System and Software Architecture Description (SSAD)

Scriptonomics

Team 07

Nicky Singh - Life Cycle Planner

Anitha Neelakantan - Software Architect

Nikhita Reddy Gade - Project Manager

Vaishnavi Venkatraman - Operational Concept Engineer

Aditya Holikatti - Feasibility Engineer

Sri Anusha Veeramachineni - Requirements Engineer

Michael Cappuccio - IIV&V & Prototyper

Alex Miller - IIV&V + Website Maintainer

Version History

Date	Author	Version	Changes made	Rationale
10/10/17	Alex	**1.0**	**Initial Template for use in project development.**	**Initial draft for use with FCR ARB**
11/25/17	Alex	**1.1**		

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the SSAD

The main purpose of this SSAD was to establish a conceptual model that defined the structure, behavior, and systems level perspective of the software. The SSAD contains a description of our project system and its representation organized in a way that supports reasoning about the structures and behaviors of the overall system.

1.2 Status of the SSAD

This version of the SSAD provides information about the project at the time of delivery to the client.

2. System Analysis

2.1 System Analysis Overview

The Scriptonomics Blog application was built to drive new users to the platform and engage current users of the Scriptonomics service. The system will allow the creation and publication of blog entries that will be viewed, shared, commented, and liked by current registered users and new non-registered users. The system will also track basic statistics associated with web traffic to monitor user engagement such as page views and time spent on page. The web traffic statistics will only be available to the Admin users.

2.1.1 System Context

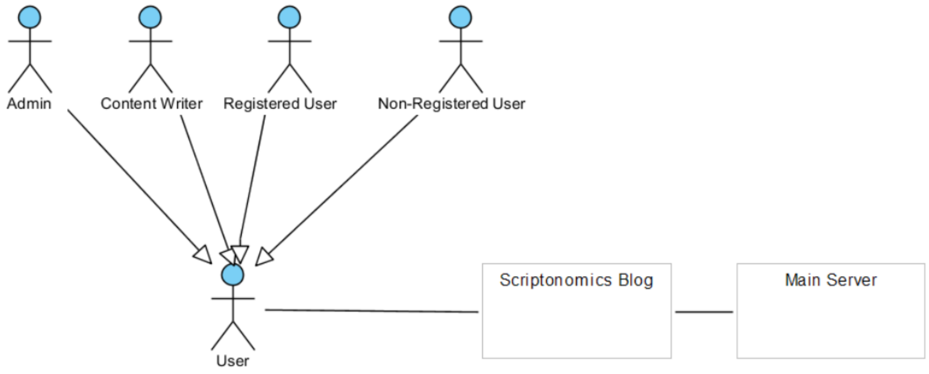


Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Admin	User with full access to tools and rights in order to manage the system.	• Control user privileges • Edit Blog Posts • Publish Blog Posts • Remove Blog Posts • Delete comments • Create Draft Blog Post • Edit Draft Blog post created by any user • Delete draft blog post created by any user
Content Writer	User with access to edit and submit draft blog posts as well as use the system as a registered reader.	• Create Draft Blog Post • Edit Draft Blog Post • Delete Draft Blog Post created by the user
Registered User	Registered user with read-only access	• View/read blog posts • Share blog posts • Like blog posts and user comments • Comment on blog posts
Non-registered User	Non-registered user with read-only access	• View/read blog posts • Share blog posts • Like blog posts and user comments

