

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/305073717>

# Software Requirements Specification (SRS) of Web Publishing System Domain

Technical Report · November 2014

DOI: 10.13140/RG.2.1.2365.5925

---

CITATIONS  
0

READS  
20,951

---

5 authors, including:



Loay Aladib

University of Malaya

7 PUBLICATIONS 10 CITATIONS

[SEE PROFILE](#)



Yamani Nair Thamutharam

3 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



Mohd Nasyim bin Rosli

University of Malaya

2 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



# Software Requirements Specification

## Web Publishing System

### Domain

**19-Nov-2014**

#### **Authors:**

Loay Aladib

Si Toh Chooi Ling

Yamani Nair Thamutharam

Mohd Nasyim bin Rosli

Emir Afiq Ridzuan

## TABLE OF CONTENTS

(SRS) Web Publishing System Domain Description.....	2
Use Case Modelling.....	2
Class Diagram.....	3
Activity Diagrams.....	3
Sequence & Communication Diagrams.....	3
State Machine Diagram .....	4
(SRS) Web Publishing System Domain Solutions .....	4
Use Case Diagrams & Use Case Descriptions.....	5
Use Case Diagrams.....	5
Use Case Description.....	6
Class Diagrams, CRC Cards & Data Dictionary .....	39
Class Diagrams .....	39
CRC Cards .....	40
Data Dictionary.....	46
Activity Diagrams.....	47
Submit Article.....	47
Search Article .....	48
Submit Review .....	49
Login .....	50
Register .....	51
Check Status .....	51
Update Reviewer .....	52
Update Author .....	53
Assign Reviewer.....	54
Receive Review .....	55
Update Article.....	56
Remove Article.....	57
Publish Article .....	58
Generate Statistic Report .....	59
Subscribe Article's Newsletter .....	60
Send Copyright .....	61
Receive Article .....	62
Send Response .....	63
Sequence & Communication Diagram .....	64
Sequence Diagrams .....	64
Communication Diagrams .....	82
State Machine Diagrams.....	97

User Lifecycle.....	97
Article Lifecycle .....	98
Author Lifecycle.....	99
Reviewer Lifecycle.....	100
Tools Used.....	101
Reference.....	101

## **(SRS) Web Publishing System Domain Description**

We analyze the software requirements and design object-oriented models for a web publishing system using UML. A Software Requirements Specification (SRS) will be provided as follows;

### **Use Case Modelling**

- Model a complete use case diagram for the web publishing system by including all the use cases and requirements described in the SRS.
- The generalization and include/exclude relationships must be considered in your diagram based on SRS.
- Use case description for each use case must be converted to the flow of events (paths) which shows the actor actions and system responses and set the priority of each use case.
- In the current SRS, the users are not required to log-in to the system. To improve this system design, you have to add new use case(s) to manage the user registration and login.
- The system should be divided into subsystems and include the relevant use cases in the subsystems.

## **Class Diagram**

- Model a domain class diagram to show the structure of the web publishing systems.
- Show explicitly the class attributes and operations and also the relationships between the classes, multiplicity specifications, and other model elements that you find appropriate.
- You need to apply CRC (Class–Responsibility–Collaboration) cards approach to model interaction between objects based on the use case scenarios. In the report, you need to include the CRC cards for your class diagram.
- Create a data dictionary to record definition of each class. Each class is written a short description defining its scope / detail of the class.

## **Activity Diagrams**

- Model activity diagrams to describe system functions that are represented by use cases.
- The pre-condition, post-condition and essential process or flow of events in the use case description can be used as the basis for drawing your activity diagrams.

## **Sequence & Communication Diagrams**

- Model the interactions that define the message passing between lifelines (e.g. objects) to capture the responses of all the objects that are included in a single use case.
- You should model the sequence and communication diagrams for each use case.
- The sequence diagrams should represent the detailed object interaction that occurs for one use case or for one operation. Show explicitly conditional interactions.
- The communication diagrams should hold the same information as sequence diagrams. Communication diagrams have many similarities to sequence diagrams. They express the same information in a different format. There is no time dimension in communication diagrams. The sequence is captured with sequence numbers. Show explicitly the links between the lifelines that participate in the collaboration.

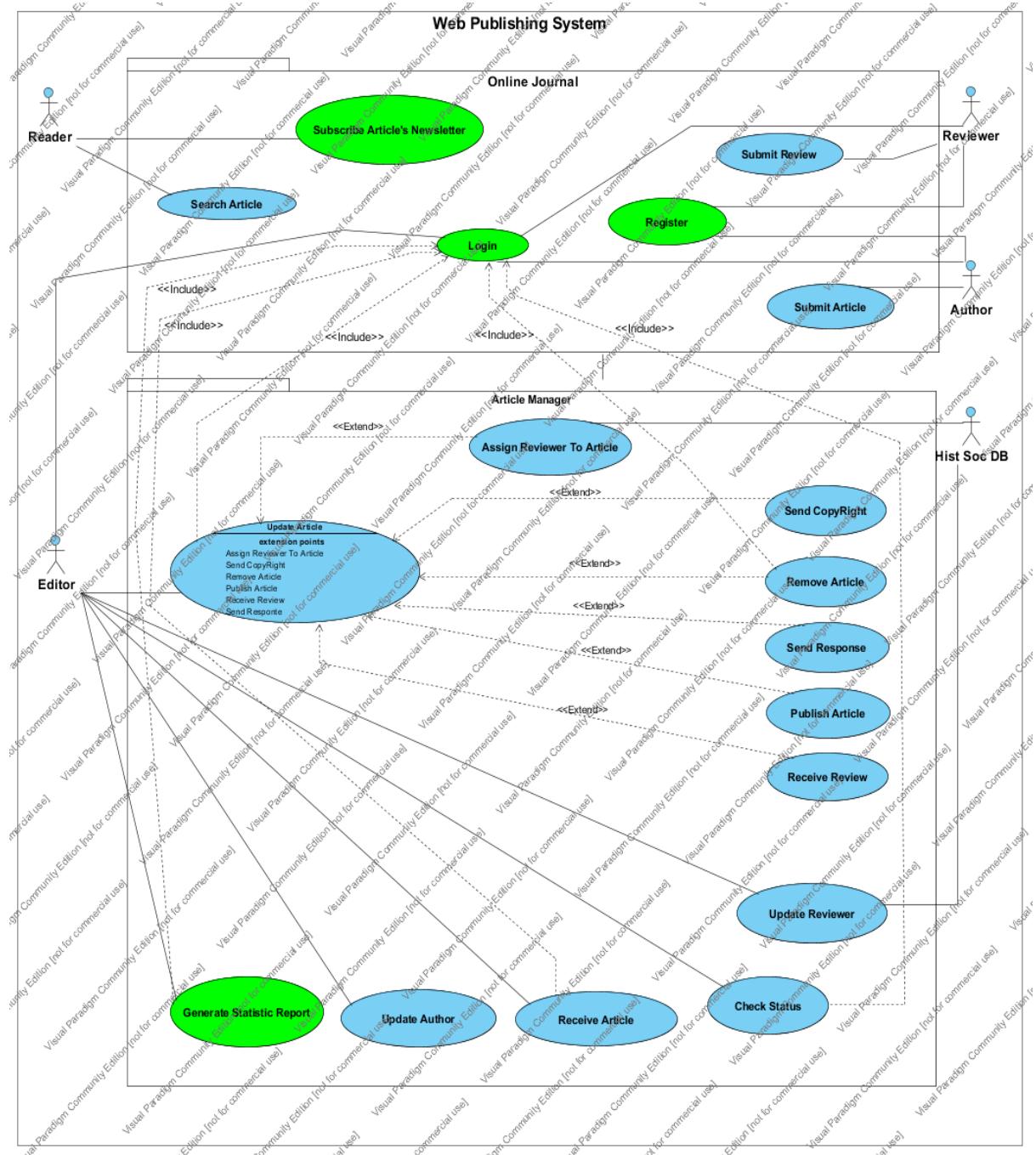
## **State Machine Diagram**

- Model state machine diagrams for each class that has more than one possible states (1 diagram for 1 class). Capture all the possible responses of a single object to all the use cases in which it is involved.
- Indicate all possible states and transitions for the class being considered. For each transition, specify relevant events, triggers, conditions and actions.
- For each diagram, you need to include a description that describes the object life cycle it is trying to show and specify how an object behavior causes it to change from one state to another state. For each transition, specify relevant events, conditions, and actions.

## **(SRS) Web Publishing System Domain Solutions**

# Use Case Diagrams & Use Case Descriptions

## Use Case Diagrams



## Use Case Description

### Submit Article

Element	Description	
<b>Use Case Name</b>	Submit article	
<b>Use Case ID</b>	UC-01	
<b>Priority</b>	High	
<b>Actor(s)</b>	Author	
<b>Description</b>	The author either submits an original article or resubmits an edited article.	
<b>Precondition(s)</b>	The form is active window, the function has been selected and all authors are already in the database so Author has already connected to the Online Journal Website.	
<b>Post-condition(s)</b>	The database has been modified.	
<b>Flow of events</b>		
Actor Action	System Response	
1. The author chooses the <i>email editor</i> button		
	2. The system uses the <i>send to</i> HTML tag to bring up the user's email system.	
	3. The author fills in the Subject line and attaches the files as directed and emails them.	
	4. This system generates and sends an email acknowledgement.	
<b>Alternative flows and exceptions</b>	May be abandoned at any time.	

## Search Article

Element	Description	
<b>Use Case Name</b>	Search article	
<b>Use Case ID</b>	UC-02	
<b>Priority</b>	High	
<b>Actor(s)</b>	Reader	
<b>Description</b>	Searches for an article and downloads it to his/her machine.	
<b>Precondition(s)</b>	The reader has already accessed the online journal website, also the web is showing the grid for searching.	
<b>Post-condition(s)</b>	The selected article is downloaded to the client machine.	
<b>Flow of events</b>	Actor Action	System Response
	1. Choose to search by author.	
		2. Create and present an alphabetical list of all authors in the database.
	3. Selects the desired author.	
		4. Create and present a list of all articles by that author in the database.
	5. Select an article.	
		6. Display the abstract of the article.
	7. Select to download the article.	
		8. Provide selected article and downloaded to the system user's machine.
<b>Alternative flows 1</b>	Actor Action	System Response
	1. Choose to search by category	
		2. Create and present

		a list of all categories in the database.
3. Select the desired category.		
	4. Create and present a list of all articles in the selected category.	
5. Select an article.		
	6. Display the abstract of the article.	
7. Select to download the article.		
	8. Provide selected article and downloaded to the System user's machine.	
<b>Alternative flows 11</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Choose to search by keywords.	
		2. Present a dialog box.
	3. Enter keyword or phrase.	
		4. Search all abstract of articles with the provided keyword or phrase and present a list of the article which matches the search result.
	5. Select an article.	
		6. Display the abstract of the

		article.
	7. Select to download the article.	
		8. Provide selected article and downloaded to the system user's machine.
<b>Exceptions</b>	At any time, the operation may abandon. The reader may choose to return to article list or previous list as well.	

## Submit Review

Element	Description	
<b>Use Case Name</b>	Submit review	
<b>Use Case ID</b>	UC-03	
<b>Priority</b>	High	
<b>Actor(s)</b>	Reviewer	
<b>Description</b>	The reviewer submits a review of an article.	
<b>Precondition(s)</b>	The reviewer is on Communication page linked from the Online Journal Main Page.	
<b>Post-Condition(s)</b>	Review will be sent to system,	
Flow of events		
	Actor Action	System Responses
	1. Choose the <i>email editor</i> button	
		2. Uses the <i>send to</i> HTML tag to invoke the user's email system.
	3. Fill in the subject line and attaches the file as directed and email it.	
		4. Generate and sends an email acknowledgment .

<b>Alternative flows</b>	Reviewer uses their own email directly with sufficient information provided as per instructed on the web page.
<b>Exceptions</b>	At any time, the operation may abandon.

## Login

Element	Description	
<b>Use Case Name</b>	Log-in	
<b>Use Case ID</b>	UC-04	
<b>Priority</b>	High	
<b>Actor(s)</b>	Editor, reviewer, author	
<b>Description</b>	The user log-in to the system.	
<b>Precondition(s)</b>	The user should land on the log-in page.	
<b>Post-condition(s)</b>	The user logged to the system.	
Flow of events	Actor Action	System Responses
	1. The user landed on the log-in page.	
		2. The system displays a login form.
	3. The user enters the valid username and password	
		4. The system authenticates the username and password.
		5. The system display greets message.
	6. User logged in to the system.	
<b>Alternative flows and exceptions</b>	1. User entered invalid credentials at the second trial.	

		2. The system validates the credentials and wrong credentials are verified. System prompt user to enter credential again.
	3. The user enters credentials again.	
		4. The system validates credential and wrong credentials are entered. A number of trials have exceeded 3 times.
		5. System prompt user to log-in again after 15 minutes.

## Register

Element	Description	
<b>Use Case Name</b>	Register	
<b>Use Case ID</b>	UC-05	
<b>Priority</b>	High	
<b>Actor(s)</b>	Reviewer, author	
<b>Description</b>	Register author or reviewer to the system.	
<b>Precondition(s)</b>	The form page is active to registration.	
<b>Post-condition(s)</b>	The user is registered with the system and profile data is stored with the system.	
Flow of events	Actor Action	System Responses
	1. Author or reviewer select registration operation.	
		2. System display registration form and prompt to fill

		in profile data.
	3. The author or reviewer fills up the form with profile data.	
	4. The system validates new user information and checks with the user is existed.	
	5. The system reverts with registration station.	
<b>Alternative flows 1</b>	1. New user registration information is not valid.	
	2. The system displays registration failure page.	
<b>Exceptions</b>	At any time, the operation may abandon.	

### Check Status

Element	Description	
<b>Use Case Name</b>	Check status	
<b>Use Case ID</b>	UC-06	
<b>Priority</b>	High	
<b>Actor(s)</b>	Editor	
<b>Description</b>	An editor checks the status of all active articles.	
<b>Precondition(s)</b>	An editor must access the main page of “article manager”.	
<b>Post-condition(s)</b>	The active article will be displayed with their status.	
<b>Flow of events</b>	<b>Actor Action</b>	<b>System Responses</b>
	1. Select “check status”.	

		2. Display a scrollable list of all active articles with their status.
		3. Return to the main page of “article manager”
<b>Alternative flows Exceptions</b>	At any time, the operation may abandon.	

### Update Reviewer

Element	Description
<b>Use Case Name</b>	Update reviewer
<b>Use Case ID</b>	UC-07
<b>Priority</b>	High
<b>Actor(s)</b>	Editor, historical society DB
<b>Description</b>	The editor updates information about a current Reviewer in the article manager database.
<b>Precondition(s)</b>	The form is active window and function has been selected.
<b>Post-condition(s)</b>	The database has been modified.

### Update Reviewer

Flow of events – Update Reviewer	Actor Action	System Response
	1. Select “add/update reviewer” on update article main page	
		2. The system presents a choice of adding or to update reviewer information
	3. The editor chooses to update reviewer.	
		4. The system links to the historical society DB.
		5. The system presents an alphabetical drop-down list of reviewer's

		names using an access query.
6. The editor selects a reviewer.		
	7. The system uses a query to retrieve the information about that reviewer and displays it in a form to update reviewer information.	
8. The editor updates the form and submits it.		
	9. The system verifies that required fields have content and enters the updated information into the database.	
	10. Display message “update reviewer complete”	
	11. Returns the editor to the article manager main page.	

#### Update reviewer alternative

Alternative flow – update reviewer	Actor Action	System Response
	1. Select “add/update reviewer” on update article main page	
	2. The system presents a	

		choice of adding or to update reviewer information
	3. The editor chooses to update reviewer.	
	4. The system links to the historical society DB.	
	5. The system presents an alphabetical drop-down list of reviewer's names using an access query.	
	6. The editor selects a reviewer.	
	7. The system uses a query to retrieve the information about that reviewer and displays it in a form to update reviewer information.	
	8. The editor does not update anything reviewer's information.	
	9. Returns the editor to the article manager main page.	
<b>Exceptions</b>	At any time, the operation may abandon.	
<b>Add reviewer</b>		
<b>Flow of events – adds reviewer</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Select	

	“add/update reviewer” on update article main page	
	2. The system presents a choice of adding or to update reviewer information	
	3. The editor chooses to add reviewer.	
	4. The system links to the historical society DB.	
	5. The system presents an alphabetical drop-down list of member’s names that register to be reviewer using an access query.	
	6. The editor selects a member.	
	7. The system uses a query to retrieve the information about that member and displays it in a form to update member information.	
	8. The editor approves member information to be as a reviewer and submits it.	
	9. The system verifies that	

		required fields have content
		10. The system link with HSDB to checks whether the reviewer is active.
		11. The system enters the updated information into the database
		12. Display message “add reviewer complete”
		13. Returns the editor to the article manager main page.

#### Add reviewer alternative flow

Alternative flows – add reviewer	Actor Action	System Response
	1. Select “add/update reviewer” on update article main page	
		2. The system presents a choice of adding or to update reviewer information
	3. The editor chooses to add reviewer.	
		4. The system links to the historical society DB.
		5. The system presents an alphabetical drop-down list

		of member's names that register to be reviewer using an access query.
6. The editor selects a member.		
	7. The system uses a query to retrieve the information about that member and displays it in a form to update member information.	
8. The editor approves member information to be as a reviewer and submits it.		
	9. The system verifies that required fields have content	
	10. The system link with HSDB to checks whether the reviewer is active.	
	11. Display message "reviewer inactive. cannot add reviewer"	
	12. Returns the editor to the Article manager main page.	
<b>Exceptions</b>	At any time, the operation may abandon.	

## Update Author

Element	Description	
<b>Use Case Name</b>	Update author	
<b>Use Case ID</b>	UC-08	
<b>Priority</b>	High	
<b>Actor(s)</b>	Editor	
<b>Description</b>	This update author information in the article manager database.	
<b>Precondition(s)</b>	The form is active window and function has been selected.	
<b>Post-condition(s)</b>	The database has been modified.	
<b>Update Author</b>		
<b>Flow of events – update author</b>	Actor Action	System Response
	1. Select “add/update author” on update article main page	
		2. The system presents a choice of adding or to update author information
	3. The editor chooses to update author.	
		4. The system presents an alphabetical drop-down list of author's names using an access query.
	5. The editor selects an author.	
		6. The system uses a query to retrieve the information about that author and

		displays it in a form to update author's information.
7. The editor updates the form and submits it.		
	8. The system verifies that required fields have content and enters the updated information into the database.	
	9. Display message "update author complete"	
	10. Returns the editor to the article manager main page.	

#### Update author alternative

Alternative flow – update author	Actor Action	System Response
	1. Select "add/update author" on update article main page	
	2. The system presents a choice of adding or to update author information	
	3. The editor chooses to update author.	
	4. The system presents an	

		alphabetical drop-down list of author's names using an access query.
	5. The editor selects an author.	
		6. The system uses a query to retrieve the information about that author and displays it in a form to update author's information.
	7. The editor does not update anything author's information	
		8. Returns the editor to the article manager main page.
<b>Exceptions</b>	At any time, the operation may abandon.	
<b>Add author</b>		
<b>Flow of events – adds author</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Select "add/update reviewer" on update article main page	
		2. The system presents a choice of adding or to update author information
	3. The editor chooses to add the	

	author.	
	4. The system presents an alphabetical drop-down list of member's names that register to be author using an access query.	
5. The editor selects a member.		
	6. The system uses a query to retrieve the information about that member and displays it in a form to update member information.	
7. The editor approves member information to be as an author and submits it.		
	8. The system verifies that required fields have content	
	9. The system checks whether the author had verified the email.	
	10. The system enters the updated information into the database.	

		11. Display message “add author complete”
		12. Returns the editor to the article manager main page.

### Add author alternative

Alternative flows – add author	Actor Action	System Response
	1. Select “add/update reviewer” on update article main page	
		2. The system presents a choice of adding or to update author information
	3. The editor chooses to add the author.	
		4. The system presents an alphabetical drop-down list of member’s names that register to be author using an access query.
	5. The editor selects a member.	
		6. The system uses a query to retrieve the information about that member and displays it in a form to update member

		information.
	7. The editor approves member information to be as an author and submits it.	
	8. The system verifies that required fields have content	
	9. The system checks whether author had verified the email	
	10. Display message "member not verified"	
	11. Returns the editor to the article manager main page.	
<b>Exceptions</b>	At any time, the operation may abandon.	

## Assign Reviewer

Element	Description	
<b>Use Case Name</b>	Assign reviewer	
<b>Use Case ID</b>	UC-09	
<b>Priority</b>	High	
<b>Actor(s)</b>	Editor, historical society DB	
<b>Description</b>	Editor assigns one or more reviewers to an article.	
<b>Precondition(s)</b>	The editor has already accessed the article using update article function.	
<b>Post-condition(s)</b>	The editor will be returned to the page of update article.	
<b>Flow of events</b>	<b>Actor Action</b> 1. Select "assign reviewer" on update article	<b>System Responses</b> This row is intentionally blank

	main page	
		2. The system presents list of reviewer with their status
	3. Select a reviewer until sufficient.	
		4. The system verifies the person that editor chooses using historical society DB.
		5. The system emails the reviewer attached with the article requesting that they do the review.
		6. The system will display message “article is sent to assigned reviewer”
		7. Returns the editor to the article manager main page.
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon.	

## Receive Review

Element	Description															
<b>Use Case Name</b>	Receive review															
<b>Use Case ID</b>	UC-10															
<b>Priority</b>	High															
<b>Actor(s)</b>	Editor															
<b>Description</b>	Editor enters a review into the system															
<b>Precondition(s)</b>	Editor has already accessed the article using update article function															
<b>Post-condition(s)</b>	Editor will be returned to the page of update article															
<b>Flow of events</b>	<table border="1"> <thead> <tr> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1. Select “receive review”.</td> <td></td></tr> <tr> <td></td> <td>2. The system presents a grid for filling with the information.</td></tr> <tr> <td>3. Editor fills the information and submits the form.</td> <td></td></tr> <tr> <td></td> <td>4. The system verifies the information.</td></tr> <tr> <td></td> <td>5. Display message “receive review article submitted”</td></tr> <tr> <td></td> <td>6. Returns the editor to the article manager main page.</td></tr> </tbody> </table>		Actor Action	System Response	1. Select “receive review”.			2. The system presents a grid for filling with the information.	3. Editor fills the information and submits the form.			4. The system verifies the information.		5. Display message “receive review article submitted”		6. Returns the editor to the article manager main page.
Actor Action	System Response															
1. Select “receive review”.																
	2. The system presents a grid for filling with the information.															
3. Editor fills the information and submits the form.																
	4. The system verifies the information.															
	5. Display message “receive review article submitted”															
	6. Returns the editor to the article manager main page.															
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon															

## Update Article

Element	Description																	
<b>Use Case Name</b>	Update article																	
<b>Use Case ID</b>	UC-11																	
<b>Priority</b>	High																	
<b>Actor(s)</b>	Editor																	
<b>Description</b>	Editor enters information about an existing article																	
<b>Precondition(s)</b>	Editor has already accessed the main page of the article manager																	
<b>Post-condition(s)</b>	Article will be updated in the database and editor will be returned to the main page of article manager																	
<b>Flow of events – update article info</b>	<table border="1"> <thead> <tr> <th style="background-color: black; color: white;">Actor Action</th> <th style="background-color: black; color: white;">System Response</th> </tr> </thead> <tbody> <tr> <td>1. Select “update article”.</td><td></td></tr> <tr> <td></td><td>2. The system presents an alphabetical drop-down list of active articles</td></tr> <tr> <td>3. Select an article to update.</td><td></td></tr> <tr> <td></td><td>4. The system presents the information about the chosen article.</td></tr> <tr> <td>5. Editor chooses to update information of the article.</td><td></td></tr> <tr> <td>6. Editor updates the existing information of the article.</td><td></td></tr> <tr> <td></td><td>7. The system verifies the updated information.</td></tr> </tbody> </table>		Actor Action	System Response	1. Select “update article”.			2. The system presents an alphabetical drop-down list of active articles	3. Select an article to update.			4. The system presents the information about the chosen article.	5. Editor chooses to update information of the article.		6. Editor updates the existing information of the article.			7. The system verifies the updated information.
Actor Action	System Response																	
1. Select “update article”.																		
	2. The system presents an alphabetical drop-down list of active articles																	
3. Select an article to update.																		
	4. The system presents the information about the chosen article.																	
5. Editor chooses to update information of the article.																		
6. Editor updates the existing information of the article.																		
	7. The system verifies the updated information.																	

		8. Returns the editor to the article manager main page.
<b>Flow of events – adds article categories</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Select “update article”.	
		2. The system presents an alphabetical drop-down list of active articles
	3. Select an article to update.	
		4. The system presents the information about the chosen article.
	5. Editor chooses to add categories of the article.	
		6. The system displays a drop-down list of categories.
	7. Editor selects one or many categories which are related to the article	
		8. The system verifies the updated categories.

		9. Returns the editor to the article manager main page.
<b>Alternative flows and exceptions</b>		If the article is not already in the database, the operation may abandon. In addition, the Editor may abandon the operation at any time

## Remove Article

Element	Description
<b>Use Case Name</b>	Remove article
<b>Use Case ID</b>	UC-12
<b>Priority</b>	High
<b>Actor(s)</b>	Editor
<b>Description</b>	An Editor removes an article from the active category.
<b>Precondition(s)</b>	An editor must access the article via update article.
<b>Post-condition(s)</b>	The article will be removed from the active article database.

<b>Flow of events</b>	<b>Actor Action</b>	<b>System Response</b>
	1. Select “remove article”	
		2. Display an alphabetized list of articles with the status of each.
	3. Select an article for removal.	
		4. Display the information about the article and article delete confirmation details.
	5. Confirm the deletion of the article	
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon.	

### Publish Article

<b>Element</b>	<b>Description</b>
<b>Use Case Name</b>	Publish article
<b>Use Case ID</b>	UC-13
<b>Priority</b>	High
<b>Actor(s)</b>	Editor
<b>Description</b>	This transfers an article from the Article manager database to the on-line journal database.
<b>Precondition(s)</b>	The form is active window and function has been selected.
<b>Post-condition(s)</b>	The article is added to an on-line journal database.
<b>Flow of events</b>	<b>Actor Action</b>
	<b>System Responses</b>
	1. The system presents a pull- down menu of authors.

	2. The editor selects an author.	
	3. The system presents a pull-down menu of articles by that author.	
	4. The editor selects an article.	
	5. The system presents a form with the article information.	
	6. The editor reviews the article information and submits to publish the article.	
	7. The system transfers the article and updates the article information to the online Journal.	
	8. The system removes the article from the active article database.	
	9. Returns the editor to the article manager main page.	
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon.	

## Generate Statistic Report

Element	Description	
<b>Use Case Name</b>	Generate statistic report	
<b>Use Case ID</b>	UC-14	
<b>Priority</b>	Medium	
<b>Actor(s)</b>	Editor	
<b>Description</b>	To generate a statistic report for operational analysis and understand user preference.	
<b>Precondition(s)</b>	The editor must access to the main page of “article manager”, and landed on ‘generate statistic report’ page.	
<b>Post-condition(s)</b>	Usage reports by desired analysis metrics are generated.	
<b>Flow of events</b>	<b>Actor Action</b> 1. Select analysis metric such as ‘total number of submitted article by current month’	<b>System Responses</b>
	2. Validate the selected metric.	
	3. The system computes the report.	

	4. The report is generated.	
<b>Alternative flows1</b>	<b>Actor Action</b>	<b>System Responses</b>
	1. Select analysis metric such as 'total number of submitted article by current year'	
		2. Validate the selected metric.
		3. The system computes the report.
	4. The report is generated.	
<b>Alternative flows 11</b>	<b>Actor Action</b>	<b>System Responses</b>
	1. Select analysis metric such as 'total number of submitted article by current week'	
		2. Validate the selected metric.
		3. The system computes the report.
	4. The report is generated.	
<b>Exceptions</b>	At any time, the operation may abandon.	

## Subscribe Article's Newsletter

Element	Description	
<b>Use Case Name</b>	Subscribe article's newsletter	
<b>Use Case ID</b>	UC-15	
<b>Priority</b>	High	
<b>Actor(s)</b>	Reader	
<b>Description</b>	This function allows the reader to receive newsletters when an article is published in web publishing system. A reader can subscribe newsletter based on specific author or article category.	
<b>Precondition(s)</b>	The reader must be able to access to the web publishing system.	
<b>Post-condition(s)</b>	Reader's email address will be stored into the database.	
Flow of events	Actor Action	System Responses
	1. Select “subscribe article's newsletters”.	
		2. An option for the subscriber is displayed, whether subscribe article's newsletter by author or by article category.
	3. Add email address.	
	4. Submit subscription.	
		5. Display that the subscription is successful saved, and send the activation key to reader's email address.
		6. Send activation key to reader's email address.
	7. Activate subscription.	
		8. Confirmed the new subscription and saved into

		the database.
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon.	

### Send Copyright

Element	Description	
<b>Use Case Name</b>	Send copyright	
<b>Use Case ID</b>	UC-16	
<b>Priority</b>	High	
<b>Actor(s)</b>	Editor	
<b>Description</b>	Editor sends a copyright form to an author.	
<b>Precondition(s)</b>	The editor has already accessed the article using update article function.	
<b>Post-condition(s)</b>	The author's article will be directed to the article manager department for managing.	
Flow of events	Actor Action	System Responses
	1. Select "send copyright".	
		2. The system presents a pull-down menu of active articles by that author.
	3. The editor selects an article.	
		4. The system presents a form with the article information.
	5. The editor reviews the article information and submits to send copyright.	
		6. The system calls the email system and puts the Author's email address in the recipient line, the name of the article on the subject line and attached copyright form.

	7. The editor fills out email text	
	8. The editor sends the copyright via email.	
		9. The system sends the copyright email to the author.
		10. Returns editor to the article manager page.
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon.	

### Receive Article

Element	Description											
<b>Use Case Name</b>	Receive article											
<b>Use Case ID</b>	UC-17											
<b>Priority</b>	High											
<b>Actor(s)</b>	Editor											
<b>Description</b>	Editor enters a new or revised article into the system											
<b>Precondition(s)</b>	The editor must access the main page of “article manager” and has a file containing the article available.											
<b>Post-condition(s)</b>	A new article or existing article will be updated in the system.											
<b>Flow of events</b>	<table border="1"> <thead> <tr> <th>Actor Action</th> <th>System Responses</th> </tr> </thead> <tbody> <tr> <td>1. Select “receive article”.</td> <td></td></tr> <tr> <td></td> <td>2. Presents a choice of entering new article or updating the existing article.</td></tr> <tr> <td>3. Select “add or update”.</td> <td></td></tr> <tr> <td></td> <td>4. Present a list of articles to choose from and a grid for filling with</td></tr> </tbody> </table>		Actor Action	System Responses	1. Select “receive article”.			2. Presents a choice of entering new article or updating the existing article.	3. Select “add or update”.			4. Present a list of articles to choose from and a grid for filling with
Actor Action	System Responses											
1. Select “receive article”.												
	2. Presents a choice of entering new article or updating the existing article.											
3. Select “add or update”.												
	4. Present a list of articles to choose from and a grid for filling with											

		the information or presents a blank grid.
	5. Fills in the information.	
	6. Submit the form.	
		7. Verify the information and return to the main page of “article manager”
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon.	

