Content Hub

Requirements Document

March 30, 2020
Version 1.0

Team Galvatron

David Guo	57734162
Pengwei Zhou	73569758
Peter Han	14912166
Tony Kong	47078150
Tyler Nee	22705157
Vickie Yen	13358156
Yuting Wen	44625168

Revision History

Date	Version	Status	Prepared by	Comments
Jan 27, 2020	0.1	Start	Team Galvatron	Initial Requirements Document
Mar 30, 2020	1.0	Final Submission	Team Galvatron	Final Requirements Document

Stakeholder Sign-off List

Role	Name	Signature	Date
Project Sponsor	Peter Smith		

Project Information

Problem Statement

Within the Vancouver software development community, there is a lack of content for individual IT organizations to share in order to drive online traffic to their respective platforms.

Description

The product will consist of a website that consolidates content and points to the content's original location from other websites and repositories. The site will have admins from contributing organizations monitoring content published to the site via an admin approval system. The main interface of the website will contain a portal to access linked articles. Any individual will be able to create an account and submit content to the site subject to admin approval.

Business Requirements Summary/Scope section

Top Level Business Requirements:

Administration: As an admin, I want to be able to add other admins, add/remove approvers, add blog categories for selection.

Search/Filter: As a user, I want to be able to search or filter posts by name of article, company or category.

Submission: As a user, I want to be able to submit blog posts or blog streams by link.

Approval System: As an approver, I want to be able to approve submitted posts to publish.

Notification System: As a user, I want to be able to subscribe and receive emails periodically of newly approved posts.

Scheduler: A scheduler should run to scan blogs for new posts and remind admins to approve content.

In Scope

- The scope of the Content Aggregator is to aggregate user submitted blog posts and blog streams on a single platform. More specifically:
  - Users can submit posts.
  - Users can search and filter posts.
  - Users can receive notifications of newly published posts.
  - Approvers can approve user submitted posts.
  - Admins can add other admins, add/remove approvers, add blog categories,
  - A scheduler runs to scan approved blogs for content to publish and reminds admins to approve unpublished blog posts.
- Multi-browser support (Chrome, Microsoft Edge, Safari, Firefox)

Out-of-Scope (Stretch Goals)

- Website usage reports
- Posting articles to a twitter feed that people can subscribe to. Similar to LinkedIn, and perhaps Facebook.

Business constraints/assumption/dependencies

Constraints

- Time (14 weeks, from Jan 7th to April 10th)
- Website builders cannot be used
- Performance constraint on our free tier AWS server

Assumptions

- There is a maximum of 30 approvers
- Maximum articles is 5000

Dependencies

- To store admin/user/approver data and blogpost, we need to use MySQL
- Technology stack (NodeJS/Express, and/or React)
- The performance of our website depends on the AWS instance
- External API (Login)

What other systems are impacted

The content aggregator does not impact any software systems.

What groups/individual impacts

Our website is designed to be a blog hub that serves tech companies based in Vancouver, but any individual around the world has access to our website. Some local companies/institutions in Vancouver will be invited as approvers/admins.

Functional requirements

Legend	
Priority	1 (highest) - 5 (lowest)

Req No.	**Description**	**MVP (Y/N)**	**Priority**
1.0.0	Users can self register with an email, the name of organization and a password	Y	1
1.0.1	Send a confirmation email to fully activate the new account	N	3
1.0.2	Users can login to the web app	Y	1
1.1.0	Allow users to submit posts into the system	Y	1
1.1.1	Submission requires a title, category, and link to website/blog	Y	1
1.2.0	Allow users to read the published posts via browsing the available blog posts	Y	1
1.2.1	Allow users to read the published posts via RSS feed	Y	2
1.3.0	Allow users to search or filter posts by name, company or category	Y	2
1.3.1	Categories can be managed in the admin modules of the system	Y	3
1.4.0	Notify users via email when admin members have approved an article	N	2
1.4.1	Users who have subscribed to the notification service will receive an email showing all newly published articles	Y	1
2.0.0	Administrators can authorize companies as approvers	Y	1
2.1.0	Allow approvers to vote and authorize content to be published	Y	1
2.1.1	Approval threshold for submissions can be changed according to user type	Y	2
2.2.0	Allow administrator to set the schedule - Schedule checking blog streams for new posts to submit	Y	2
2.3.0	Allow administrator to delete posts	Y	1
2.4.0	Allow administrator to delete blog streams	Y	1