2.1.2 Artifacts & Information

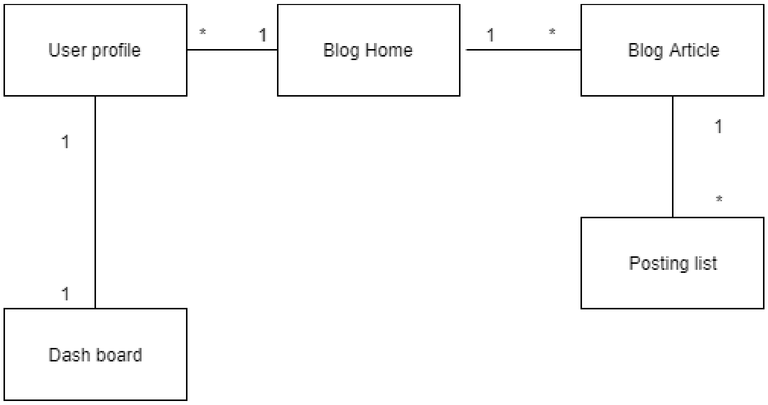


Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
Posting List	Posting List contains the actual post
Blog Article	Blog Article which display all available posts
Blog Home	Blog Homepage which display all available blogs
User Profile	Platform for user to share information about oneself
Dashboard	Contains web traffic analytics

2.1.3 Behavior

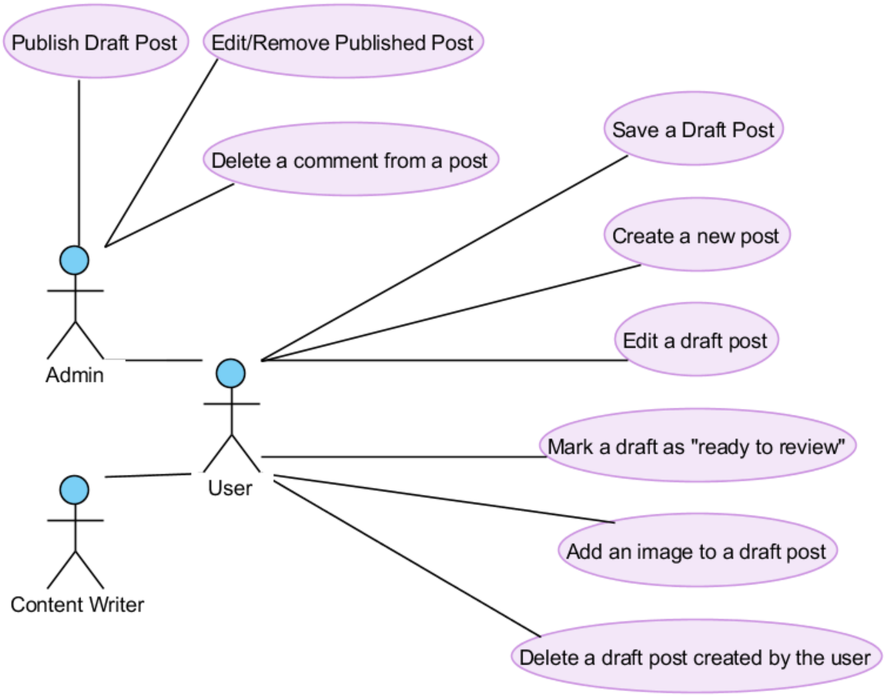

Publish Draft Post
Edit/Remove Published Post
Delete a comment from a post
Save a Draft Post
Create a new post
Edit a draft post
Mark a draft as "ready to review"
Add an image to a draft post
Delete a draft post created by the user
Admin
User
Content Writer

Figure 3: Process Diagram for Admin and Content Writers

2.1.3.1 Capability: Blog Post Creation, Editing, and Publishing

2.1.3.2.1 Process Descriptions

Table 3: Blog Post Creation Process Description

Identifier	Create a new post
Purpose	Allows the user to generate a new blog post and associated webpage.
Requirements	User rights and privileges
Development Risks	It may be difficult to implement a user access control system that can integrate with the existing site.
Pre-conditions	User is logged in
Post-conditions	None

Table 4: Publish Draft Blog Post Process Description

Identifier	Publish Draft Post
Purpose	Allows the user with admin rights to make a draft post visible to the public.
Requirements	Accessed by user with appropriate rights and privileges (admin)
Development Risks	None
Pre-conditions	Marked as “ready to review” by the author (user)
Post-conditions	None

Table 5: Edit/Remove Published Post Process Description

Identifier	Edit/Remove Published Post
Purpose	Allows the user with admin rights to alter a published post.
Requirements	Accessed by user with appropriate rights and privileges (admin)
Development Risks	None
Pre-conditions	Post is already publicly visible
Post-conditions	Post will remain in original address in the website.

