

Use Case Title	Join Newsletter	
Primary Actor	User	
Level	Full Description	
Stakeholders:	Publishers, Users	
Related use cases	<i>Read Content, Manage Notifications, Send Message</i>	
Precondition:	User is not already subscribed to the newsletter	
Postcondition	User is subscribed to the newsletter and starts receiving updates	
Trigger:	User thinks about subscribing to a publisher's newsletter	
Main Success Scenario:	Actor <ol style="list-style-type: none"> <li>1 User navigates to the subscription page</li> <li>2 User enters email and clicks "Subscribe"</li> <li>3 User is redirected to a confirmation page</li> <li>4 User clicks on the confirmation link in their email</li> </ol>	System <ol style="list-style-type: none"> <li>1.1 Displays subscription page with newsletter details</li> <li>2.1 Validate email format and check for availability</li> <li>3.1 Send a confirmation to the user</li> <li>4.1 The user is marked as subscribed and starts receiving the newsletter</li> </ol>
Exceptions	<ol style="list-style-type: none"> <li>1.1 If the Users email is in an invalid format, send an error message and ask them to add a valid one.</li> <li>2.1 If the email is already subscribed, display a message that says "You are already subscribed" to notify user and prevent further attempts</li> <li>3.1 If User does not click the confirmation link in their email, they will not be subscribed. Send a reminder after a set amount of time (between 24-48 hours)</li> </ol>	

Use Case Title	Read Content	
Primary Actor	User	
Level	Full Description	
Stakeholders:	Users, Publishers	
Related use cases	<i>Join Newsletter, Schedule Post, Manage Notifications</i>	
Precondition:	User is subscribed to a newsletter, User is logged in	
Postcondition	User has read the newsletter's latest content	
Trigger:	User thinks to read a newsletters content	
Main Success Scenario:	Actor <ol style="list-style-type: none"> <li>1 User navigates to the content section of a newsletter</li> <li>2 User clicks on a specific post to read</li> <li>3 User reads and scrolls through the entire content</li> </ol>	System <ol style="list-style-type: none"> <li>1.1 Display a list of recent posts or articles</li> <li>2.1 Display the content of the selected post</li> <li>3.1 Displays all/addition content on the related post</li> </ol>
Exceptions	<ol style="list-style-type: none"> <li>1.1 If the content is unavailable/removed, display a message saying "This content is not available" and provide other suggestions</li> <li>2.1 If the user is not subscribed to the newsletter, prompt the user to subscribe before allowing them access to the content</li> <li>3.1 If there is a loading error, display an error message and prompt user to retry</li> </ol>	

Use Case Title	Schedule a Post	
Primary Actor	Publisher	
Level	Full Description	
Stakeholders:	Publishers, Users	
Related use cases	<i>Create Post, Read Content, Manage Notifications</i>	
Precondition:	Publisher is logged in, Publisher has permission to post content, Publisher has drafted a post ready to schedule	
Postcondition	The post is scheduled to be published at a specific time	
Trigger:	Publisher thinks schedule a new post	
Main Success Scenario:	<p>Actor</p> <ol style="list-style-type: none"> <li>1 Publisher clicks "Create Post" + button</li> <li>2 Publisher creates or selects an existing post draft</li> <li>3 Publisher clicks "Schedule Post" option</li> <li>4 Publisher selects the desired date and time and clicks "Schedule"</li> <li>5 Publisher is notified that the post is successfully scheduled</li> </ol>	<p>System</p> <ol style="list-style-type: none"> <li>1.1 Display an option to create a new post</li> <li>2.1 Save the post as a draft</li> <li>3.1 Prompt the Publisher for a date and time for scheduling</li> <li>4.1 Save the post and schedule it to be published at the chosen time</li> <li>5.1 Add the post to the publishers "Scheduled Posts" list</li> </ol>
Exceptions	<ol style="list-style-type: none"> <li>1.1 If the Publisher enters an invalid date/time, display an error message and prompt them to select a valid date/time</li> <li>2.1 If the post is incomplete (missing title or body), prevent the user from scheduling and display a messgae asking the publisher to complete the post before scheduling</li> <li>3.1 If there is already another post scheduled from the same date &amp; time, prompt the publisher to choose a different date/time and display available slots.</li> </ol>	

