



# Reputation System For System of Collaborative Community

Shubhendra Pal Singhal - NIT Trichy  
Kaushik Sridhar - Osmania University  
Punit Agrawal - NIT Patna

# Acknowledgement

We would like to extend our sincere thanks PBL team - Rajat , Miraj , Anshuman , Amey for testing our system.

We are highly indebted to Prof. D.B. Phatak , Nagesh Karmali and Abhijeet and Nitin for their guidance and constant supervision as well as for providing necessary information regarding the project also for their support in completing the project.

Our thanks and appreciations also go to all the team members in developing the project and people who have willingly helped us out with their abilities.

# Introduction

- Presents the actual statistics about the active participation of the user<sup>[1]</sup>.
- Automates the role selection.
- The reputation factor increases with level of contribution only.

# Technologies used

- Django-1.11.7 LTS
- Docker
- MySQL

# System of Collaboration Community

- Community: Request Creation, Subscribe, Unsubscribe, Manage
- Group(Public/Private): Create, Subscribe, Unsubscribe, Manage, Invite
- Roles (Community/Group based): Author, Publisher, Admin
- Resources (Basic Article, Course): Create and contribute to others in Community
- Revisions of Article Forum: For every community
- Workflow: Publishing process for resources like Article (Different for community and group)
- Dashboard: Users, Community, and Groups

# Motivation

- Changing the system to “**DECENTRALISED**” Reputation system.

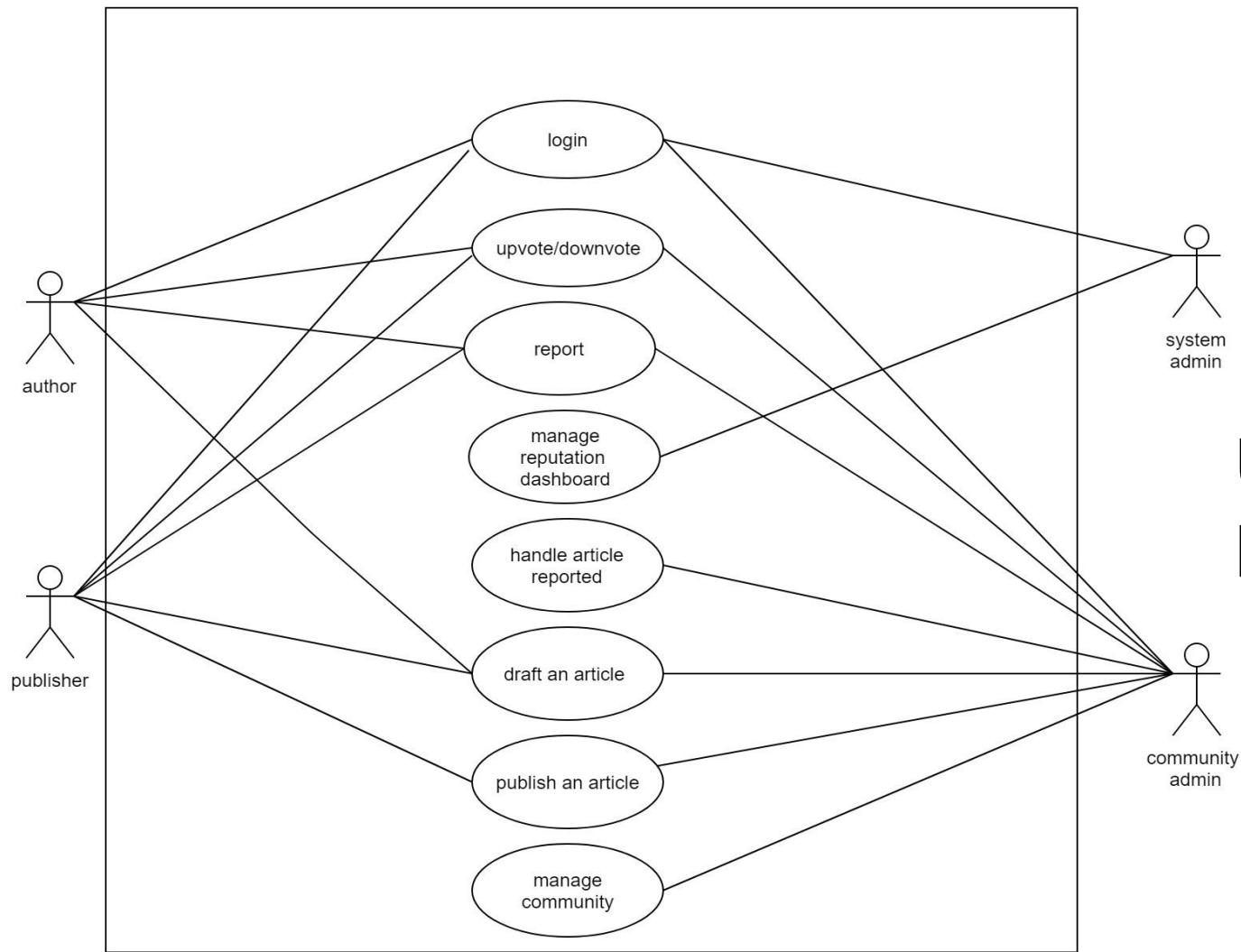
There is no accurate measure ,rather than the trust of super-admin, that can promote the users to the next level - publisher and admin.