Table 6: Delete Comment Process Description

Identifier	Delete a comment from a post
Purpose	Allows admin to remove inappropriate comments
Requirements	Accessed by user with appropriate rights and privileges (admin)
Development Risks	None
Pre-conditions	Admin is logged in and has selected the “Edit Post” option
Post-conditions	Post will remain in original address in the website

Table 7: Save Draft Blog Post Process Description

Identifier	Save a draft post
Purpose	Allows user to save progress prior to closing browser (logging out)
Requirements	Accessed by user with appropriate rights and privileges (admin or content writer)
Development Risks	None
Pre-conditions	User is logged in and has selected the “save post” button
Post-conditions	The post (page) will be updated with the current information.

Table 8: Edit Draft Post Process Description

Identifier	Edit a draft post
Purpose	Allows the user to edit a draft before it is published.
Requirements	Accessed by user with appropriate rights and privileges (admin or content writer)
Development Risks	None
Pre-conditions	User is logged in, has selected "edit draft post", and the post has already been saved on the site.
Post-conditions	None

Table 9: Indicate Draft Post as "ready to review" Process Description

Identifier	Mark a draft as "ready to review"
Purpose	Allows user to indicate to the admin that a post is no longer being edited prior to release.
Requirements	Accessed by user with appropriate rights and privileges (admin or content writer)
Development Risks	None
Pre-conditions	User is logged in, has selected "edit draft post", and the post has already been saved on the site.
Post-conditions	The post will be tagged as "ready to review". The tag is viewable by content writers and admins only.

Table 10: Blog Post Image Upload Process Description

Identifier	Add an image to a draft post
Purpose	Allows the user to upload images to blog post drafts.
Requirements	Accessed by user with appropriate rights and privileges (admin or content writer)
Development Risks	None
Pre-conditions	User is logged in, and is in the “post editing” page
Post-conditions	The image will be uploaded and linked to the post it was uploaded to.

Table 11: Draft Blog Post Deletion Process Description

Identifier	Delete a draft post created by the user
Purpose	Allows users to delete works in progress.
Requirements	User is logged in, and has access to the draft post list.
Development Risks	None
Pre-conditions	User created the draft post that is being deleted (unless user is admin). Draft is saved in system.
Post-conditions	Draft post moves to “recycle bin” for 72 hours. After 72 hours, it is automatically deleted.

Table 12: Typical Course of Action

Seq#	Actor's Action	System's Response
1	Admin login	System displays Admin Dashboard Page
2	Admin navigates to Blog Posting List	System displays Blog Posting List
3	Admin deletes most recent blog post	System removes the post from the public listing but does not delete the post from the system.

Table 13: Alternate Course of Action

Seq#	Actor's Action	System's Response
1	Admin login	System displays Admin Dashboard Page
2	Admin navigates to Blog Posting List	System displays Blog Posting List
3	Admin selects post draft to publish	System makes the post viewable to the public in the blog home page and the individual post's page

Table 14: Alternate Course of Action

Seq#	Actor's Action	System's Response
1	User Accesses Blog Home Page	System displays Blog home page
2	User clicks a blog headline	System displays selected Blog Post Page
3	User clicks "comment"	System prompts user to login/sign up
4	User logs in with existing ID	System returns to user's previously selected comment link.

Table 15: Alternate Course of Action

Seq#	Actor's Action	System's Response
1	User Accesses Blog Home Page	System displays Blog home page
2	User clicks a blog headline	System displays selected Blog Post Page
3	User clicks “comment”	System prompts user to login/sign up
4	User signs up with the Scriptonomics Site	System returns to user's previously selected comment link.

