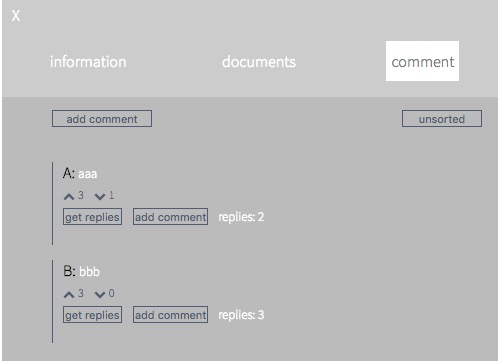
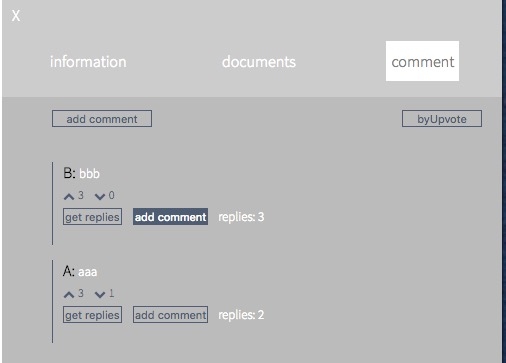
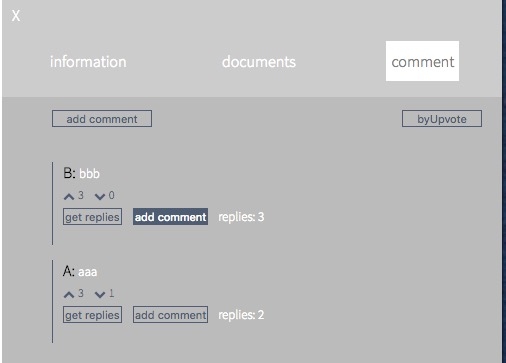
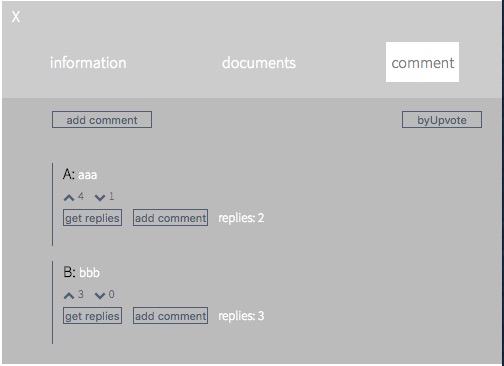
1. Test on: Soring by Upvote when having same upvote
   1. Action: switch the status from unsorted to byUpvote
   2. Expected Result: Commment B will move to top
   3. Explanation: since A and B has same number of upvote, the order will be decided by downvote. Whoever has less downvote will be ordered prior.





1. Test on: Soring by Upvote when having different upvote
   1. Action: upvote comment A
   2. Expected Result: Commment A will move to top
   3. Explanation: since A have higher upvotes than comment B, it should move top





1. Test on: Soring by Replies number
   1. Action: switch sorting method from byUpvote to byReplies
   2. Expected Result: Commment B will move to top, replies in commentB will also reordered according to the number of replies they have
   3. Explanation: comment with higher replies will move to the top no matter it’s a direct comment of the project or replies of another comment





1. Test on: sorting by replies number when add new comment
   1. Action: add one comment to B and then add another comment to B
   2. Expected Result: comment A will move to top when comment add to B first, then comment B will move to top when another comment add to B
   3. Explanation: At the beginning, B has more commments than A, so B is placed top. When only add one reply to A, A will become the most most newly-updated comment, so it will move to top for user’s better reference. After adding another comment to B, B will has more comments than A, so B will move back to top again