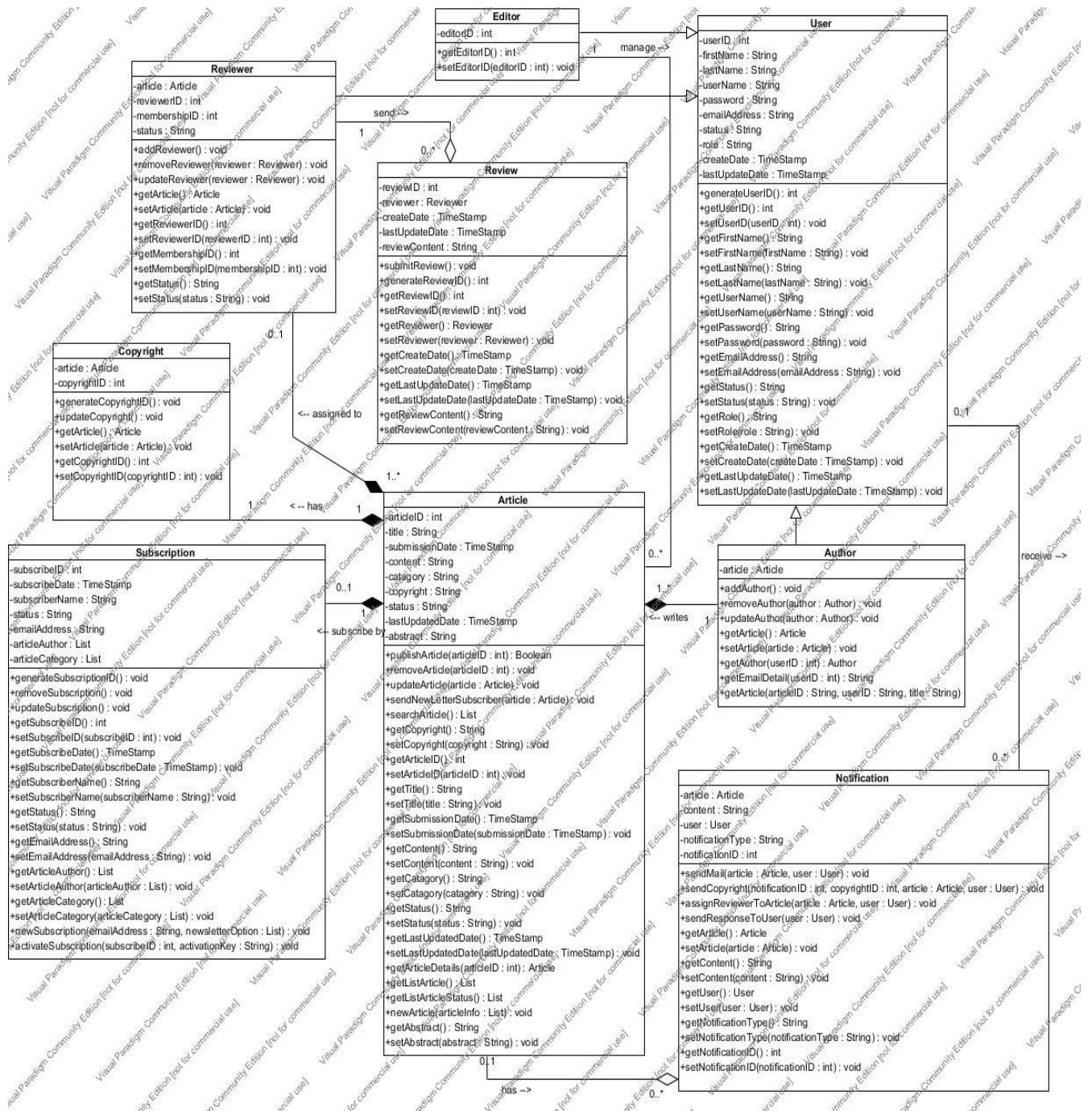
## Send Response

Element	Description	
<b>Use Case Name</b>	Send response	
<b>Use Case ID</b>	UC-18	
<b>Priority</b>	High	
<b>Actor(s)</b>	Editor	
<b>Description</b>	Editor sends a response to an Author.	
<b>Precondition(s)</b>	The editor has already accessed the article using update article function.	
<b>Post-condition(s)</b>	The author's article will be directed to the article manager department for managing.	
Flow of events	Actor Action	System Responses
	1. The editor selects to “send response”.	
		2. The system presents a pull-down menu of the author.
	3. The editor selects an author from the list.	

		4. The system calls the email system and puts the Author's email address in the recipient line and the name of the article on the subject line.
	5. Editor fills out the email text.	
	6. The editor sends the response via email.	
		7. The system sends the response email to the author.
		8. The system returns the editor to the article manager main page.
<b>Alternative flows and exceptions</b>	At any time, the operation may abandon.	

# Class Diagrams, CRC Cards & Data Dictionary

## Class Diagrams



## CRC Cards

Article	
<b>Description:</b> Hold article detail	
Attributes:	
Name	Description
ArticleID	The article ID
Title	The article title
SubmissionDate	The date of submission
LastUpdateDate	The date of update
Abstract	The abstract
Content	The content
Category	The category
Status	The status
Copyright	The copyright
Responsibilities:	
Name	Collaborator
GetArticleID	Subscription
SetArticleID	Reviewer
GetTitle	Copyright
SetTitle	Author
GetAbstract	
SetAbstract	
GetContent	
SetContent	
GetCopyright	
SetCopyright	
GetSubmissionDate	
SetSubmissionDate	
GetUpdateDate	
SetUpdateDate	
GetStatus	
SetStatus	
UpdateArticle	
SearchArticle	
SendNewsLetterSubscriber	
PublishArticle	
RemoveArticle	
UpdateArticle	
SendNewLetterSubscriber	
SearchArticle	
GetLastUpdateDate	
SetLastUpdateDate	
GetArticleDetails	
GetListArticle	
GetListArticle	
NewArticle	
GetCategory	
SetCategory	

Author	
<b>Super Classes:</b> User	
<b>Description:</b> Hold Author detail.	
<b>Attributes:</b>	
Name	Description
Article	
<b>Responsibilities:</b>	
Name	Collaborator
AddAuthor	User
UpdateAuthor	Notification
RemoveAuthor	Article
GetArticle	
SetArticle	
GetAuthor	
GetEmailDetail	
GetArticle	
<b>CopyRight</b>	
<b>Description:</b> Hold copyright detail	
<b>Attributes:</b>	
Name	Description
CopyrightID	
Copyright	
Article	
<b>Responsibilities:</b>	
Name	Collaborator
GenerateCopyrightID	Notification
UpdateCopyright	Article
SetCopyrightID	
GetCopyrightID	
SetArticle	
GetArticle	

Notification		
Name	Description	
NotificationID		
NotificationType		
User		
Content		
Article		
Responsibilities:		
Name	Collaborator	
SendMail	User	
SendCopyRight	Article	
SendResponseToAuthor		
AssignReviewerToArticle		
GetArticle		
SetArticle		
GetContent		
SetContent		
GetUser		
SetUser		
GetNotificationType		
SetNotificationType		
GetNotificationID		
SetNotificationID		
Review		
Description:	To hold the review details	
Attributes:		
Name	Description	
ReviewID	The review ID	
ReviewContent	The review	
CreateDate	The date of creation	
LastUpdateDate	The date of update	
Reviewer	The reviewer	
Responsibilities:		
Name	Collaborator	
GetReviewer	Reviewer	
SetReviewer		
GetReviewContent		
SetReviewContent		
GetReviewerID		
SetReviewerID		
GetCreateDate		
SetCreateDate		
GetLastUpdateDate		
SetLastUpdateDate		
GenerateReviewerID		
SubmitReview		

Reviewer	
<b>Description:</b> Hold Reviewer Detail	
<b>Attributes:</b>	
Article	Name
MembershipID	Description
ReviewerID	
Status	
<b>Responsibilities:</b>	
Add Reviewer	Name
UpdateReviewer	Collaborator
RemoveReviewer	User
GetArticle	Notification
SetArticle	Article
GetMembershipID	Review
SetMembershipID	
GetReviewerID	
SetReviewerID	
GetStatus	
SetStatus	

Editor	
<b>Super Classes:</b> User	
<b>Description:</b> To hold the detail of user	
<b>Attributes:</b>	
editorID	Name
	Description
getEditorID	editorID
setEditorID	The id of editor
<b>Responsibilities:</b>	
getEditorID	Name
setEditorID	Collaborator

Subscription	
<b>Description:</b> Hold Subscription detail	
<b>Attributes:</b>	
SubscribeID	Name
SubscriberName	The subscription ID
EmailAddress	The subscriber name
ArticleAuthor	The subscriber email
ArticleCategory	The subscribed author
SubscribeDate	The subscribed category
Status	The subscription date
<b>Responsibilities:</b>	
RemoveSubscription	Description
UpdateSubscription	The subscription ID
GetSubscribeID	The subscriber name
SetSubscribeID	The subscriber email
GetSubscriberName	The subscribed author
SetSubscriberName	The subscribed category
GetEmailAddress	The subscription date
SetEmailAddress	The subscription status
GetArticleAuthor	
SetArticleAuthor	
GenerateSubscriptionID	
GetArticleCategory	
SetArticleCategory	
GetSubscribeDate	
SetSubscribeDate	
GetStatus	
SetStatus	
NewSubscription	
ActiveSubscription	

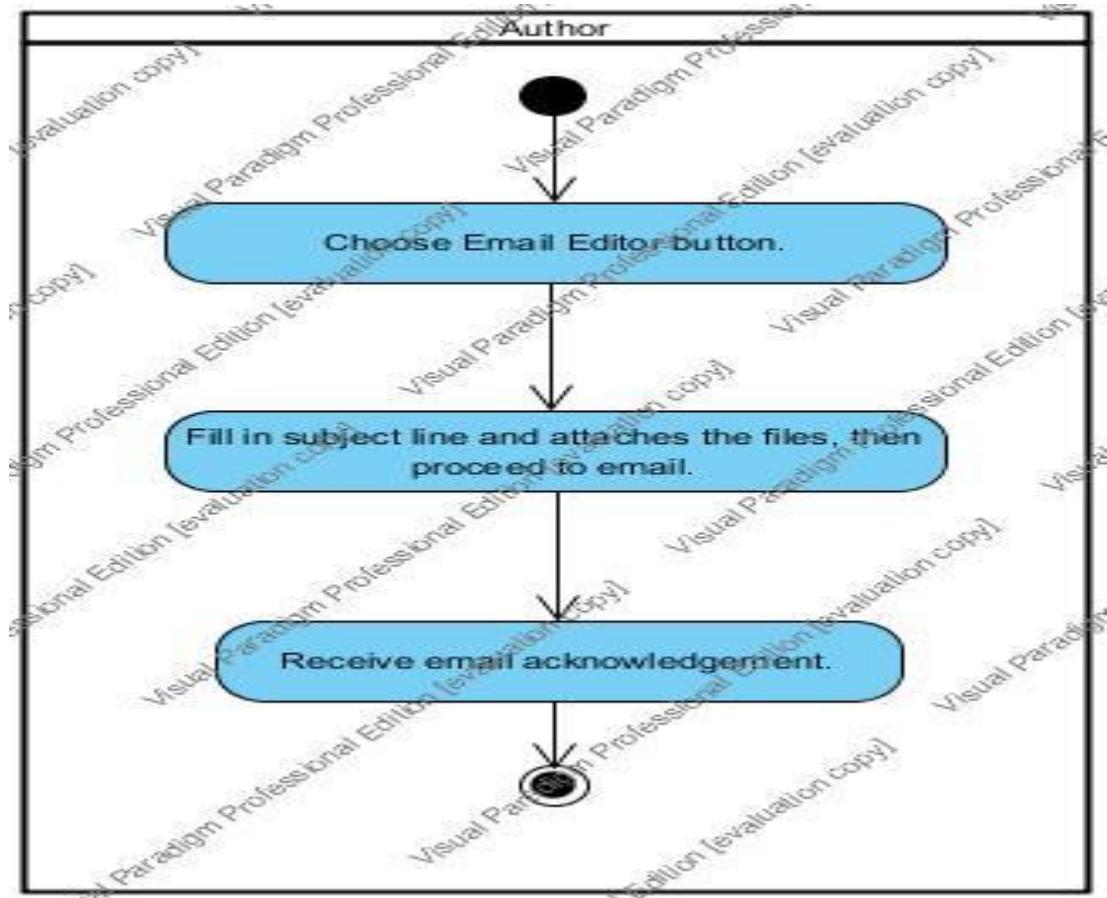
User		
Attributes:		
Name	Description	
UserID	The user ID	
Password	The password	
EmailAddress	The email	
Status	The user status (Active or Deactivated)	
Role	The role of user (Author or Reviewer)	
CreateDate	The date of creation	
UpdateDate	The date of update	
FirstName	The first name of user	
LastName	The last name of user	
UserName	The log on name	
Responsibilities:		
Name	Collaborator	
GenerateUserID	Reviewer	
GetUserID	Author	
SetUserID		
GetPassword		
SetPassword		
GetEmailAddress		
SetEmailAddress		
GetStatus		
SetStatus		
UpdateUser		
GetUserID		
SetUserID		
GetFirstName		
SetFirstName		
GetLastName		
SetLastName		
GetCreateDate		
SetCreateDate		
GetUpdateDate		
SetUpdateDate		
GetUserName		
SetUserName		

## Data Dictionary

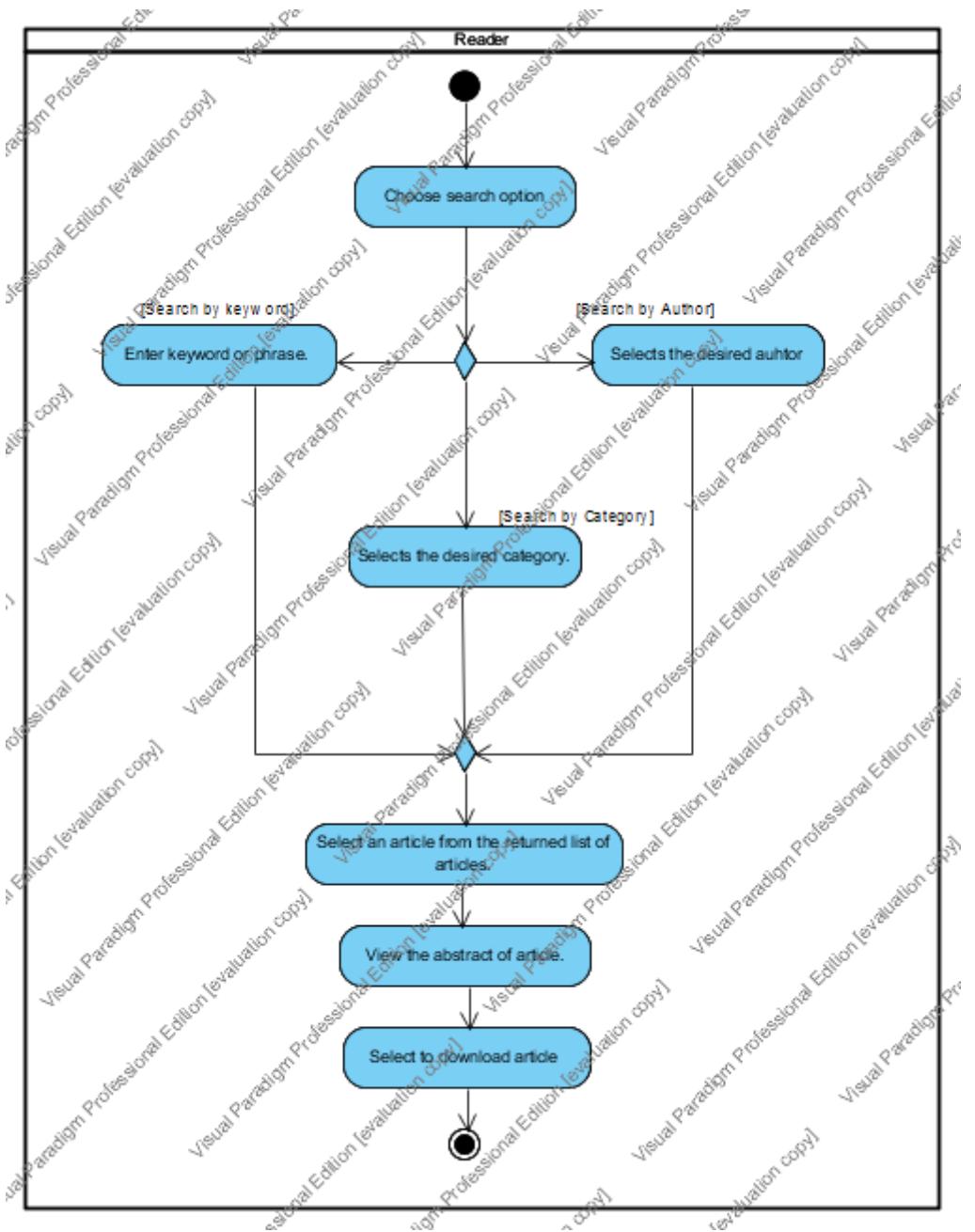
Class	Definition
<b>Author</b>	<ul style="list-style-type: none"> <li>-A person who submits an article to be reviewed.</li> <li>-The class has attributes of article class.</li> <li>-The author may write one or more article.</li> <li>-The author shares some same attributes and operation with User class.</li> </ul>
<b>Article</b>	<ul style="list-style-type: none"> <li>-A document writes by author and reviewed by reviewer which tracked by the system.</li> <li>-The class has attributes articleID, title, submissionDate, abstract, content, category, copyright, status, and lastUpdatedDate.</li> <li>-Multiple articles may belong to one author. Multiple articles may assign to one reviewer.</li> <li>-The article may subscribe by zero or more subscriber.</li> <li>-Every article may have one copyright.</li> <li>-The article may have zero or more notification.</li> </ul>
<b>Editor</b>	<ul style="list-style-type: none"> <li>- A person who manage articles in the web publishing system.</li> <li>- The class has attributes editorID.</li> <li>- The editor may manage more than one article.</li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>-A person that examines an article and has the ability to send a review for the article.</li> <li>-The class has attributes of article class, reviewerID, and membershipID.</li> <li>-A reviewer may assign with one or more article.</li> <li>-A reviewer may send zero or more review for one article.</li> <li>-A reviewer shares some same attributes and operation with User class.</li> </ul>
<b>Review</b>	<ul style="list-style-type: none"> <li>-A written recommendation about the appropriateness of an article for publication; may include suggestions for improvement.</li> <li>-The class has attributes of the reviewer: review ID, createdDate, updateDate, status, and reviewContent.</li> <li>-There will be zero or review send by one reviewer.</li> </ul>
<b>User</b>	<ul style="list-style-type: none"> <li>-Author, Reviewer, and Editor.</li> <li>-A person who receive notification.</li> <li>-The class attributes userID, firstName, lastName, userName, password, emailAddress, status, role, createdDate and updatedDate.</li> <li>-A user may receive Zero or more notifications.</li> </ul>
<b>Copyright</b>	<ul style="list-style-type: none"> <li>-A written exclusive and assignable legal right, given to the article and send it to the author.</li> <li>-The class has attributes of article class and copyrightID.</li> <li>-The article has one copyright only.</li> </ul>
<b>Subscription</b>	<ul style="list-style-type: none"> <li>-An arrangement to receive the newsletter by email for an article based on article category and author.</li> <li>-The class has attributes subscribeID, subscribeDate, subscriberName, status, emailAddress, articleAuthor, and articleCategory.</li> <li>-There will be zero or more subscriptions may subscribe to an article.</li> </ul>
<b>Notification</b>	<ul style="list-style-type: none"> <li>-The action of notifying author or reviewer regarding article updates.</li> <li>-The class attributes of article class and user class, notificationID, notificationType, and content.</li> <li>-The article may or may not have notifications.</li> <li>-Zero or one notification may receive by zero or more than one user.</li> </ul>

