CHAPTER II: REQUIREMENT ANALYSIS

2.1 Literature Review

Content Management system is a software for the creation, organization, and distribution of digital content in a digital aspect. This system provides authors with an opportunity to publish content and helps users reach out to that content in a single platform. Content Management System has become an important platform for managing content in an organized way and delivering users with specific content. With the growth of tools and technologies, there is a global change in how users are receiving content. From books to pdf files, and pdf files to videos, users expect the content to be delivered in a more appealing form. As a result, there is a need for an efficient and effective content management system that addresses the needs of the users. This literature review aims to provide a thorough overview of key concepts, evolutions, challenges, and trends of the content management system.

There is a drastic change in the evolution of the Content Management System. The Content Management System evolved with the development of the first static CMS called Jekyll. Jekyll is a simple CMS that helps in generating static content that does not require any database and server-side scripts. As the pages are static in Jekyll, it loads fast. Due to the blooming of the large content and widening of the business, dynamic content becomes obvious. The development of the Video Text Management System provided an opportunity for the user to experience the dynamic content. Many open-source dynamic Content Management Systems like Wix, Drupal, Joomla, and WordPress [1] came to the hype because they provided a platform to create customizable content with rich user interactivity. WordPress is now very popular for its flexibility and user-friendly interface. It stores the content in a database and retrieves the content easily and dynamically. WordPress provides a theme that is responsive to many devices. It provides regular updates to avoid any kind of susceptibility. It also provides multilingual support for supporting users in different languages. [2] Similarly, the platform of the content management system is evolving continuously with underway enhancement of content management, user interactivity, security, and responsiveness.

Although these Content Management Systems provide user-friendly content with responsive design, Content Management Systems are still not able to address the problem of blind users. Up to date, no CMS is friendly to blind users. [3] In addition, there is a trend

in people that they like to search through voice. They don’t like to type. Excessive typing may result in strained muscles, eye strain, and poor posture. That's why users try to avoid typing. The website called listening.io accepts the input texts from the users and converts them into the audio file that reads out the text. [4] The development of this website sets the proof that users are now in need of a Content Management System that drifts the CMS towards a text-to-speech converter. Though this is the era of the internet, users are not always accessible on the internet. Users want to download the content and access the content even if it does not connect them to the internet. Likewise, users want to interact with the author or editor easily whenever they are using a CMS. A website like GeeksforGeek [5] does not give the facilities for downloading the content and interacting with the author or editor. Although Wikipedia allows users to interact with the author or editor, the process of interacting is long and difficult.

Floating throughout the evolution of the Content Management System, we are planning to make an SEO-friendly Content Management System that will fulfill the shortcomings of the previous CMS. Our Content Management System will allow users to create, post, and organize content that will be in the form of text, images, and video by strictly addressing the needs of blind users. Our CMS will use a speech-to-text converter that allows the users to write content through the voice command. In the similar way, this system will include a text-to-speech converter that will be helpful for blind users to read the pdf files. This system will be friendly to blind people, which will help in taking education one step closer towards inclusive education. Likewise, users will download and access the PDF files even in the absence of the internet. This will help users to access the content from anywhere at any time. Our system will provide rich interactivity by providing users with a facility to add a comment and interact with the author and editor. This system provides an opportunity for the users to apply for the author and can create and post content like an author after the editor or admin has accepted the request. This system will store the content in the database and retrieve it dynamically. Overall, our Content Management System will address diverse needs, prioritize accessibility, and encourage user engagement through innovative features.

2.2 Problem Definition

In this time, students are often bored to read from the same textbooks and it is very difficult to find the right content from the Google like search engine. Finding the appropriate and right content according to their syllabus is a huge task. Similarly, teachers are not able to

provide note to every student and writing on the board could be really painful. While students cannot just sit and read the notes given by the teachers. The collaboration and interaction about the educators and learners are a huge problem during content creation and content delivery. Other external persons are not able to provide some help to the students by providing note to each and every student.