Non-Functional Requirements

Req No.	**Description**	**Priority**

3.0.0	Limit response times for searches and filters be less than 4 seconds.	2
3.1.0	Provide support for at least 10000 concurrent users	2
3.2.0	Multi-browser support (Chrome, Microsoft Edge, Safari, Firefox)	1

Costs

Items	Cost Breakdown
AWS EC2 Instance	Free Tier

Use Cases

Submit a Blog Post/Stream (See Diagram Below - Figure 1A.)	
Description	A user is able to submit a blog post or blog stream to the content aggregator.
Primary Actor	User
Preconditions	User is logged into the system
Postconditions	A new post submission is created with the state “Pending Approval”
Main scenario	1. User selects “Submit a Post” on the home page 2. Website displays a “New Post” form - Title - Link - Category - Type (Blog Post/Blog Stream) 3. Website creates the new post submission with state “Pending Approval”
Alternate scenario	- User does not fill in required fields - App displays appropriate validation error upon submit - User submits invalid link - App displays invalid link error upon submit

Approve a Blog Post/Stream (See Diagram Below - Figure 2A.)	
Description	An admin approves a blog post/stream.
Primary Actor	Admin
Preconditions	- Admin is logged into the system - Submissions in “Pending Approval” state is non-empty
Postconditions	The unpublished submission’s number of approvals increases by one and is published to the home page if the number of approvals equals the approval threshold.
Main scenario	1. Admin navigates to unpublished posts page

	2. Admin approves a blog post/stream by selecting “Approve” for a specific submission 3. The unpublished submission’s number of approvals increases by one
Alternate scenario	- If the unpublished submission’s number of approvals equals the approval threshold, the post is automatically published to the home page - Different authorizations such as admin or user require different approvals thresholds

Delete a Blog Post (See Diagram Below - Figure 3A.)	
Description	An admin deletes an unpublished blog post on the unpublished posts page or deletes a published blog post on the home page.
Primary Actor	Admin
Preconditions	Admin is logged into the system
Postconditions	The blog post selected is removed from the unpublished posts page or home page.
Main scenario	**Delete an Unpublished Post** 1. Admin navigates to the unpublished posts page. 2. Admin presses the delete button on the unpublished blog post 3. A confirmation dialog pops up 4. User confirms their deletion action **Delete a Published Post** 1. Admin navigates to the home page. 2. Admin hovers over the published post they want to delete. 3. A delete button will appear upon hover and the admin clicks the delete button. 4. A confirmation dialog pops up. 5. User confirms their deletion action

Delete a Blog Stream (See Diagram Below - Figure 3B.)	
Description	An admin deletes a blog stream.
Primary Actor	Admin
Preconditions	Admin is logged into the system
Postconditions	The blog stream is removed from the content aggregator..
Main scenario	1. Admin navigates to the Admin panel page 2. Admin navigates to Blog Stream Manager tab 3. Admin deletes the blog stream 4. A confirmation dialog pops up 5. User confirms their deletion action

Scheduler reading recently approved posts and sending email notifications (See Diagram Below - Figure 4A.)	
Description	A scheduler checks for new posts and sends email notification to all subscribers
Primary Actor	Scheduler
Preconditions	Scheduler is turned on

Postconditions	Newly approved posts are retrieved and sent in an email to subscribers
Main scenario	1. Scheduler reads all approved posts in a given frequency (past day or past week) 2. Scheduler gathers all emails of all the subscribers 3. Scheduler composes an email to all of the subscribers containing the new posts

Scheduler reading blog streams (See Diagram Below - Figure 5A.)	
Description	A scheduler checks blog streams for any newly published posts and automatically submits the posts for approval.
Primary Actor	Scheduler
Preconditions	Scheduler is turned on.
Postconditions	Newly published posts from the blog streams are submitted for approval
Main scenario	1. Scheduler reads all approved blog streams 2. Scheduler gathers all the newly published posts since the last time it was executed 3. Scheduler submits new posts for approval