Use Case Title	Create Post	
Primary Actor	Publisher	
Level	Full Description	
Stakeholders:	Publishers, Users	
Related use cases	<i>Schedule Post, Manage Notifications, Read Content</i>	
Precondition:	Publisher is logged in, Publisher has access to create a post	
Postcondition	The post is created and saved as a draft or schedules for publishing	
Trigger:	Publisher thinks to upload an article and writes it	
Main Success Scenario:	Actor	System
	1 Publisher navigates to "Create Post" + button	1.1 Display the post-creation interface with input for title, body, images, etc.
	2 Publisher enters the title and content for the post	2.1 Validate input fields to ensure no required fields are empty
	3 (optional) Publisher adds attachments/media to the post	3.1 Uploads and previews images or attachments.
	4 Publisher reviews and finalizes the content	4.1 Display a preview of the post formatting
	5 Publisher clicks "Save Draft", "Schedule Post" or "Publish Now" to save the post	5.1 Save the post as a draft, schedule post for specific date/time, or publish immediately
Exceptions	1.1 If a input field is left blank, display an error message and prompt the publisher to complete it before saving or publishing 2.1 If there is an upload issue with images, video, or attachments, display an error message and prompt the publisher to try reuploading 3.1 If the content contains invalid file formats/types, display an error message and prompt the publisher to select a proper format	

Use Case Title	Manage Notifications	
Primary Actor	User	
Level	Full Description	
Stakeholders:	Publishers, Users	
Related use cases	<i>Join Newsletter, Read Content, Send Message</i>	
Precondition:	User is logged in, User is subscribed to at least one newsletter	
Postcondition	Users notification preferences are updated, and notifications are adjusted accordingly	
Trigger:	User think to change their notification preferences/settings	
Main Success Scenario:	Actor <ol style="list-style-type: none"> <li>1 User navigates to "Notification Settings" section</li> <li>2 User selects or deselects the settings they wish to recieve</li> <li>3 User saves their preferences</li> </ol>	System <ol style="list-style-type: none"> <li>1.1 Display current notifications preferences</li> <li>2.1 Update the notification preferences per the users selections</li> <li>3.1 Confirm the changes and update the notification settings</li> </ol>
Exceptions	<ol style="list-style-type: none"> <li>1.1 If the User selects conflicting preferences, display an error message and prompt user to correct it</li> <li>2.1 If the settings cannot be saved to to network issues, display error message and allow user to retry saving</li> <li>3.1 If User navigates to the settings without making changes, confirm no updates are necessary and redirect them back to the previous page.</li> </ol>	

Use Case Title	Send Messages	
Primary Actor	User	
Level	Full Description	
Stakeholders:	Publishers, Users	
Related use cases	<i>Send Audio Message, Send Video Message, Send Attachment, Moderate Chats</i>	
Precondition:	User is logged in, User has access to a chat or comments within a newsletter	
Postcondition	The message is sent to the selected recipient(s) and stored in their conversation history	
Trigger:	User thinks to create a message within a chat or comment section	
Main Success Scenario:	Actor <ol style="list-style-type: none"> <li>1 User navigates to the chat or message interface</li> <li>2 User types a message</li> <li>3 User clicks "Send" button</li> <li>4 The message is posted in the chat or message section</li> <li>5 User receives confirmation the message was sent</li> </ol>	System <ol style="list-style-type: none"> <li>1.1 Display the chat or message history interface</li> <li>2.1 Activate the send button once the User starts typing</li> <li>3.1 Add sender's name, profile image, and timestamp to their message</li> <li>4.1 Message is visible ot all members in the chat</li> <li>5.1 System updates the chat with the new message</li> </ol>
Exceptions	<ol style="list-style-type: none"> <li>1.1 If the User is not logged in, prompt them to log in before they can send a message</li> <li>2.1 If the message exceeds the character limit, display an error message and ask the user to shorten the content</li> <li>3.1 If the User attempts to send an empty message, disable the send button and prompt the User to enter text</li> <li>4.1 If the message cannot be delivered, display an error message and allow User to retry</li> </ol>	

Use Case Title	Moderate Chats	
Primary Actor	Publisher	
Level	Full Description	
Stakeholders:	Publishers, Users	
Related use cases	<i>Send Message, Manage Notifications, Filter Content</i>	
Precondition:	Publisher is logged in, Publisher has permission to moderate chats, Publisher is in an active chat	
Postcondition	The message/content is approved, removed, or flagged based on the Publisher's moderation actions	
Trigger:	Publisher receives a content flag or thinks to moderate a chat/comment section	
Main Success Scenario:	Actor	System
	1 Publisher navigates to the chat/comment moderation panel	1.1 Display a list of recent messages or comments flagged for review
	2 Publisher reviews any flagged content or other message content	2.1 Display the message along with the sender's name, timestamp, and any flags/reports
	3 Publishers decides to either approve, delete, or flag a message for further review	3.1 Update the message status according to the publishers actions
	4 Publisher submits their action and receives a confirmation	4.1 The changes immediately show in the chat/comment section based on the moderation actions
Exceptions	1.1 If the publisher does not have moderation permissions, deny access to the moderation panel and display an error message	
	2.1 If there is an error while flagging or deleting a message, display an error message and prompt the publisher/moderator to try again	
	3.1 If a message has already been moderated, notify the publisher and prevent them from duplicating an action on the same message	

