2.4 Ranking

2.4.1 *Overview*

This section details an overview of the ranking scheme developed to enhance the user experience and ensure visibility to the most recognized papers within MyOpenJournal. In the following sections, the replacement and refinement of the current ranking system is explained.

2.4.2 *Replacement of Previous System*

The ranking system that was put in place in the last effort would not work as is with the updated Java architecture. The ranking system that was previously constructed gave a certain weight to every object of the site and calculated a ranking for every individual object. It was deemed that this system would be too taxing on the system and would be beyond the scope of the group to complete along with the other aspects of the project that needed to be completed. Another reason that the ranking system had to be changed is to give a fair chance to all users. If users have ratings and their papers are displayed based on this rating, newer users would never be able to have their papers viewed by others because they would be buried.

2.4.3 *Goals of Ranking*

The ranking algorithm idea that is going to be implemented in the new version of MyOpenJournal uses up and down votes for users to rate papers and reviews. When a new paper is uploaded, it will appear at the top of the website. Over time, its rating will be determined by how long the paper has been up on the site combined with the net score (up/down votes) that it has received over that time period. This allows all users to have a fair chance at their paper being seen and rated.

2.4.4 *Refinement*

Knowing that our rankings would be determined solely on the up-vote count, the down-vote count, and the time since the paper had been uploaded to MyOpenJournal, we set out to make an efficient, simple, and functional ranking system.

Our first few prototype ranking systems calculated the proportion of up-votes to down-votes and then divided this value by some function of the number of days since the paper had been uploaded. The problem with this approach is that the older papers would sometimes have higher priority than the newer papers. Also, as papers got older, their scores would keep decreasing, so even really good but old papers would essentially become invisible after enough time passed.

After reevaluating these prototypes, we made an algorithm that would still find the proportion of up-votes to down-votes, but it would add some constant divided by the number of days to this value. This method fixed the problem of older papers dropping off, since after many days had passed the additional term would essential become 0 (the time this would take would be affected by the size of the constant). We still were not satisfied with this method since we realized that the proportion of up-votes to down-votes could give us highly varying values.

Our current iteration of the ranking system uses the proportion of up-votes to total votes and then adds a constant divided by the number of days. We like this method so far since it can be scaled nicely to a value between 0 and 1. New papers will be shown at the top and (with the constant we have chosen) the regency of the paper won't be factor past roughly 30 days of its initial upload. One factor our ranking doesn't take into account is papers with many votes will have as much rating as papers with few votes, but the same proportion of votes. This may be troublesome, since controversial, well-known papers deserve more attention than obscure ones, but this all depends on how users decide to vote on the papers.

$$\frac{upvotes + 1}{upvotes + downvotes + 2} + \frac{9}{days + 1}$$

(value between 0 and 1) (value between 0 and 9)

2.4.5 *Test Data*

Paper	up	down	n	(up/down)/(2^n)	(up/down)/(n^2+1)	(up/down)+(600/(n+1))	(up+1)/(up+down+2)+(9/(n+1))	<- scaled
1	120	66	0	1.806	1.806	601.806	9.644	0.964
2	0	0	0	1.000	1.000	601.000	9.500	0.950
3	3	4	0	0.800	0.800	600.800	9.444	0.944
4	2	1254	0	0.002	0.002	600.002	9.002	0.900
5	2234	85	1	12.994	12.994	325.988	5.463	0.546
6	1534	23	7	0.500	1.279	138.958	2.110	0.211
7	256	758	4	0.021	0.020	120.339	2.053	0.205
8	5000	4999	9	0.002	0.012	61.000	1.400	0.140
9	15000	15001	9	0.002	0.012	61.000	1.400	0.140
10	5563	334	42	0.000	0.009	30.562	1.153	0.115
11	2342	32375	18	0.000	0.000	31.651	0.541	0.054

n = # of Days since Upload

up = # of Upvotes

down = # of Downvotes