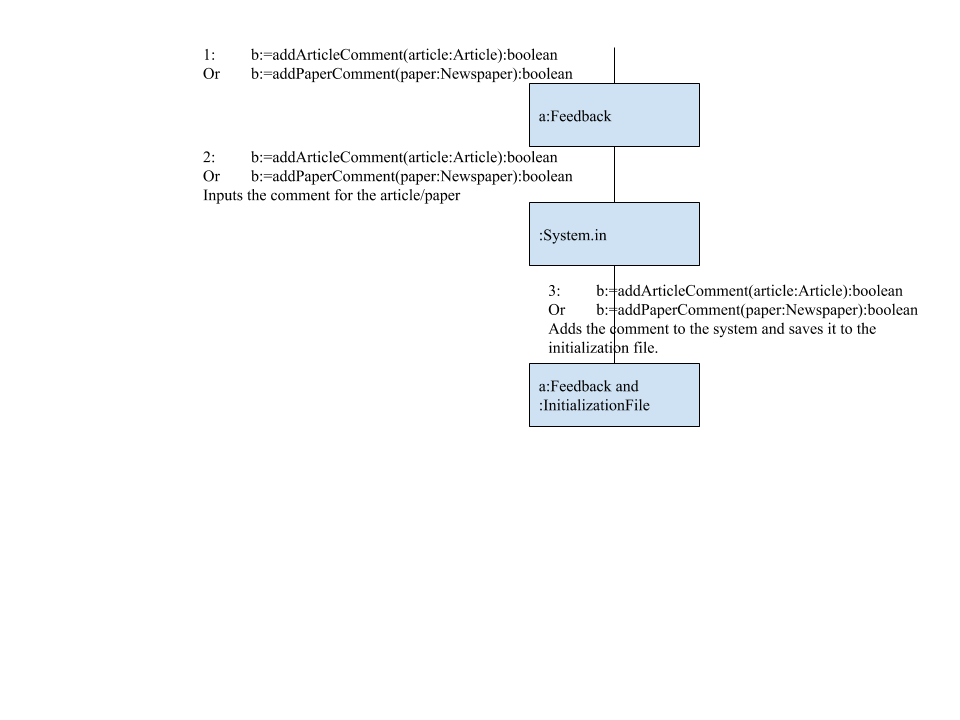
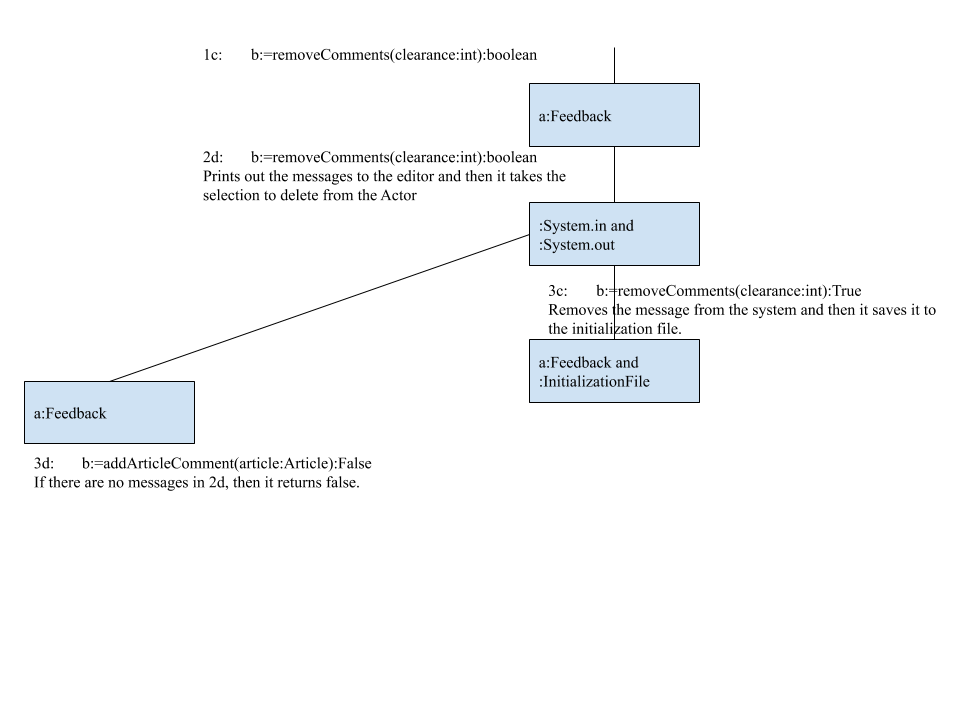
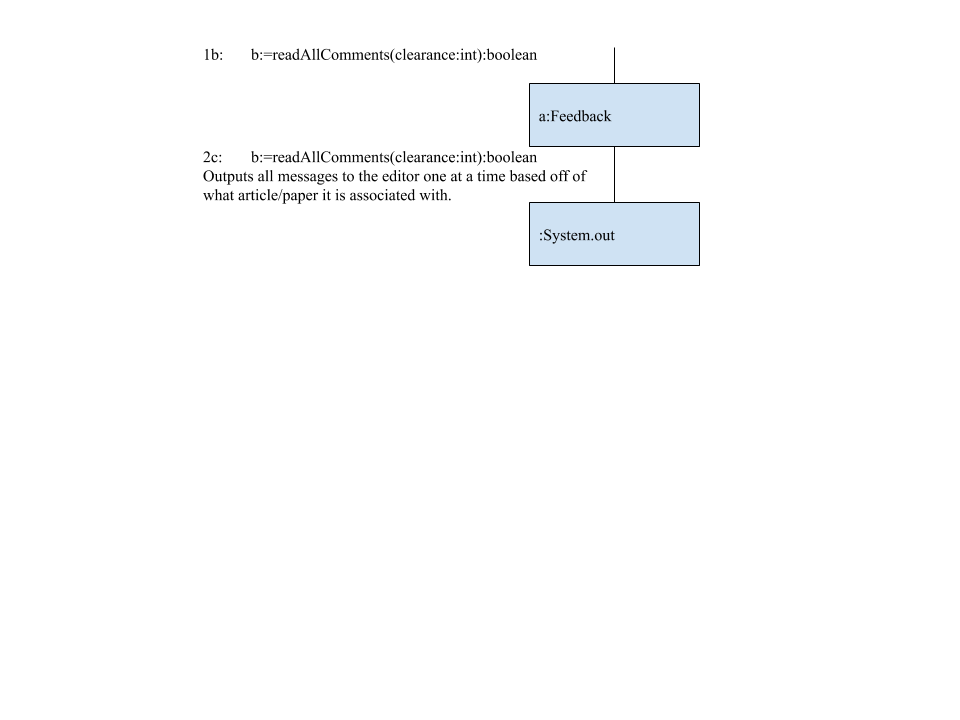
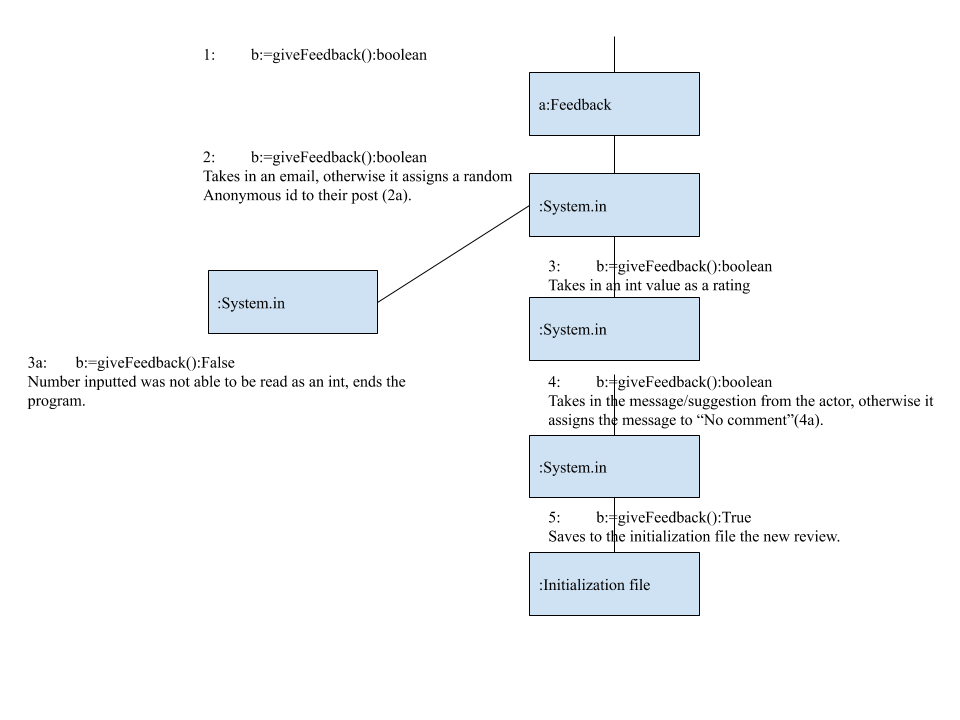
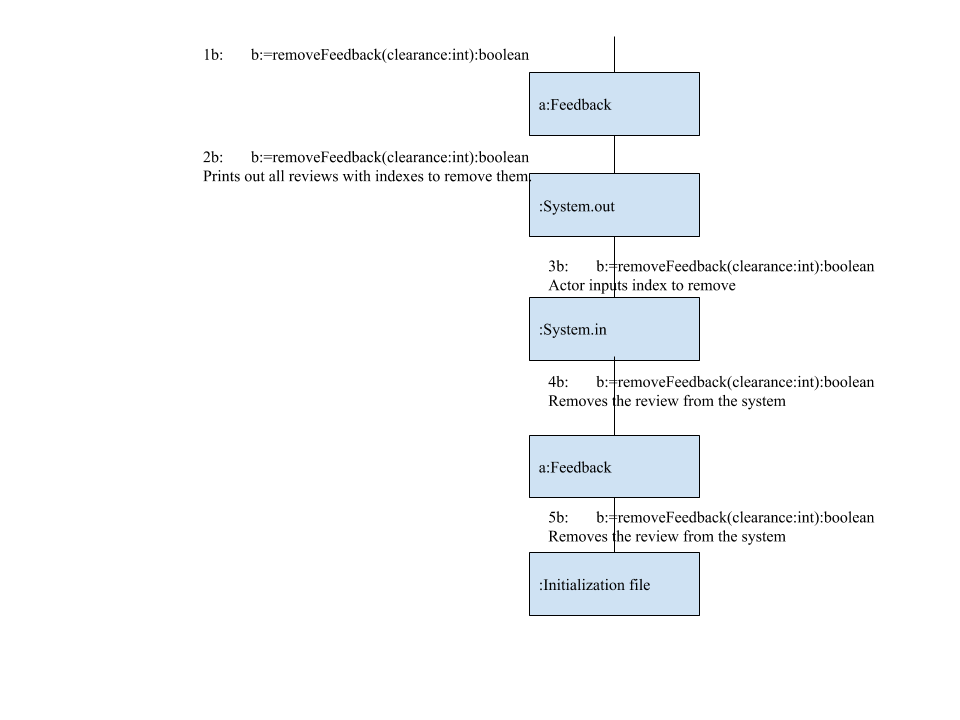
|  |  |
| --- | --- |
| Use Case: | Add/read/remove messages to the editor for articles/newspapers. |
| Actor: | Customer, Employees, Managers, Owners |
| Precondition: | The database exists how it was initially designed, and the actor has clearance to do the command. |
| Main Success  Scenario: | 1: The Actor chooses to add a message to the editor on an article/newspaper after reading it.  2: The Actor gives them a message line by line in the console, however many lines it needs.  3: The Actor enters q on the last line when they are done, and hits enter to quit. |
| Extensions: | 1a: The Actor chooses to not give them a message.  2a: Does not happen if 1a occurs.  3a: Does not happen if 1a occurs.  1b: The Actor chooses to read messages to the editor.  2c: If 1b occurs, then the program prints out all messages to the editor in the console.  3b: If 1b and 2c occur, then the program does not have a third step.  1c: The actor chooses to remove a message to the editor.  2d: If 1c occurs, then it prints out the identifier for the message (the name of the article or volume/issue of the newspaper) along with an index, which the actor chooses from.  3c: If 1c and 2d occur, then it removes the message and saves the change to the initialization file.  3d: If 2d shows that there are no messages, then it does not allow any selections or saves. |

