

Term Project - Deliverable 2

Use Cases

- Upload an article
- Navigate through different web pages
- Filter/sort through articles
- Share to social media
- Track website analytics
- View webpage on mobile
- Alter website aesthetics by interacting with it

Textual descriptions of key use cases

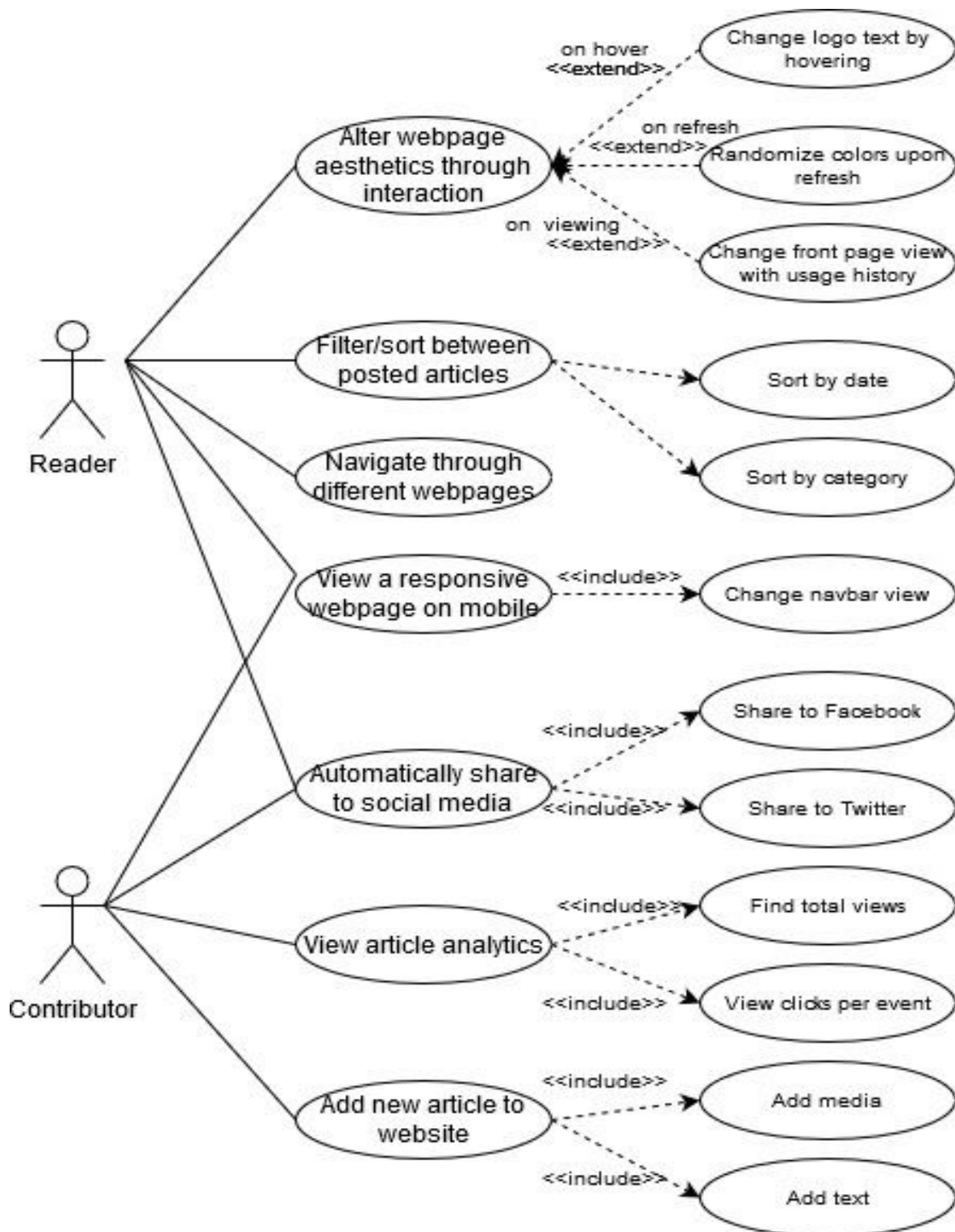
CASE NAME:	Upload an article		USE CASE TYPE
PRIORITY:	High		Functional requirement
PRIMARY BUSINESS ACTOR:	Publication contributing writer		
OTHER PARTICIPATING ACTORS:	Publication team, admin, system		
OTHER INTERESTED STAKEHOLDERS:	N/A		
SHORT DESCRIPTION:	A contributor should have an easy interface to be able to upload their articles to the website with little to no coding/technical knowledge		
PRE-CONDITION:	Submitted articles have already been edited and finalized by the team		
TRIGGER:	Writer submits their article text		
TYPICAL COURSE	Writer Action	System Response (website)	
OF EVENTS:	Step 1: Writer splits their article into title, byline, author, date, and text		