## Activity Diagrams

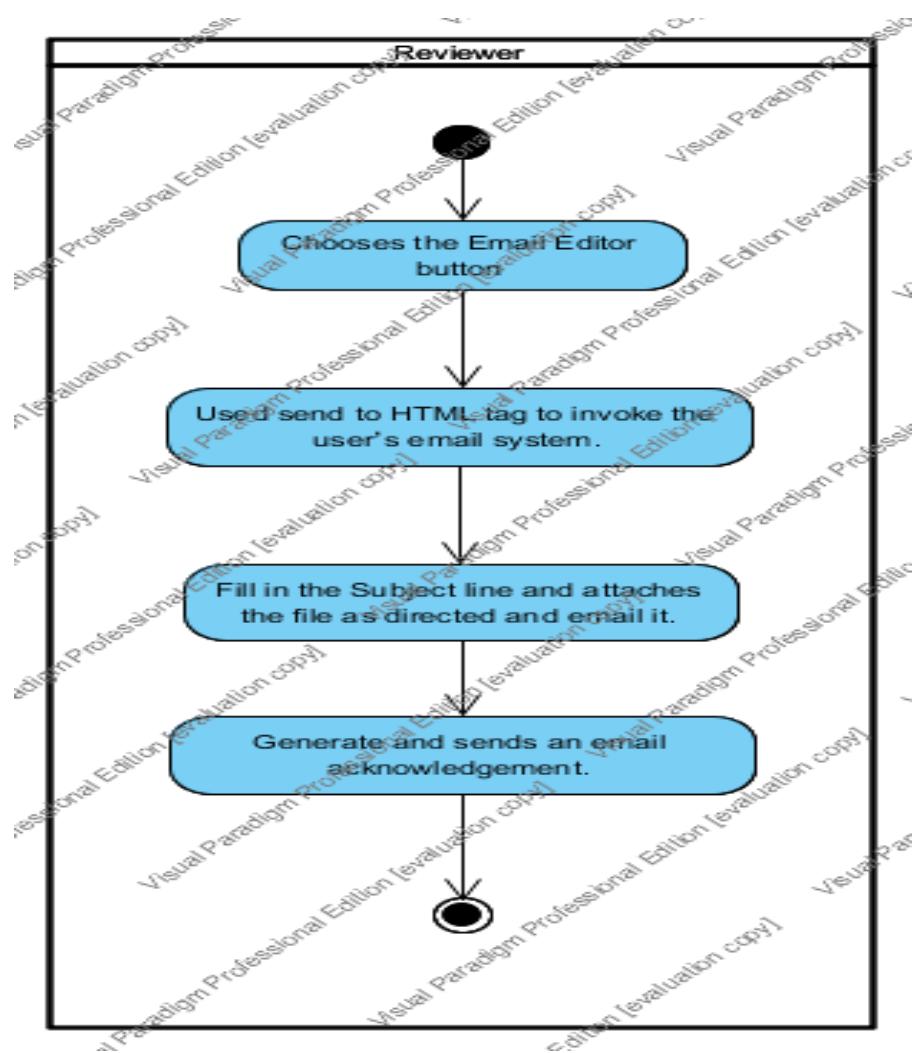
### Submit Article



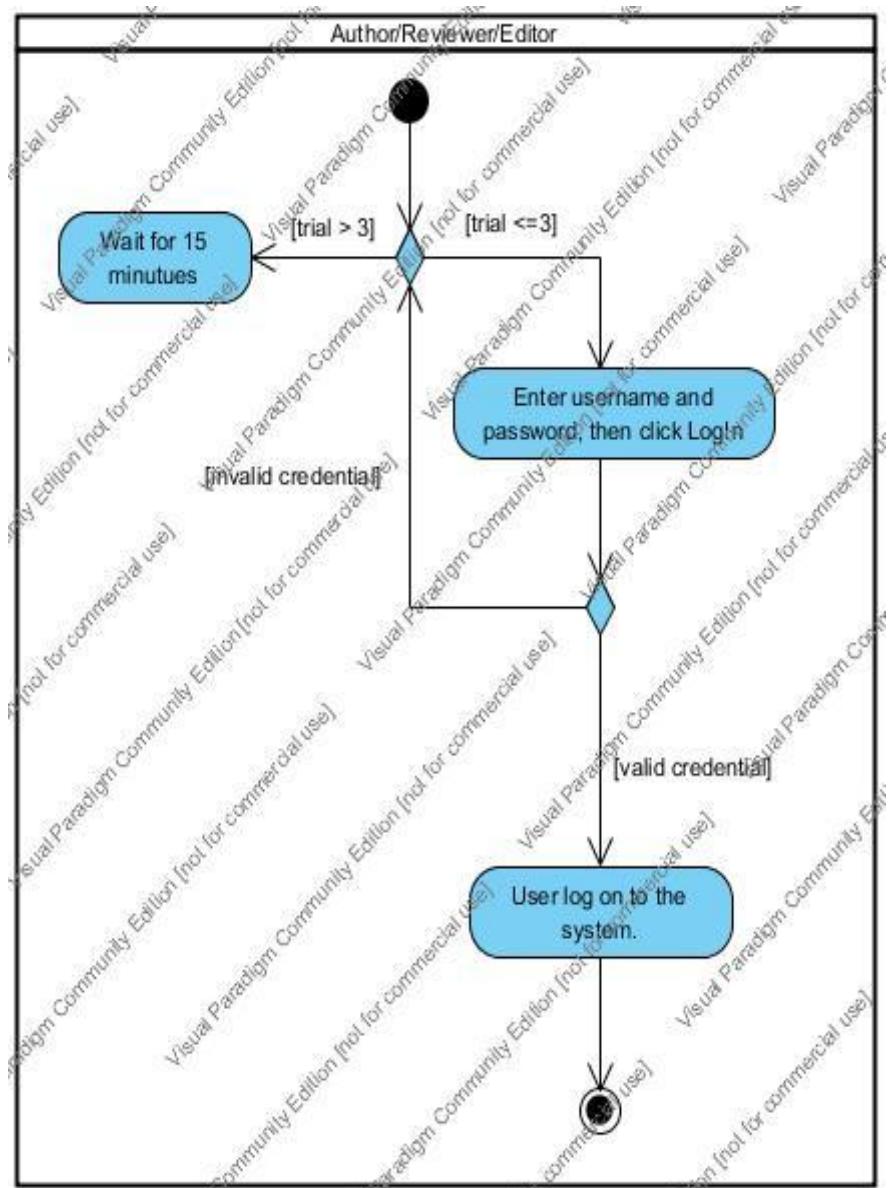
## Search Article



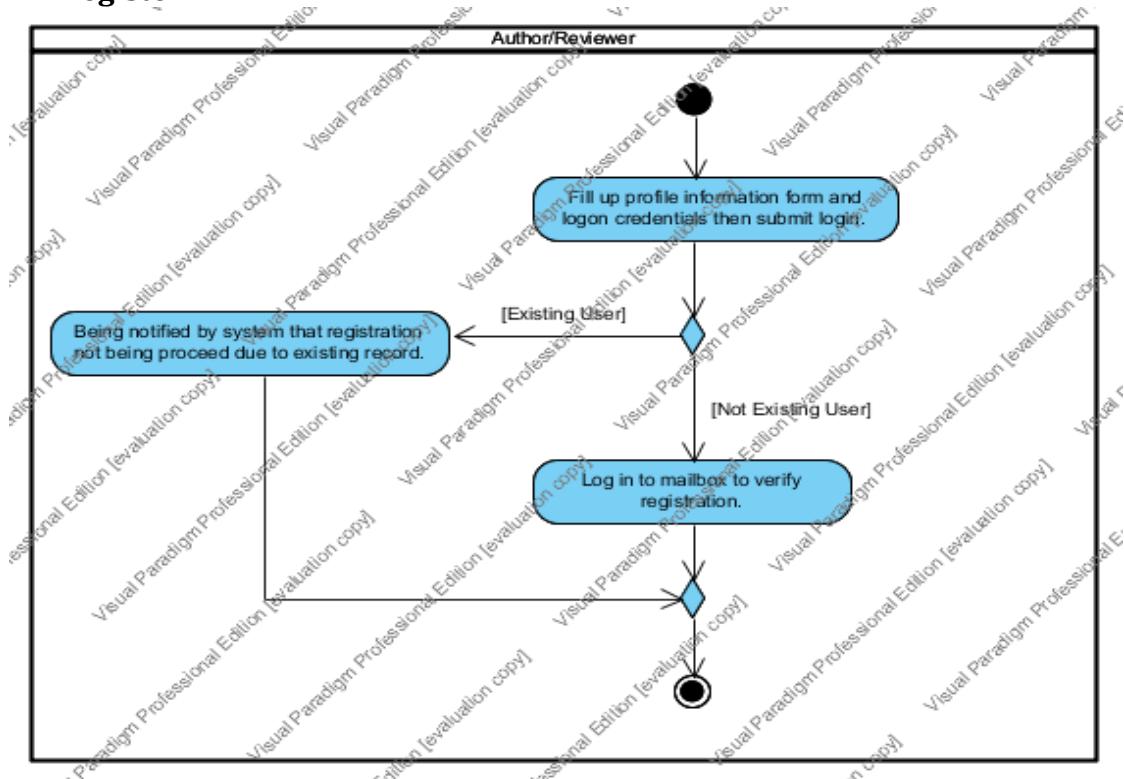
## Submit Review



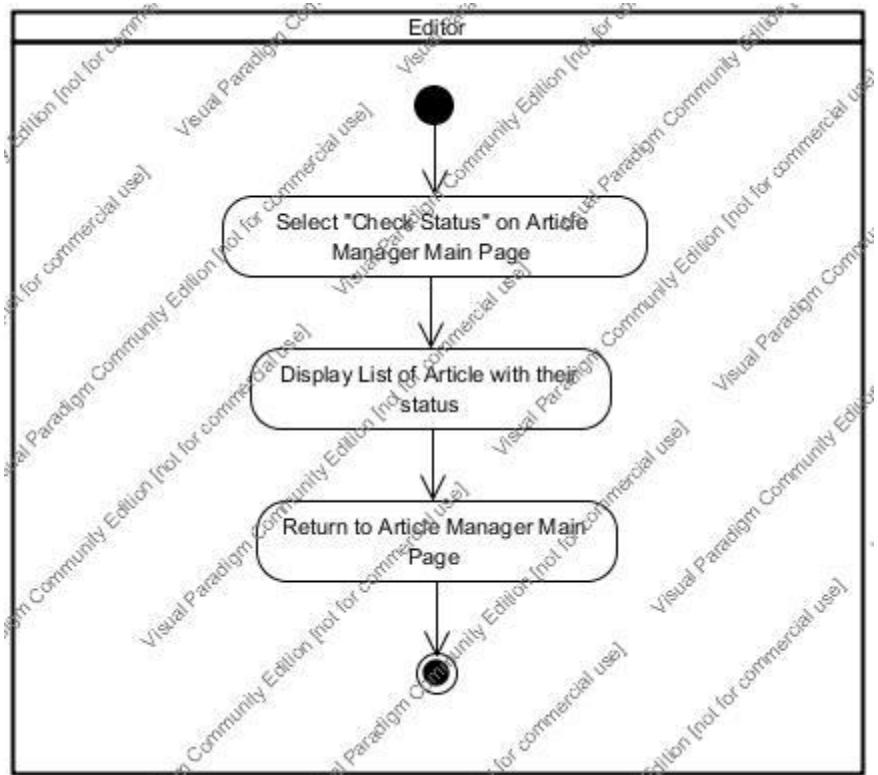
## Login



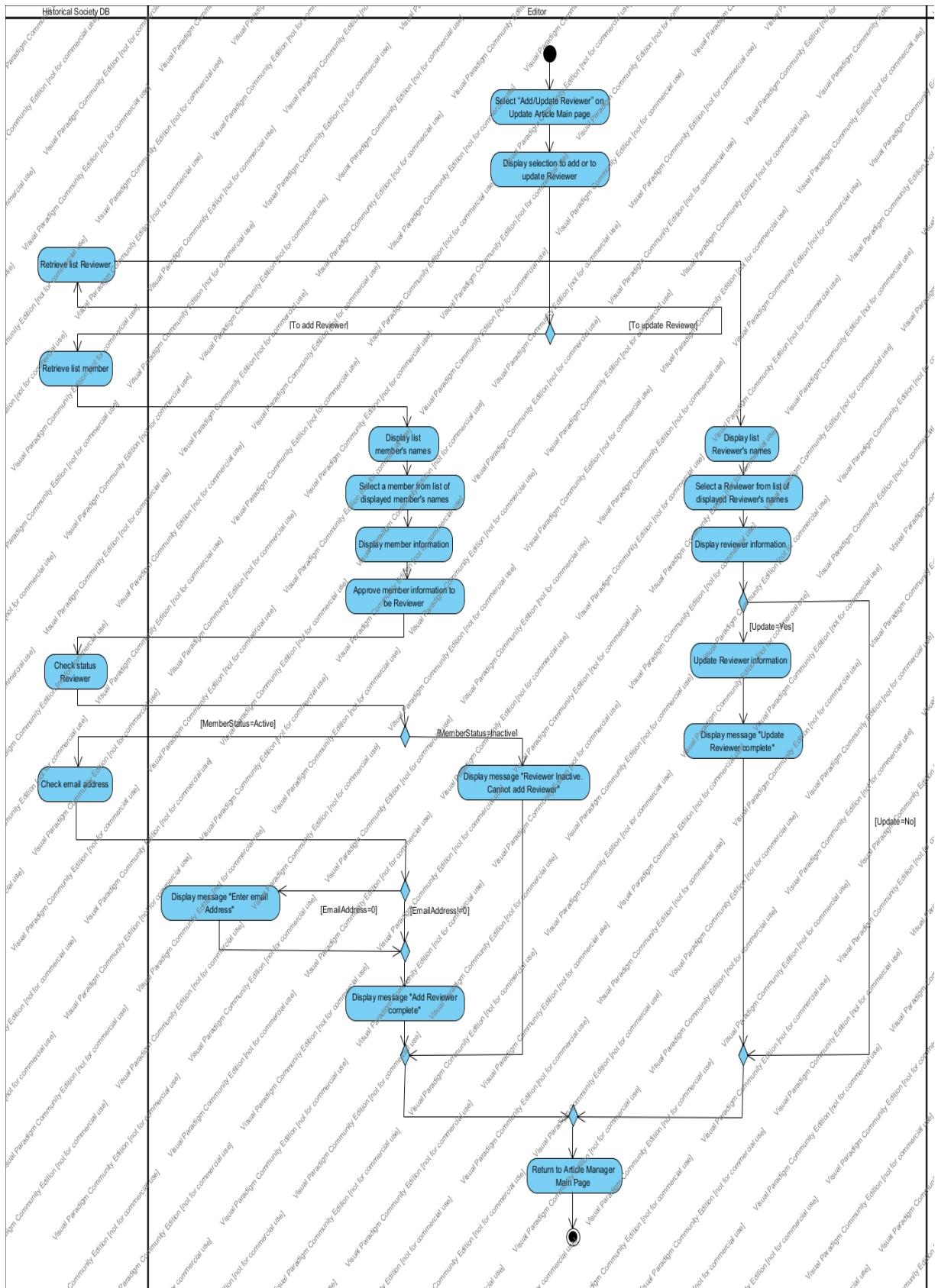
## Register



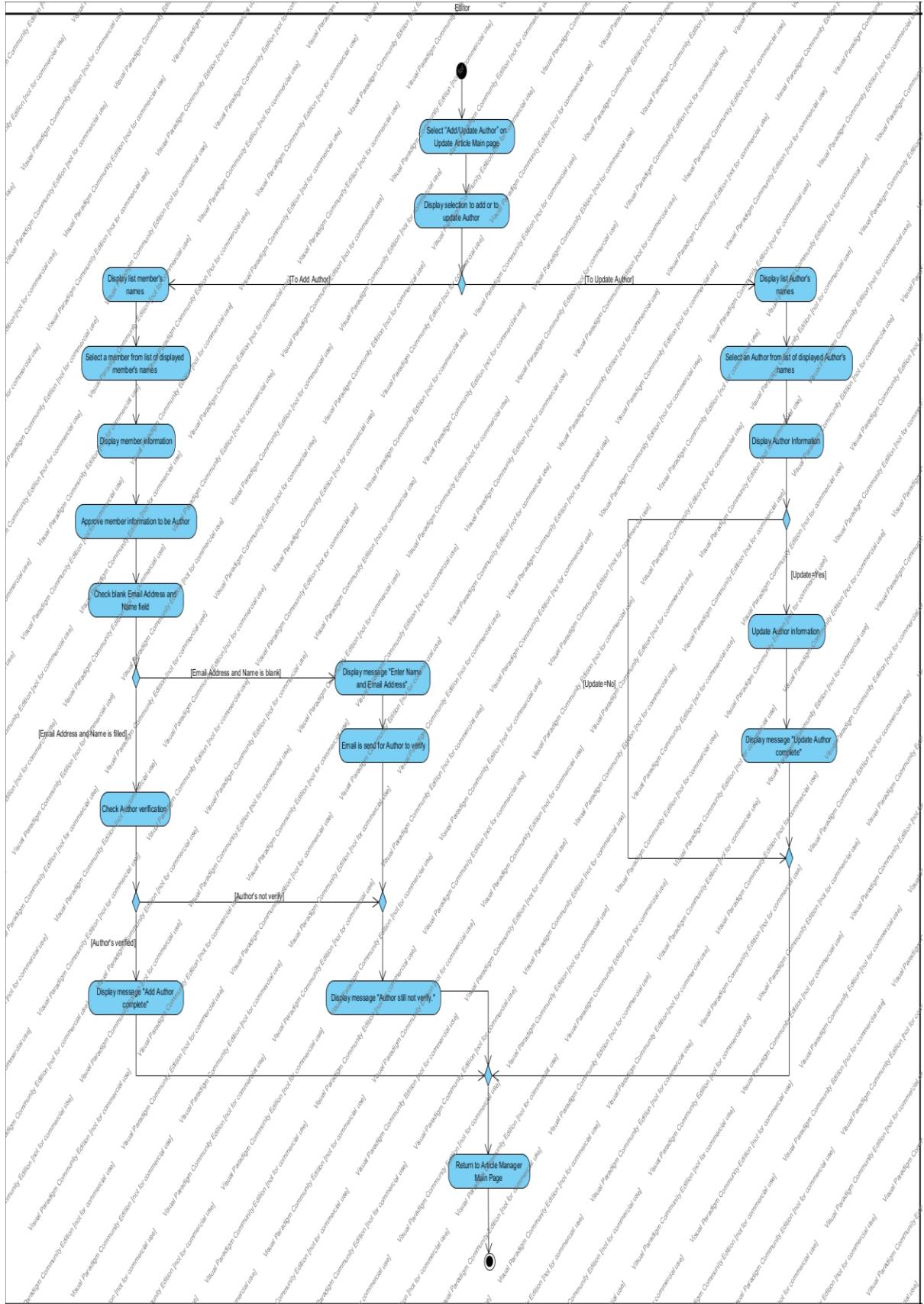
## Check Status



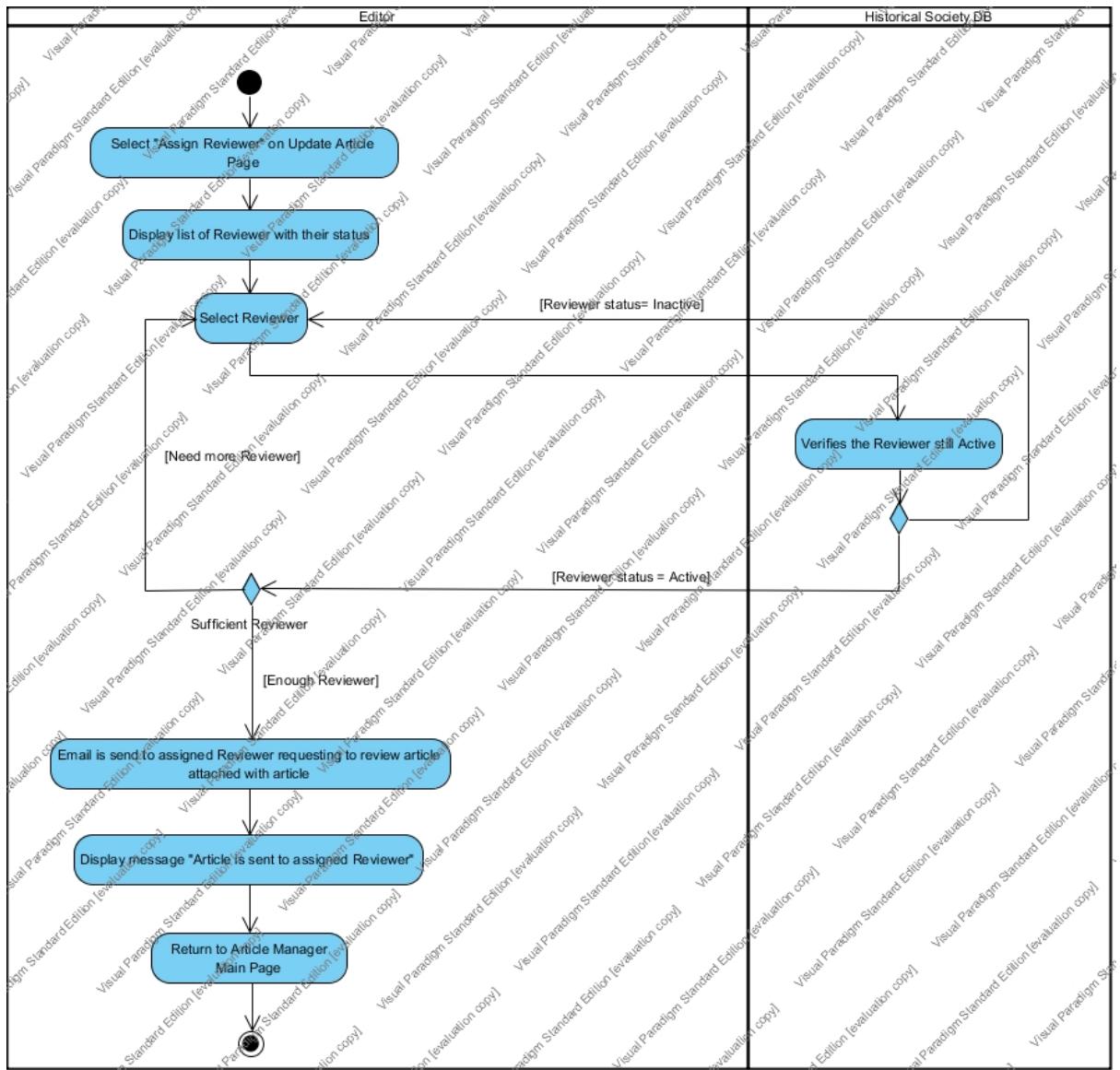
## Update Reviewer



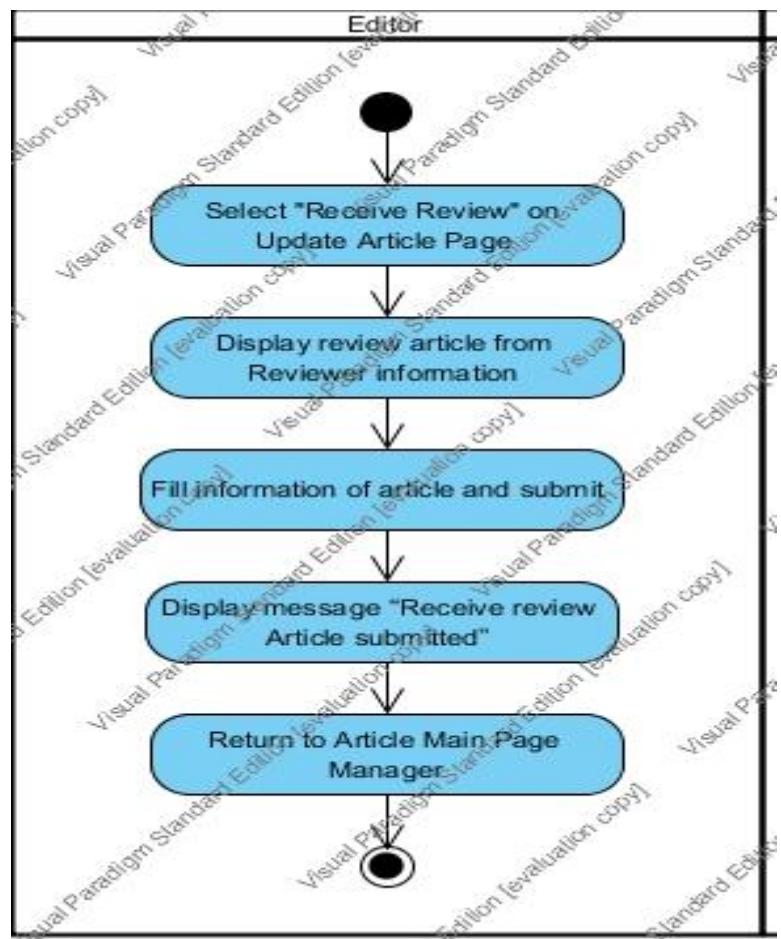
## Update Author



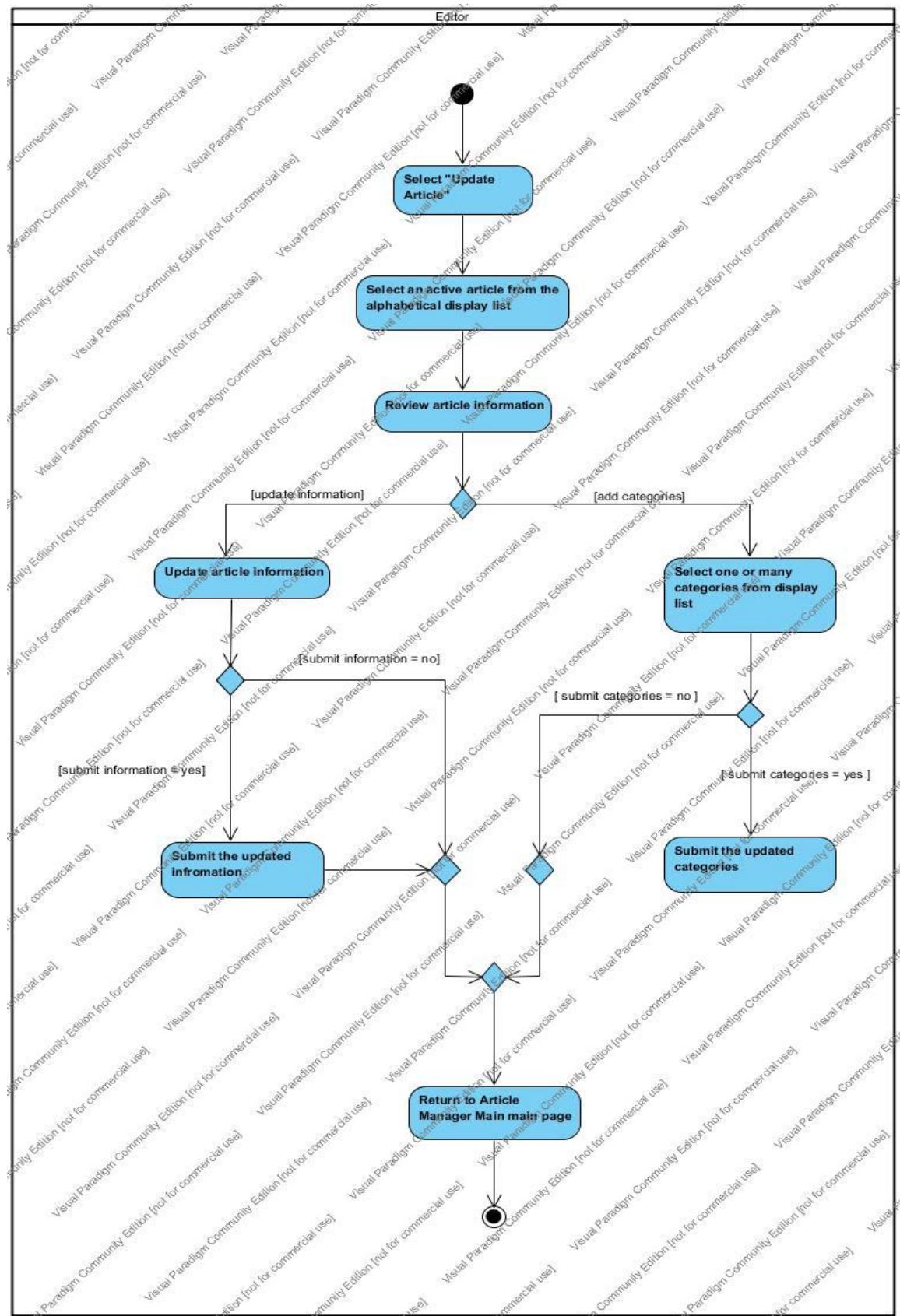
## Assign Reviewer



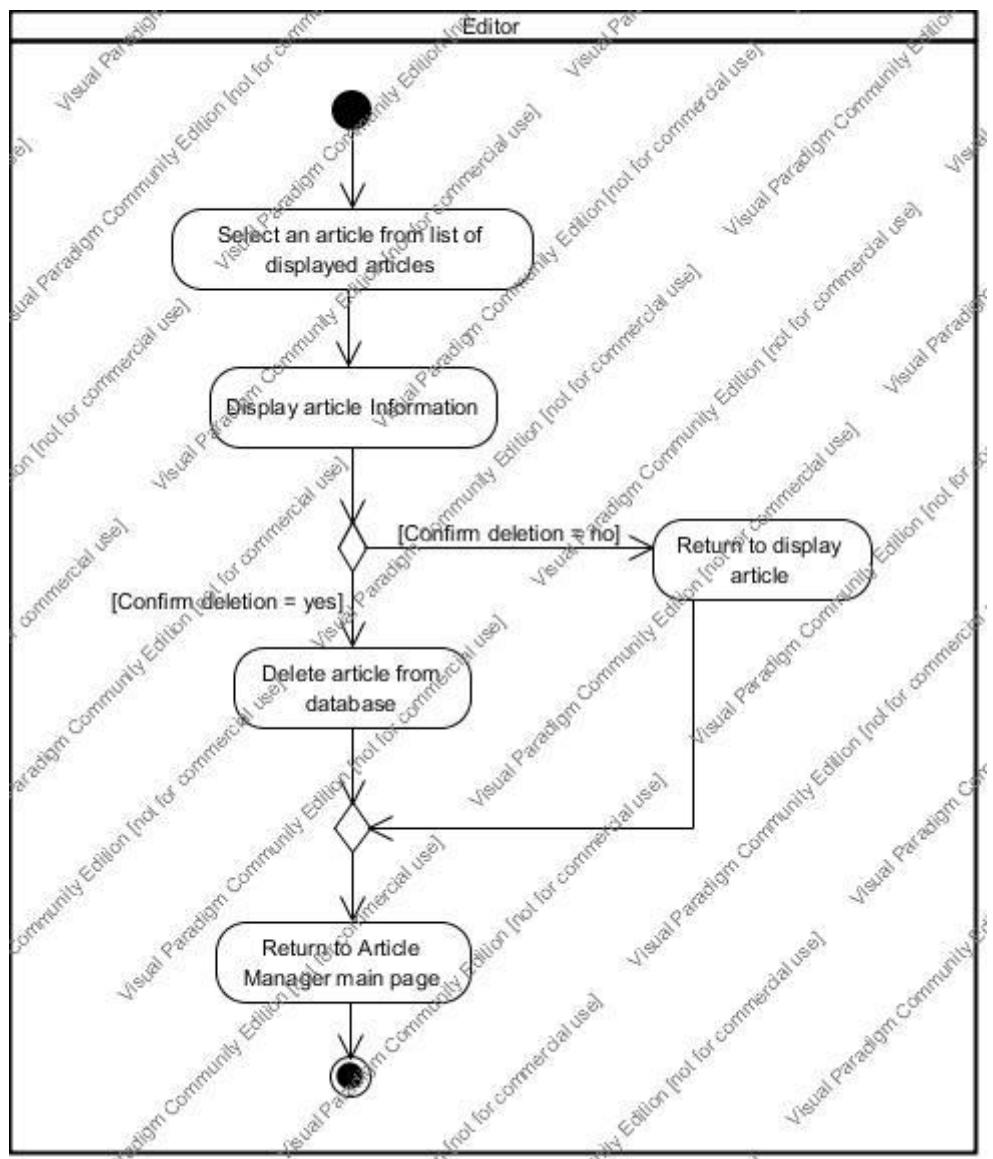
## Receive Review



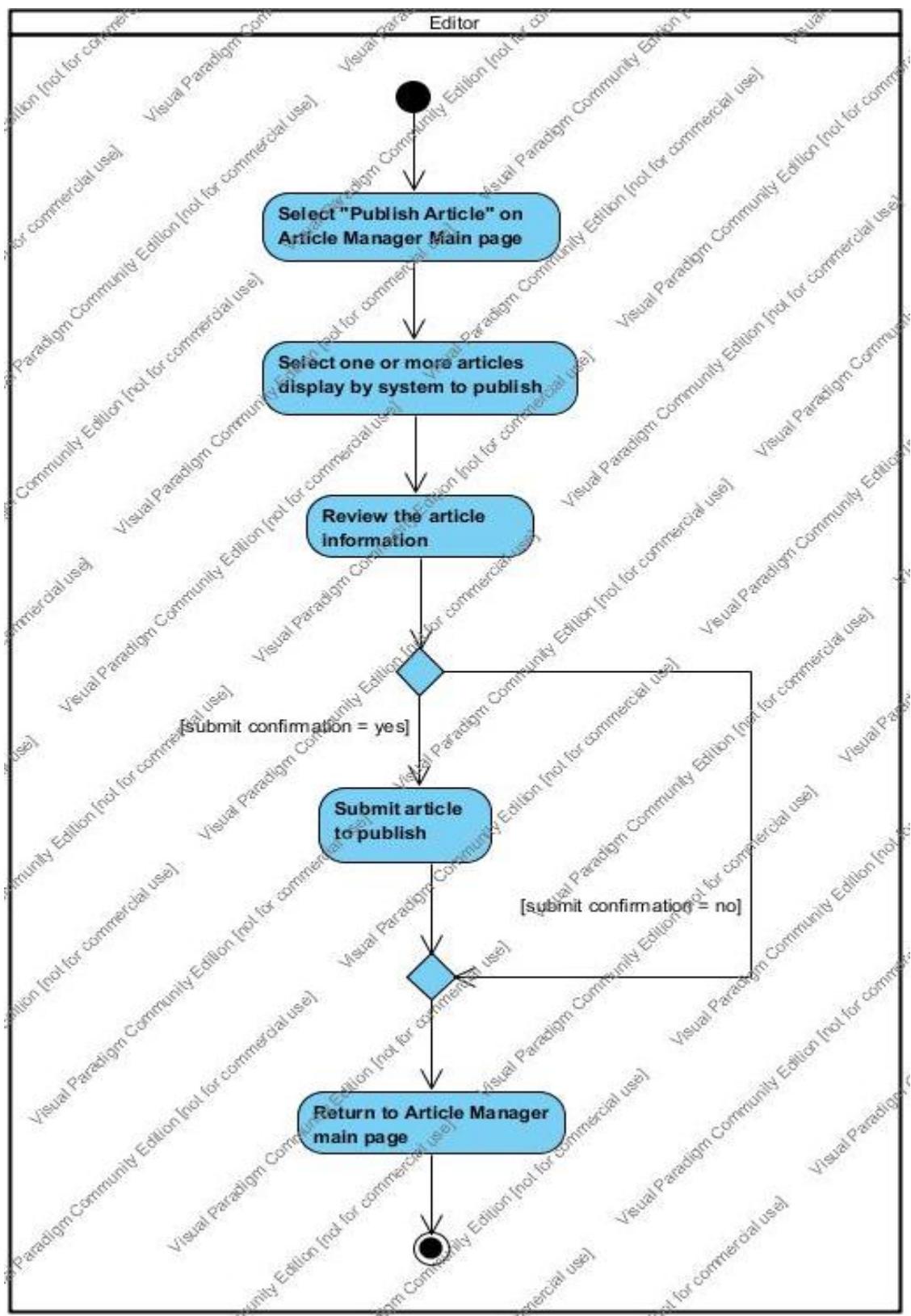
## Update Article



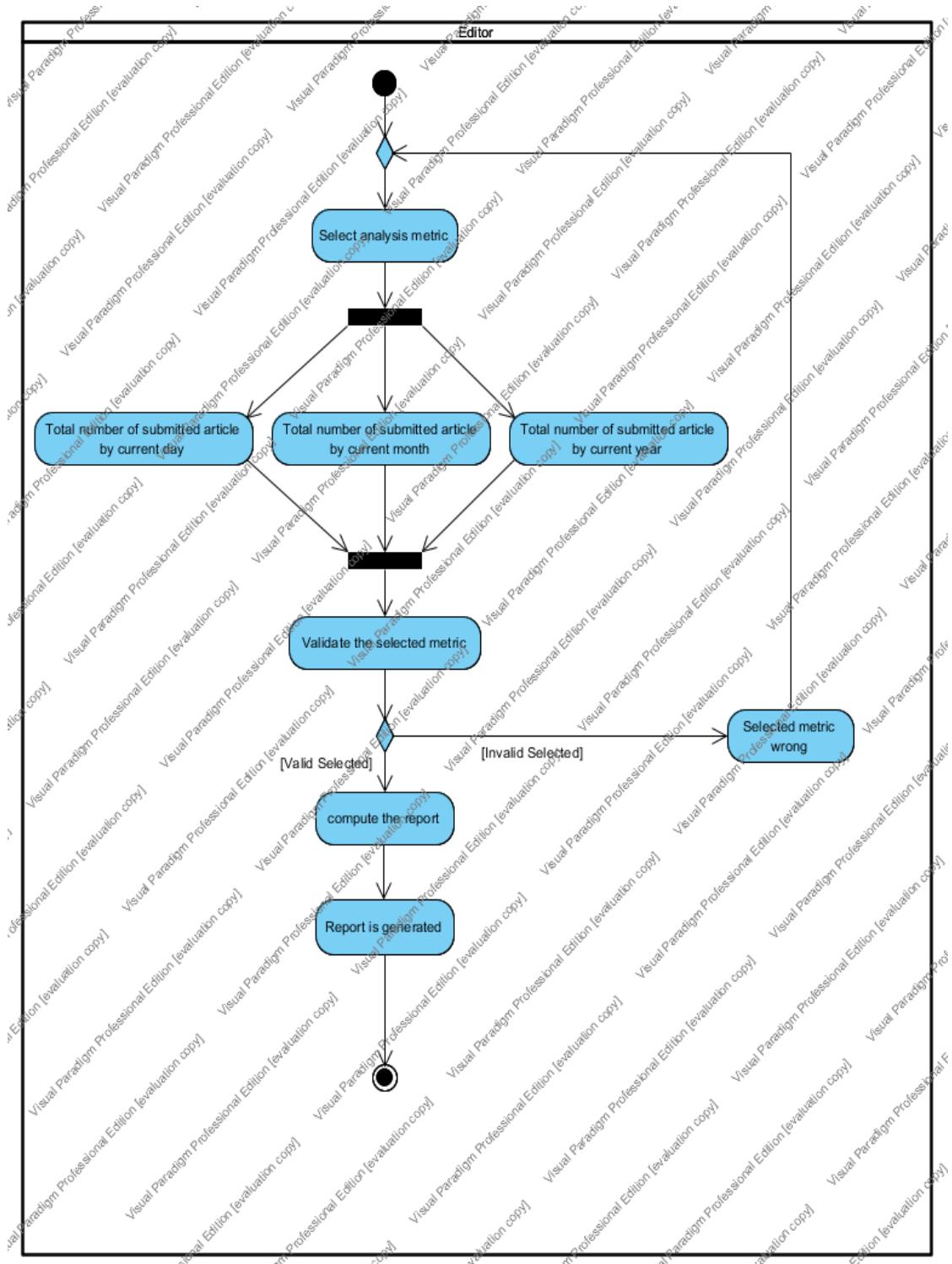
## Remove Article



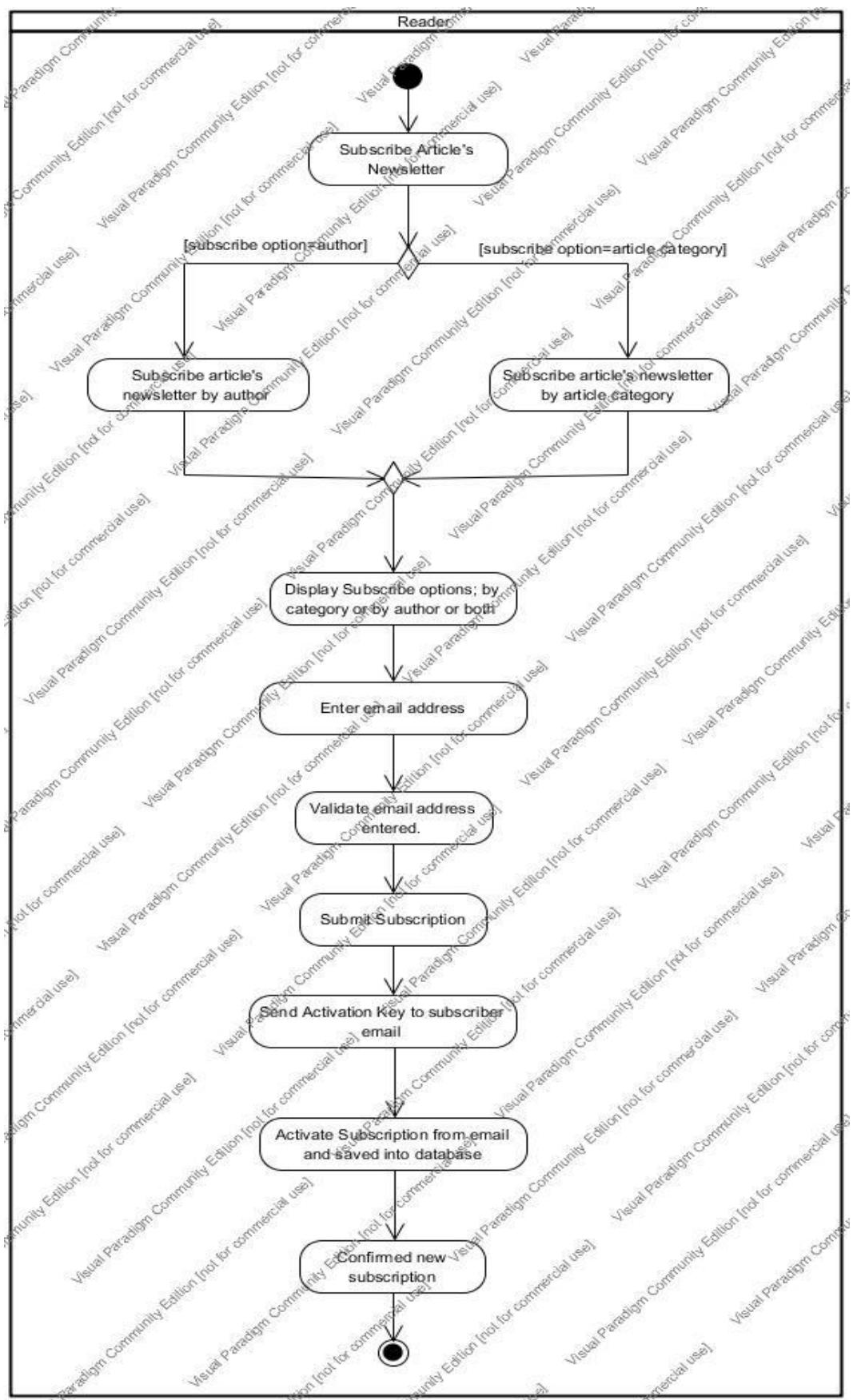
## Publish Article



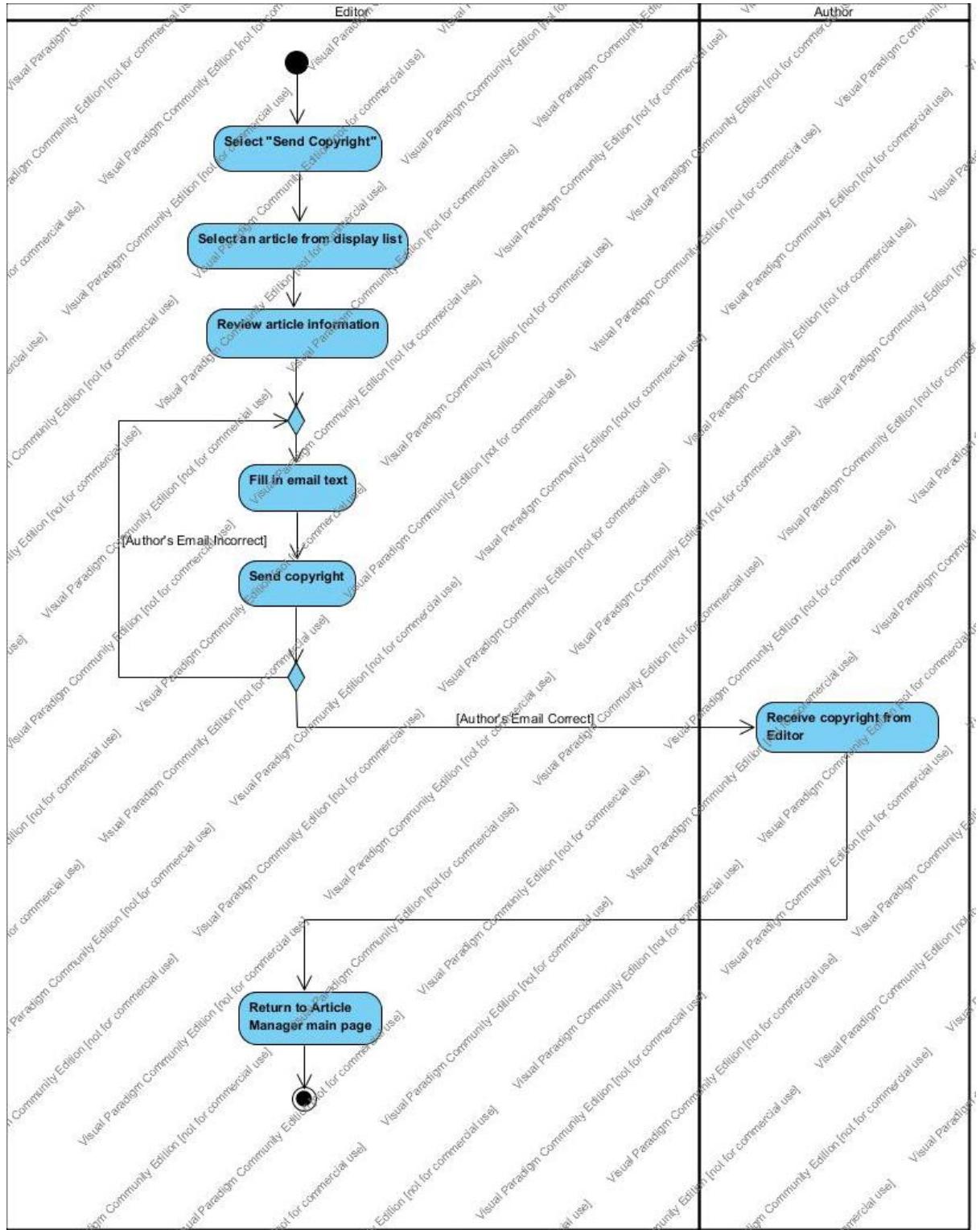
## Generate Statistic Report



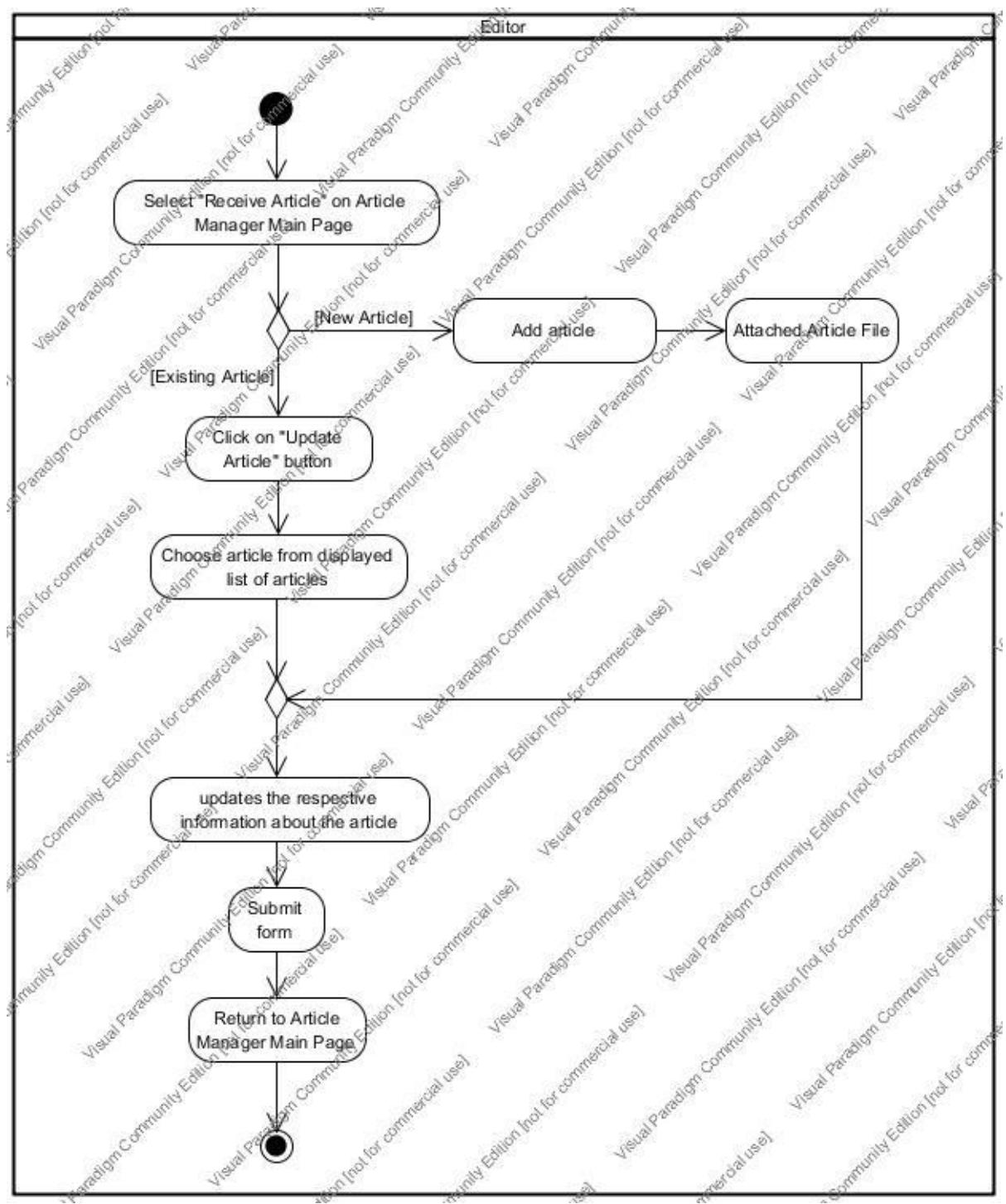
## Subscribe Article's Newsletter



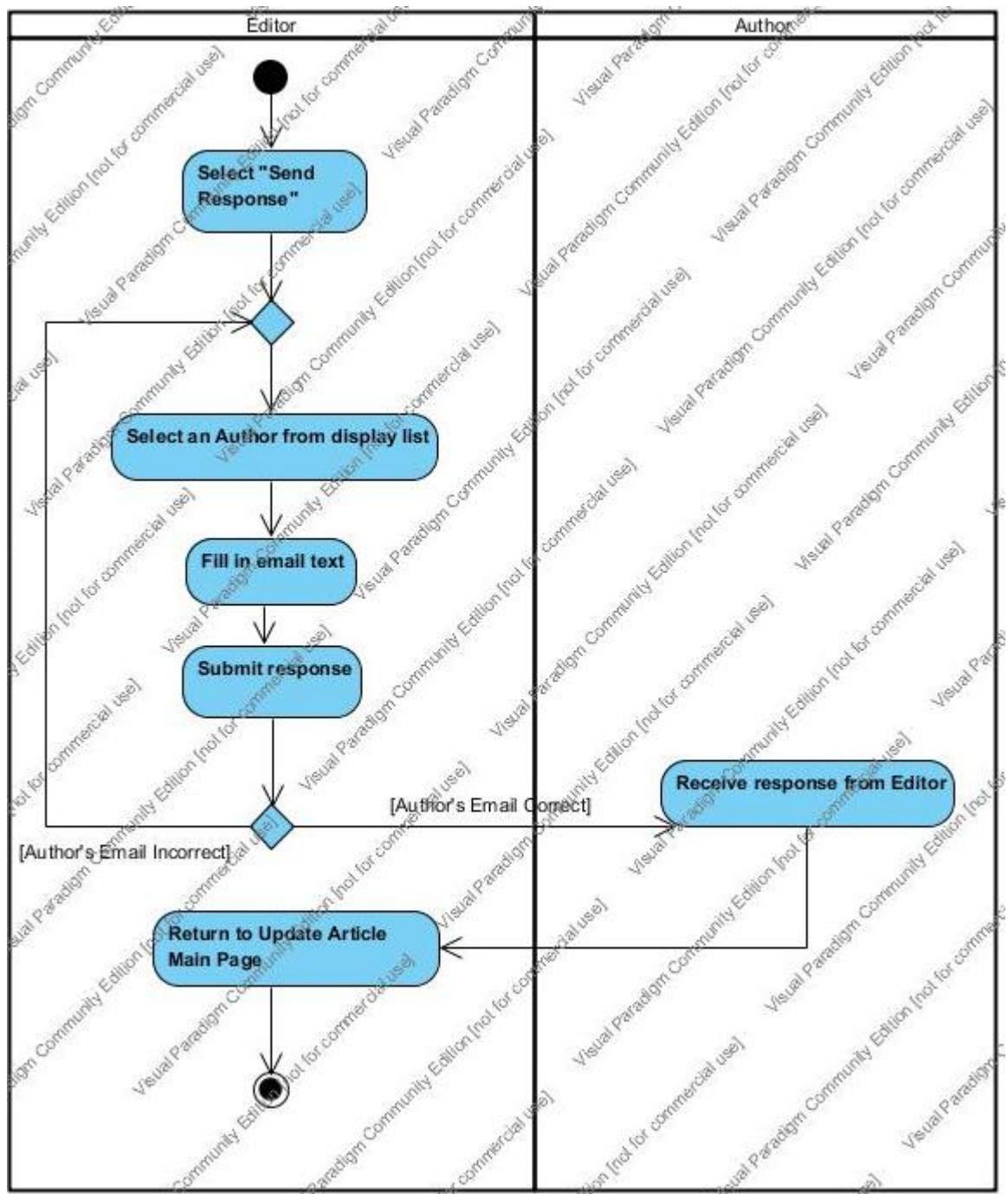
## Send Copyright



## Receive Article



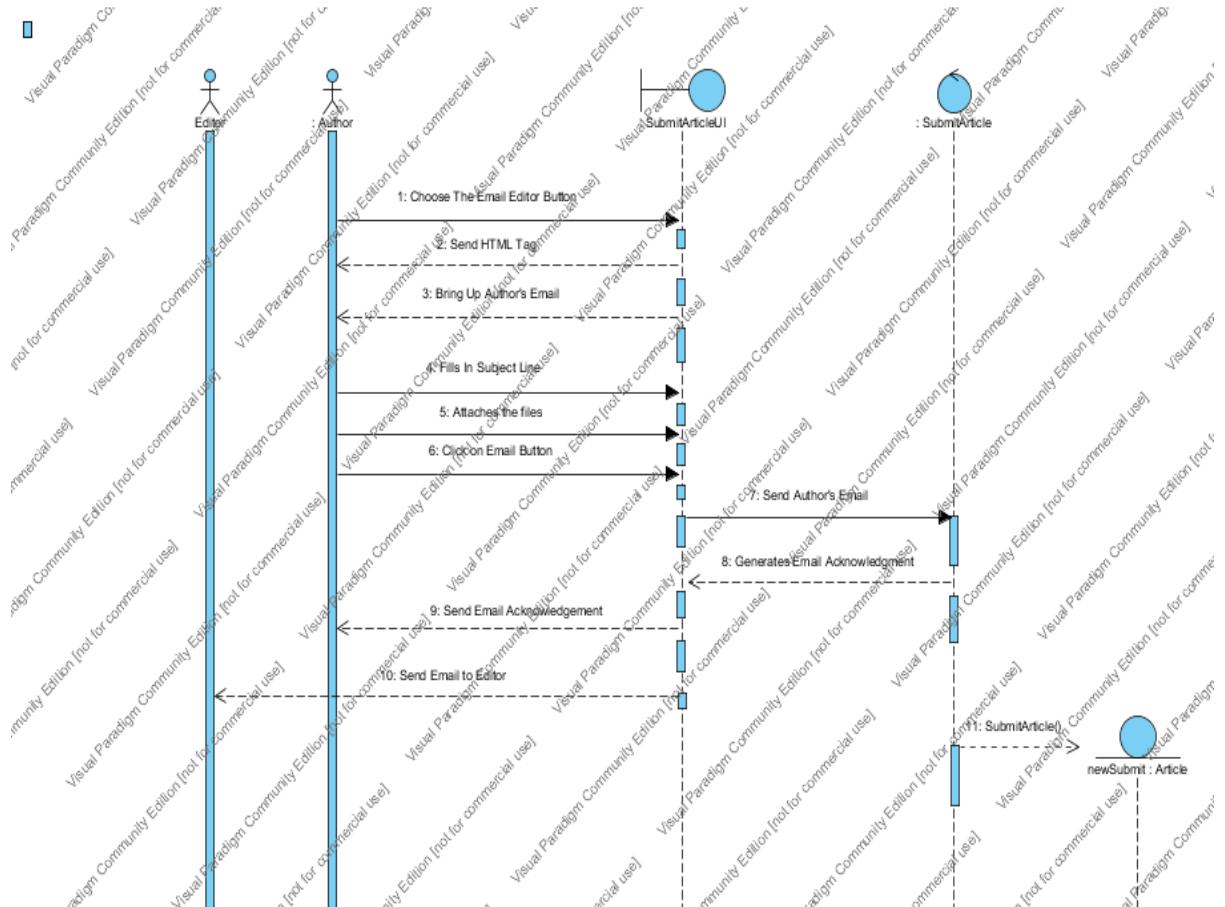
## Send Response



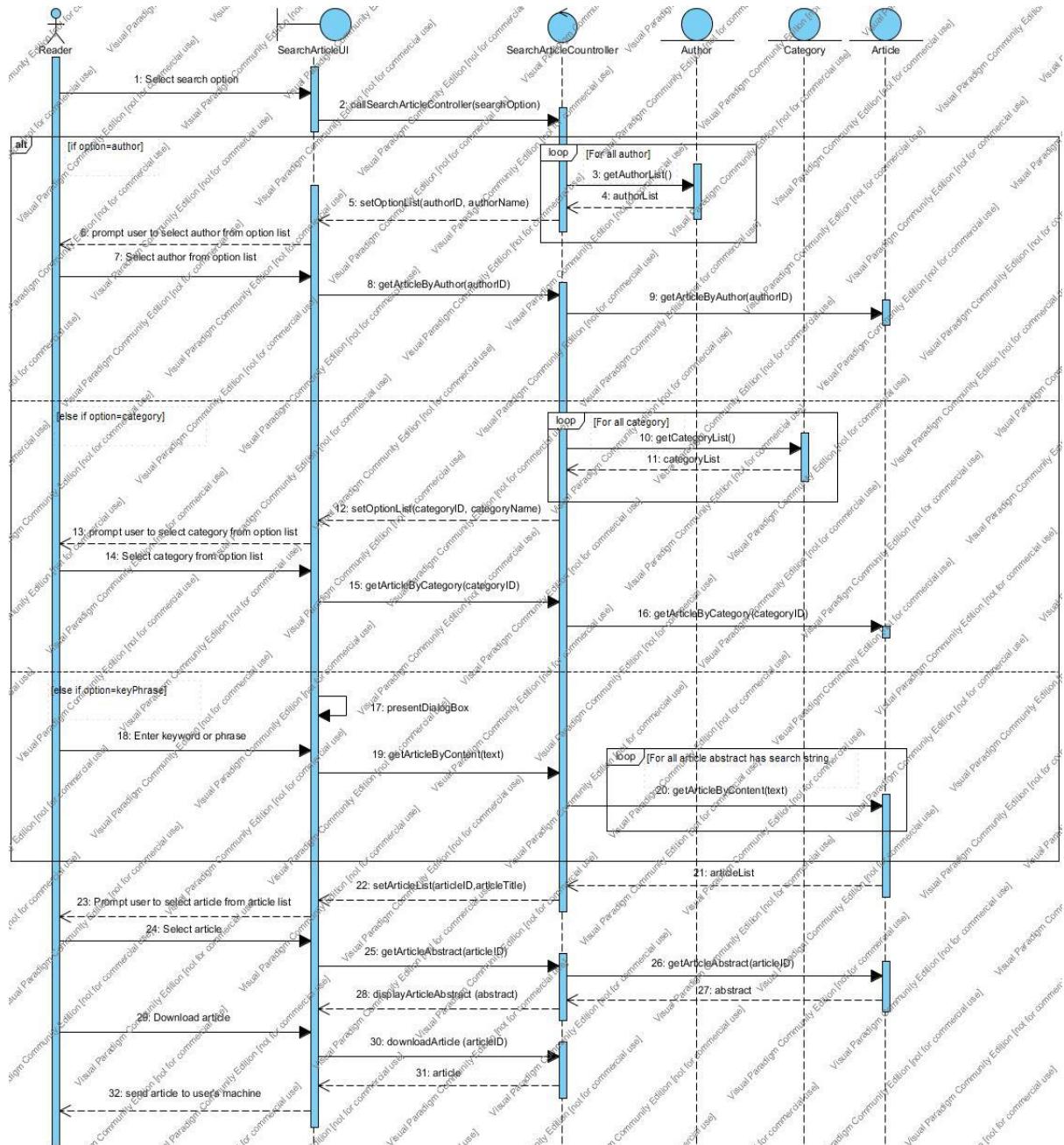
# Sequence & Communication Diagram

## Sequence Diagrams

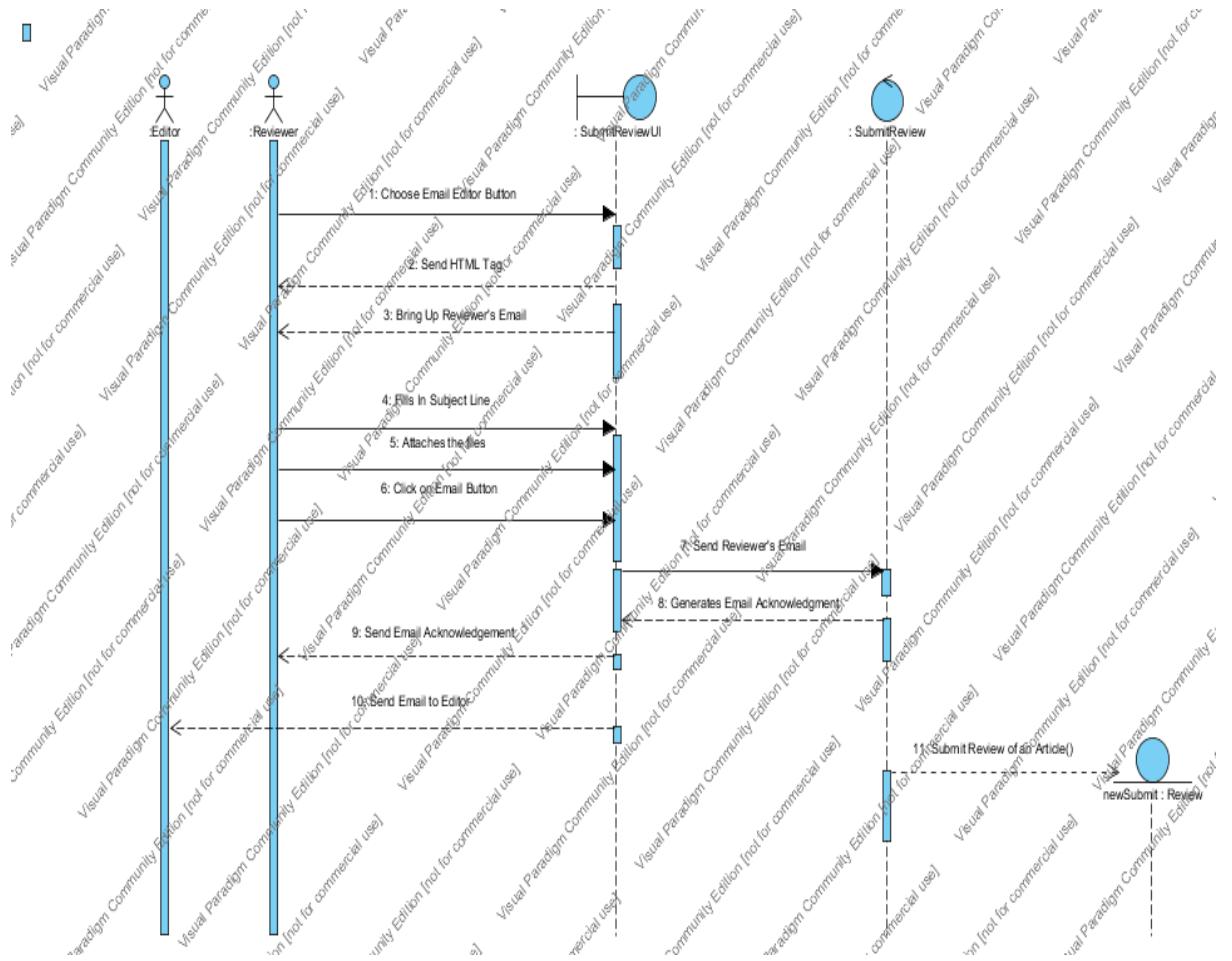
### Submit Article



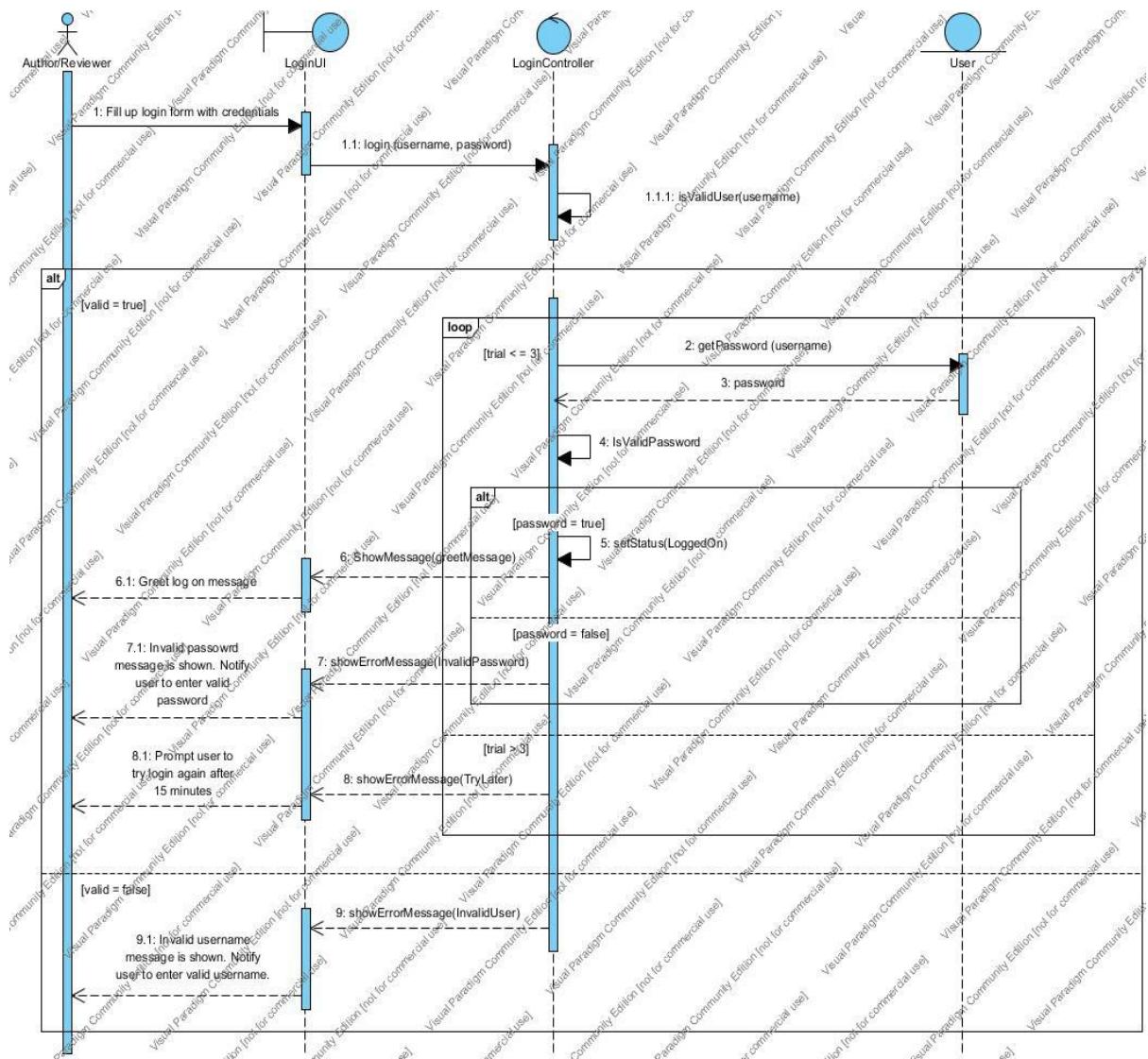
