**Use Case Descriptions**

*1. Create an Account*

*Abstract:*A student or professor uses this use case to create their account on JMU-Share.

*Precondition:* Student or professor has a valid JMU email.

*Postcondition:* The student or professor has an account on JMU-Share.

*Steps:*

1. Student or professor chooses to create an account on JMU-Share

2. System requests a valid JMU e-mail to use as the account username

3. Student or professor provides an email.

4. System sends confirmation email.

5. Student or professor clicks link in confirmation email.

6. Account is created.

*Alternatives:*

3a1. Student or professor provides an invalid email

3a2. System returns to step 2

*2. Search Through Posts By Class*

*Abstract:* A student uses this use case to search through note postings by a specific class

*Precondition:* Student is logged into JMU-Share

*Postcondition:* Student views all results for their search

*Steps:*

1. Student selects a class from the options and requests search results.

2. System returns any postings matching the searched class.

*3. Search Through Posts By Professor*

*Abstract:* A student uses this use case to search through note postings by a specific professor.

*Precondition:* Student is logged into JMU-Share.

*Postcondition:* Student views all results for their search.

*Steps:*

1. Student enters a professor’s name and requests search results.

2. System returns any postings matching the searched professor.

*4. Search Through Posts By Post Author*

*Abstract:* A student uses this use case to search through note postings posted by a specific author

*Precondition:* Student is logged into JMU-Share.

*Postcondition:* Student views all results for their search.

*Steps:*

1. Student enters an author’s name and requests search results.

2. System returns any postings matching the search author.

*5. Add a Post*

*Abstract:* A student uses this use case post a note to the system

*Precondition:* Student is logged into JMU-Share.

*Postcondition:* Student adds their note into the system.

*Steps:*

1. Student enters all relevant information into the boxes provided.

2. System creates the posting with the students information.

*Alternatives:*

1a1. Student chooses an image file to upload

*6. Delete a Post(student)*

*Abstract:* A student uses this use case to delete a note from the system

*Precondition:* Student is logged into JMU-Share.

*Postcondition:* Student’s note is removed from the system.

*Steps:*

1. Student chooses a note that they uploaded to the system.

2. System asks for confirmation.

3. System removes the note.

*7. Delete a Post(administrator)*

*Abstract:* An administrator uses this use case to delete a note from the system

*Precondition:* Administrator is logged into JMU-Share.

*Postcondition:* The note is removed from the system.

*Steps:*

1. The Administrator chooses a note that has been uploaded to the system.

2. System asks for confirmation.

3. System removes the note.

*8. Comment on pages*

*Abstract:* A student uses this use case to add a comment to a note

*Precondition:* Student is logged into JMU-Share.

*Postcondition:* Student’s comment is logged to the note.

*Steps:*

1. Student chooses a note that has been uploaded to the system.

2. Student provides the comment.

3. System adds the comment to the note selected.

*9. Rate Posts*

*Abstract:* A user uses this use case to rate a note.

*Precondition:* User is logged into JMU-Share.

*Postcondition:* User’s rating is stored in the system.

*Steps:*

1. Student chooses a note that has been uploaded to the system.

2. Student selects up-vote or down-vote.

3. System stores the choice.

*10. Crop Note Sections*

*11. Favorite Posts*

*Abstract:* A student uses this use case to delete a note from the system

*Precondition:* Student is logged into JMU-Share.

*Postcondition:* Student’s note is removed from the system.

*Steps:*

1. Student chooses a note that has been uploaded to the system.

2. System asks for confirmation.

3. System removes the note.