2.1.4 Modes of Operation

The software has a single mode of operation; blogs are created by content writer and are sent to Admin to be published. The admin can evaluate and publish the blog on the main blog home page.

2.2 System Analysis Rationale

Based on an analysis of user-system interaction for blog post creation and publishing, there were two main classes of operation stakeholders:

- *Scriptonomics Admins:* These users are Scriptonomics team representatives that manually control which user blogs will be displayed on the blog homepage and control various functions of the blogs and home page. These users are responsible for publishing the blog posts in draft state, as well as removing and editing published posts. These functionality can be controlled by the admin login page.
- *Scriptonomics content writer/User:* These users are content writers with system privileges that allow the creation of a post, marking a post as a “draft” (not ready for review), marking a draft post as in the “to be reviewed” state so admin can review them and subsequently review and publish it. They can also edit post in draft mode and like other posts available on blog homepage. They are responsible for providing new content and also enjoy other people post and provide feedback to them.

3. Technology-Independent Model

3.1 Design Overview

3.1.1 System Structure

Commented [MAM1]: Needs to be updated with .vpp

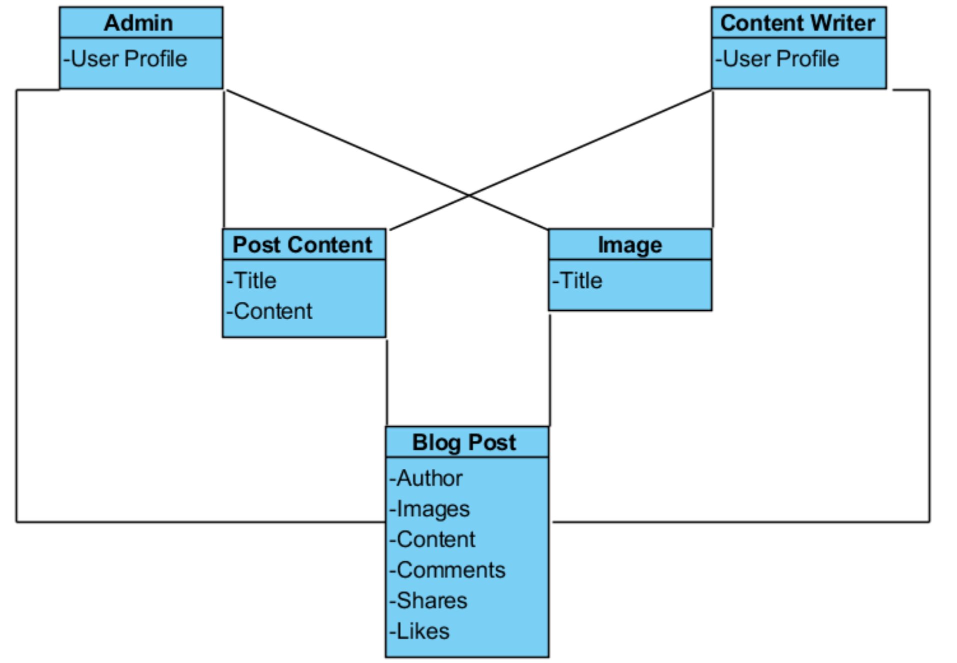


Figure 4: Conceptual Domain Model for Blog Post Creation

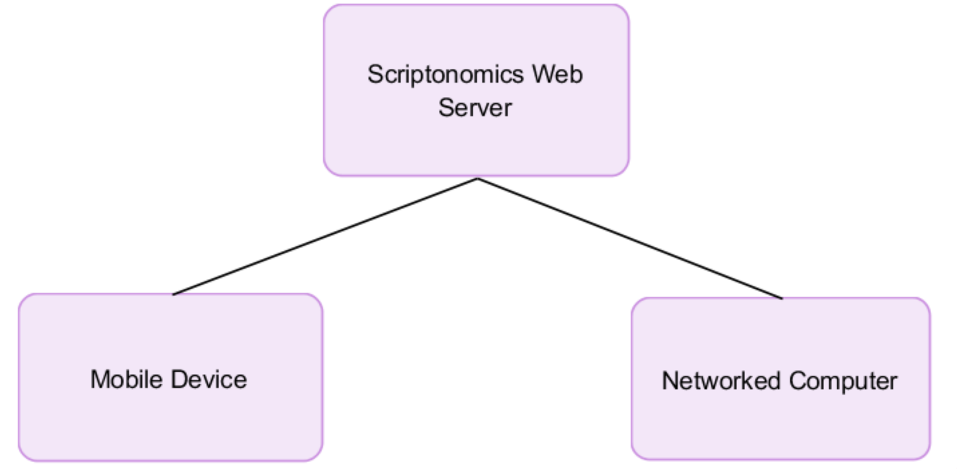