Use Case Title	Filter Content	
Primary Actor	Publisher	
Level	Full Description	
Stakeholders:	Publishers, Users	
Related use cases	<i>Send Message, Moderate Chats, Send Content</i>	
Precondition:	Publisher is logged in, Publisher has permission to filter content, Content is available for filtering	
Postcondition	Content is filtered according to the publishers criteria	
Trigger:	Publisher thinks to filter chat/comment/post content	
Main Success Scenario:	Actor	System
	1 Publisher navigates to the content filter settings	1.1 Display available filter options
	2 Publisher selects the filter criteria they want to apply	2.1 Validate filter criteria
	3 Publisher click "Apply Filters"	3.1 Apply selected filters to the content, hiding anything that does not match the criteria
	4 Publisher recieves confirmation the filter has been applied	4.1 Update the content display based on the new filter settings
Exceptions	2.1 If the filter cannot be applied to the content, show an error message and prompt the publisher to retry	
	3.1 If the filter criteria are too broad, causing all content to be hidden, alert the publisher and ask if they would like to refine their settings	



Use Case Title	Send Content	
Primary Actor	Publisher	
Level	Full Description	
Stakeholders:	Publisher, User	
Related use cases	<i>Create Post, Filter Content, Moderate Chats</i>	
Precondition:	Publisher is logged in, Publisher has content ready to send or publish, Content meets the format guideline requirements	
Postcondition	Content is successfully sent to the users or published on their newsletter	
Trigger:	Publisher thinks to send content to subscribers/users or publish content	
Main Success Scenario:	<p>Actor</p> <ol style="list-style-type: none"> <li>1 Publisher navigates to the content creation or post section</li> <li>2 Publisher selects or creates content</li> <li>3 Publisher selects "Send" or "Publish" buttons</li> <li>4 Publisher confirms the action</li> <li>5 Publisher is notified the content was sent or published successfully</li> </ol>	<p>System</p> <ol style="list-style-type: none"> <li>1.1 Display options to create or select content to send</li> <li>2.1 Validate the content format</li> <li>3.1 Prompt the publisher to confirm their action</li> <li>4.1 Send the content to the selected audience</li> <li>5.1 Display a confirmation message and log content</li> </ol>
Exceptions	<ol style="list-style-type: none"> <li>1.1 If the content is in an unsupported format, prompt the publisher to correct it</li> <li>2.1 If the content exceeds size/character limit, alert the publisher and prompt them to fix it</li> <li>3.1 If the Publisher tries to send/publish content that violates platform guidelines, reject the content and notify the publisher of the violation</li> <li>4.1 If the content fails to send, display an error message and allow the publisher to retry</li> </ol>	

Use Case Title	Manage Inbox	
Primary Actor	User	
Level	Full Description	
Stakeholders:	Users, Publishers	
Related use cases	<i>Send Message, Read Content, Manage Notification</i>	
Precondition:	User is logged in, User has received messages, notifications, or other content in their inbox	
Postcondition	User has read, organized, or deleted messages, and their inbox status is updated	
Trigger:	User thinks to view, organize, or manage their inbox messages	
Main Success Scenario:	Actor <ol style="list-style-type: none"> <li>1 User navigates to their inbox</li> <li>2 User selects a message or notification to view</li> <li>3 User decides to manage the message or their inbox</li> <li>4 User selects "Delete", "Archive",etc</li> <li>5 User receives confirmation the action was successful</li> </ol>	System <ol style="list-style-type: none"> <li>1.1 Display a list of received messages, notifications, or content</li> <li>2.1 Display full content of the selected message or notification</li> <li>3.1 Allow the User to perform the selected action</li> <li>4.1 Remove, archive, etc. the selected messages and update the Users inbox</li> <li>5.1 Update the inbox, reflecting all changes</li> </ol>
Exceptions	<ol style="list-style-type: none"> <li>1.1 If the user attempts to delete or archive a message that is already removed or archive, notify the User that no action is needed</li> <li>2.1 If the inbox is full and the user attempts to recieve new messages, prompt the User to manage their inbox before allowing further action</li> <li>3.1 If a message content cannot be loaded, display a loading error and allow the user to retry.</li> </ol>	