- Requirement of an automation.
- Increase in reputation just on the basis of one factor - "Contribution".
- The level of contribution can be reflected by :

The count of votes.

Articles edited, reported and published.

Creation of community/group.



# Use Case Diagram

# Grading System

The values given below can be dynamically changed by using the form created.

**The initial values are given below :**

Creating community (approved by the superadmin):+25

Article published : +5 to the user of last edit and to others and +5 to publisher.

Article Reported(Approved) : -5 to the user of that version and publisher gets -5.

Article Reported(Rejected) : -5 for the user who reported.

Article/comment upvote/downvote: +/-2



# Badge System

Community Author 0 to 2000

Community Publisher 2001 onwards.

- 5 color scheme representing different levels of contribution.
- Badges have divided for publishers and author.
- Contribution means drafting and editing.

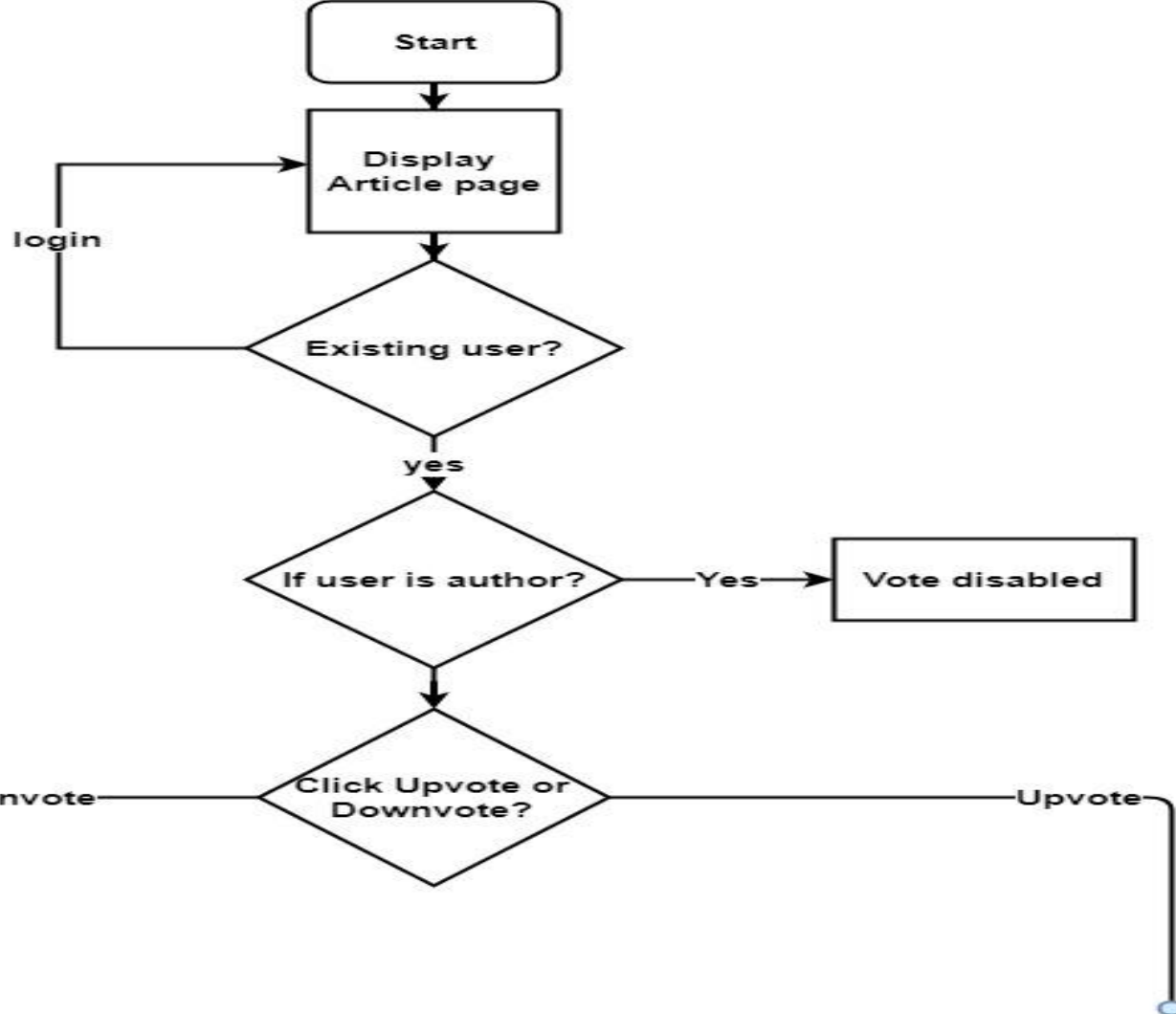
# Badge System

Activity	Badges				
	1 to 5	6 to 15	16 to 30	31 to 50	Above 50
	Author / Publisher				
Articles Contributed	AC	AC	AC	AC	AC
My Articles Published	AP	AP	AP	AP	AP
	Publisher				
Articles Revised by me	RA	RA	RA	RA	RA
Articles Published by me	PA	PA	PA	PA	PA

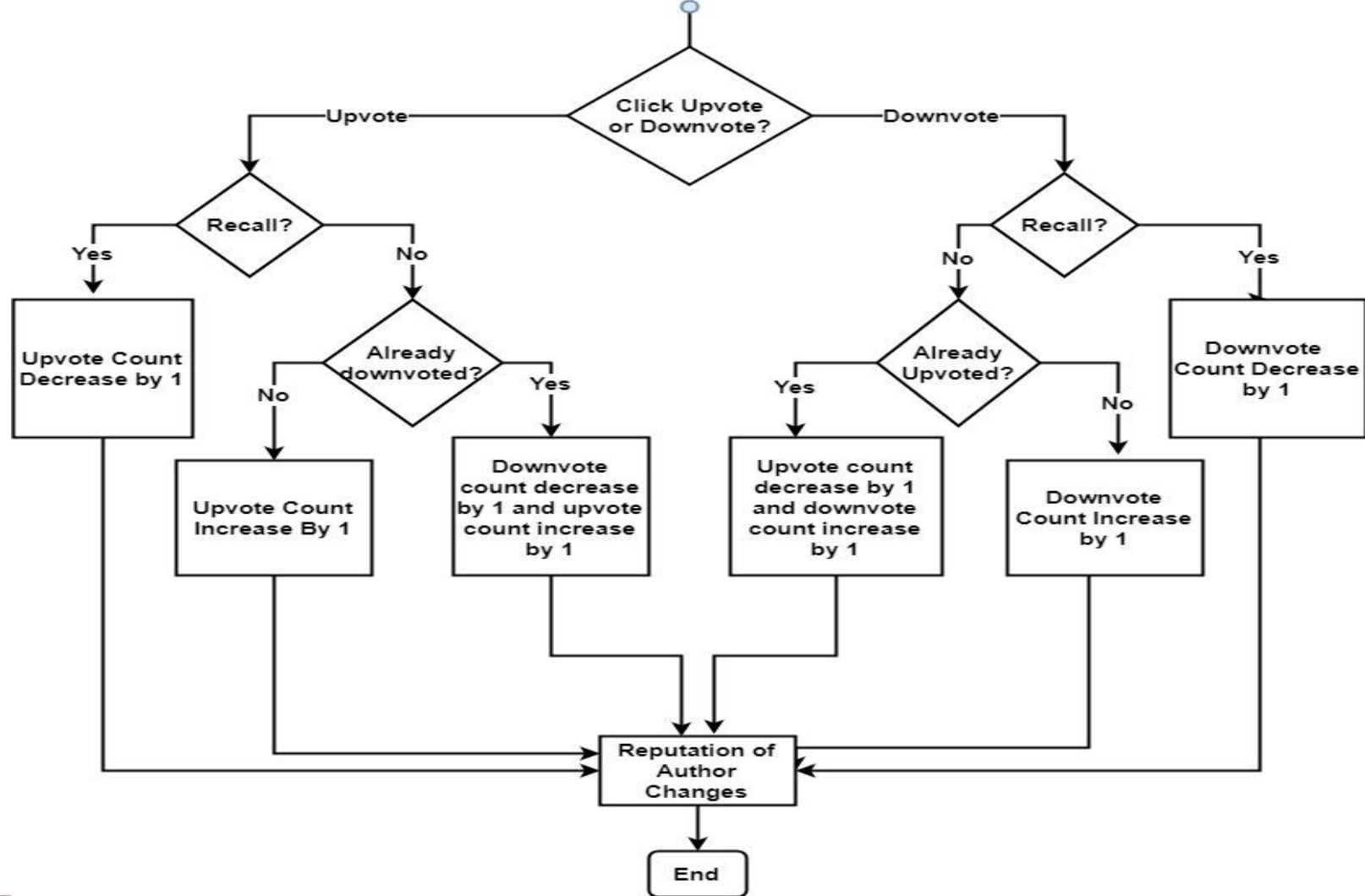
# What changes the reputation?