Figure 5: Hardware Component Class Diagram

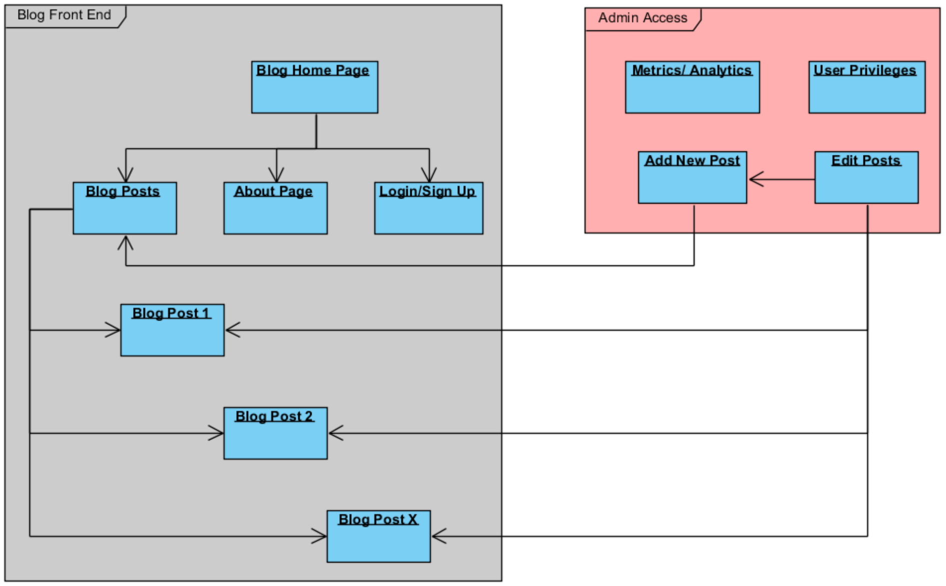


Figure 6: Software Component Class Diagram

Commented [MAM2]: Needs to be updated with .vpp

Commented [MAM3]: Needs

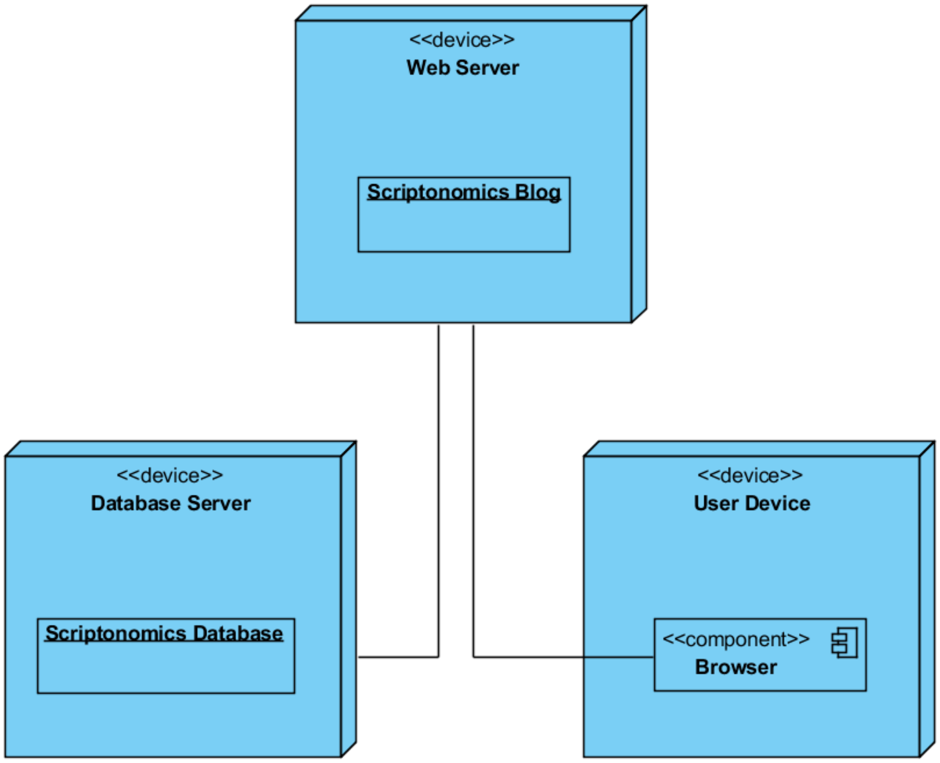


Figure 7: Deployment Diagram

Commented [MAM4]: Needs to be updated with .vpp

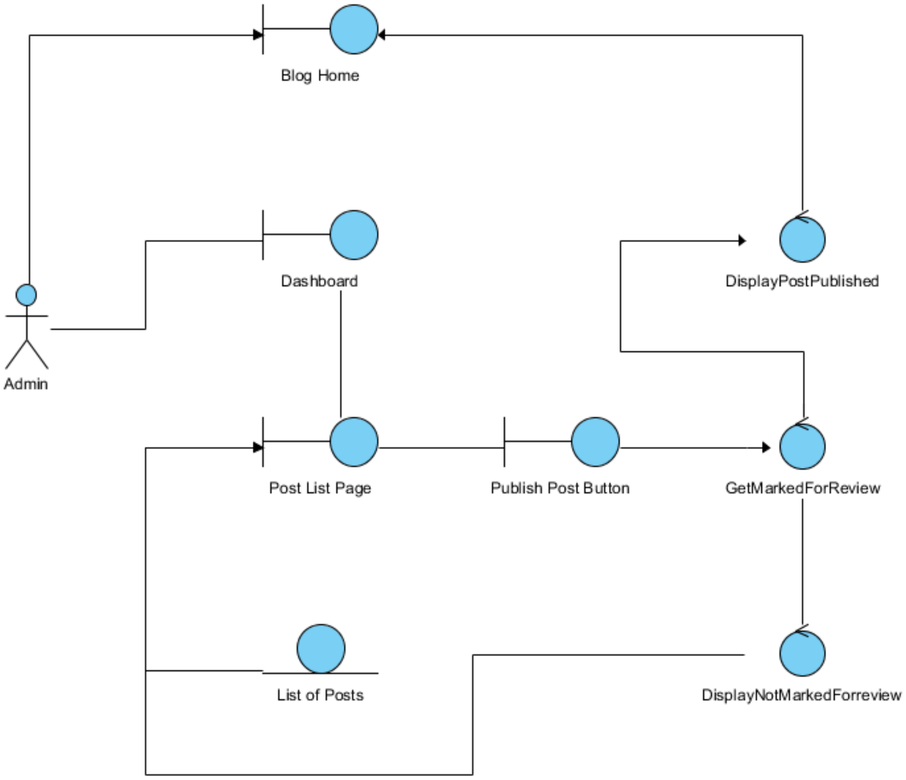
Table 16: Hardware Component Description

Hardware Component	Description
Web Server	Hosts all of the Scriptonomics data and website information
Database Server	Hosts the Scriptonomics database
Networked Computer	User's personal computer (desktop or laptop). Used to interact with the Scriptonomics website from a browser.
Mobile Device	User's mobile device (i.e. phone, tablet). Used to interact with the Scriptonomics website from a browser.