Diagrams

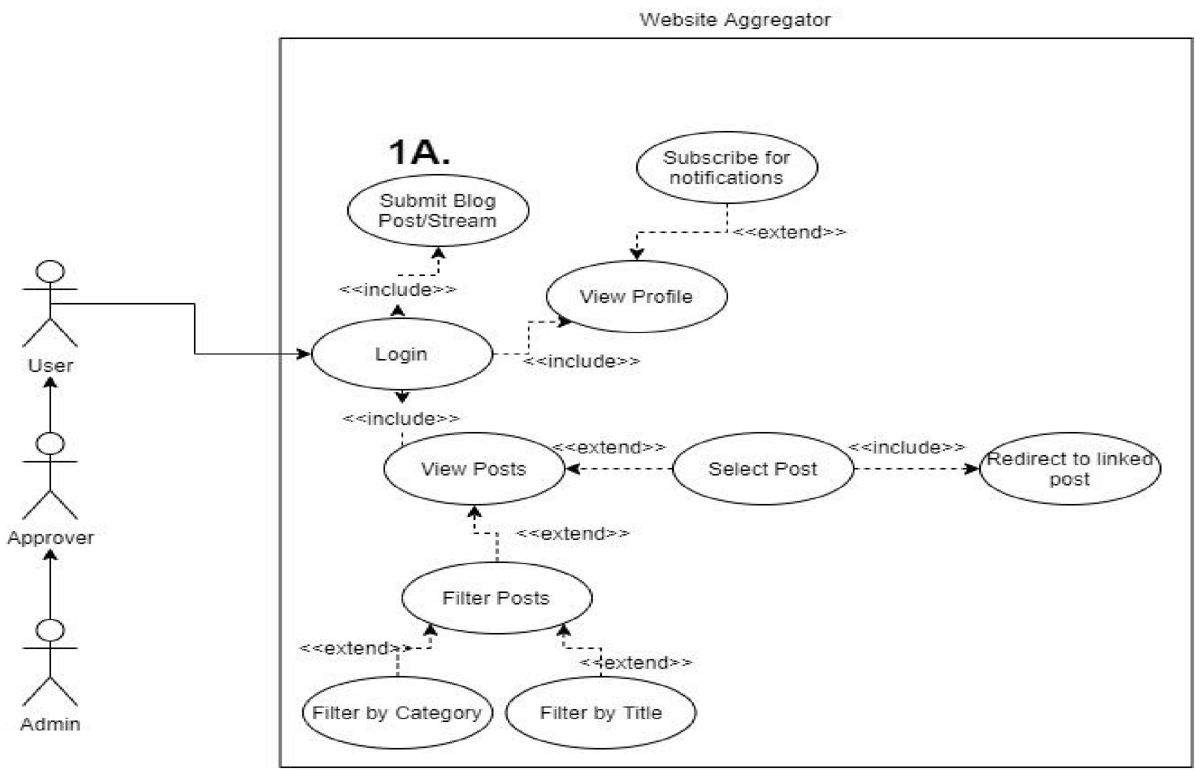

Figure 1. Use case diagram for Users

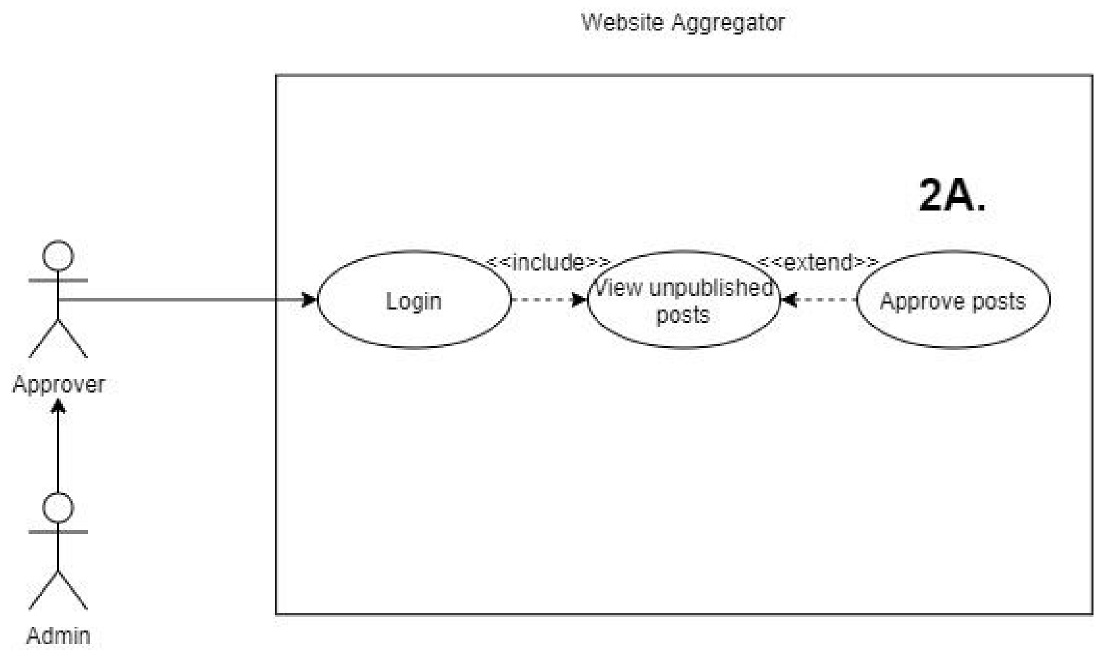