## Search Article



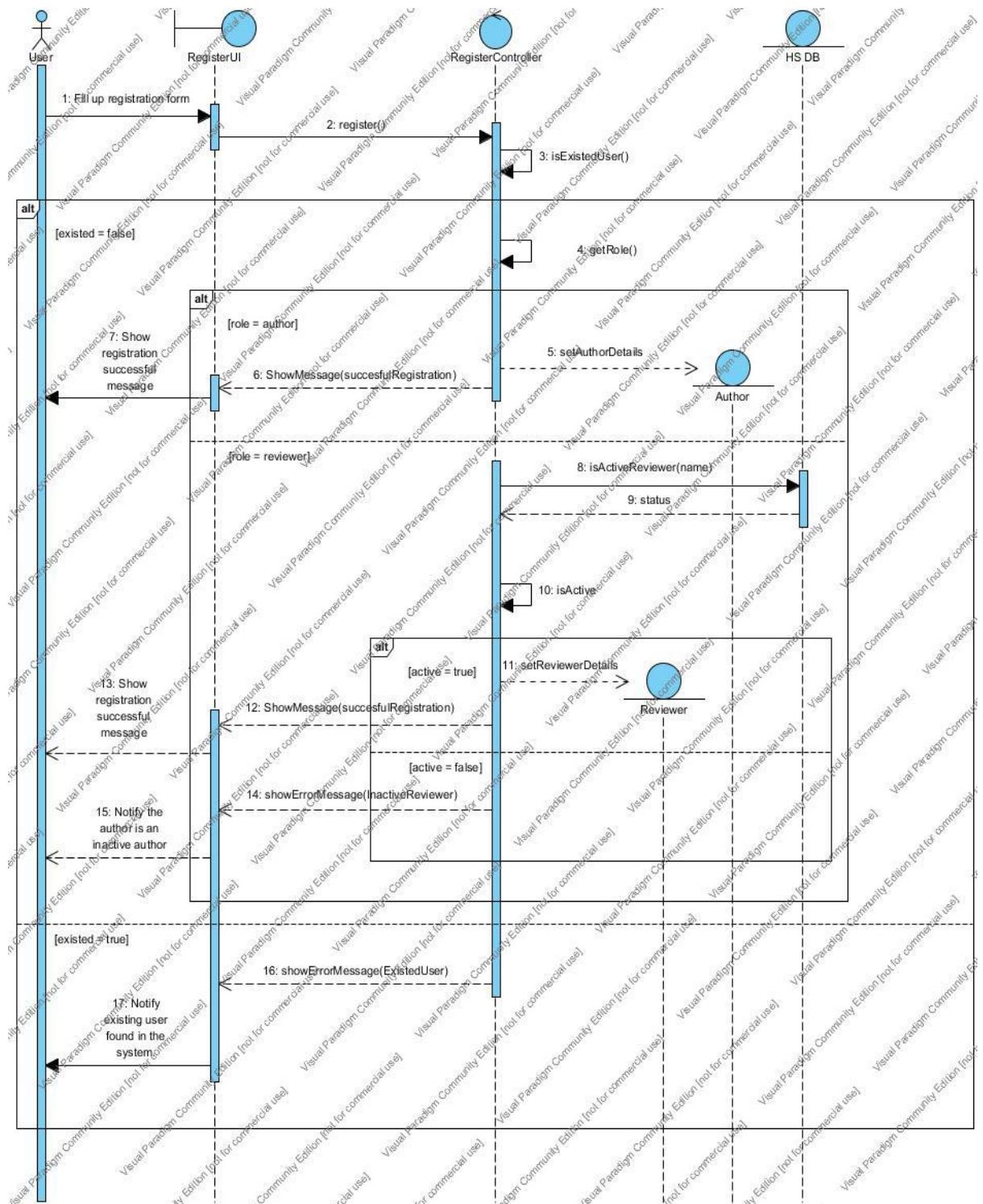
## Submit Review



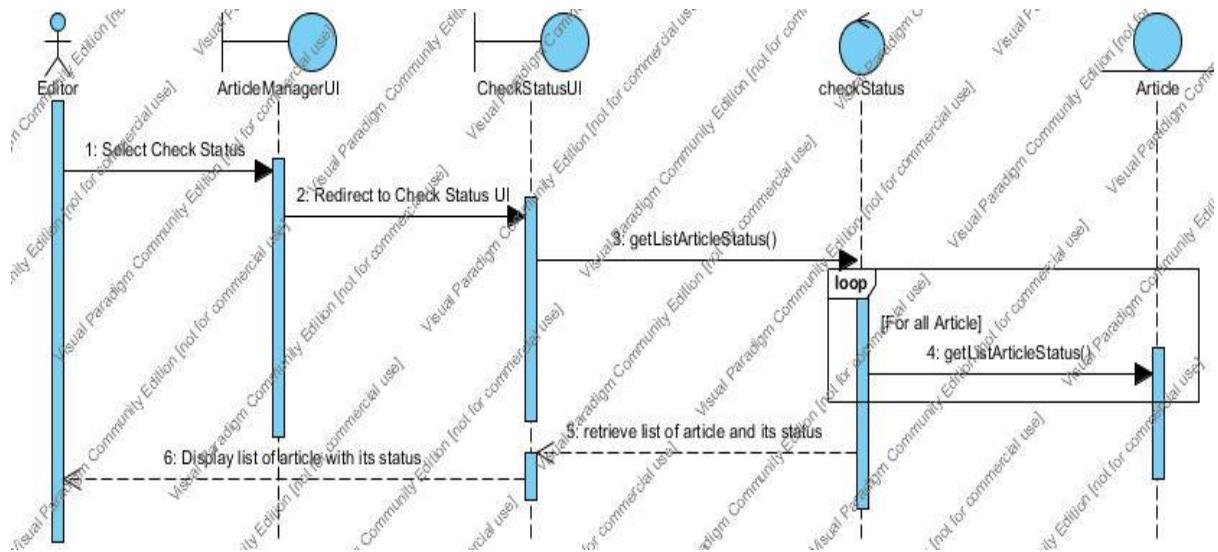
## Login



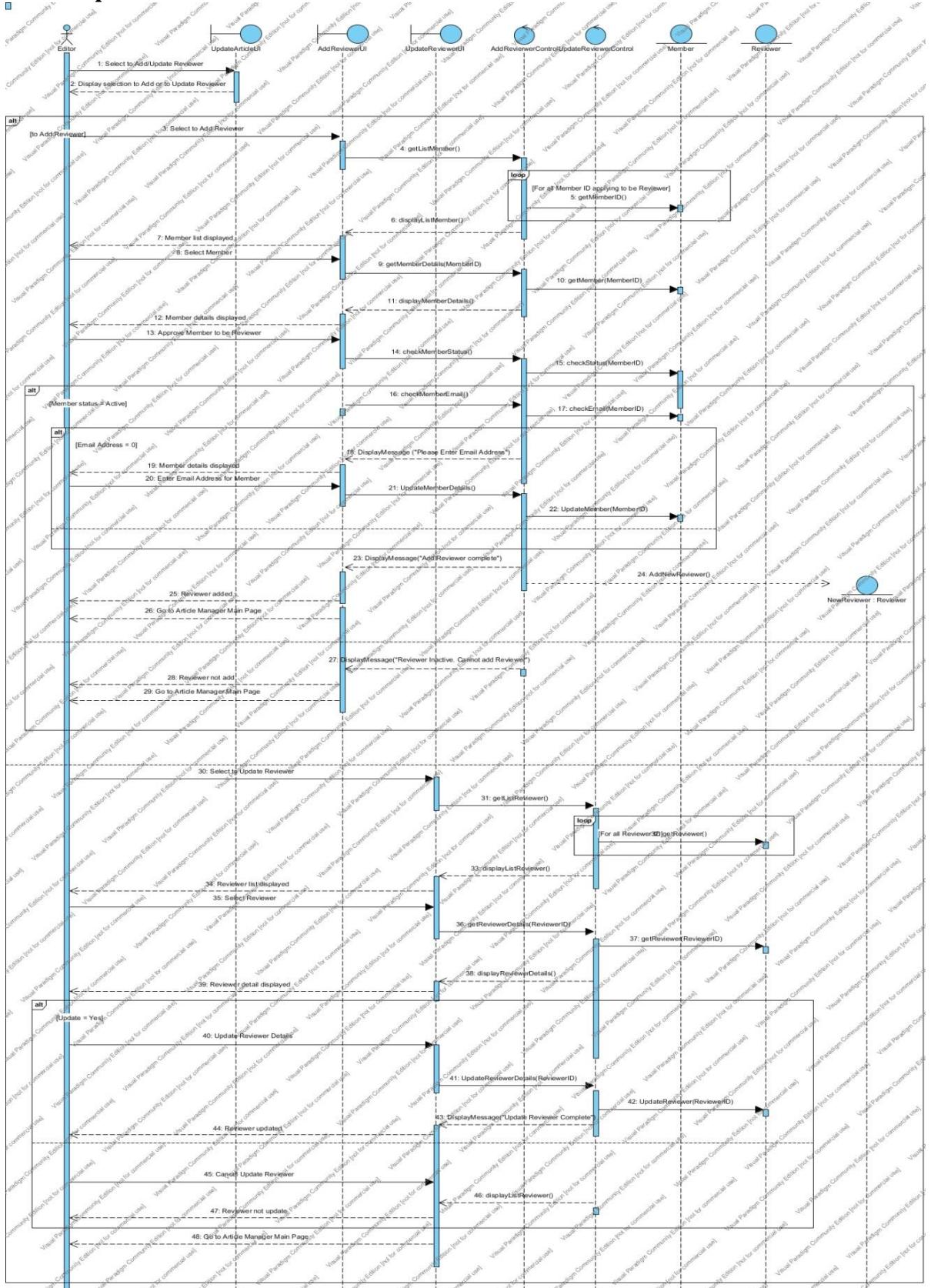
## Register



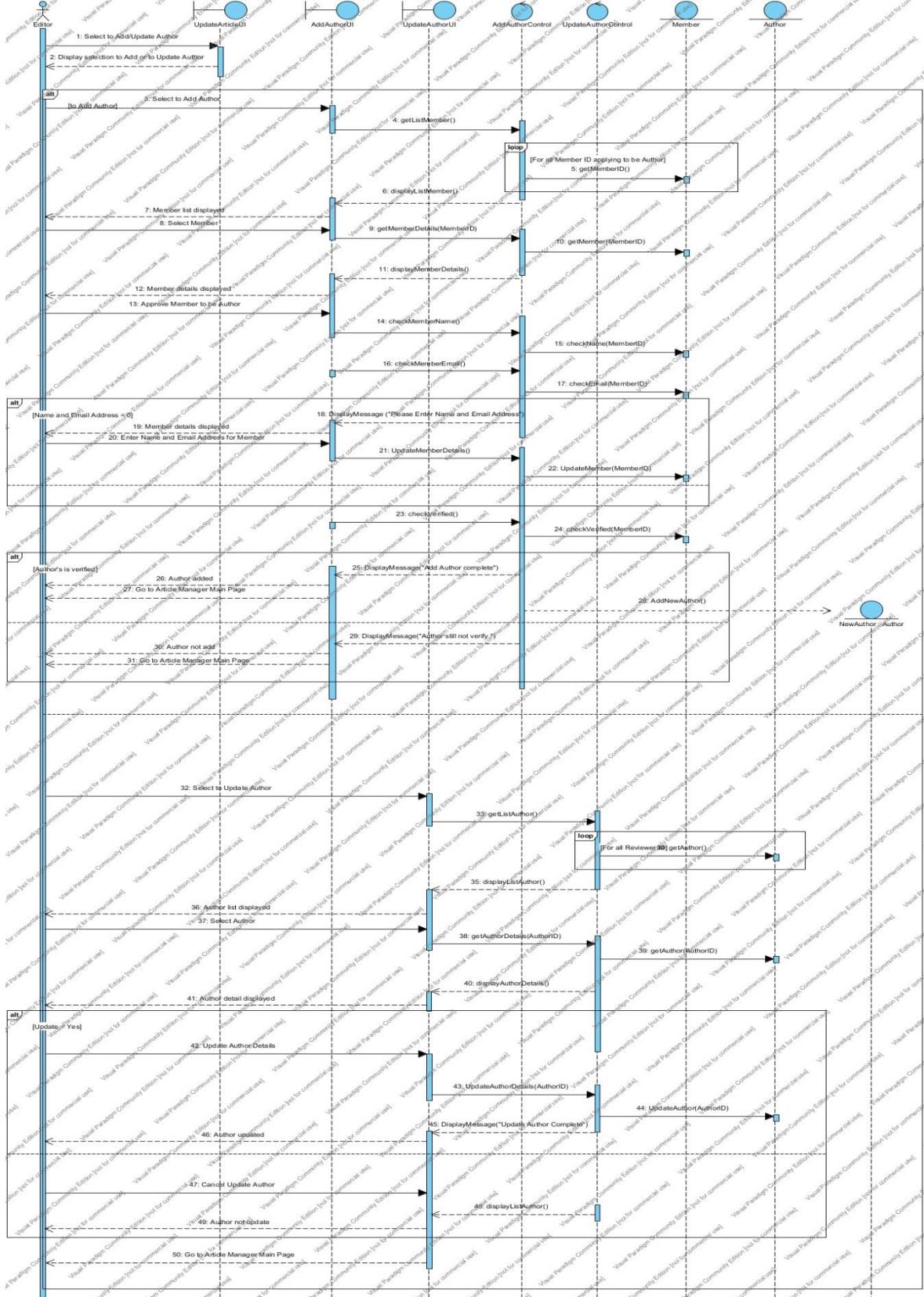
## Check Status



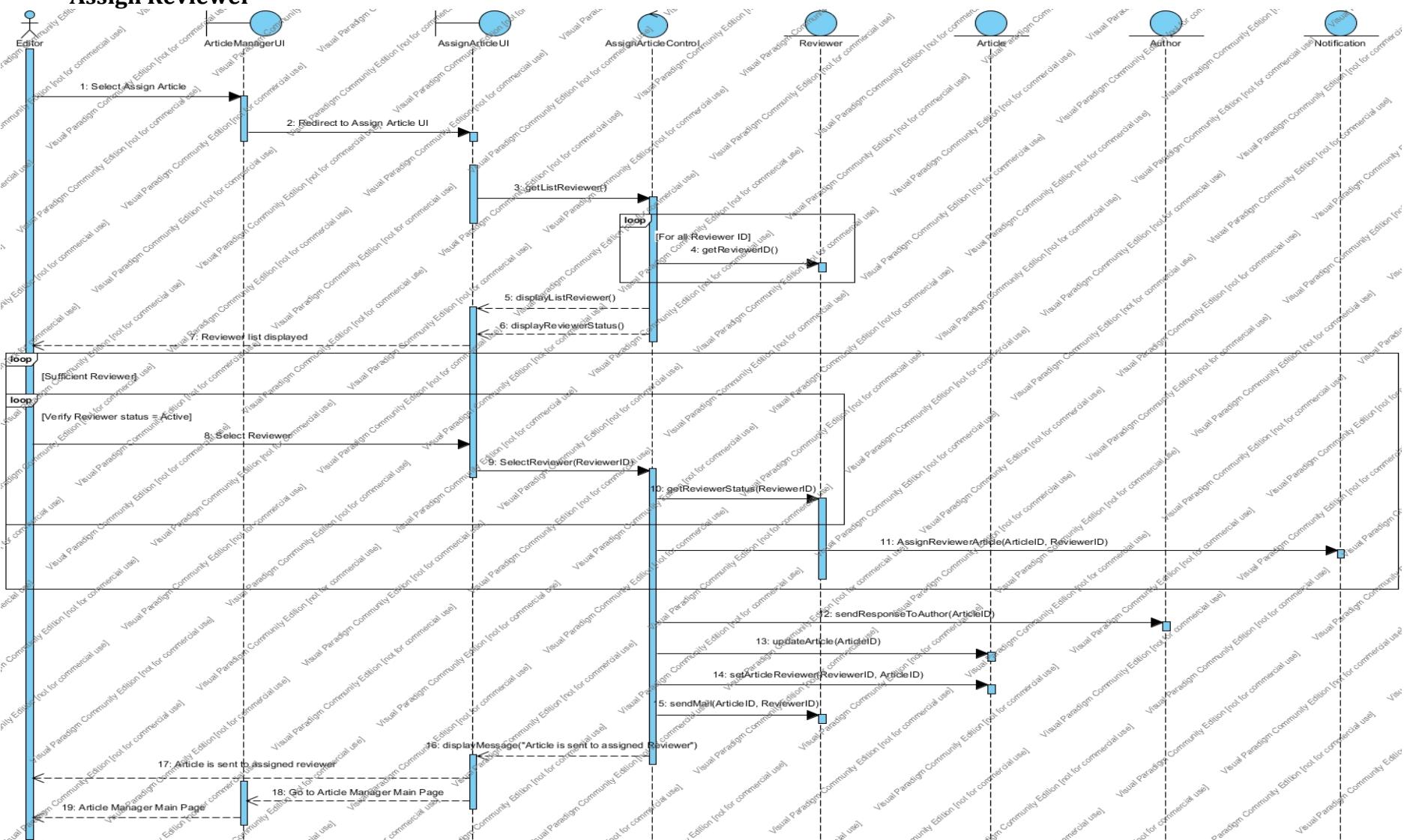
## Update Reviewer

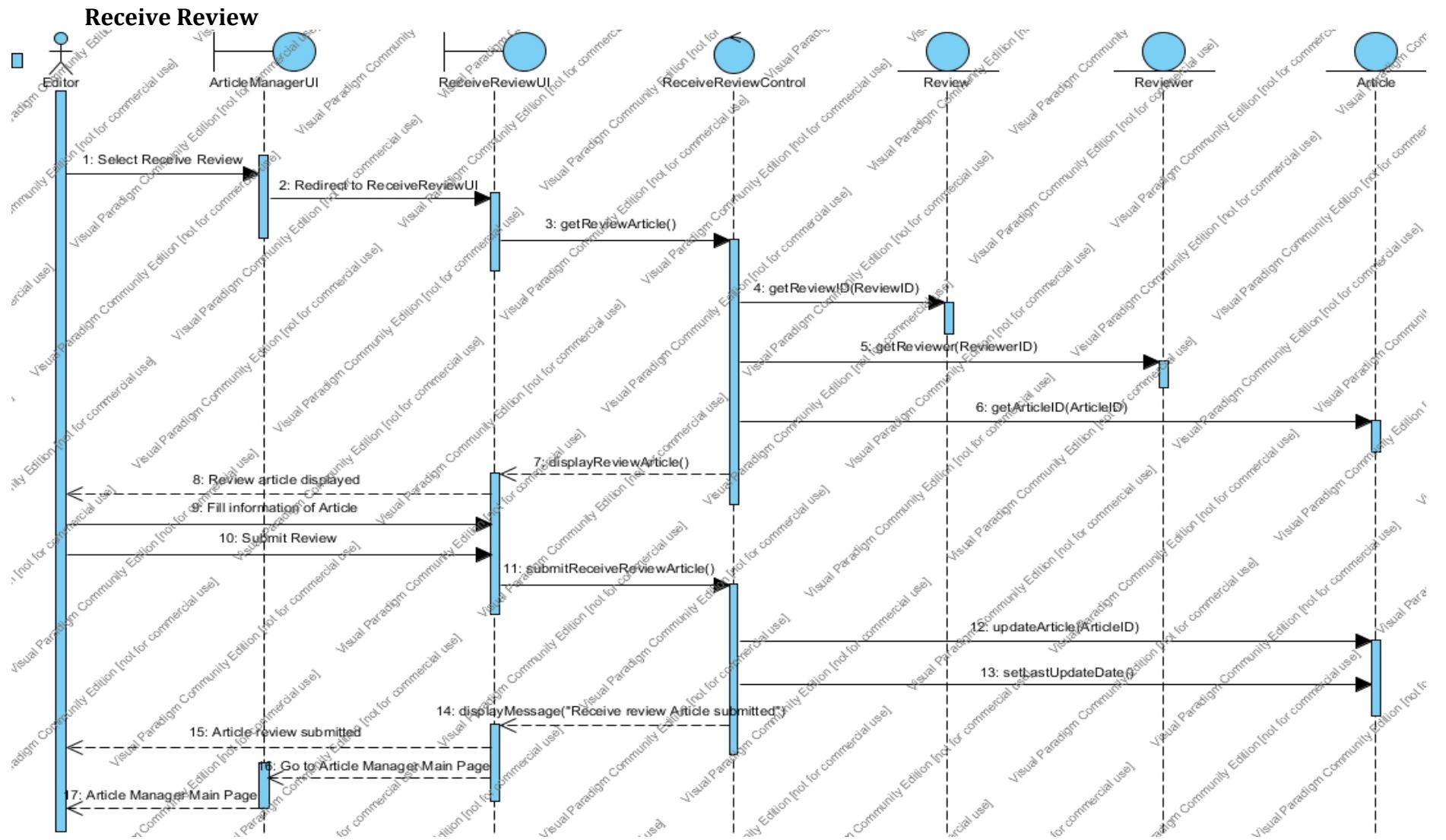


## Update Author

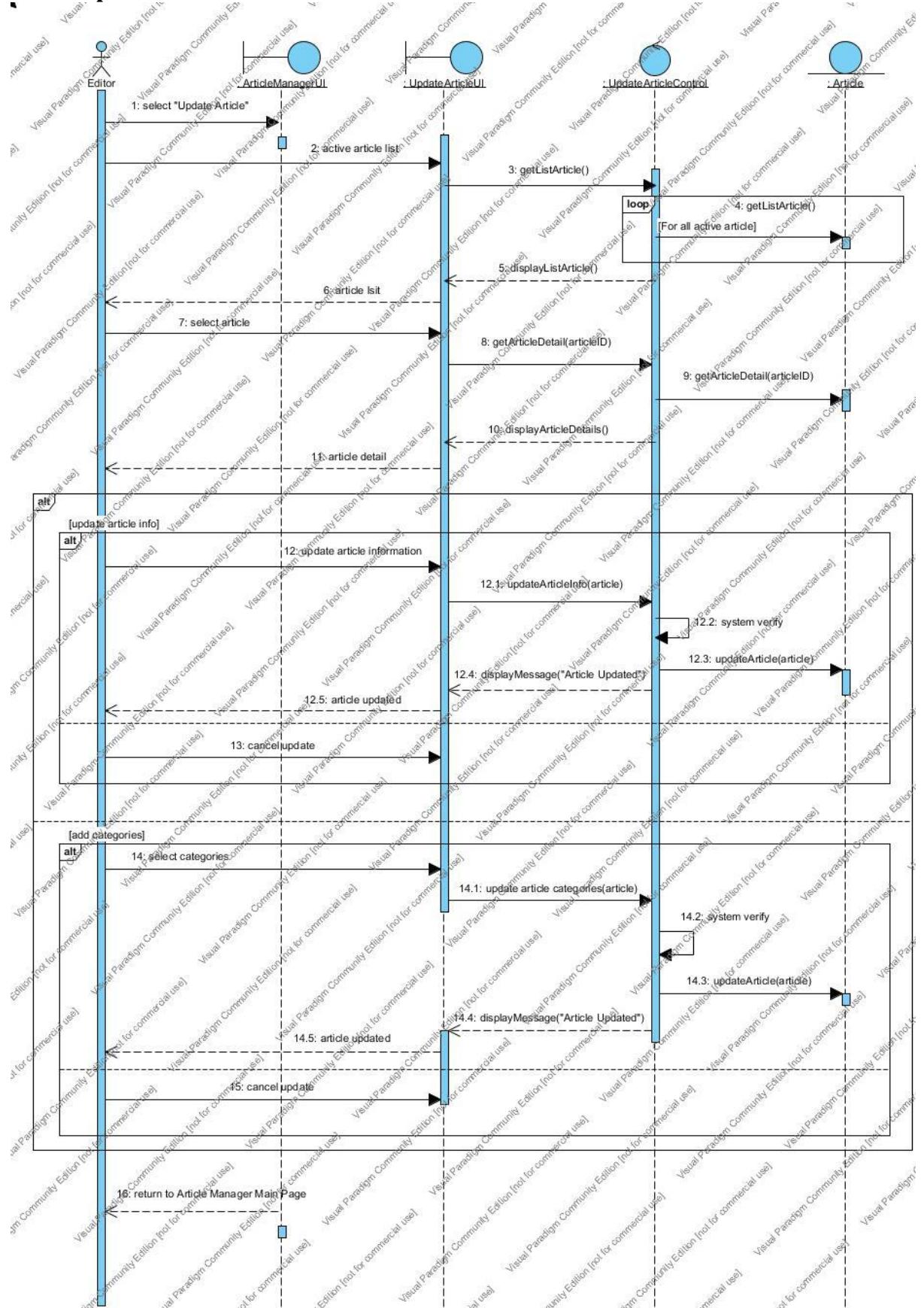


## Assign Reviewer

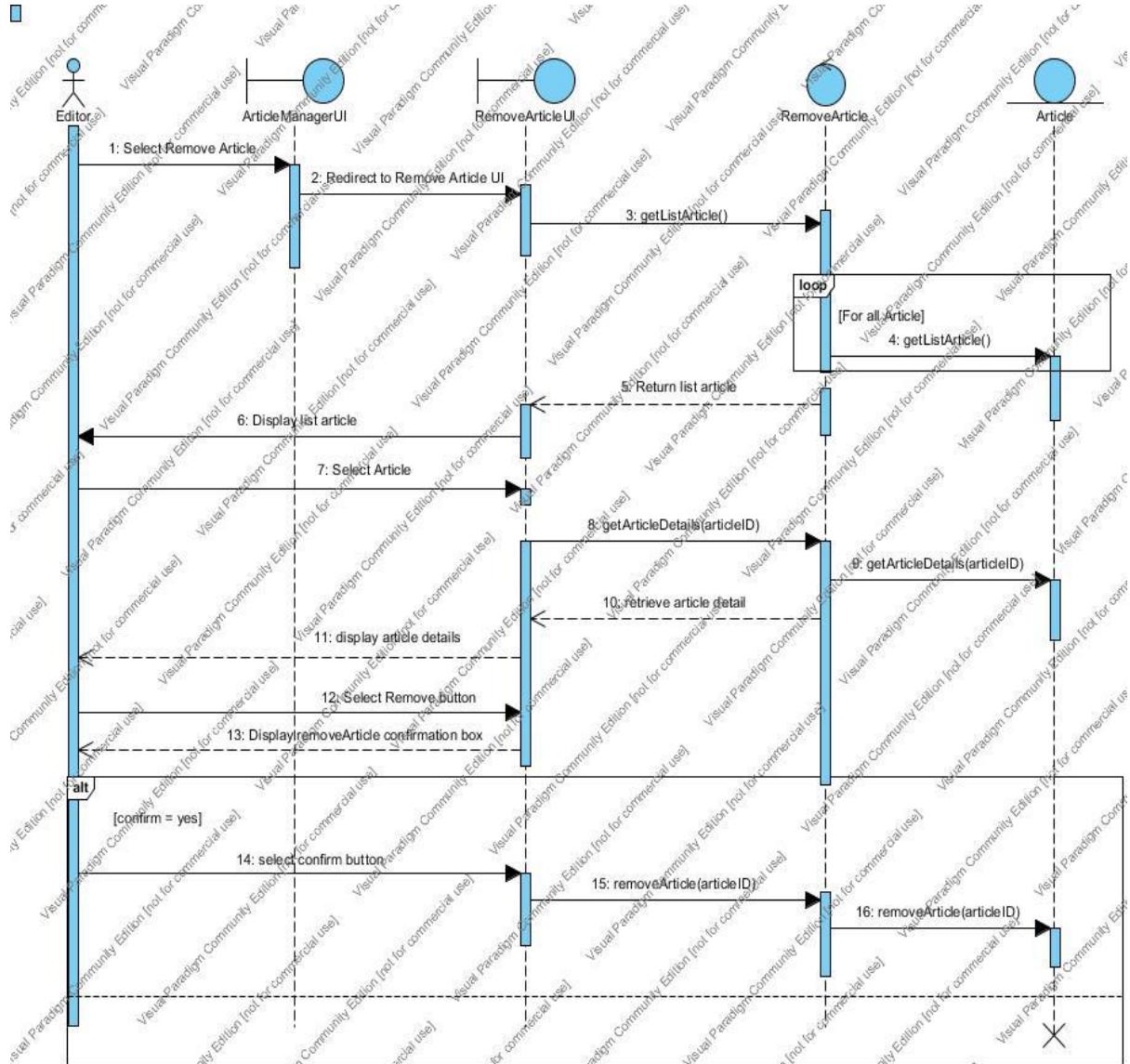




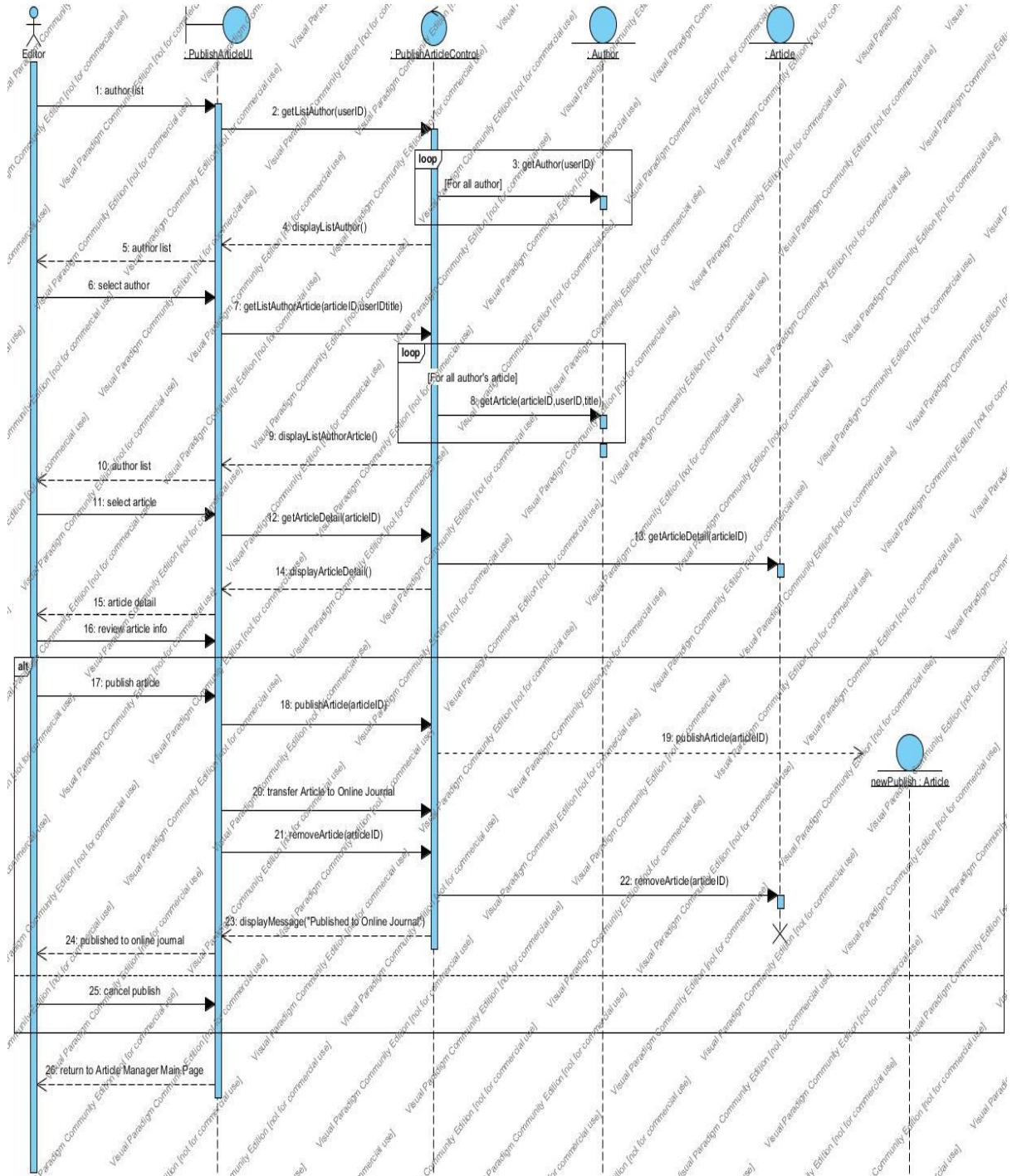
## Update Article



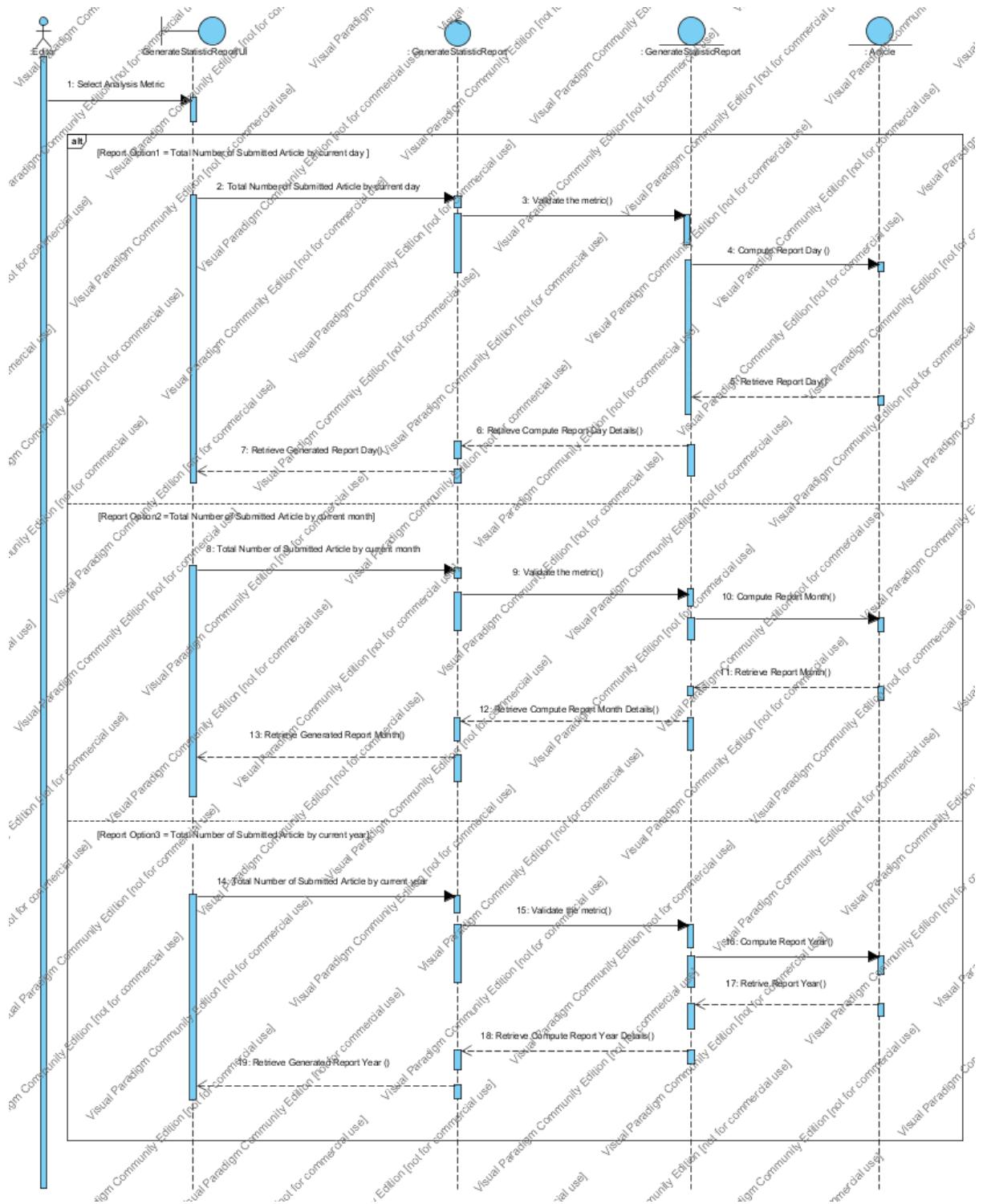
## Remove Article



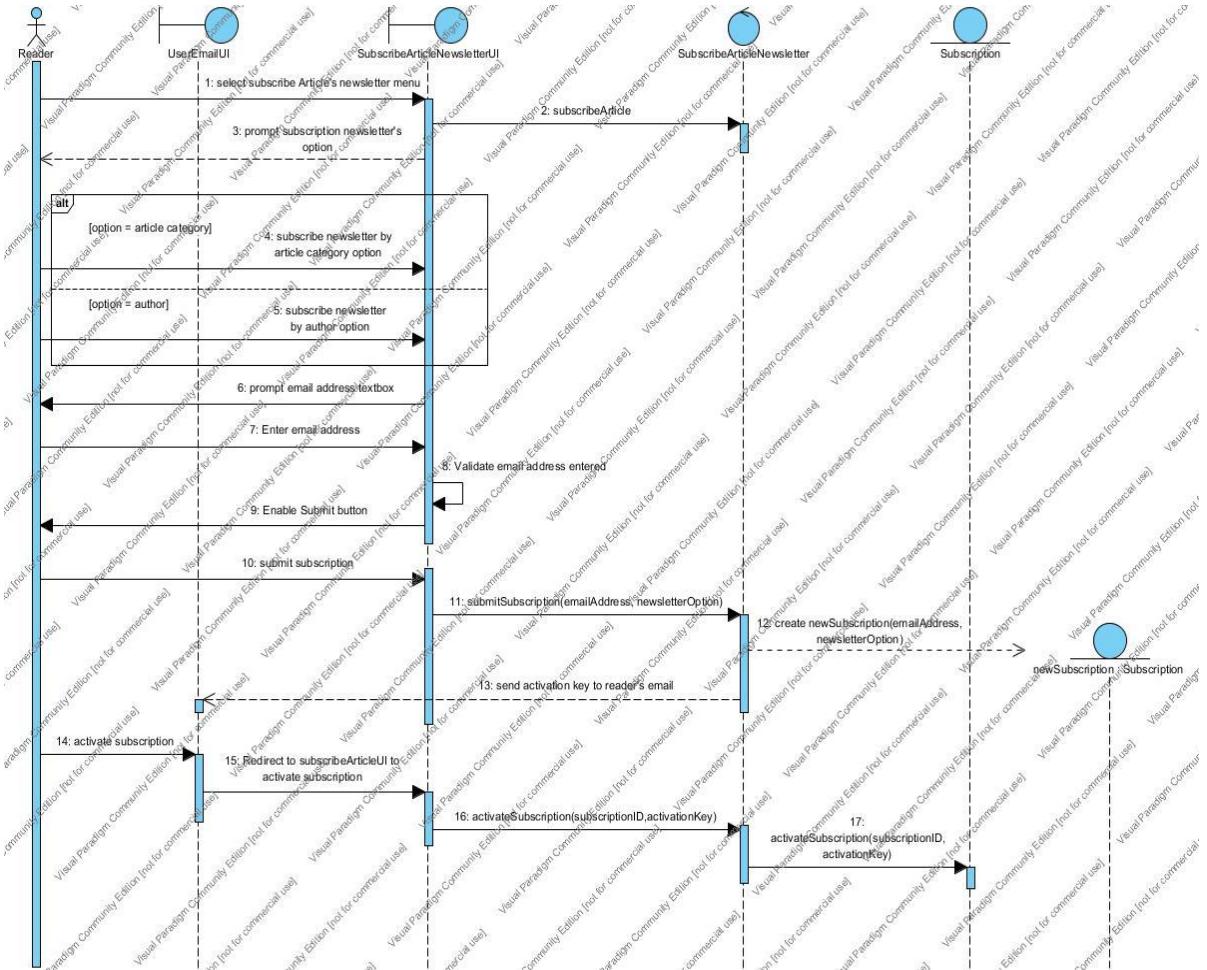
## Publish Article



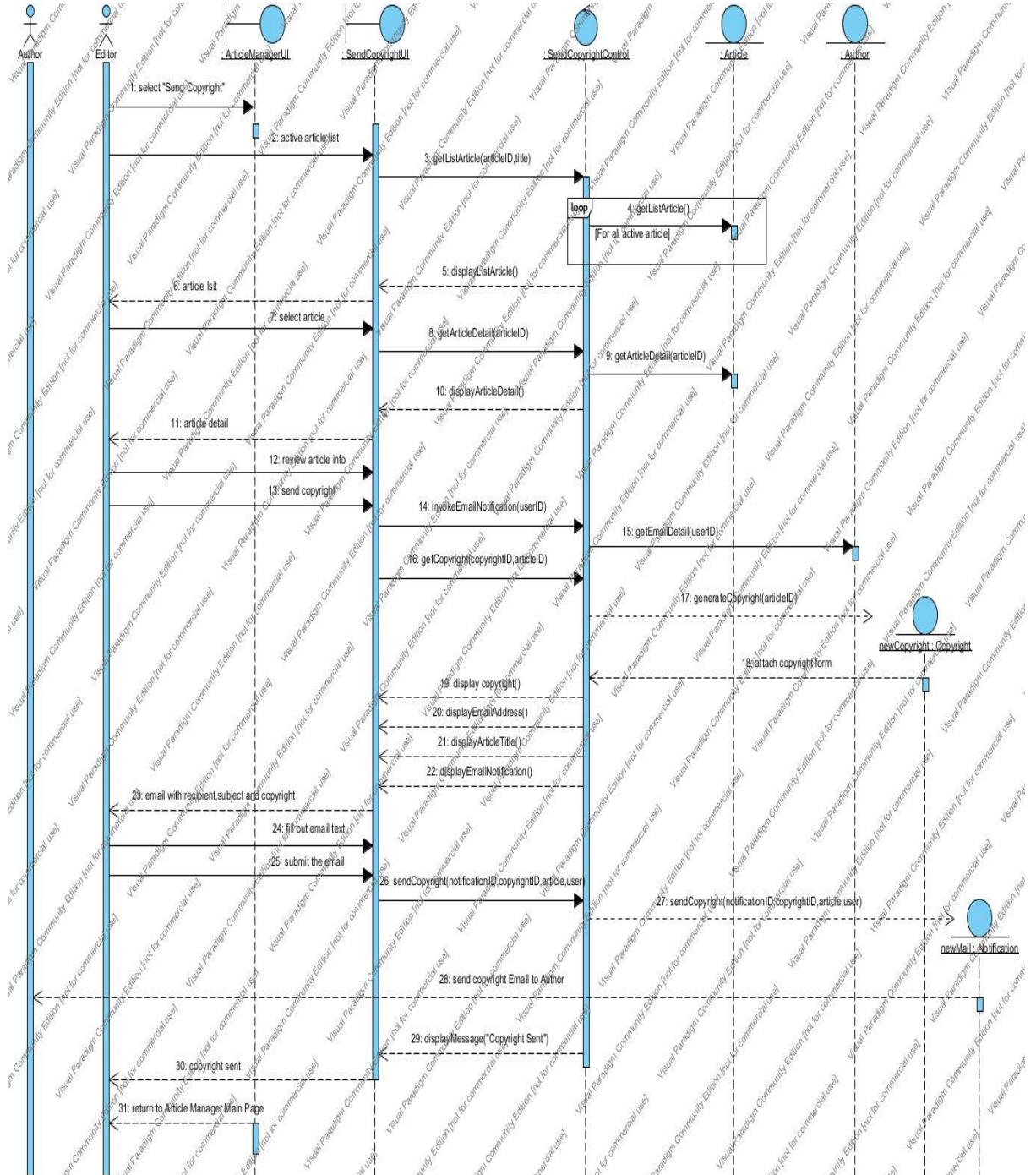
## Generate Statistic Report



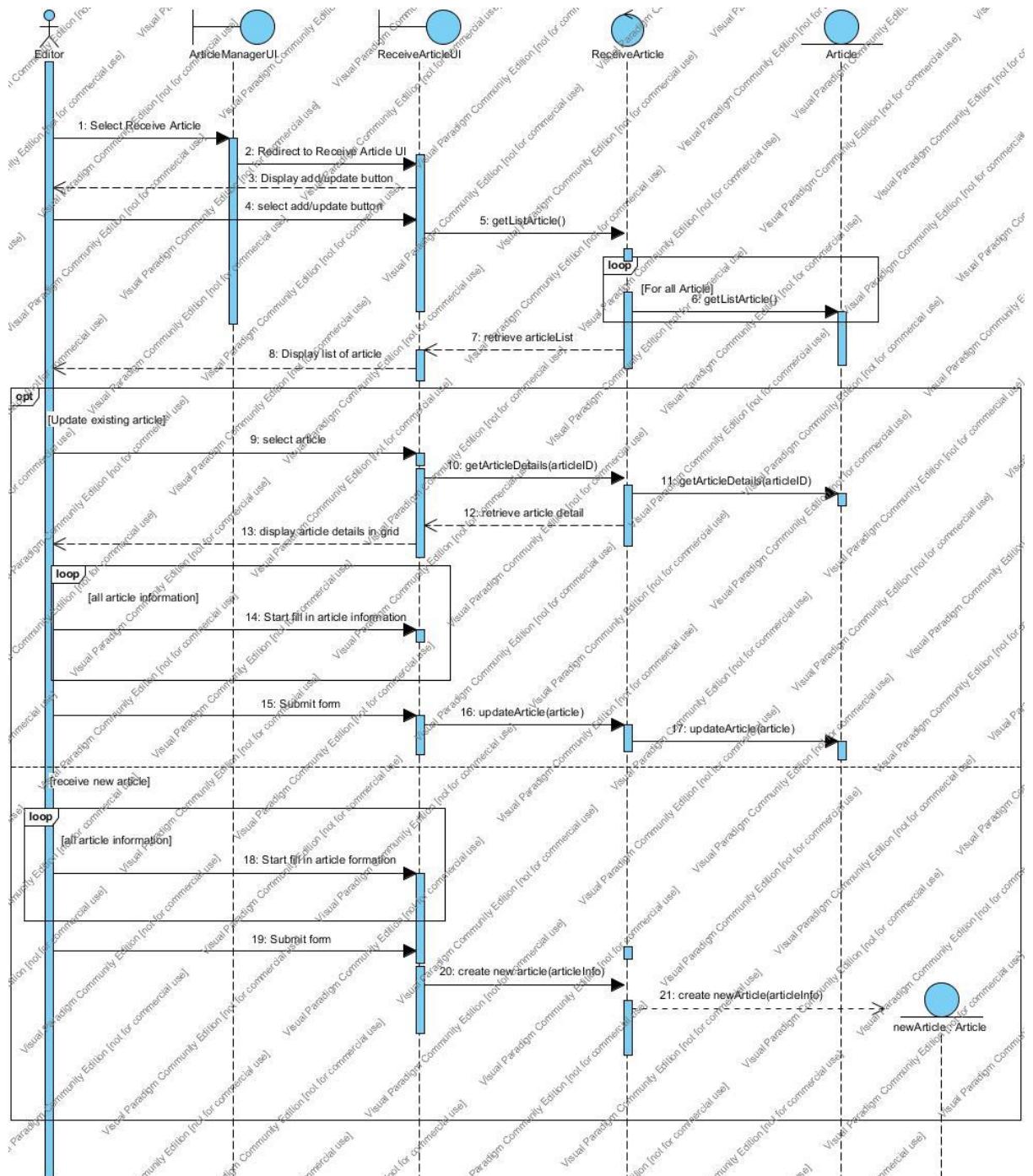
**Subscribe Article's Newsletter**



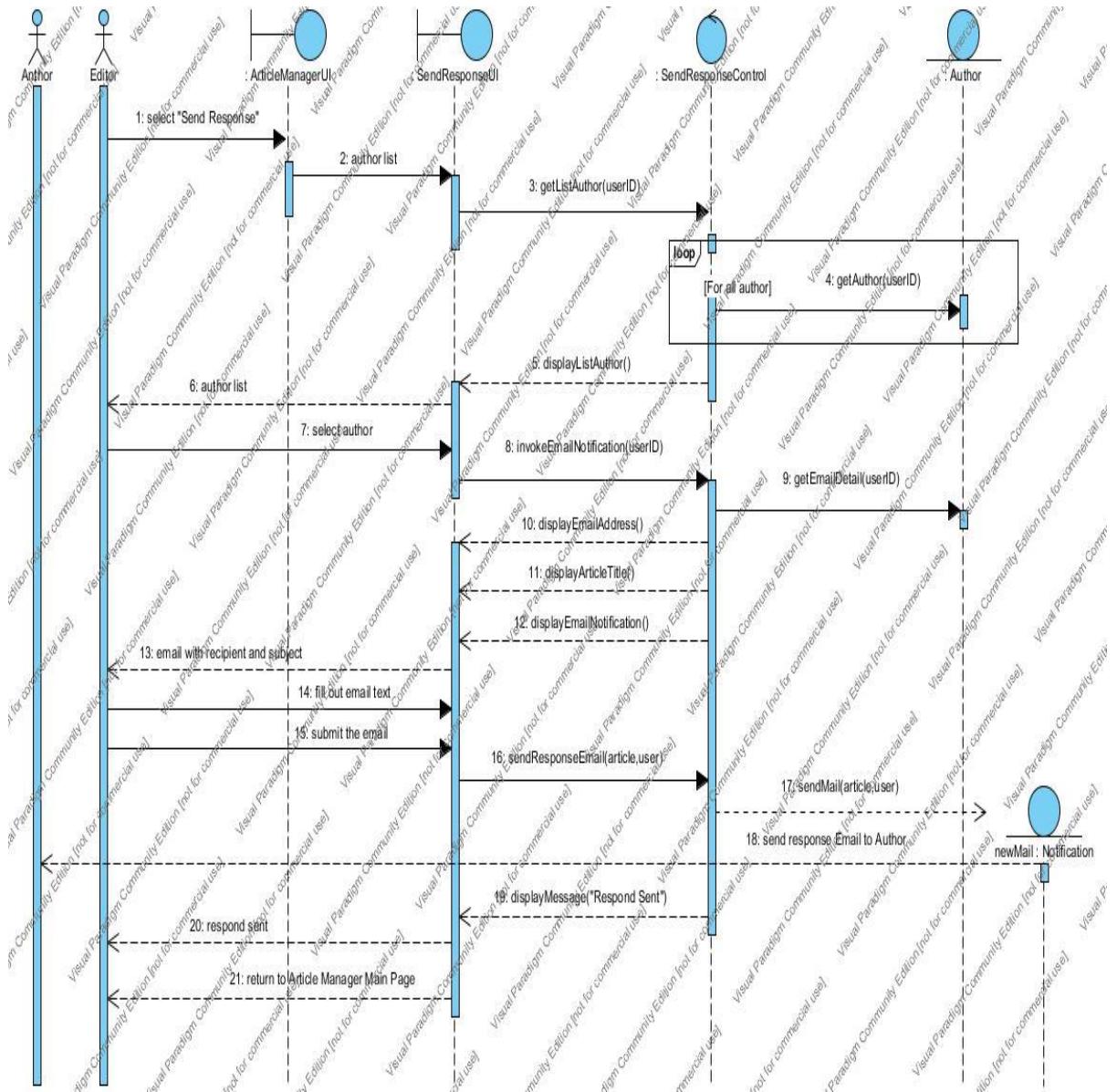
## Send Copyright



## Receive Article

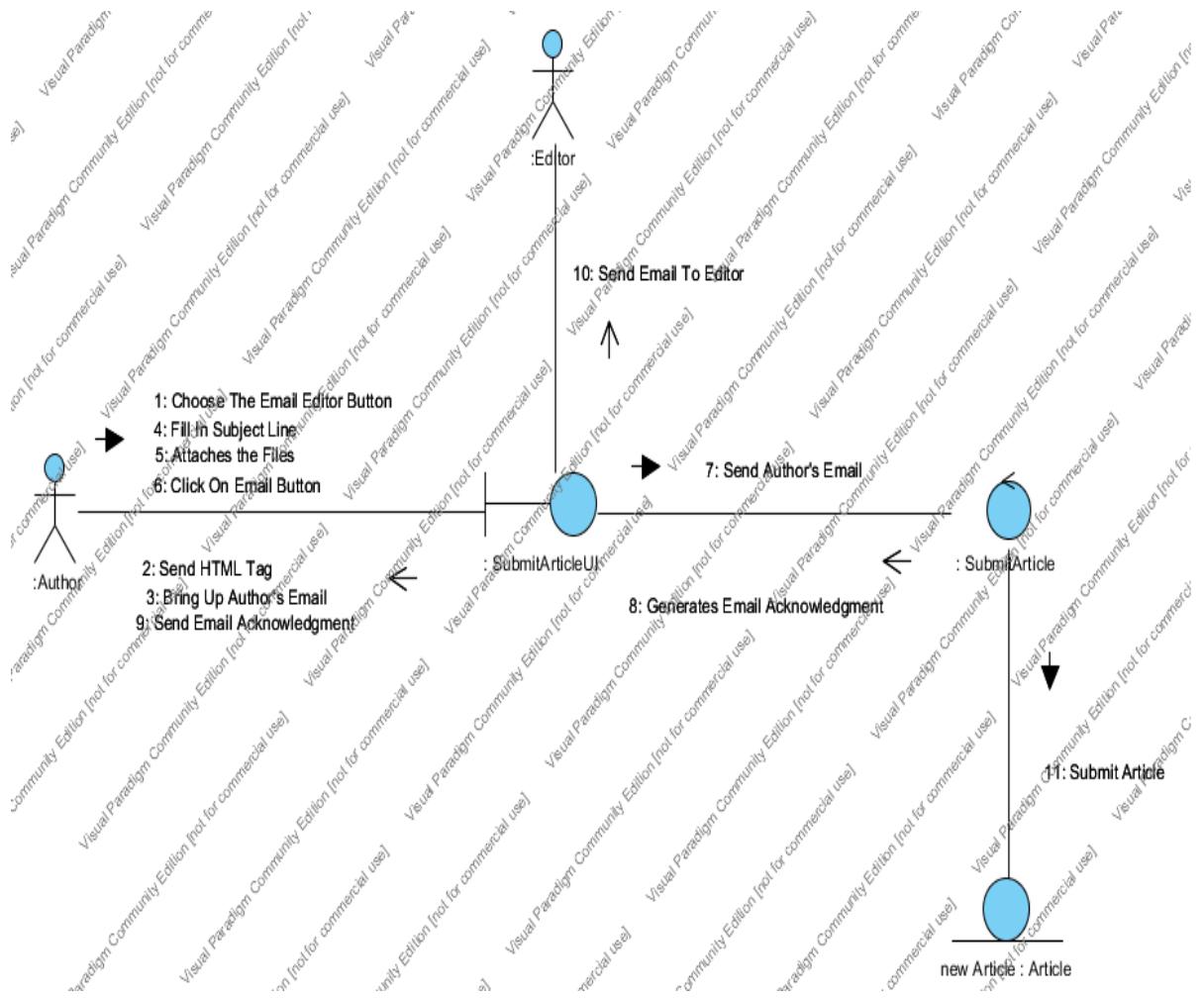


## Send Response

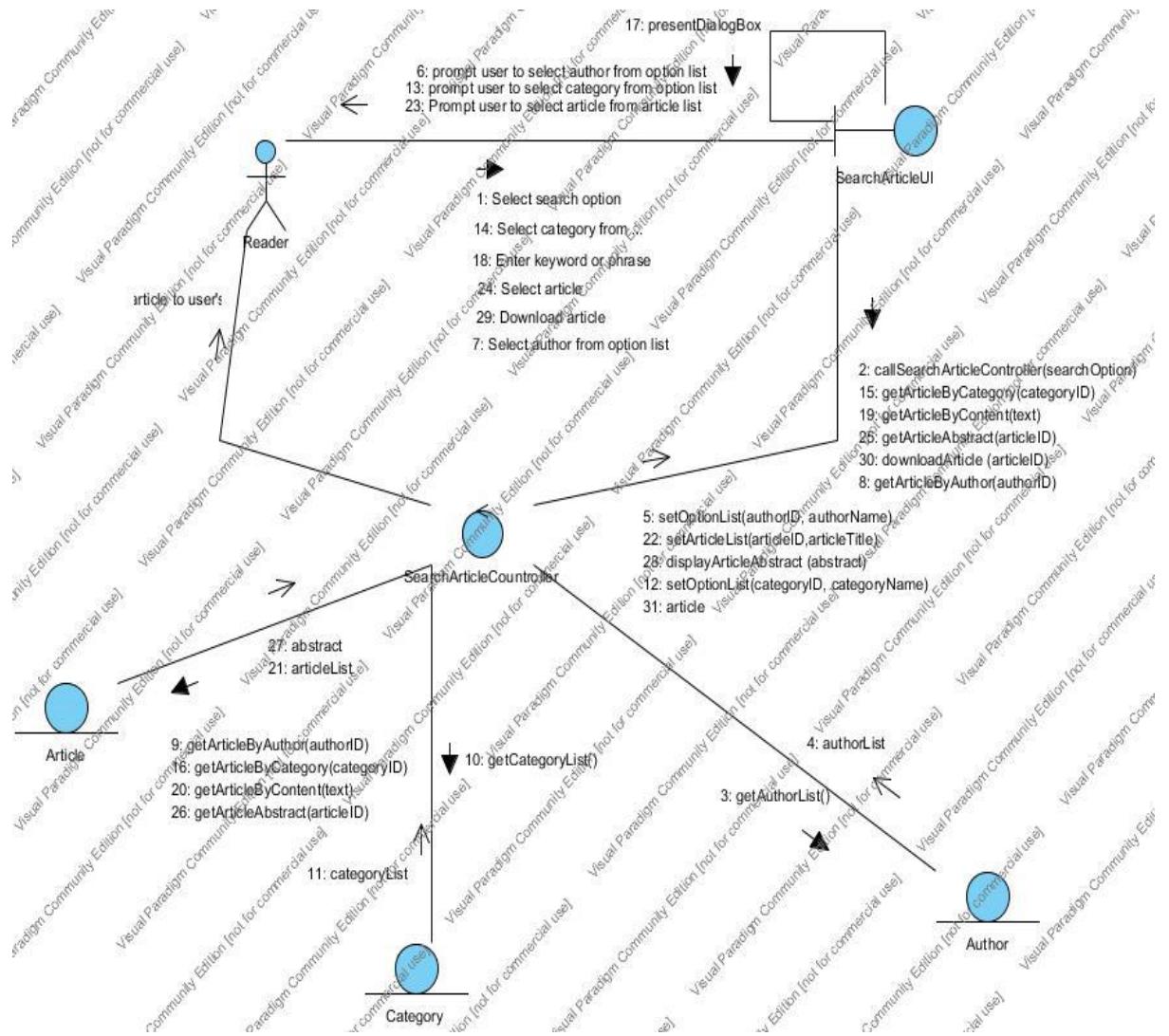


## Communication Diagrams

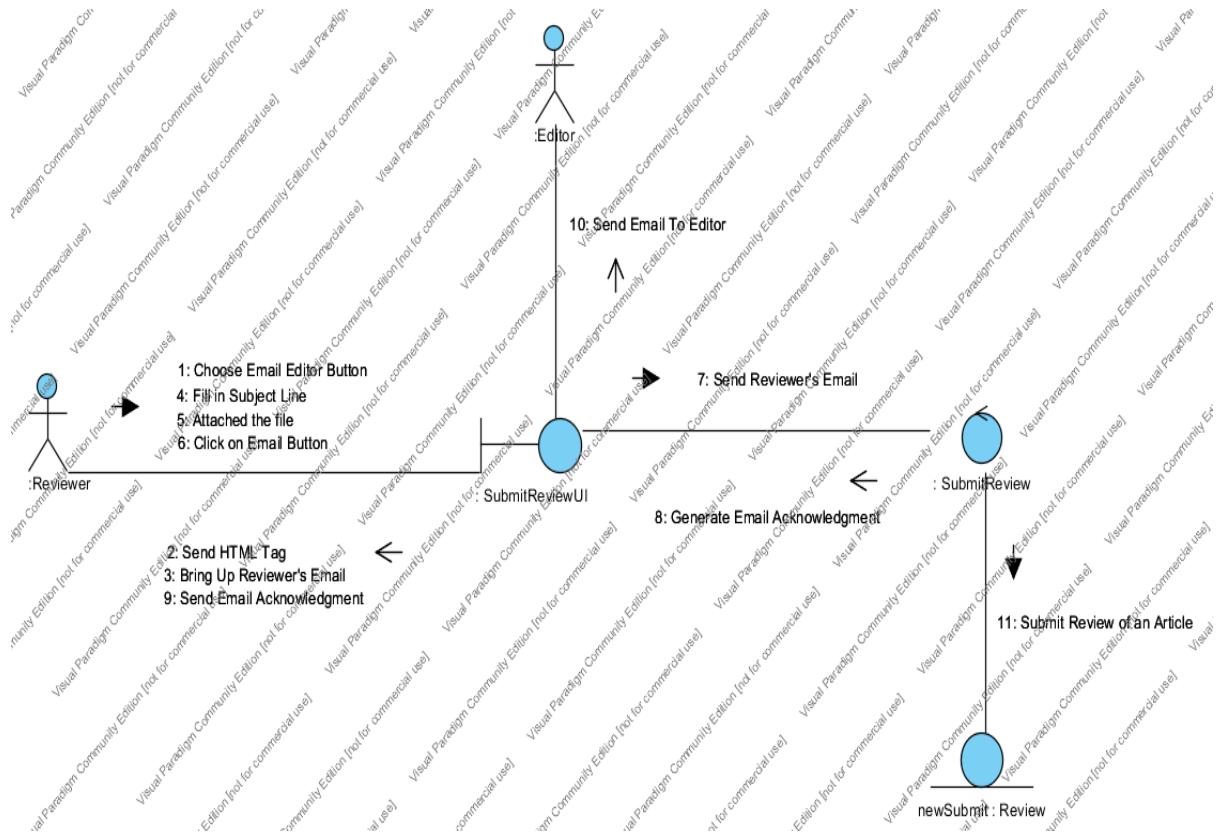
### Submit Article



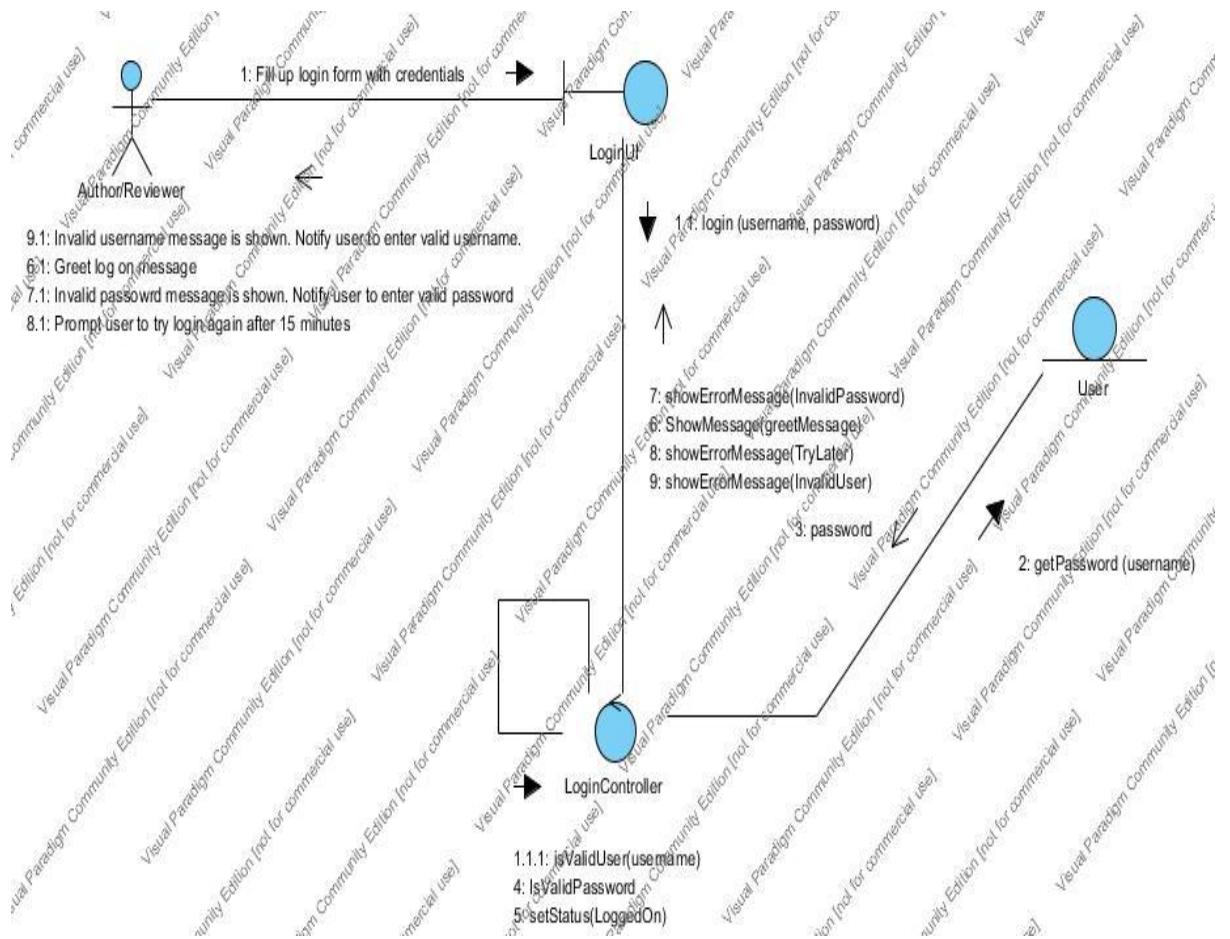
## Search Article

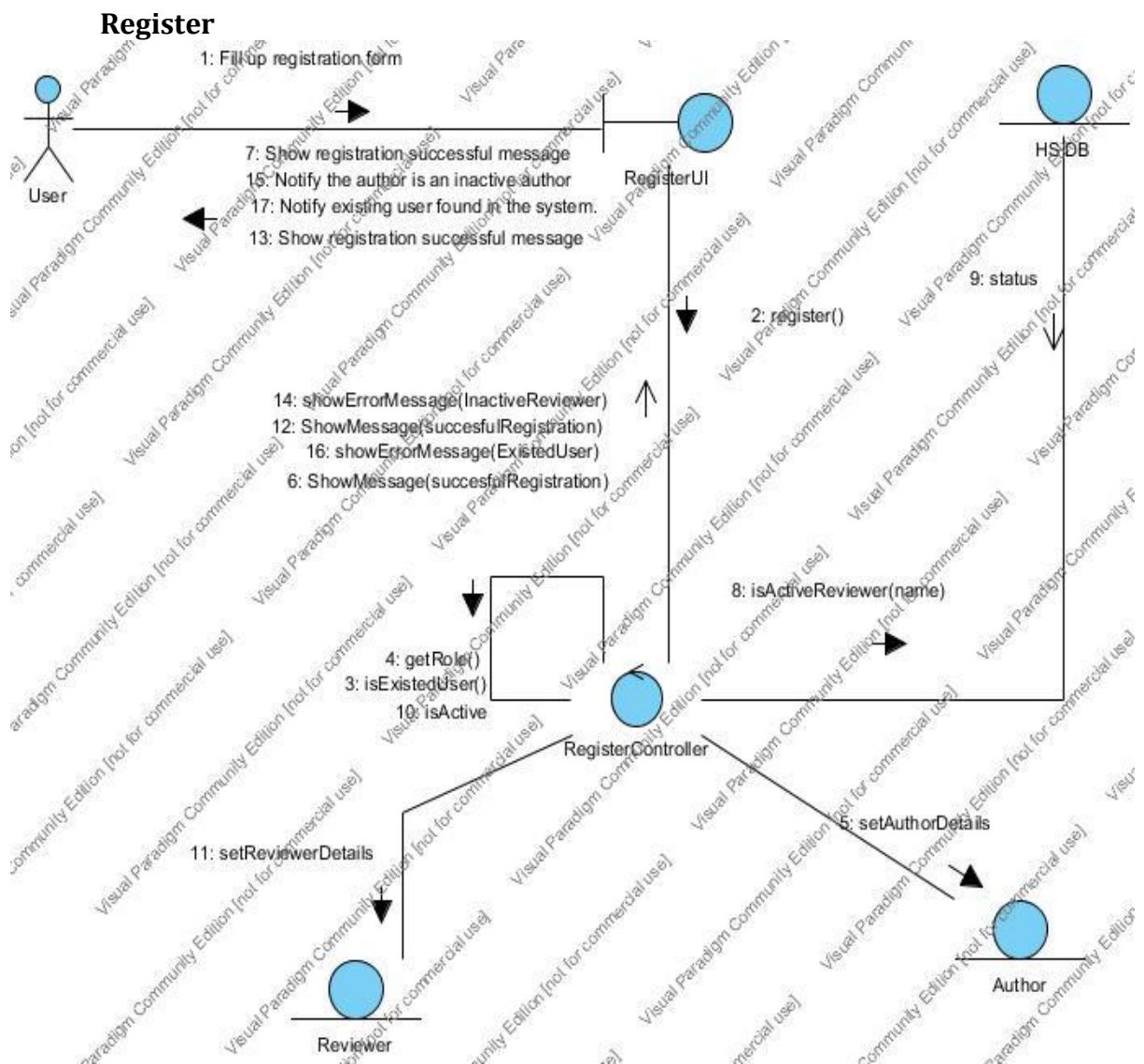


## Submit Review

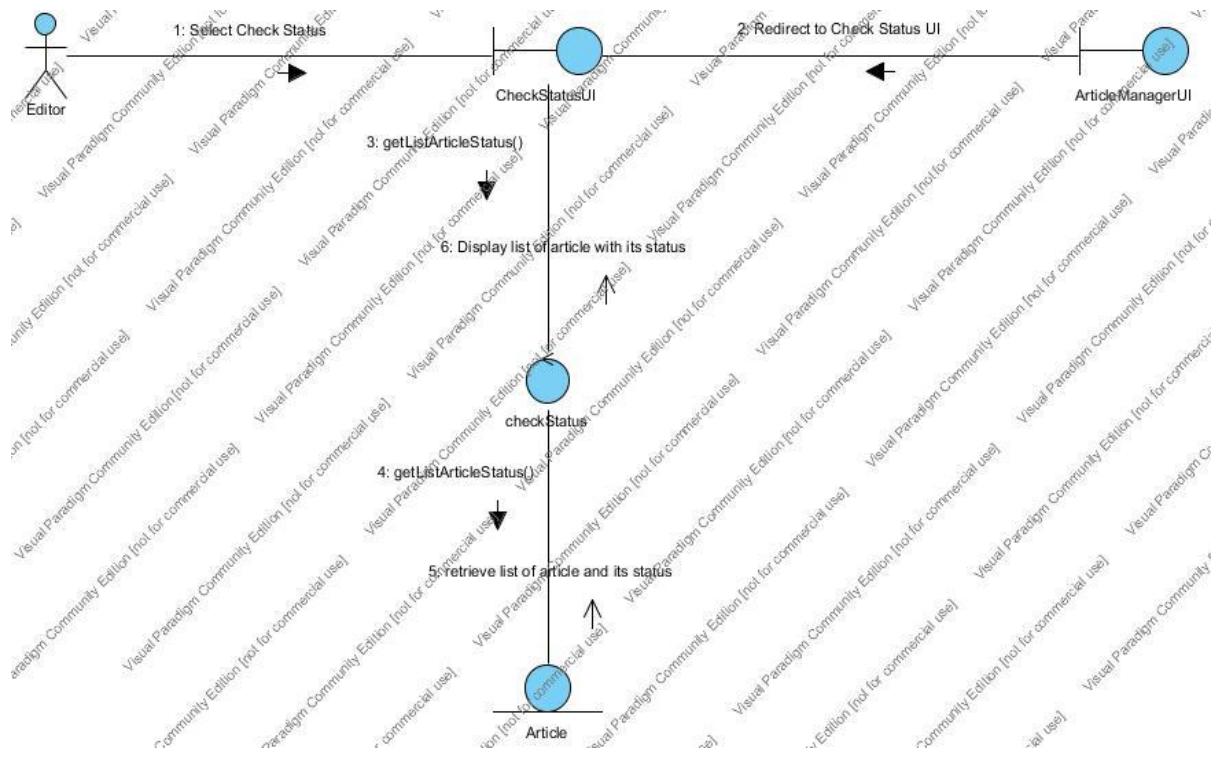


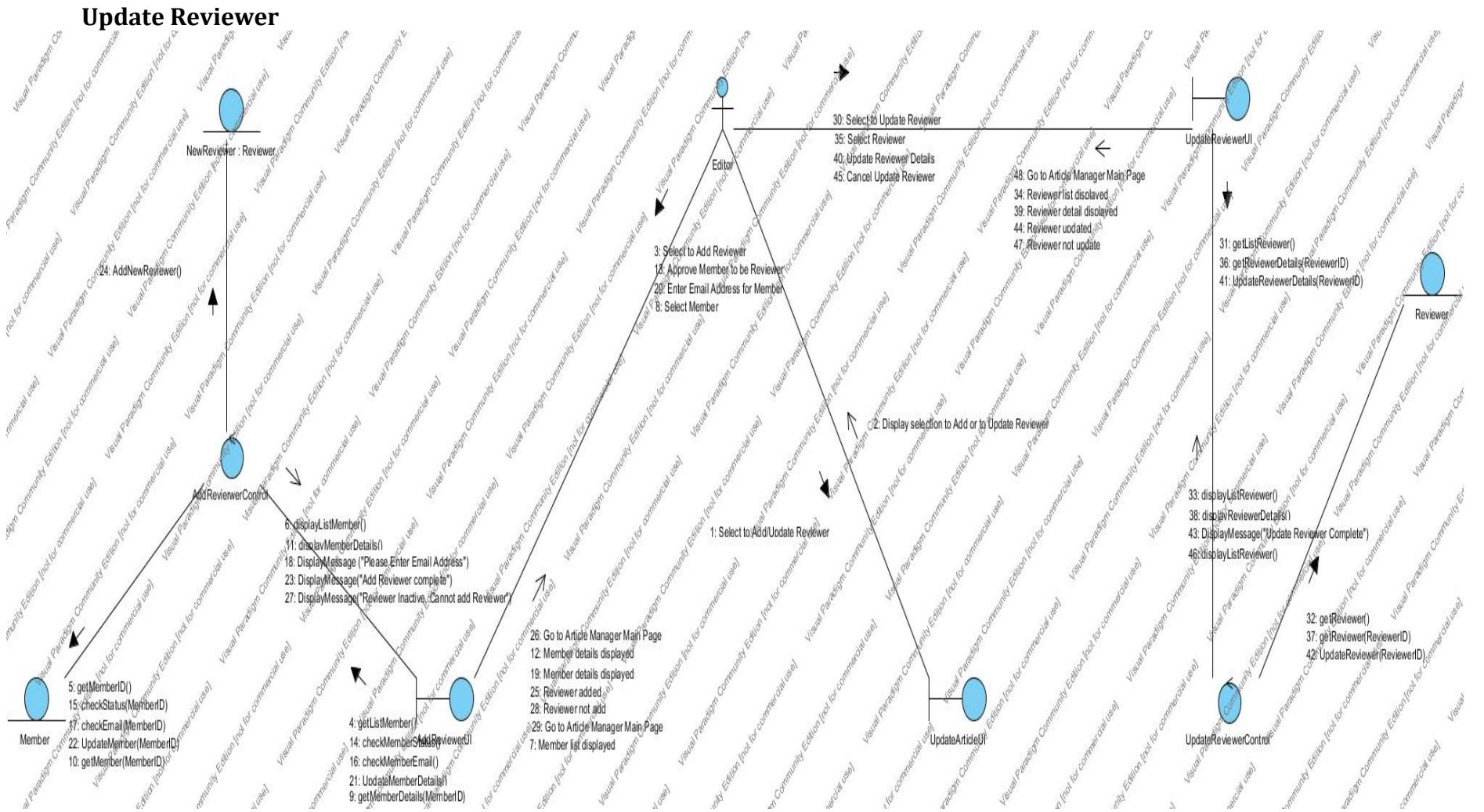
## Login

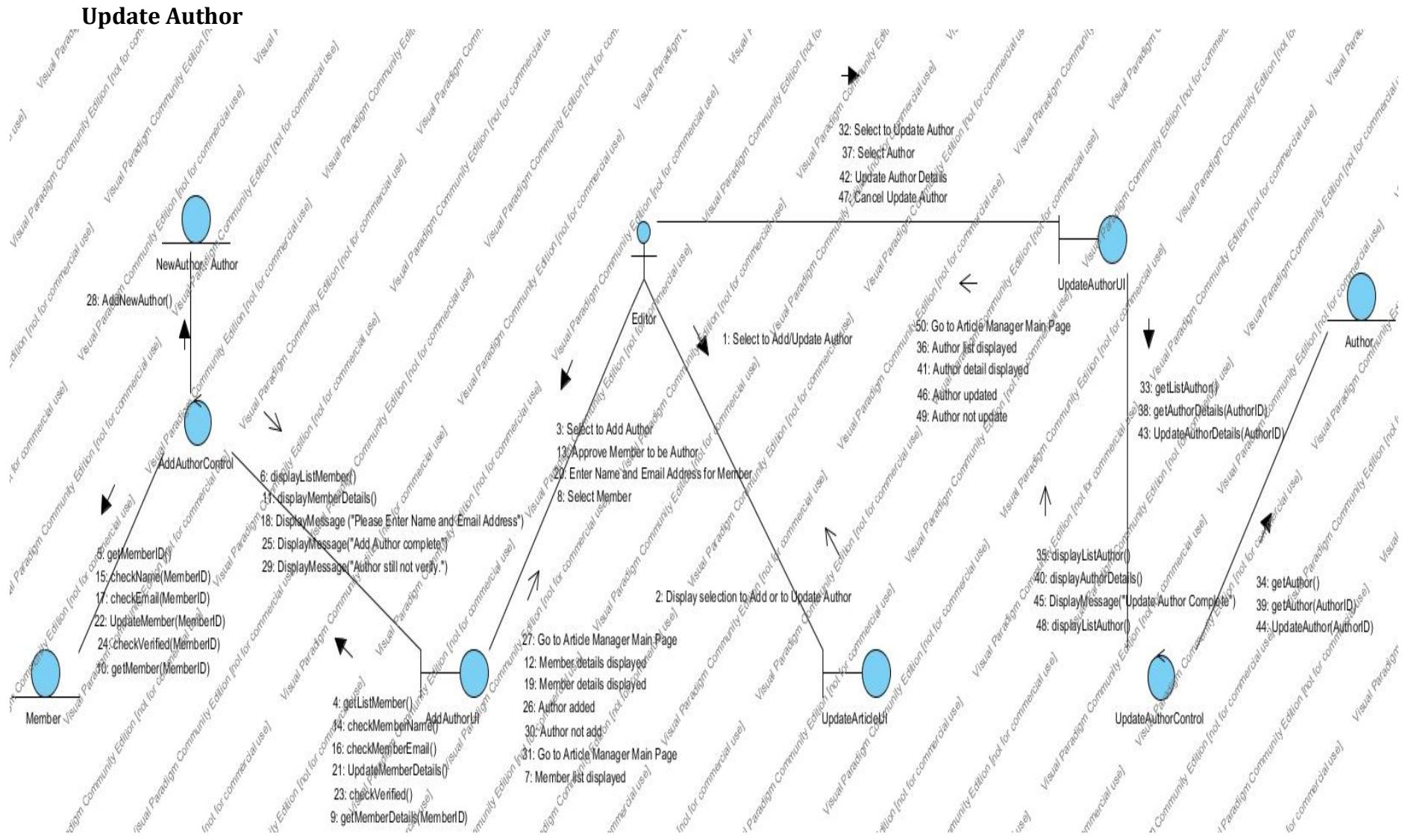


