改或确认自己的评分。

11. 发布初次评审结果

用户以 chair 身份进入会议，若所有稿件三个 pc 都已确认或修改自己的评分，则可点击发布初次结果按钮，发布一轮论文录用结果。

12. Rebuttal

Chair 发布一轮结果后，用户可以 author 的身份进入会议，查看自己的稿件是否被录用。若稿件未被录用，用户可提交 rebuttal 信息进行反驳，稿件进入二轮讨论状态，等待发布最终结果。

13. 发布最终结果

用户以 chair 身份进入某个会议，若所有稿件的三个 pc 都已对 rebuttal 的稿件进行重新讨论、评分，则用户可点击发布最终结果，发布最终论文录用情况。

三、 功能需求

A. 用户注册

1. 用户注册时需提供所需要的相关信息。

1.1 用户需提供用户名、真实姓名、密码、邮箱、公司、所在地信息。

1.1.1 任意项不能为空，若为空要标红并给出不能为空提示。

1.1.2 用户名不少于五位，必须以字母开头，不能包含特殊字符。若不符合要标红并给出正确格式提示。

1.1.3 密码不少于六位，必须以数字、字母或下划线组成，不能包含用户名。若不符合要标红并给出正确格式提示。

1.1.4 确认密码必须与密码相同。若不匹配要标红并给出密码不相同提示。

1.1.5 邮箱必须符合标准邮箱格式。若不符合要标红并给出正确邮箱格式。

1.2 用户名、邮箱不能与系统中已注册用户重复，若不符合需给出用户名已存在提示，或邮箱已存在提示。

1.3 密码需要设置为隐式，不能直接显示在页面上。

2. 用户提交注册表单后要给出是否成功的提示。

B. 用户登录

1. 用户登录时需提供所需要的相关信息。

1.1 用户需提供用户名、密码。

1.1.1 任意项不能为空，若为空要标红并给出不能为空的提示。

1.2 用户名与密码必须匹配，若不匹配需给出密码错误提示。

1.3 用户名必须存在，若不存在需给出用户名不存在提示。

2. 用户提交登录表单后要给出是否成功的提示。

C. 会议申请

1. 用户申请会议时需提供相关信息。
 - 1.1 用户需要提供写会议的简称、全称、举办时间、举办地点、投稿截止日期、评审结果发布日期。
 - 1.1.1 所有信息不能为空，若为空需要标红并给出不能为空提示。
 - 1.1.2 举办时间<投稿截止日期<评审结果发布日期。
 - 1.2 用户提交会议申请表单后需给出成功与否提示。
2. 用户查看会议申请。
 - 2.1 用户可以方便地找到已提交的会议申请。
 - 2.2 用户可以查看已提交的会议申请的具体信息及审核状态。
 - 2.2.1 若通过会议申请，用户成为该会议的 chair 和 pc，可在我参与的会议中以 chair 或 pc 的身份进入该会议。
3. 管理员审核会议申请。
 - 3.1 管理员可以方便的找到待审核会议。
 - 3.2 管理员可以查看待审核会议的具体信息。
 - 3.3 管理员可以通过会议申请或拒绝会议申请。
 - 3.3.1 若通过会议申请，则申请人自动成为该会议的 chair 和 pc。
 - 3.4 管理员通过或拒绝会议申请后要给出成功与否的提示。

D. 邀请

1. 用户以 chair 的身份进入会议后可邀请 pc。
 - 1.1 chair 可以在搜索框中搜索系统用户的真实姓名。
 - 1.2 chair 可以查看搜索结果中用户的具体信息，包括用户名、真实姓名、邮箱、公司、所在地。
 - 1.3 chair 可以选择搜索结果中的用户，并发送 pc 邀请。
 - 1.3.1 chair 发送邀请后可在邀请界面-我发送的邀请中查看邀请的具体信息，包括会议信息、被邀请人信息及该邀请的状态（通过、拒绝或代查看）。
2. 用户查看受到的邀请。
 - 2.1 用户登录后可进入邀请界面-我收到的邀请查看收到的邀请。
 - 2.1.1 可以查看邀请的具体信息，包括会议信息、邀请人信息。
 - 2.2 用户可以通过或拒绝收到的邀请。