2.3 Proposed Solution

Our proposed solution for a Content Management System is to develop a web-based platform that addresses key challenges in content creation, delivery, and collaboration for educators, students, and contributors. To enhance accessibility, the system incorporates voice-based input, streamlining content creation and allowing educators to focus on delivering high-quality educational materials. This transition reduces time spent on administrative tasks while supporting students with diverse learning needs.

To promote self-paced learning, the platform allows students to listen to course content during their free time, promoting self-paced learning, reinforcing concepts, and improving retention. Teachers can efficiently upload and share notes, videos, and other learning resources, ensuring that students can easily access syllabus-specific content without the distraction of irrelevant materials from generic search engines. The system also fosters collaboration through interactive features, enabling users to comment, interact with authors, and apply to contribute content after approval from editors or admins.

Additionally, it supports offline learning by allowing users to download notes and PDFs, ensuring continuity even without internet access. With tools for category and user management, as well as a focus on voice-based innovation and interactivity, this Content Management System transforms the educational experience into a dynamic, engaging, and accessible journey for all users.

2.4 Requirement Analysis

In our project, we have collected a list of documents with sufficient and necessary requirements for the project development. To derive the requirements, we have done better understanding of the products under the development which we achieved through detailed and continuous communications with the project team throughout the software development process.

2.4.1 Requirement Identification

2.4.1.1 Study of Existing System

GyanShristi: Content Management System is a web application to automate the content creation and delivery process easier, faster, and interactive. The purpose of this system is to provide a better environment for students to learn.

2.4.1.2 Requirement collection

There are two types of requirement collection i.e. functional and non-functional requirement.