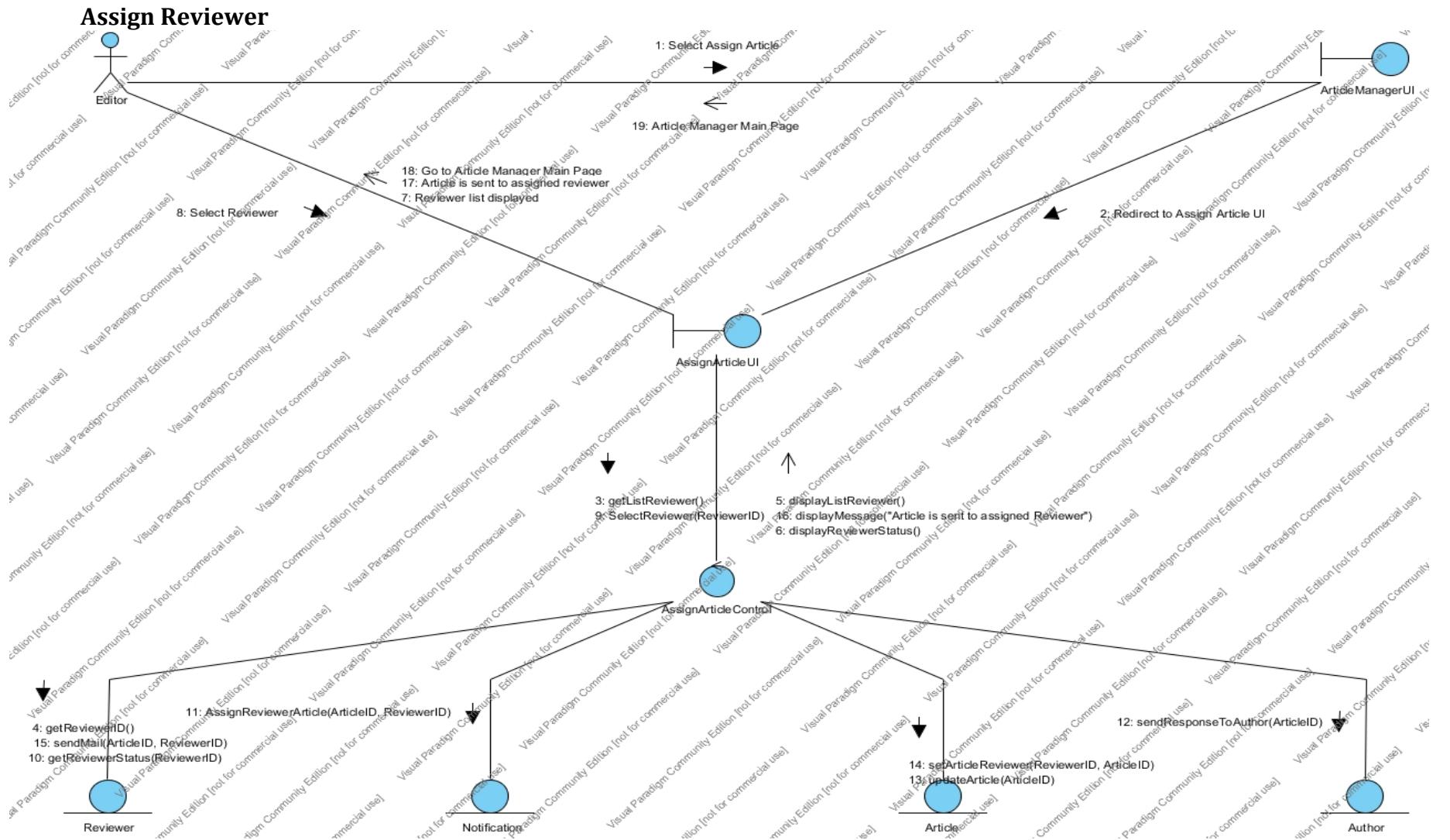


## Check Status

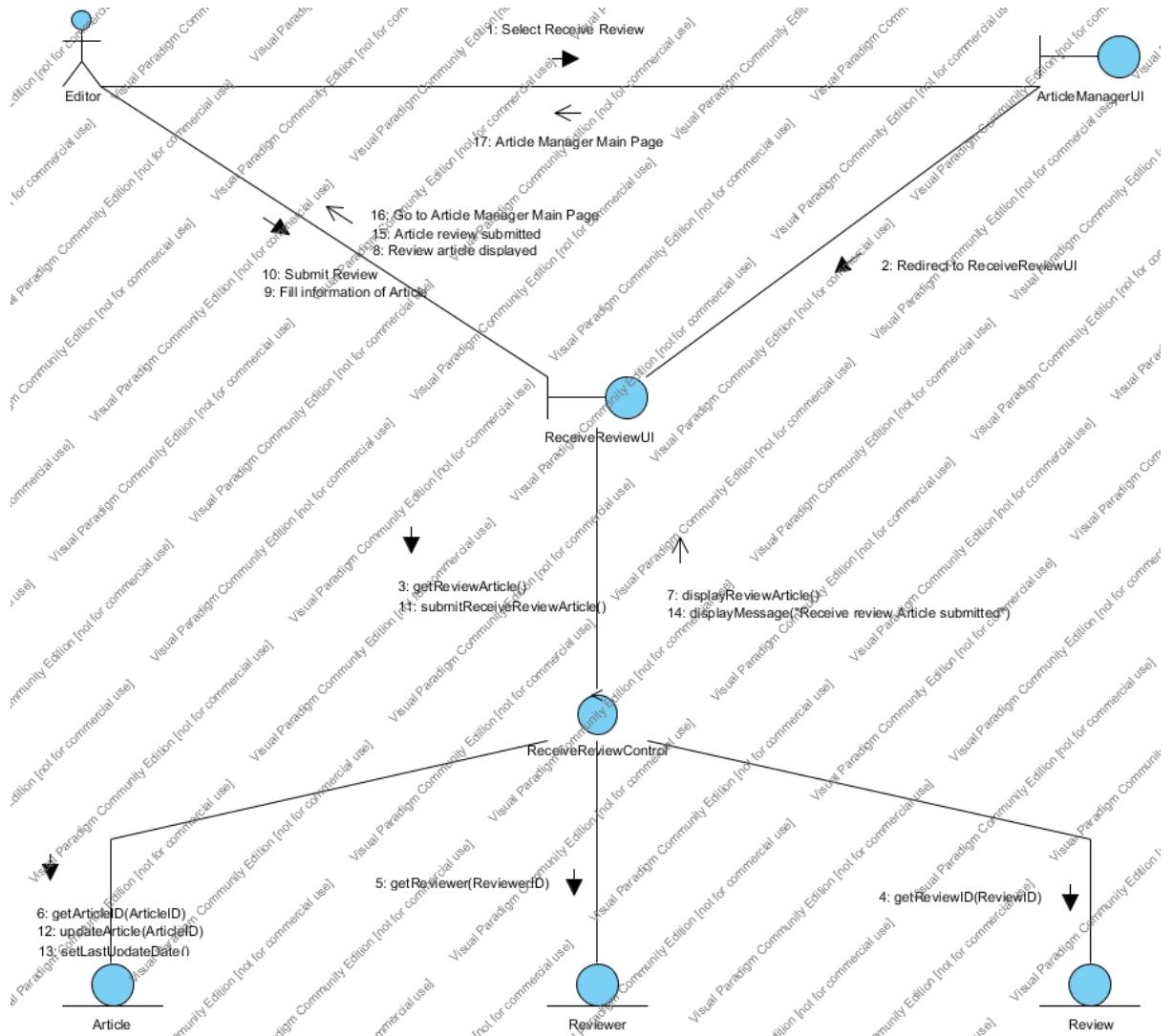




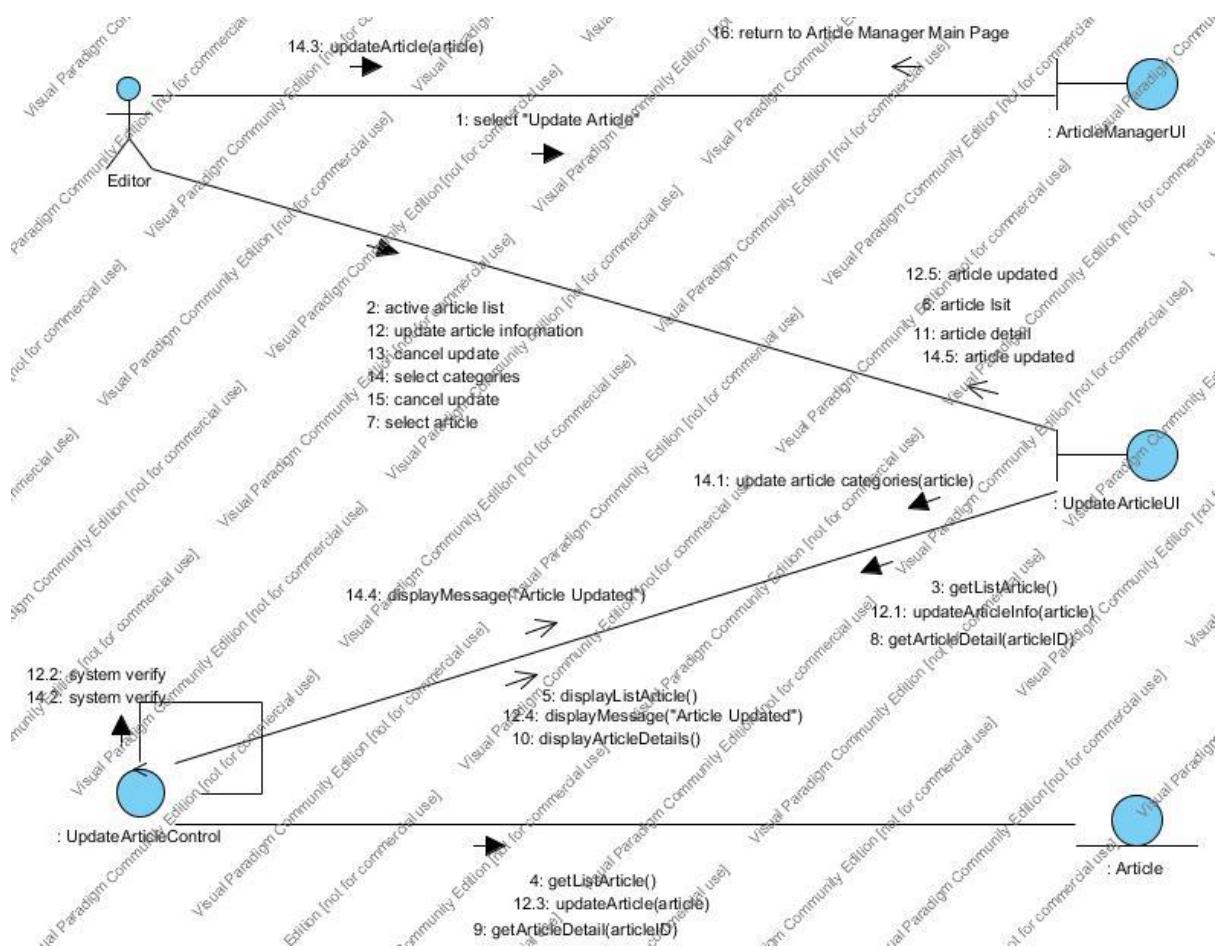




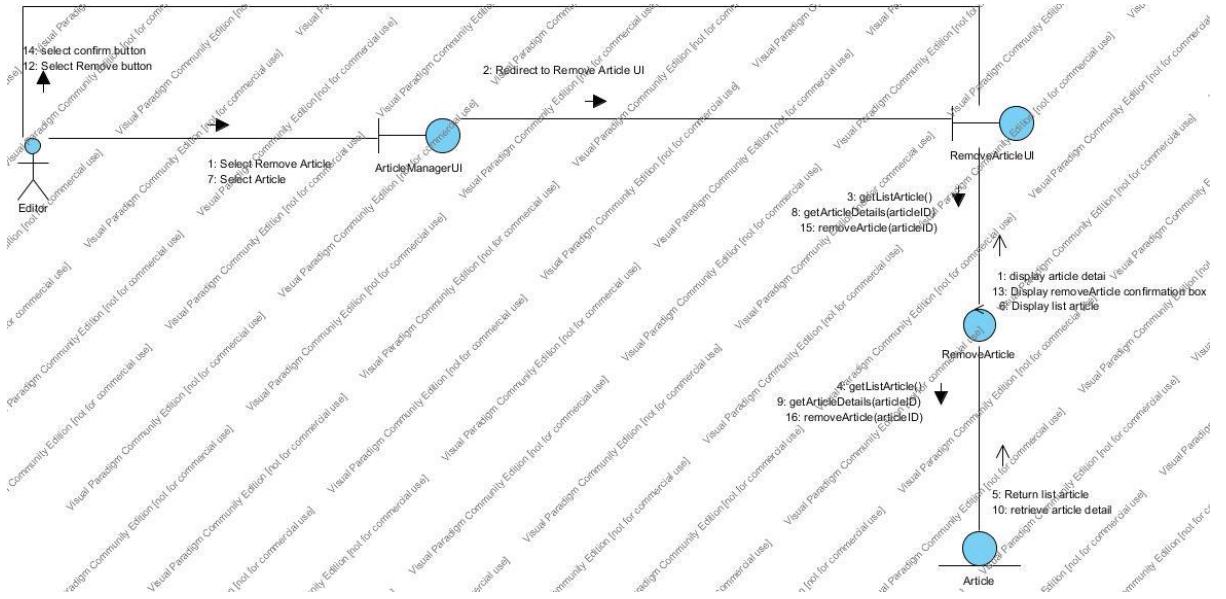
## Receive Review



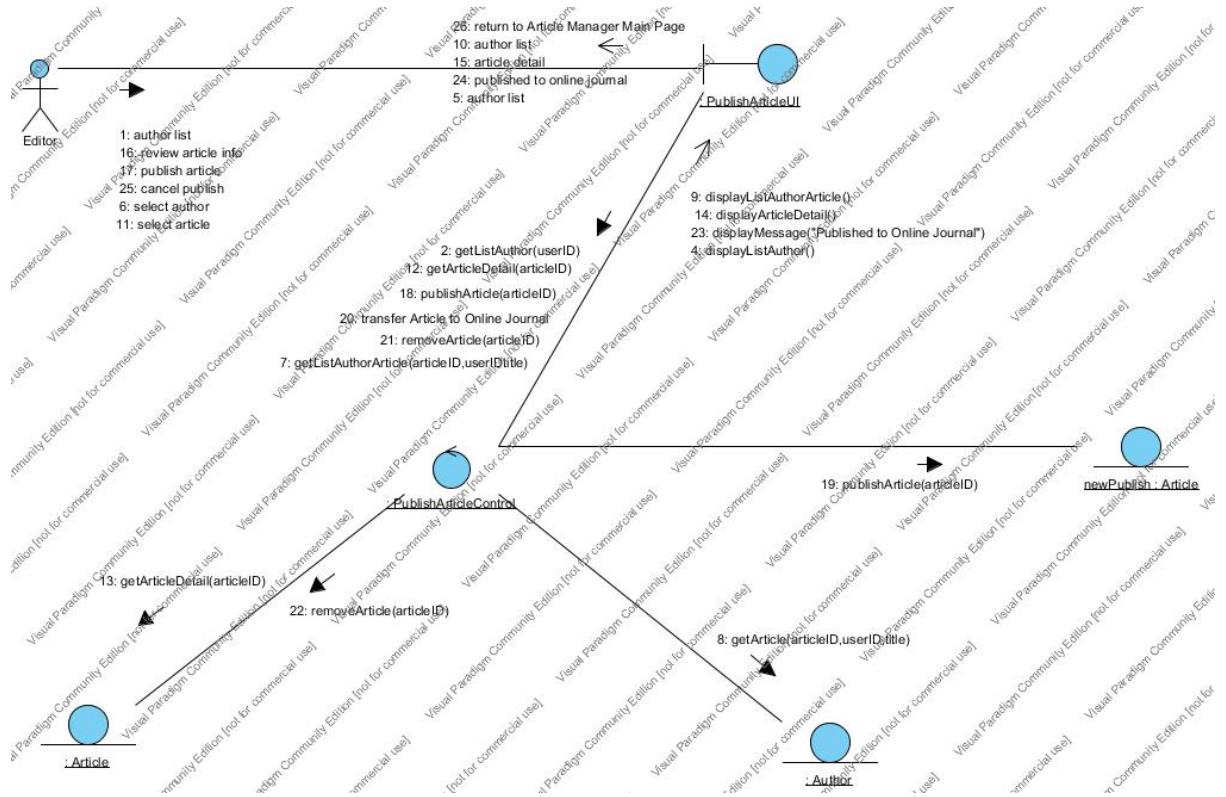
## Update Article



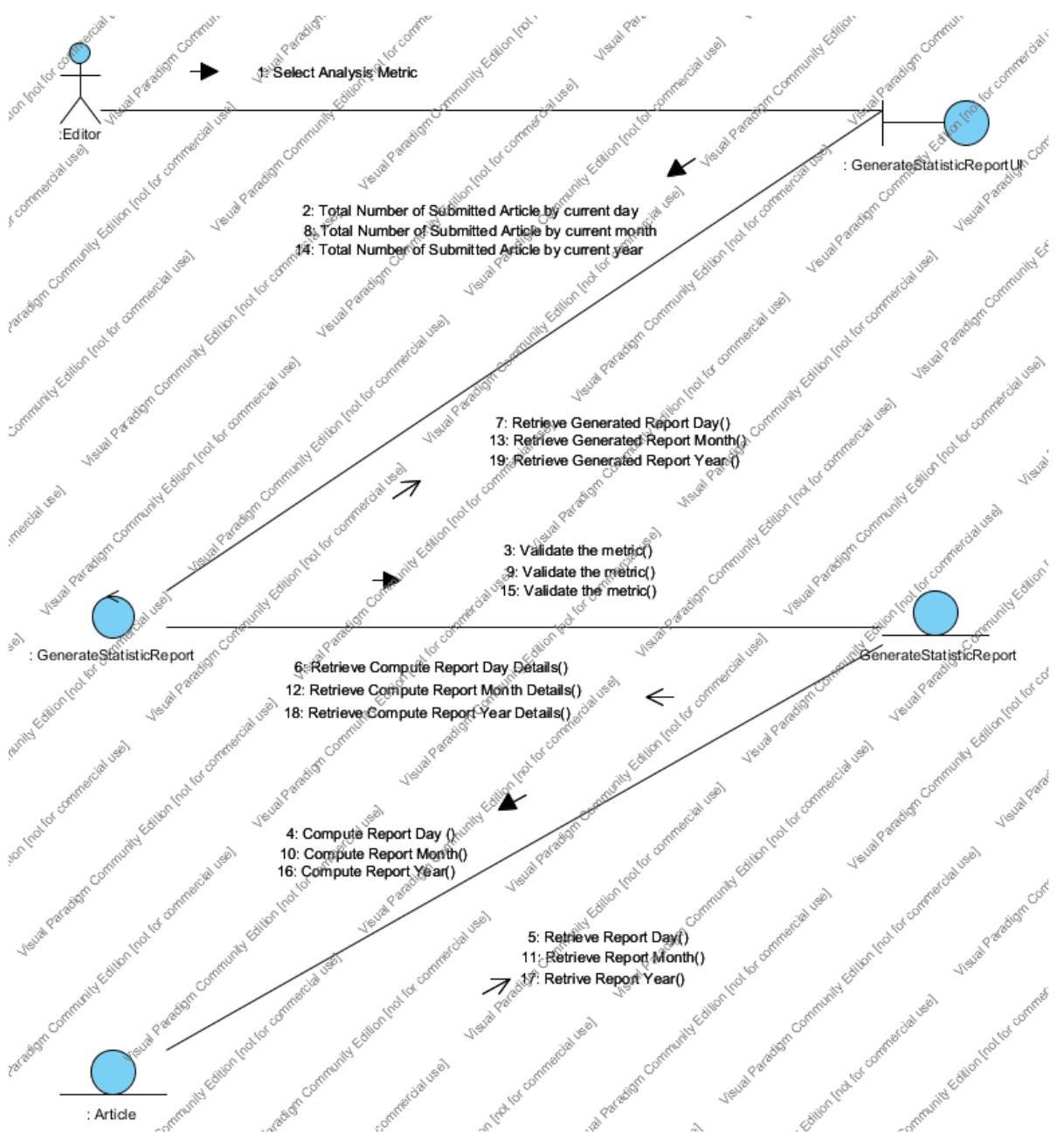
## Remove Article



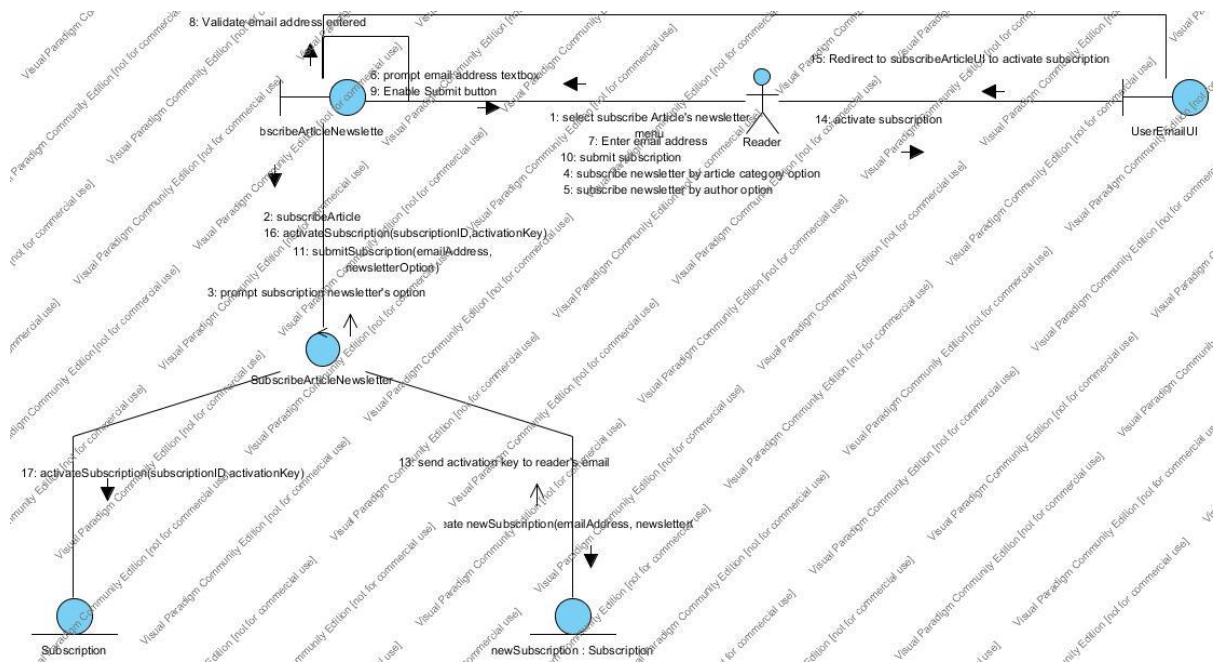
## Publish Article



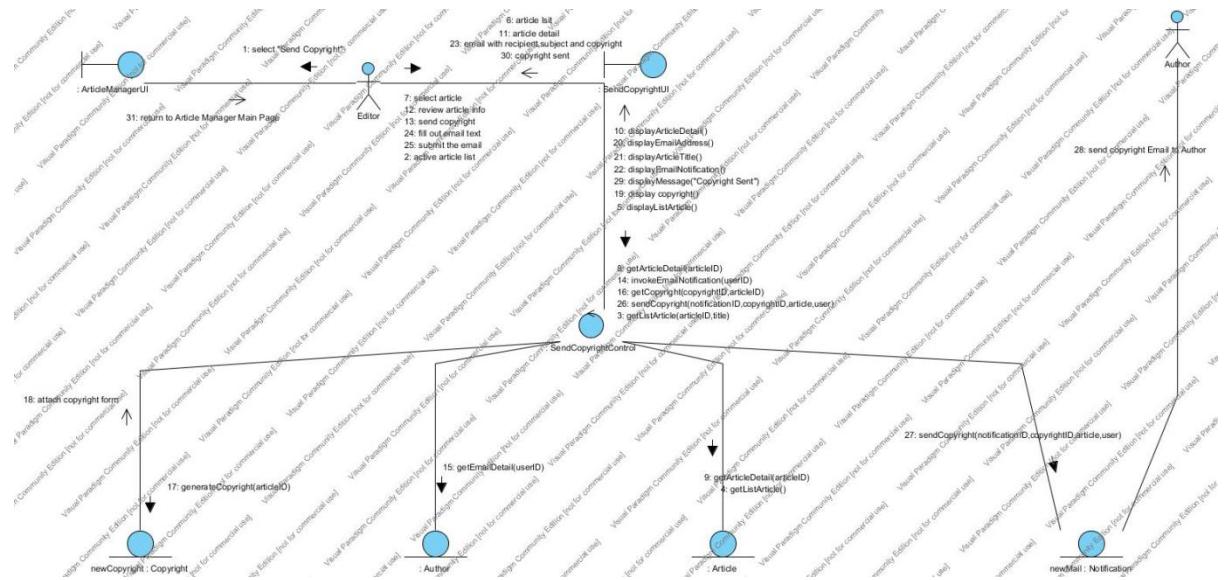
# Generate Statistic Report



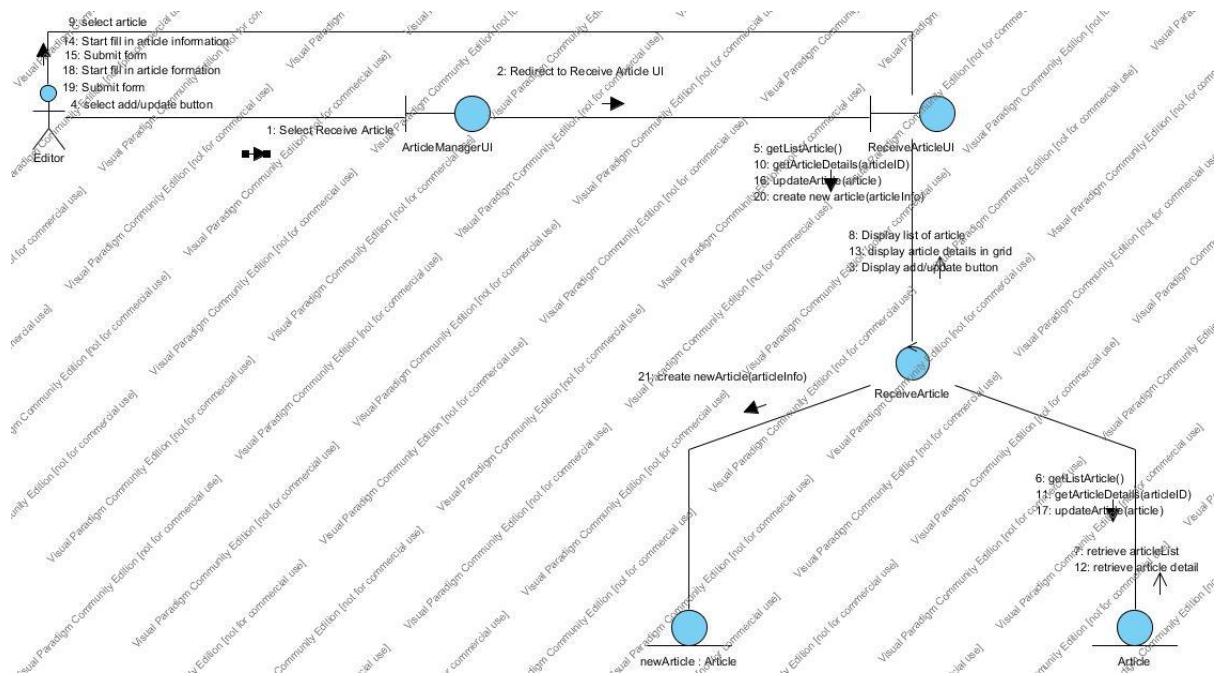
## Subscribe Article's Newsletter



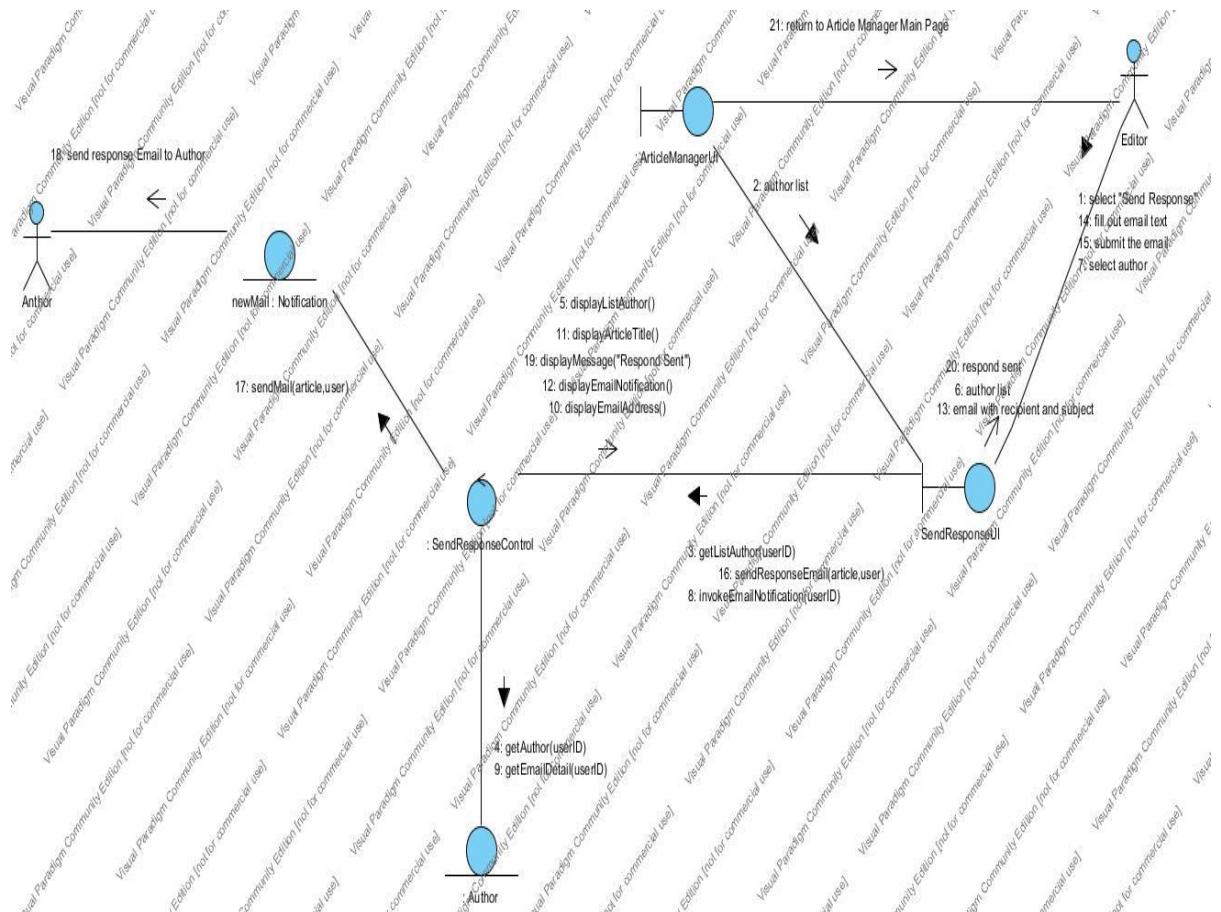
## Send Copyright



## Receive Article

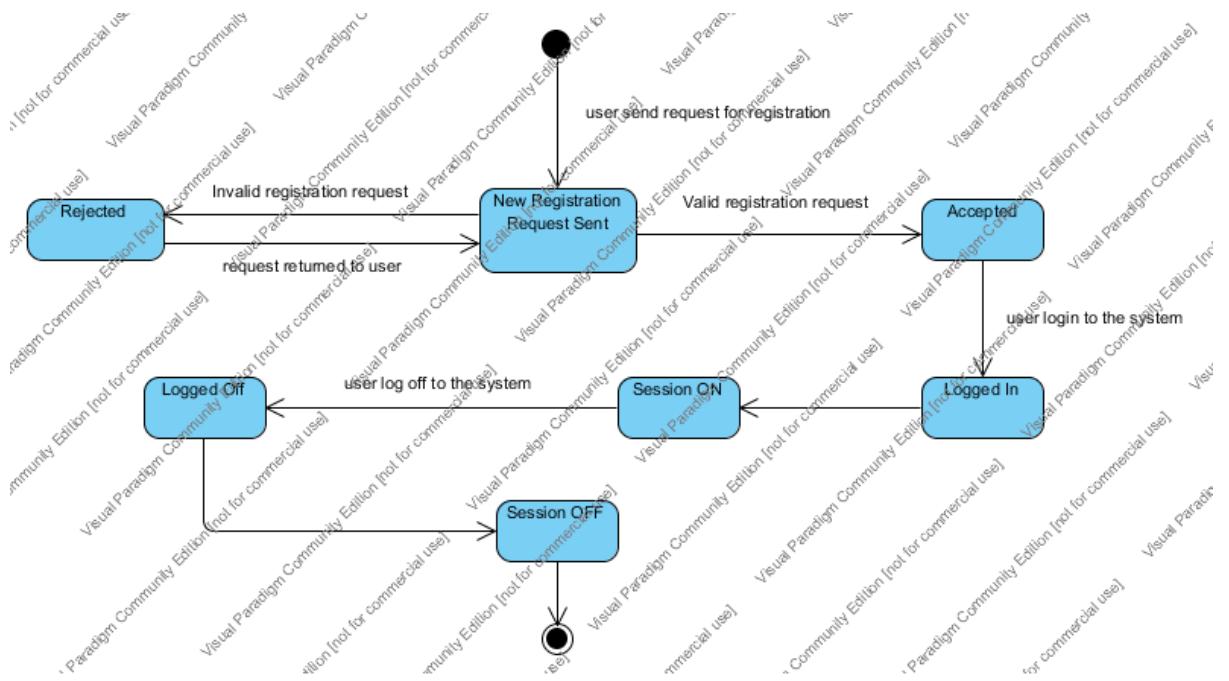


## Send Response



## State Machine Diagrams

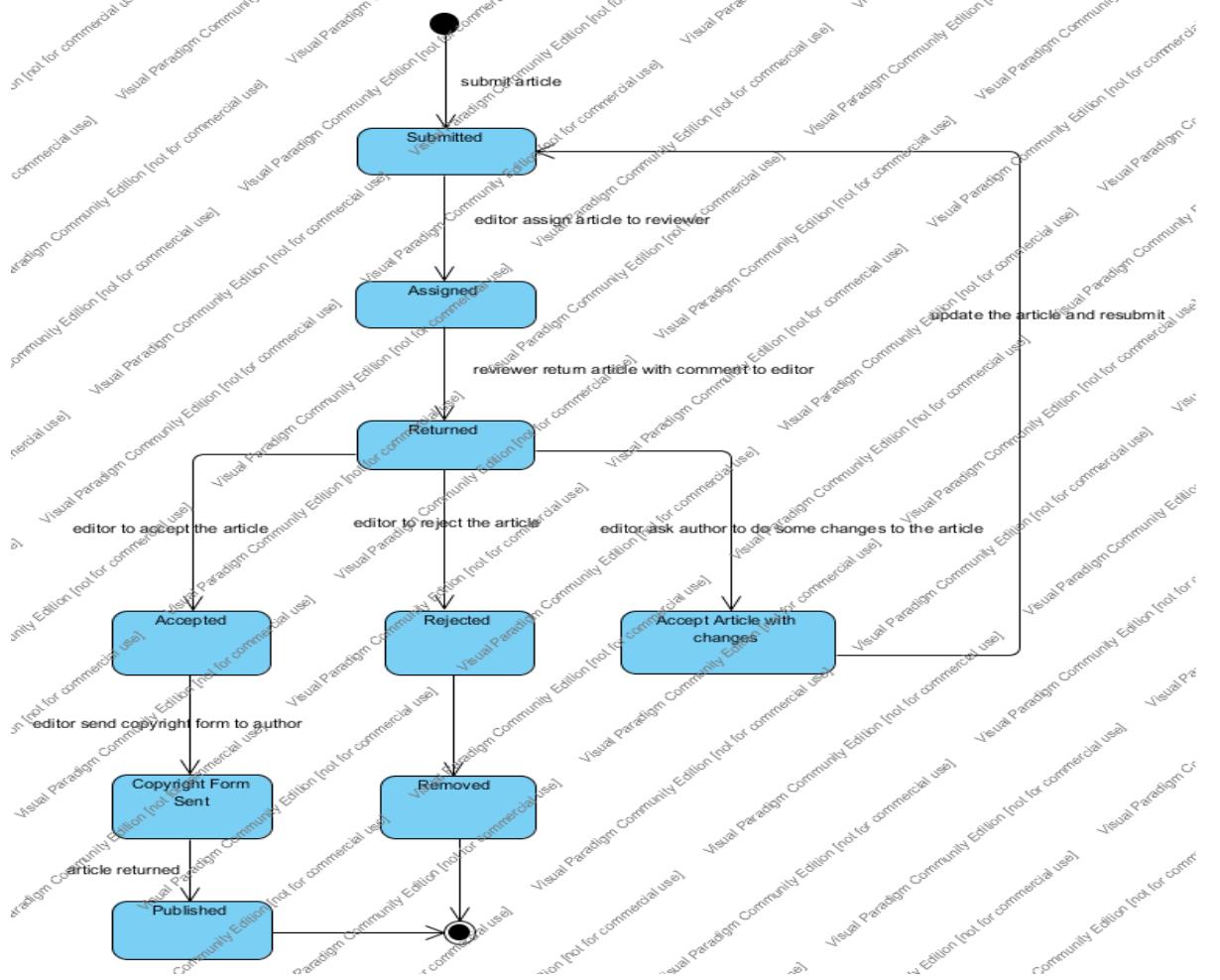
### User Lifecycle



### Description:

- The object is instantiated as a **User** in the *New Registration Request Sent* state. The registration process started when a user sent a new registration request.
- The registration will be accepted in the *Accepted* state if the new registration request is valid.
- If the new registration request contains invalid information, the request will be rejected in the *Rejected* state and request will then be returned to the user to make some necessary changes and sent the registration request back.