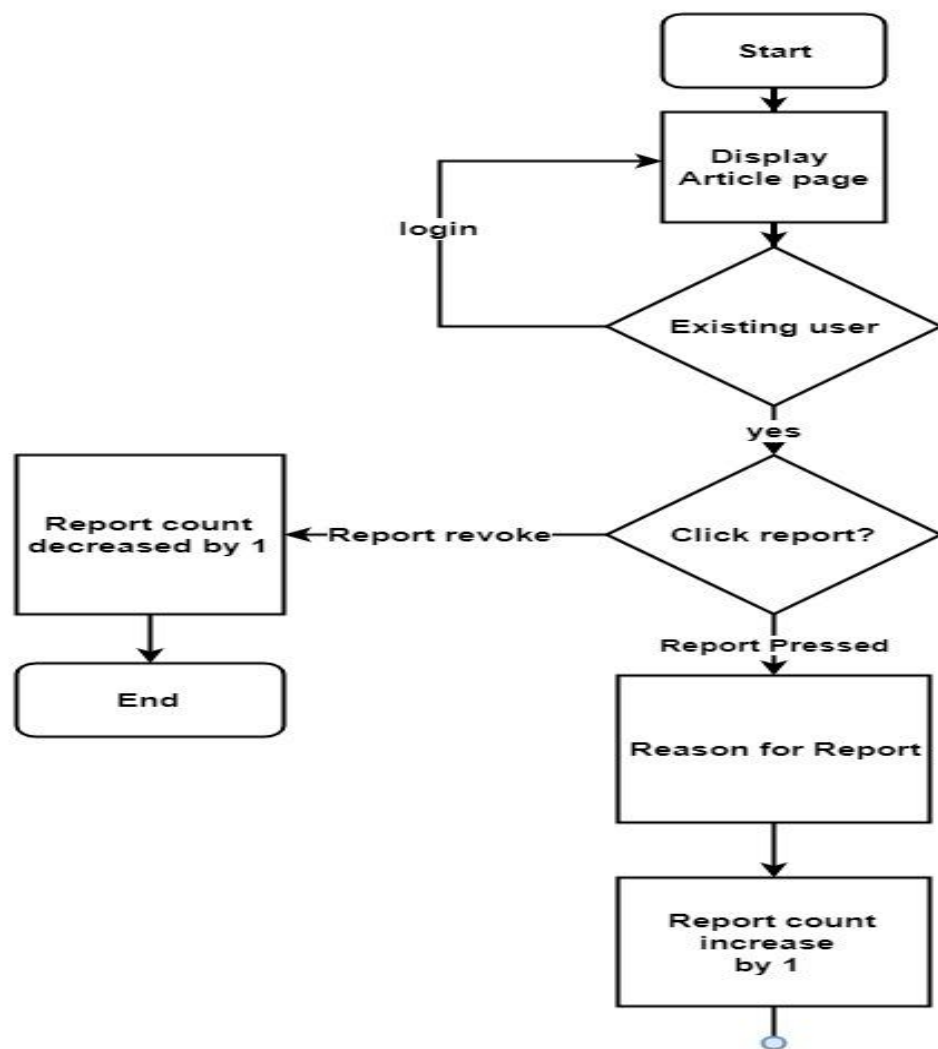
- Person Joining the community.
- Person voting the Article
- Person editing the article
- Person reporting the article
- Person changing the article to published state