2.2.1 若通过邀请，则成为该会议的 pc。

2.2.2 若通过邀请，可在我参与的会议中以 pc 的身份进入该会议。

2.3 用户通过或拒绝邀请后要给出操作成功与否的提示。

E. 投稿

1. 用户投稿必须满足以下条件。

1.1 只允许注册用户进行投稿，并且登录成功之后才允许投稿。

1.2 只能在会议投稿开启后，会议会议投稿截止时间之前进行投稿。会议主席可以手动设置会议开启投稿状态。

1.3 一个会议的主席不能对本会议进行投稿。

2. 用户能够方便快捷地找到投稿的会议。

3. 用户进入一个会议后，能够方便地浏览在本会议中已投稿的论文列表。

3.1 已投稿的论文列表显示每篇论文的论文 ID、名称、投稿时间、最后更新时间、当前状态。

3.2 点击论文标题后进入论文详细信息页，展示论文题目（不超过 50 个字）、摘要（不超过 300 个字）、作者列表、论文全文信息。

3.3 在论文详细信息页上可以进行论文信息更新。

4. 用户投稿时需要提供所需要的相关信息。

4.1 需要提供论文题目（不超过 50 个字）、摘要（不超过 300 个字）、作者列表、论文全文。

4.2 用户可以方便地添加作者信息

4.2.1 论文允许有多个作者，投稿人必须是作者之一，其他作者可以不是系统注册用户，但投稿时需提供姓名、单位、国籍和邮箱。

4.2.2 论文的作者需要按照论文署名顺序进行输入和显示。

4.2.3 除用户本人外，每个论文作者输入邮箱后自动判断是否系统注册用户，如果是则自动填充其他信息，如果不是则由用户填写其他信息。

4.3 论文全文必须以 PDF 格式上传，文件大小不超过 5M。

4.4 论文投稿后在会议截稿时间之前，用户可以随时进行信息的更新或撤稿。

4.5.1 用户可以更新投稿信息，包括论文题目（不超过 50 个字）、摘要（不超过 300 个字）、作者列表、论文全文。

5. 系统应当对于用户投稿信息进行妥善的保存和管理。

5.1 系统为每个投稿分配唯一的 ID。

5.2 系统应当确保所提交的投稿信息（包括论文原文）进行适当的存储，确保不会丢失。

F. 审稿

1. Chair 开启审稿

1.1 Chair 可点击开启审稿按钮，选择按 topic 相关度分配稿件或按审稿平均负担分配稿件。

1.1.1 按 topic 相关度分配。在所有与该稿件 topic 相关的 pc 中随机分配。

1.1.1.1 若与该稿件 topic 相关的 pc 小于 3 人，则在会议所有的 pc 中随机分配。

1.1.2 按审稿平均负担分配。在会议所有的 pc 中随机分配。

1.1.3 Paper 的作者和投稿人不能审该稿。

1.1.4 若审稿的 pc 小于 3 人则提示分配失败。

1.2 提示开启审稿是否成功。

1.3 Chair 可以查看稿件分配详情，包括稿件信息和 Pc 信息。

2. Pc 可以 pc 身份进入会议进行审稿。

2.1 Pc 可查看分配到的稿件详情。

2.1.1 Pc 可在线预览稿件 pdf。

2.1.2 Pc 可下载稿件 pdf。

2.2 Pc 需对分配到的稿件进行评分。

2.2.1 Pc 需提供稿件的 score、comment 和 confidence。

2.2.2 Pc 可查看已提交的分数详情。

3. Chair 结束审稿。

3.1 所有 Pc 提交分数后，Chair 才可以结束审稿。

3.2 Chair 可以查看稿件的分数。

3.3 结束审稿的同时为每篇稿件开启讨论贴。

G. 讨论

1. 可以参与讨论的人员。

1.1 Chair 可以参与该会议中所有稿件的讨论。

- 1.2 每篇稿件的 Pc 可以参与该稿件的讨论。
 2. 讨论内容。
 - 2.1 参与讨论的人员可以发布回贴、回复、及评论回复。