- After the **User object** has been successfully registered, the **User object** can now log in successfully in the *Logged In* state
- The **User object** will granted access to the system in the *Session On* state
- The **User object** will be logged out to the system in the *Logged Off* state
- The session will be terminated in the *Session Off* state.

## Article Lifecycle

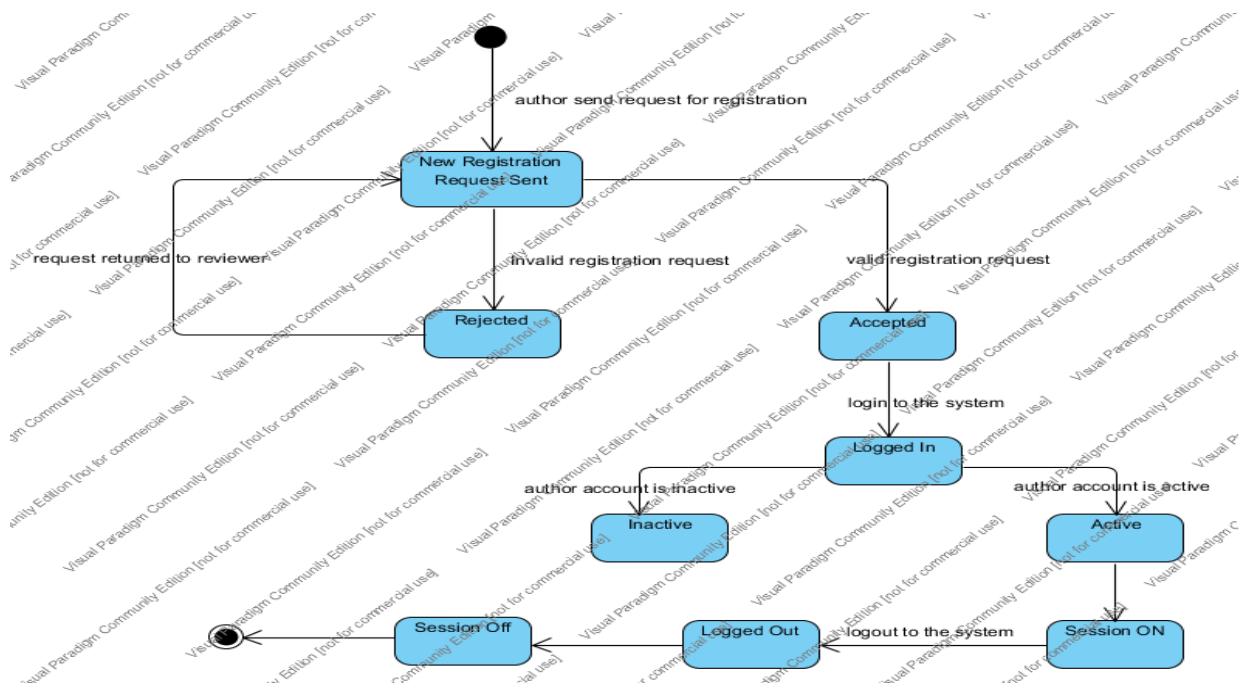


### Description:

- The object is instantiated as an **article** in the *Submitted* state.
- Then the **article object** has been assigned and sent to Reviewer in the *Assigned* state
- Reviewer will then return the article to the editor with its comment in the *Returned* state
- Then the **article object** could be fully accepted in the *Accepted* state
- The **article object** can also be declined in the *Rejected* state and article will be removed in the *Removed* state
- The **article object** can be returned back to the author to make some changes in the *Accept Article with changes* state and **article** will be submitted again and loop back to the *Submitted* state
- When the **article object** has been accepted in the *Accepted* state, the Editor will then send the copyright form to the author in the *Copyright form sent* state