Table 17: Software Component Description

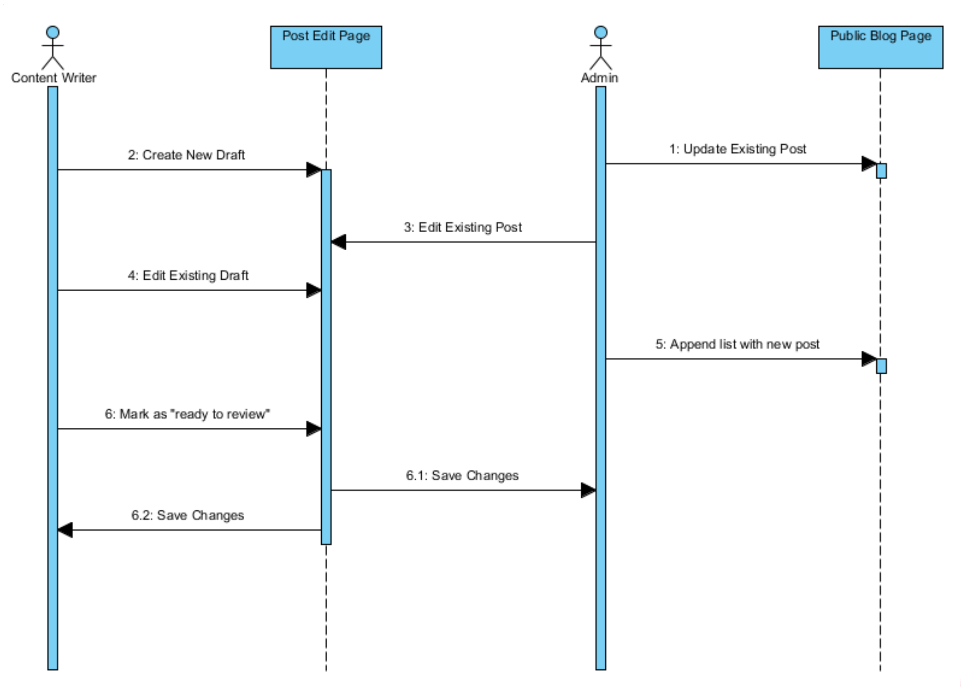
Software Component	Description
Web Browser	Displays a GUI of the Scriptonomics website and facilitates communication between the user and the web server
Web Server OS	Responsible for routing traffic and prioritizing access

3.1.2 Process Realization



Commented [MAM5]: Needs to be updated with .vpp

Figure 8: Robustness Diagram for Post Publishing



Commented [MAM6]: Needs to be updated with .vpp

Figure 9: Sequence Diagram for Blog Creation, Editing, and Publishing

3.2 Design Rationale

The main motivation to choose the architecture and design of our project was to make sure that it’s not platform dependent. The system design was inter-dependable which means functioning of one module do reflect its impact on other modules of the system. For example, Admin persona controls the right to add and remove blogs submitted to it and user persona

Commented [MAM7]: I don’t understand this sentence.

Commented [MAM8]: This example is incomplete.

4. Technology-Specific System Design

4.1 Design Overview

4.1.1 System Structure

Table 18: Supporting Software Component Description

Support Software Component	Description
Zinnia	Open source Blog application for build on top of Django.

4.2 Design Rationale

Zinnia was used because it is easy to install inside of the already build Scriptonomics website and architecture. Both are built on the Django framework (Table 19). Zinnia offers an all in one customizable blogging application which several desirable features such as comments, pre-publication, and templates.

5. Architectural Styles, Patterns and Frameworks

Table 19: Architectural Styles, Patterns, and Frameworks

Name	Description	Benefits, Costs, and Limitations
Django	Python based web framework.	● Open Source ● Many pre-built applications and features ● Scalable