Figure 2. Use case diagram for Approvers

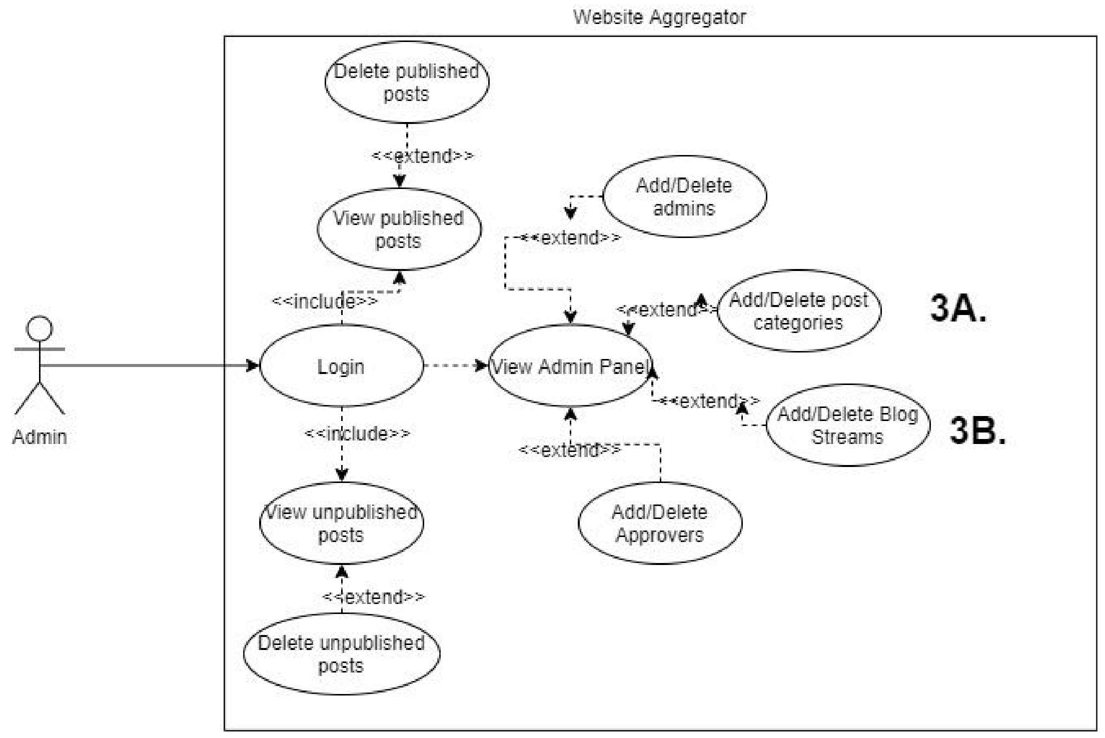

Figure 3. Use case diagram for Admins

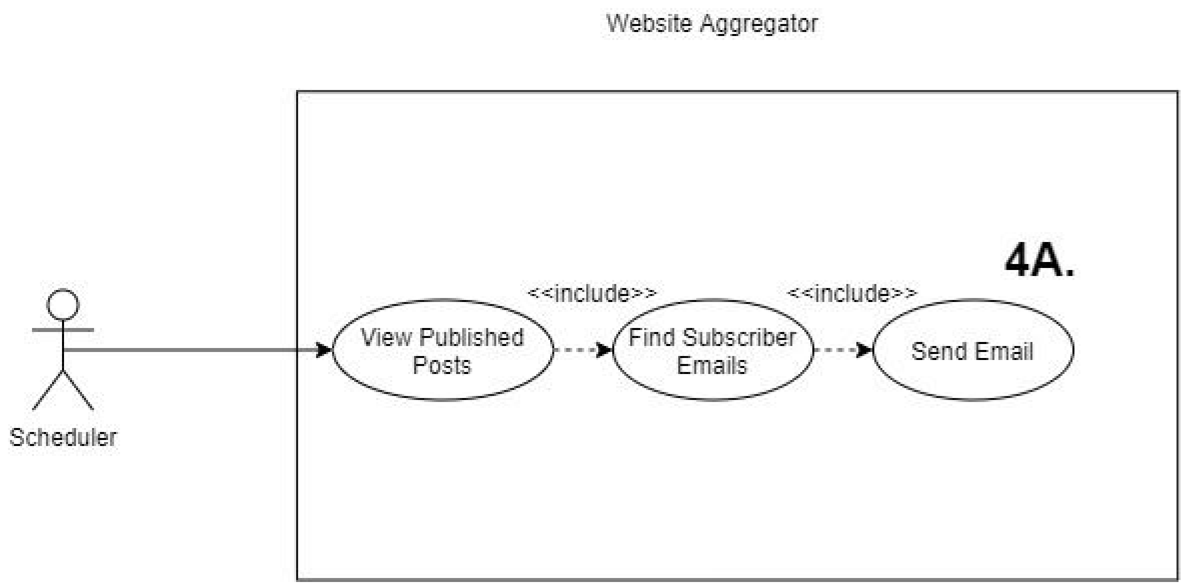