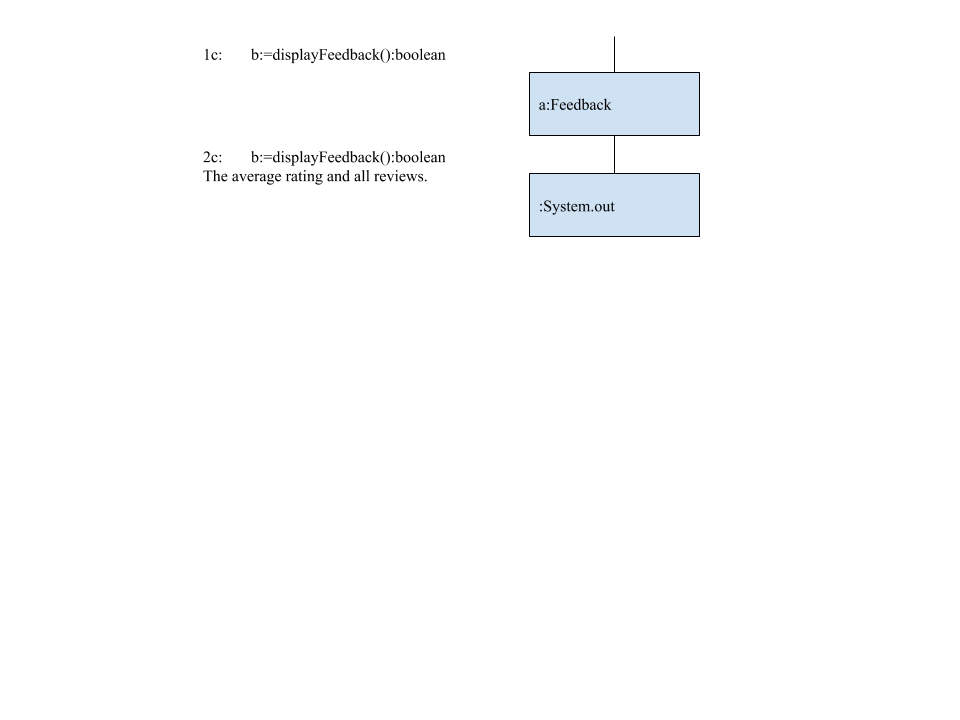




|  |  |
| --- | --- |
| Use Case: | Add, remove, or read the feedback for the company |
| Actor: | Customer, Employees, Managers, Owners |
| Precondition: | The database exists how it was initially designed, and the actor has clearance to do the command. |
| Main Success  Scenario: | 1: The Actor chooses to add feedback to the company.  2: The Actor chooses to give the program an email.  3: The Actor gives an integer rating.  4: The Actor gives a suggestion/message along with all this to be part of the review.  5: The program saves this all to the initialization file and immediately makes it available. |
| Extensions: | 1a: The Actor chooses to not give feedback, so none of the following steps will happen.  2a: If the Actor chooses to not give an email, it says the email is Anonymous with a random number after it.  3a: If the number inputted can not be read as an integer, it kicks them out.  4a: If they choose to not give a suggestion/message, it just says no comment.  5a: If 3a happens, then this step does not happen.  1b: If the Actor chooses to remove a review.  2b: It lists out all reviews with indexes.  3b: The Actor chooses an index to get removed.  4b: The program removes it from immediate access.  5b: After leaving, the program saves all changes to the file.  1c: The Actor chooses to read all reviews.  2c: The program prints out all reviews along with an average rating. |







Class Diagrams:

|  |
| --- |
| Feedback |
|  |
| b:=removeFeedback(clearance:int):boolean  b:=giveFeedback():boolean  b:=displayFeedback():boolean |

|  |
| --- |
| DMEditor |
| Has access to the database, article abstract class, and newspaper class because it directly manipulates a file in the database and is supposed to comment on specific articles/newspapers. |
| b:=addArticleComment(article:Article):boolean  b:=addPaperComment(paper:Newspaper):boolean  b:=readAllComments(clearance:int):boolean  b:=removeComments(clearance:int):boolean |

Justification for Feedback class: Only has access to the database, because it has no need to see any other part of the system. It needs access to the database to save all reviews to a file so the reviews are consistently present. The class should exist on its own because unlike the messages to the editor class (DMEditor), it has no need to see articles or newspapers.

Justification for DMEditor class: Has access to the database, article abstract class, and newspaper class because it directly manipulates a file in the database and is supposed to comment on specific articles/newspapers. This should exist as its own class due to the fact that it requires access to Article and Newspaper, and should have access to no other classes other than that. While its methods could be implemented in Article and Newspaper, the code would be extremely similar in the two, resulting in extra code. DMEditor exists to cut down on this duplicate code.