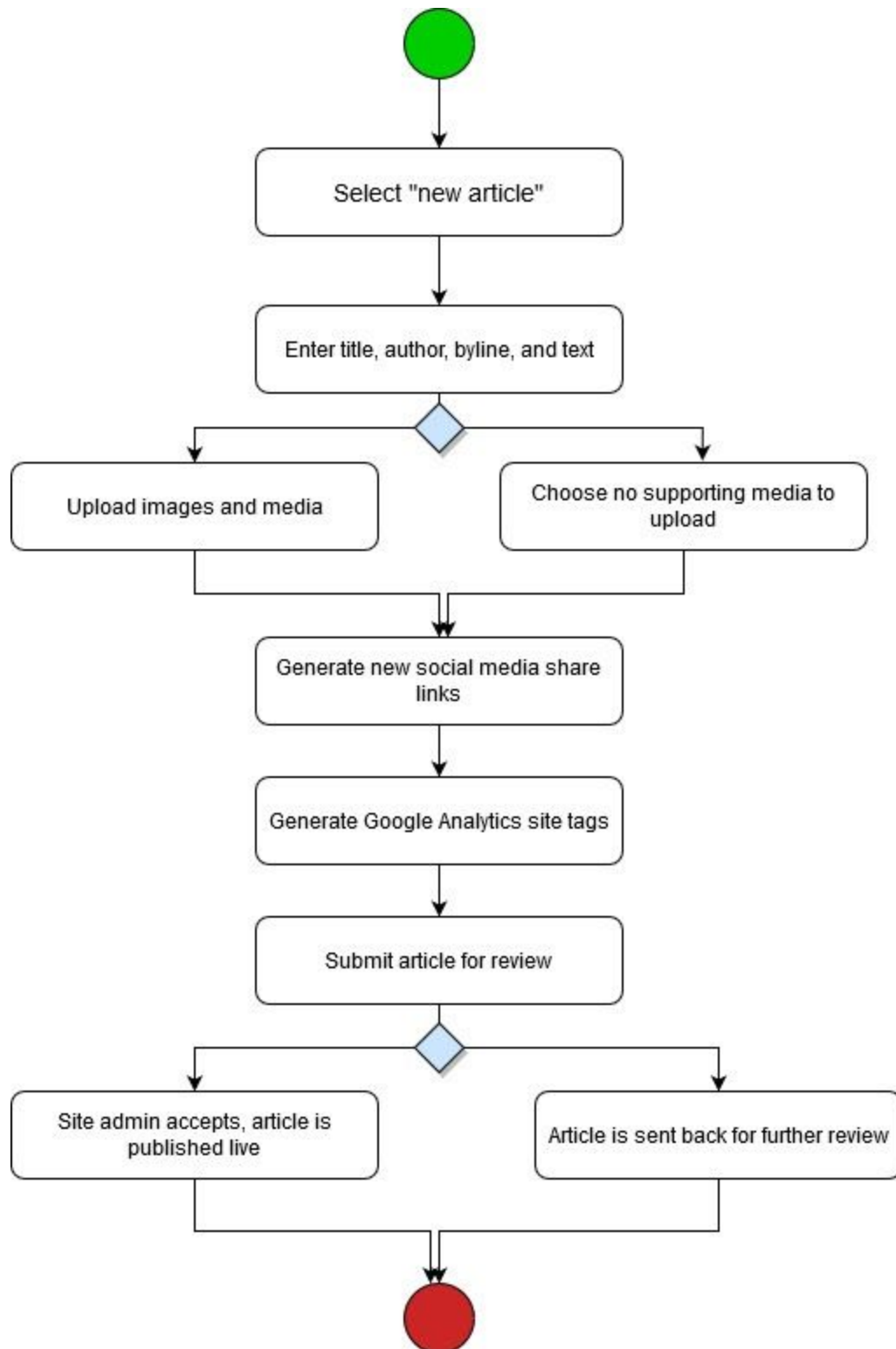
	2. Writer enters each relevant info into the specific field	3. Site parses the relevant text and processes it using JavaScript into an HTML template
		4: Site generates social media share and google analytics tags for the article page
		5. Site pushes the updated site to master, and the new article page becomes live.
ALTERNATE COURSES:	Alt step 3: If the text is somehow invalid, site throws an error and does not create a new article page.	
CONCLUSION:	Article posted to the website using a simple template that is easy for contributing writers to work with.	
POST-CONDITION:	Site admin must confirm that the article has been outputted correctly	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	Due to time constraints, we may not be able to develop a GUI for writers to submit their articles, but they can submit it through a text form that they have to fill out and that is parsed into a JSON that is read by the website.	
ASSUMPTIONS:	Site admin are able to continuously facilitate this process and ensure it is operating smoothly	