# Mechanism of Subsystems

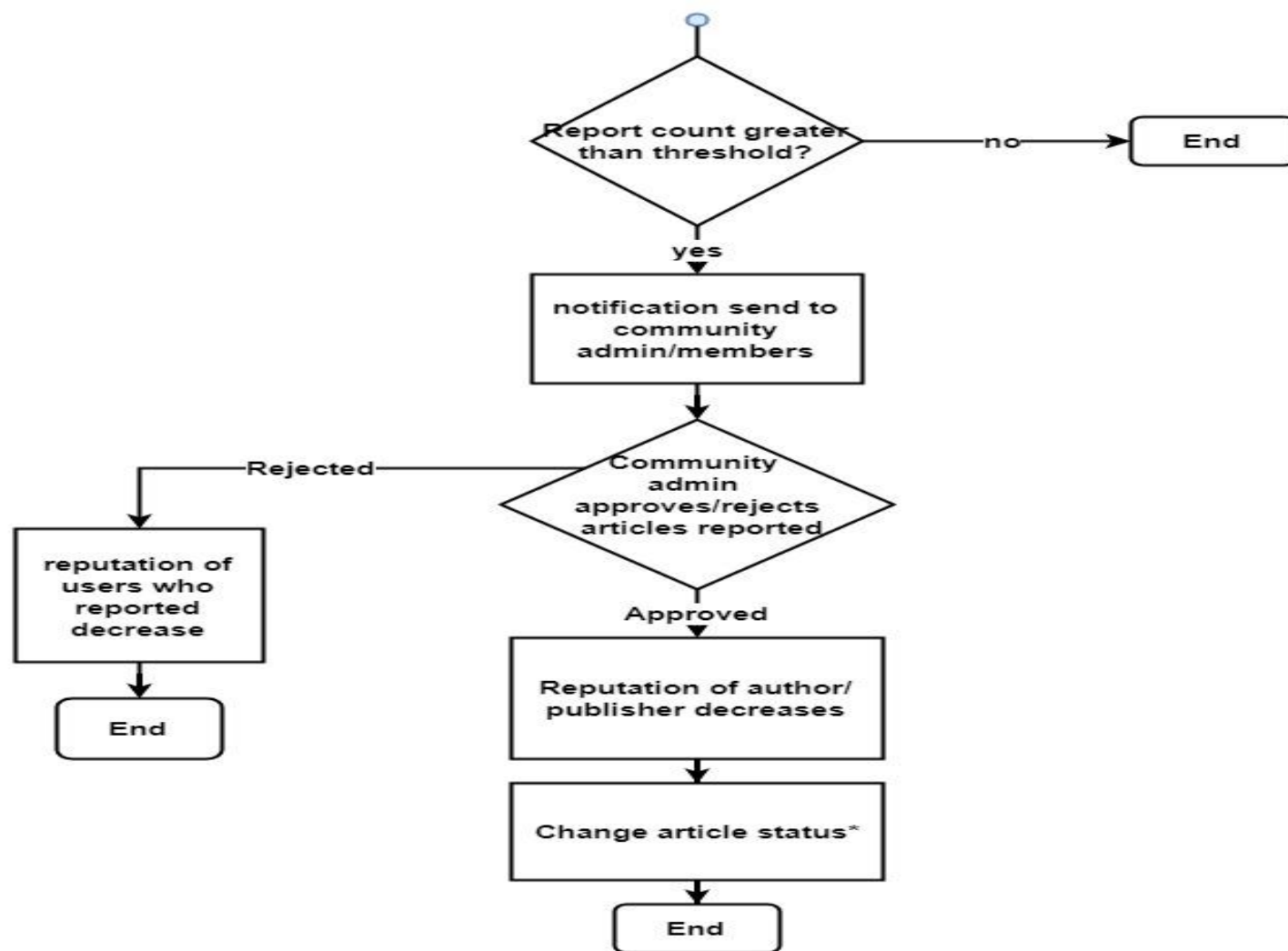


# Voting Mechanism

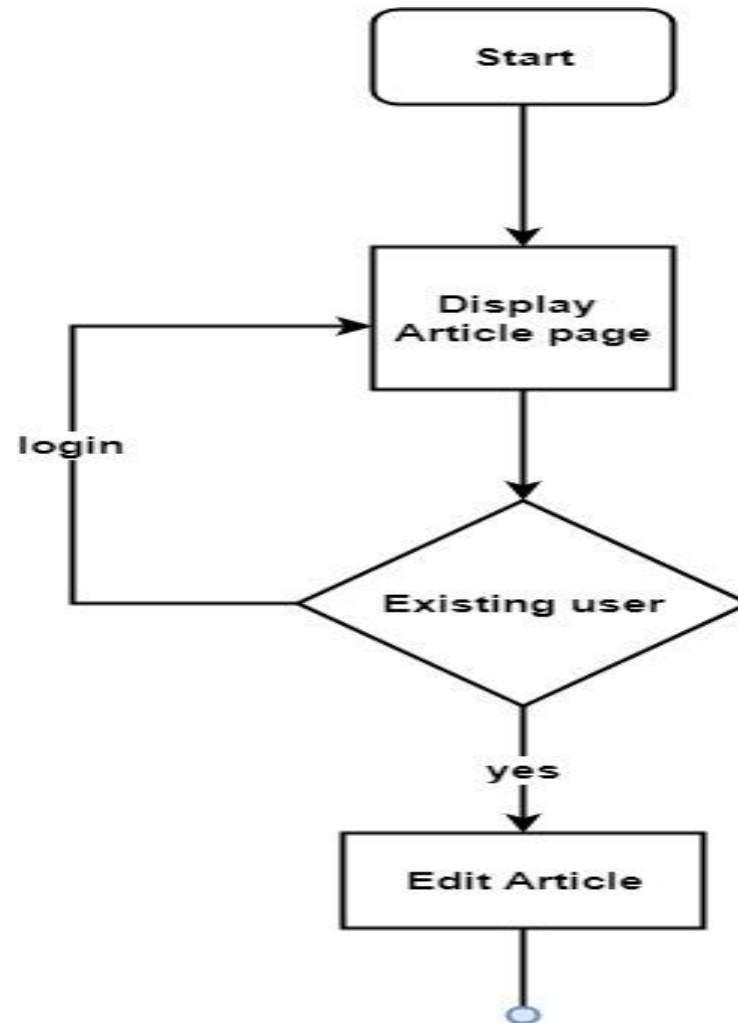




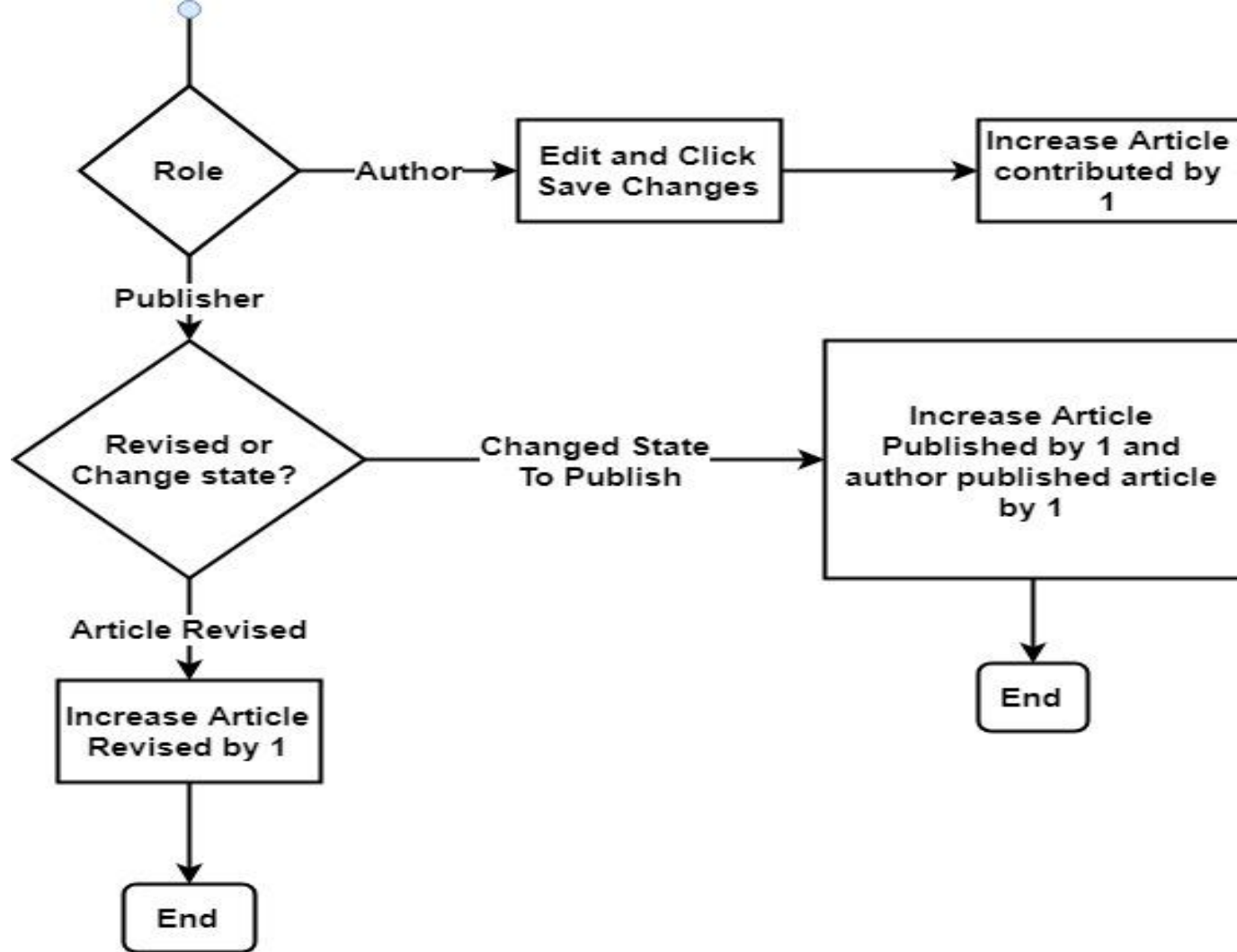
# Reporting Mechanism







# Badge System



# The implementation of Reputation System

The link to implementation using **Docker** :

<https://github.com/fresearchgroup/Collaboration-System/tree/reputation>

The link to the Documentation :

<https://github.com/singhalshubh/Reputation-system-Documentation-and-Research>

# Extra work - Selenium Testing

The selenium Docker was installed for Testing.

The link to Selenium Testing and Report including “How to perform a test case” is included.

<https://github.com/singhalshubh/Notification-system-Testing-using-selenium>

# Selenium Testing

Testing of Reputation System :

- Upvote and downvote
- Reputation change
- Badge change
- Reporting the article

# Limitations

- Why all contributors who helped in editing should get equal points?
- How to avoid unnecessary edits by users?
- How to ensure that admins don't remove each other every time?
- Develop model which understands user behaviour.
- Economic Model of reputation system?<sup>[4]</sup>

# Improvements in Future

# The implementation

- <https://github.com/singhalshubh/Reputation-system-Documentation-and-Research>

The implementation with detailed analysis is there in the link for more reference.