Figure 4. Use case diagram for Scheduler Emails

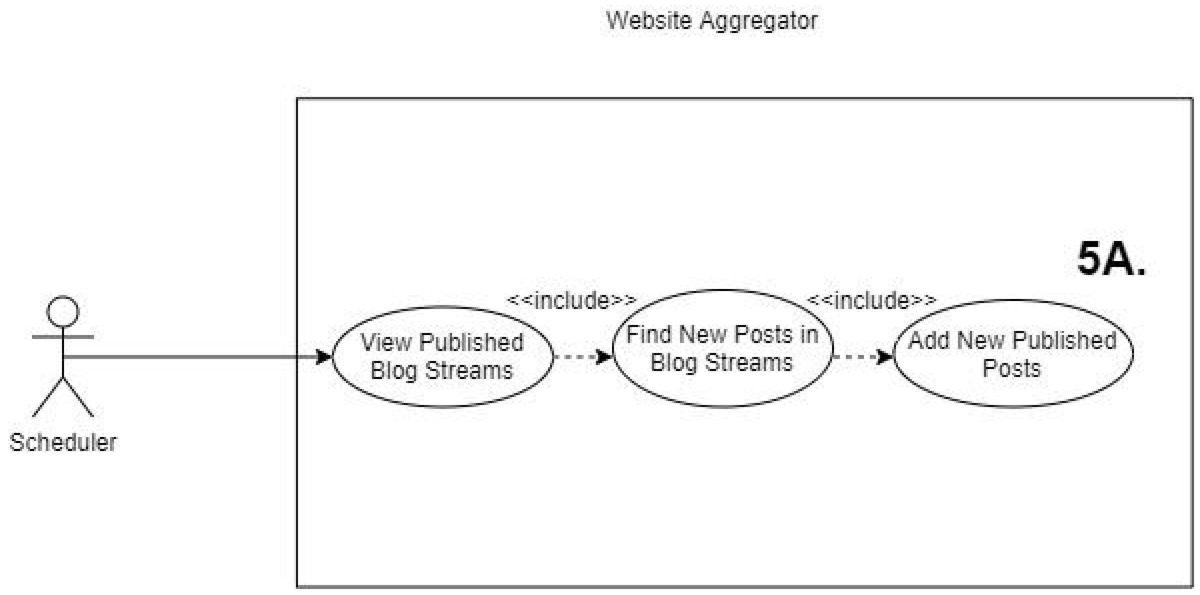

Figure 5. Use case diagram for Scheduler Blog Stream Posts