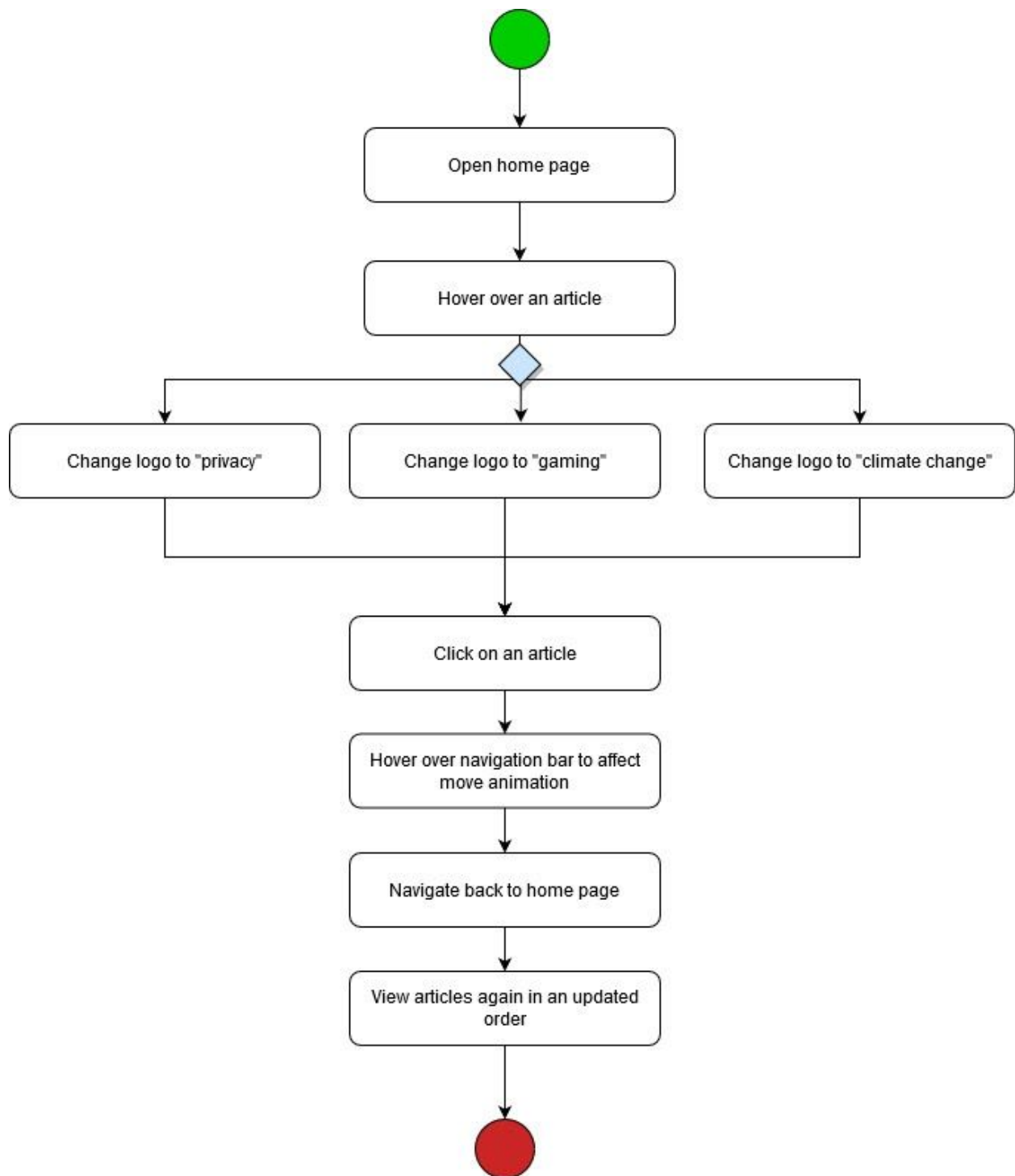
CASE NAME:	Alter website aesthetics by interacting with it	USE CASE TYPE
PRIORITY:	Medium	Functional requirement
PRIMARY BUSINESS ACTOR:	Website reader	
OTHER PARTICIPATING ACTORS:	Website team, writers	
OTHER INTERESTED STAKEHOLDERS:	Website managers	