 - 2.2 讨论贴中要显示讨论的具体内容、发表者、发表时间。
 - 2.3 讨论贴中的内容需要按照发表时间排序。
 3. 讨论必须进行。
 - 3.1 所有讨论参与者必须在讨论贴中发表至少一次言论。
 4. Pc 可以根据讨论的内容修改或确认评分。
- H. Rebuttal
1. 只有一轮没有被录用的稿件才可以 rebuttal。
 2. Author 可以确认分数或选择 rebuttal。
 - 2.1 若确认分数，则该稿件将不会被录用。
 - 2.2 若选择 rebuttal，需要提供 rebuttal 信息。
 - 2.2.1 提交 rebuttal 信息后 author 可在稿件详情中查看到。
 - 2.3 系统需提示操作是否成功。
 3. Rebuttal 的稿件必须经过二轮讨论。
- I. 发布评审结果
1. 只有 chair 可以发布评审结果。
 2. 发布一轮评审结果。
 - 2.1 只有在所有 pc 都确认分数后才能发布。
 - 2.2 三个 pc 的分数中只要有一个小于 0，则该稿件不被录用。
 - 2.3 发布评审结果后，author 可在稿件详情中查看分数。
 3. 发布最终评审结果。
 - 3.1 只有在所有 pc 都确认 rebuttal 稿件的分数后才能发布。
 - 3.2 三个 pc 的分数中只要有一个小于 0，则该稿件不被录用。
 - 3.3 发布评审结果后，author 可在稿件详情中查看分数。

四、 非功能性需求

1. 运行环境
 - 1) 支持在 Windows、Linux 系统上运行。

2) 采用 H2Database.

2. 响应时间

- 1) 在 95%的情况下，一般时段响应时间不超过 1.5 秒，高峰时段不超过 4 秒。
- 2) 定位系统从点击到第一个界面显示出来所需要的时间不超过 1 秒。
- 3) 登录响应时间在 2 秒内，刷新栏目响应时间在 2 秒内，刷新条目分页列表响应时间 2 秒内，打开信息条目响应时间 1 秒内。

3. 安全性

- 1) 严格权限访问控制，用户在经过身份认证后，只能访问其权限范围内的数据，只能进行其权限范围内的操作。
- 2) 不同的用户具有不同的身份和权限，需要在用户身份真实可信的前提下，提供可信的授权管理服务，保护数据不被非法/越权访问和篡改，要确保数据的机密性和完整性。
- 3) 提供运行日志管理及安全审计功能，可追踪系统的历史使用情况。
- 4) 对表单等输入的参数进行检查，防止恶意攻击。

4. 可靠性

- 1) 对输入有提示，数据有检查，防止数据异常。
- 2) 系统健壮性强，应该能处理系统运行过程中出现的各种异常情况，如：人为操作错误、输入非法数据、硬件设备失败等，系统应该能正确的处理，恰当的回避。
- 3) 因软件系统的失效而造成不能完成业务的概率要小于 5%。
- 4) 系统缺陷率每 1,000 小时最多发生 1 次故障。

5. 易用性

- 1) 用户友好的界面。
- 2) 系统易操作。

6. 可测试性

- 1) 华为云的代码检查工具，对前后端的代码进行质量检查，使用代码检查工具，并根据质量评估结果，对代码问题，圈复杂度，代码重复率进行修改。最后要保证代码问题数为‘0’全部解决，质量门禁显示‘passed’，而圈复杂度和代码重复率尽可能低。
- 2) SonarQube 代码检测，进一步解决代码中 Bad Smell 的问题

3) 使用 junit 进行单元测试，并保证测试覆盖率不低于 80%，

7. 可维护性

1) 90%的 BUG 修改时间不超过 1 天，其他不超过 2 天。

2) 代码的圈复杂度必须在 10 以内。

3) 任何对象的任何方法都不允许超过 200 行代码。

4) 安装新版本必须保持所有的数据库内容和所有个人设置不变。