- Once copyright form has been returned **article object** will be published in the *Published* state.

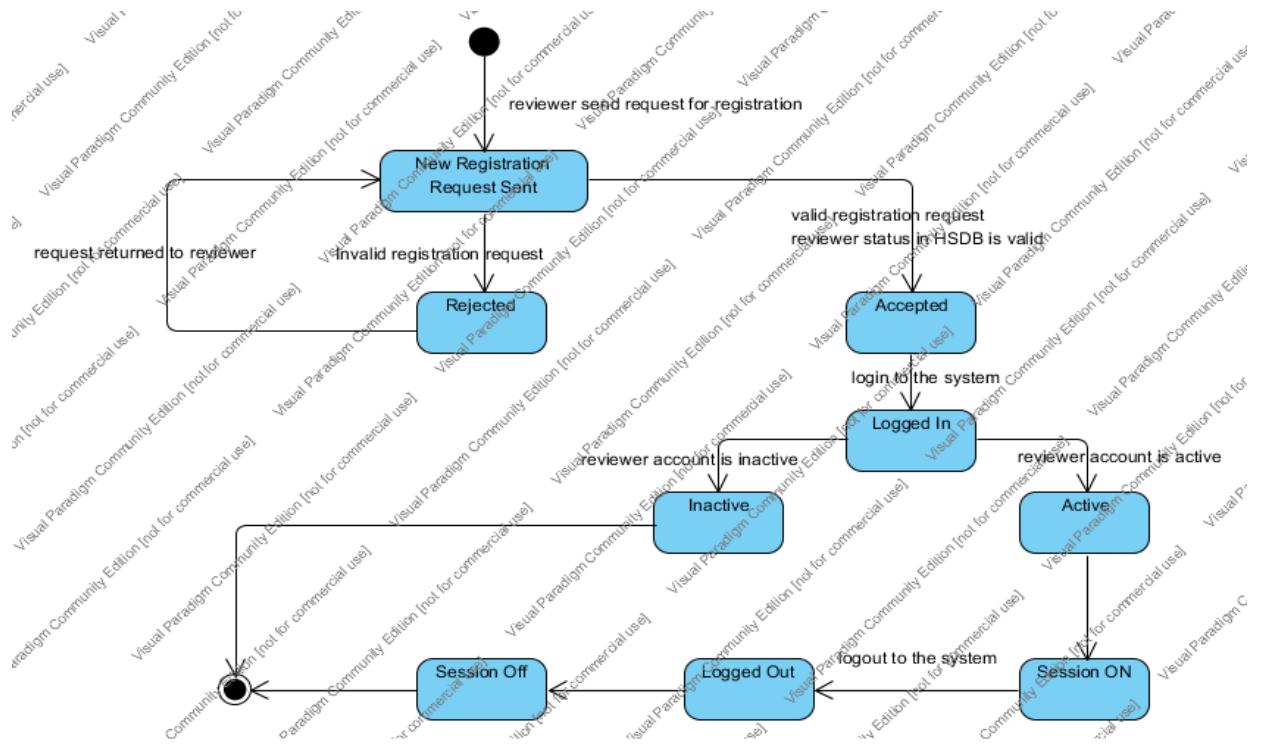
## Author Lifecycle



## Description:

- The object is instantiated as an **author** in the *New Registration Request Sent* state. The registration process started when author sent a new registration request.
- The registration will be accepted in the *Accepted* state if the new registration request is valid and author status in HSDB is valid
- If the new registration request contains invalid information, the request will be rejected in the *Rejected* state and request will then be returned to the author to make some necessary changes and sent the registration request back.
- After the **author object** has been successfully registered, the **author object** can now log in successfully in the system in the *Logged In* state
- System will check If the Author account is active in the *Active* state
- If the author account is inactive system should not grant access to the Author and session will be terminated in the *Inactive* state
- The **author object** will granted access to the system in the *Session On* state
- The **author object** will be logged out to the system in the *Logged Off* state
- Session will be terminated in the *Session Off* state

## Reviewer Lifecycle



## Description:

- The object is instantiated as a **reviewer** in the *New Registration Request Sent* state. The registration process started when Reviewer sent a new registration request.
- The registration will be accepted in the *Accepted* state if the new registration request is valid and reviewer status in HSDB is valid
- If the new registration request contains invalid information, the request will be rejected in the *Rejected* state and request will then be returned to the reviewer to make some necessary changes and sent the registration request back.
- After the **reviewer object** has been successfully registered, the **reviewer object** can now log in successfully in the system in the *Logged In* state
- System will check If the reviewer account is active in the *Active* state
- If the reviewer account is inactive system should not grant access to the reviewer and session will be terminated in the *Inactive* state
- The **reviewer object** will granted access to the system in the *Session On* state
- The **reviewer object** will be logged out to the system in the *Logged Off* state
- The session will be terminated in the *Session Off* state.

## **Tools Used**

Visual Paradigm  
Enterprise Architecture  
Microsoft Visio

## **Reference**

Object Oriented Systems Analysis and Design Using UML, (4th Edition), McGraw Hill, 2010.