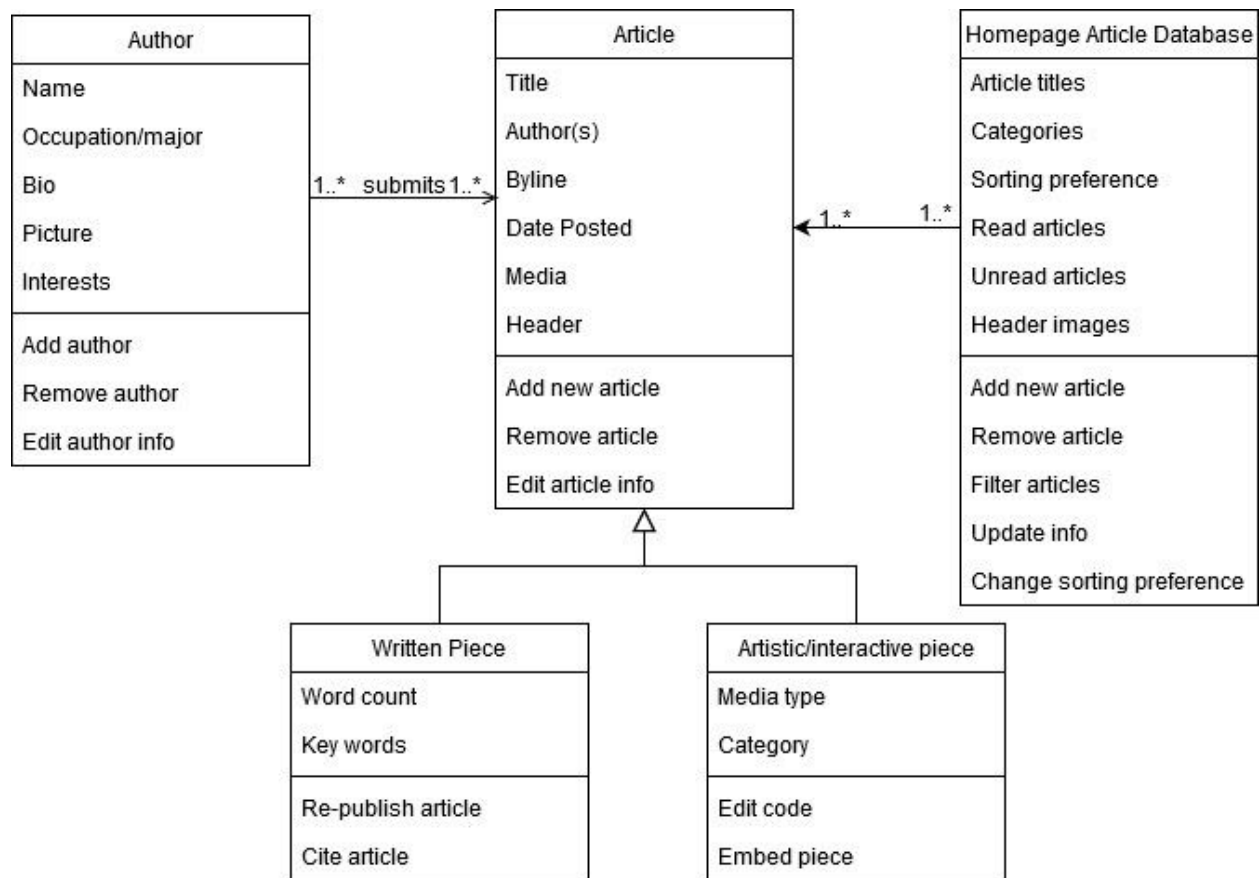
SHORT DESCRIPTION:	Since the team is working on a website called “Multiplicity” representing different perspectives (like the UML method) they want the aesthetic of the website to represent change and customization.	
PRE-CONDITION:	Website static CSS and design must be first complete.	
TRIGGER:	User starts moving mouse	
TYPICAL COURSE	User Action	System Response (website)
OF EVENTS:	Step 1: After site loads, user hovers over an article card on the front page	2. Website title “multiplicity. <i>tech</i> ” changes its second extension to represent the category of the article hovered over (e.g. “multiplicity. <i>privacy</i> ”)
	2. User clicks on article	3. Article web page opens with a randomized color scheme
	4. User hovers over navigation bar	5. Simple 2D animation shows behind the navigation bar that moves around the user’s mouse
	6. User navigates back to home page by clicking on logo	5. Home page displays a different ordering of the articles
ALTERNATE COURSES:	N/A - but sequence of events could be different	
CONCLUSION:	User is able to see various parts of the website’s design change depending on their interaction and usage of the website	
POST-CONDITION:	NA	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	The amount of customization we offer the website will depend on our own time and technical constraints, so we will first focus on altering just the logos and navbar, then proceed with the remaining events.	
ASSUMPTIONS:	Designer has provided preferable matching color schemes we can switch between so the randomization does not look too jarring.	