## Why all contributors who helped in editing should get equal points?

The system consists of various versions of article. Our basic aim to select those versions which actually improvised the article in a better fashion. Let us consider  $x_1, x_2, \dots, x_n$  as the number of versions after the version of first draft as  $x_0$ .

Consider  $u_k$  : upvote of the  $k$ th version And  $d_k$  : downvote of the  $k$ th version.

Let the selected versions be denoted by  $s_i$ 's.  $U()$  is upvote and  $D()$  is the downvote.

If the factor of  $U(x_i - s_{i-1})/D(s_{i-1} - x_i)$  is positive then this means that there is a substantial increase in upvotes or decrease in downvotes. So we select only those versions which have the positive value.

Let the set  $S$  denote the selected version of article for point allocation including  $x_0$ .

STEP 1 :  $k=0, j=0$

STEP 2 : For  $i = 1$  to  $n$ :

STEP 3 : if(  $U(x_i > S(s_k))$  ) if(  $U(x_i - S(s_k))/D(S(s_k) - x_i) > 0$  ):

STEP 4 :  $S = S \cup x_i$

STEP 5 :  $k=k+1$

STEP 6 : else:

STEP 7 :  $i=i+2$

STEP 8 : Endloop

## Point Allocation for version of article

Now the ratio of  $(u_j+d_j)$ :Number of views is compared and seen that if the ratio is close to one then it is a good article else the article is not worth because number of views are way higher than  $u_j+d_j$ .

Obviously  $u_j+d_j$  can never be  $>$  Number of views. If this is true then something is wrong in code.

So according to the tested statistics the allocation is : We maintain a bank where the number of versions selected as  $s_i$ 's will each have +5 point so bank has  $5n$  points if  $n$  are selected.

Case 1: So those articles which have ratio close to one will have 70% of  $5n$ .

Case 2: Publisher will have 20 % of  $5n$ .

Case 3: And rest goes to remaining versions of article.

Now further if there are two or more articles in every case then equal distribution will be followed.

## How to avoid unnecessary edits by users?

- In the system , the "save-changes" button after editing ,is clickable even if the user has not edited anything.
- So the aim is to improve it.
- This problem is being addressed in the NLP research where they are trying to interpret the meaning of the article through certain algorithms and then identifying whether, the article makes any sense or not after or before editing.

# User Participation

- This field has been widely studied under the topic “Game Theory”.
- Studying the “Nash Equilibrium” of the system can solve this issue.
- Reputation system can be considered a game where all users are trying to increase their payoff i.e. reputation factor.

# Conclusion

- Decentralization of power.
- Grading and Badge system.
- Implementation of system and community reputation.

# References

[1] Dana Movshovitz-Attias, Yair Movshovitz-Attias, Peter Steenkiste, Christos Faloutsos, "Analysis of the Reputation System and User Contributions on a Question Answering Website: StackOverflow", 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.

[2] Documentation of Collaboration Community, GNU Licensed.

[https://docs.google.com/document/d/1ENp\\\_UAspLg\\\_c291cJvQxqlC1bnbyjxelaoxCdI4xqes](https://docs.google.com/document/d/1ENp\_UAspLg\_c291cJvQxqlC1bnbyjxelaoxCdI4xqes).

[3] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, Jure Leskovec, "Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow", Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012.

[4] Andrew Marder, "Stack Overflow Badges and User Behavior: An Econometric Approach", Proceedings of the 12th Working Conference on Mining Software, 2015.

# References

[5] Stack overflow reputation and moderation which describes about the various scenarios of voting, User contributions licensed under cc by-sa 3.0 with attribution required. rev 2018.6.11.30708, <https://stackoverflow.com/help/whats-reputation>.

[6] M Gupta, P Judge, M Ammar - ... of the 13th international workshop, "A reputation system for peer-to-peer networks", 2003- dl.acm.org.

[7] C. Boyd A. Jøsang, R. Ismail. A survey of trust and reputation systems for online service provision. Decision Support Systems, 43:618–644, 2007.

[8] B.T. Adler, K. Chatterjee, L. de Alfaro, M. Faella, I. Pye, and V. Raman. Assigning trust to Wikipedia content. In Proc. of WikiSym 08: International Symposium on Wikis. ACM Press, 2008.

Questions??



THANK YOU