Use Case Diagram



Activity Diagrams**A. Post a new article**

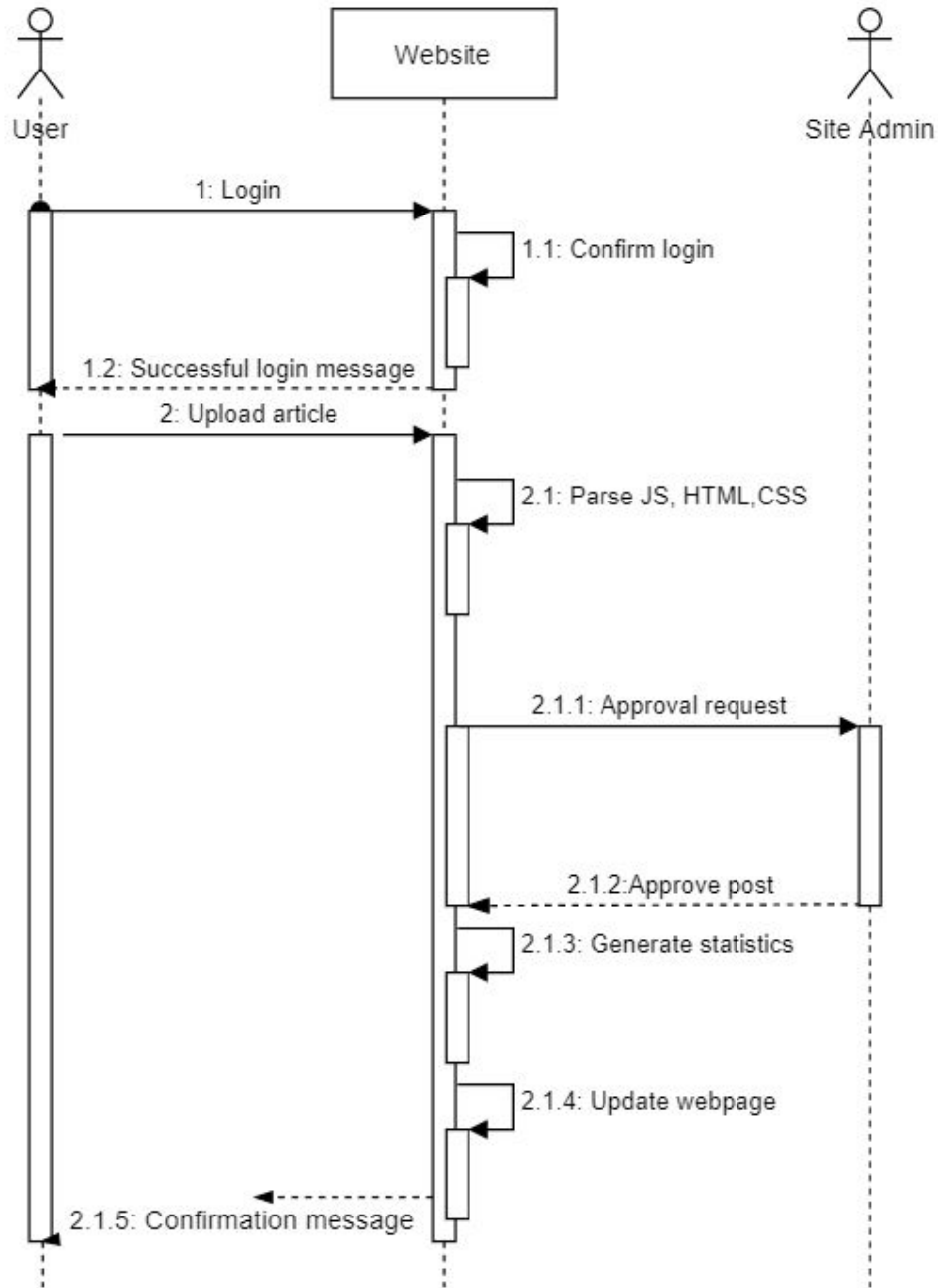
B. Interact with website design

Class diagram**Design patterns used:**

Polymorphism: We used polymorphism to break down the article class into written and interactive pieces, since there will be both forms of content, and it would be more organized and easier to work with if we identify and create a distinction between the two and the different ways they operate.

Creator: The article database object creates the articles, which gives it the responsibilities necessary and allows us to keep track of the addition of new articles and be able to collaborate with future classes.

High coupling: There is high coupling (i.e. a direct link) between the author and the article classes, since we want that process to be as seamless as possible, and any background processing or information that needs to be handled will be taken care of by the article database class without concerning the contributing authors.

Sequence Diagrams**A. Upload an article****Upload an article****B. Share to social media**

Share to social media