A. Functional Requirement

1. Viewer

1.1 Search for the Content

1.1.1 Search based on categories. Each content has a category which might have sub categories.

1.1.2 Search for the result based on specific keyword of the heading.

1.1.3 Search the content based on voice recognition.

1.1.4 View the contents even without logging in.

1.1.5 Listen to the text content through text to speech.

1.2 Provide Comment

1.2.1 Create an account and comment on the content or post to provide feedback.

1.2.2 Reply to the comments on the posts/contents.

1.3 Manage the content

1.3.1 Apply as an author and wait for approval from author or editor.

1.3.2 Upload contents which need approval from admin or editor after approval.

1.4 Apply as Author

1.4.1 Apply as author to create posts/contents

1.5 Login

1.5.1 Login into the system.

1.6 Register

1.6.1 Register to use different functionalities.

1.7 Logout

1.7.1 Viewers can logout from the system.

2. Author

2.1 Manage the contents

2.1.1 Add categories to manage the content properly according to category and sub category

2.1.2 Upload the contents which needs approval from admin or editor.

2.1.3 Update the contents and needs approval from admin or editor.

2.1.4 Create content based on voice recognition.

2.2 Provide Comment

2.2.1 Create an account and comment on the content or post to provide feedback.

2.2.2 Reply to the comments on the post/comments.

2.3 Login

2.3.1 Login into the system.

2.4 Register

2.4.1 Register to use different functionalities.

2.5 Logout

2.5.1 Viewers can logout from the system.

3. Editor

3.1 Manage the contents

3.1.1 Add categories to manage the content properly according to category and sub category

3.1.2 Approve or unapproved content written by the author.

3.1.3 Create content based on voice recognition.

3.1.4 Upload contents without approval.

3.1.5 Update the contents without approval.

3.2 Create and manage author

3.2.1 Create authors.

3.2.2 Update authors.

3.2.3 Delete authors.

3.2.4 Approve viewer to be author.

3.3 Provide Comment

3.3.1 Create an account and comment on the content or post to provide feedback.

3.3.2 Reply to the comments on the post/comments.

3.4 Login

3.4.1 Login into the system.

3.5 Register

3.5.1 Register to use different functionalities.

3.6 Logout

3.6.1 Viewers can logout from the system.

4. Admin

4.1 Manage users

4.1.1 Create every user in the system.

4.1.2 View users list.

4.1.3 Update every user in the system.

4.1.4 Delete every user in the system.

4.1.5 Approve the viewer to be the author.

4.2 Manage the content

4.2.1 Create content.

4.2.2 Update content.

4.2.3 Delete content.

4.2.4 View content.

4.2.5 Approve contents written by the authors.

4.2.6 Add categories to manage the content properly according to category and sub category

4.2.7 Create content based on voice recognition.

4.3 Provide Comment

4.3.1 Comment on the content or post to provide feedback.

4.3.2 Reply to the comment on the posts/contents.

4.4 Login

4.4.1 Login into the system

4.5 Register

4.5.1 Register to use different functionalities

4.6 Logout

4.6.1 Viewers can logout from the system.

B. Non-Functional Requirement

- ➢ **Performance:** The performance of the website depends upon the system specification and network strength.
- ➢ **Security:** The details of the users are well managed and accountable to every user.
- ➢ **Scalability:** The system can be extended later due to well managed codes and documents.
- ➢ **Maintainability:** The system is easy to maintain.
- ➢ **Accountability:** Every user has their own responsibilities which are tracked.
- ➢ **Supportability:** The system is supported by every device.

2.5 Feasibility Study

2.5.1 Economic Feasibility Study

Economic feasibility evaluates the financial aspects of the "GyanShristi" CMS project to determine whether the benefits cross the costs or not.

2.5.1.1 Cost Analysis

Table 2.1: Cost estimation table

SN	Category	Estimated Cost (in Rs)	Frequency
1	Paper print cost	5,000	One time
2	Development charge per person	3*5000=15,000	One time
3	Hosting and Domain	500*12=6,000	Annual
4	User support and Maintenance	5,000	Annual
5	Miscellaneous Expense (Training, Minor Fixing, Travel)	3,100	Annual
	Total Cost	33,500	

The total estimated cost for implementing and maintaining the project is Rs. 33,500, covering one-time expenses such as paper printing and development charges, alongside

annual costs for hosting, user support, and miscellaneous expenses. This budget ensures efficient project execution and ongoing support.

2.5.1.2 Payback Period

Payback period is a financial metric used to assess the time it takes for an investment to recoup its initial cost through the cash flows it generates.

Net annual cash inflow = Total Annual Cash Inflows - Total Annual Cash Outflow

= 60,000 - 10,000

= 50,000

Payback period = initial investment / net annual cash inflow

=33,500/50,000

= 8.1 months

The payback period for the project is calculated to be 8.1 months, indicating that the initial investment of Rs. 33,500 will be recovered through net annual cash inflows of Rs. 50,000 within this time frame. This short payback period demonstrates the project's financial viability and quick return on investment.

2.5.1.3 Return on Investment (ROI)

The return on investment (ROI) is a financial metric used to evaluate the profitability of an investment relative to its cost. It measures the percentage return or profit generated from an investment over a specific period of time.

ROI = (Net profit/initial investment) *100

Here,

Net profit = (Net annual cash inflow - Initial Investment)

= 50,000 - 33,500

= 16,500

So, ROI = Net profit / initial investment * 100%

= 16,500/33,500 * 100%

= 49.25%

The return on investment (ROI) for the project is calculated to be 49.25%, indicating that the project generates a profit of nearly half its initial investment. This high ROI demonstrates the project's strong profitability and its potential to deliver significant financial benefits over time.

2.5.2 Technical Feasibility Study

The technical feasibility of "GyanShristi" give us the information where the system is technically feasible. It lets us know whether we have enough resources or not to complete the project.

1. Technology Stack

Front-end Technologies

- **Languages:** HTML5, CSS3, Tailwind CSS, JavaScript
- **TypeScript:** For starting static typing to JavaScript, improving code quality and maintainability.
- **Frameworks:** NextJS to simplify the server-side rendering and static site generation, enhancing performance and SEO.

Back-end Technologies

- **Language:** JavaScript
- **Framework:** Express.js for server-side development
- **API architecture:** Restful APIs to ensure modularity, scalability and ease of integration.

Database

- **Type:** NoSQL
- **Database System:** MongoDB for flexible and schema less data management
- **Database Management Tool:** MongoDB GUI Compass for a user-friendly interface to explore and manipulate data.

Hosting and Development:

- **Cloud Provider:** Vercel for hosting the website that offers automatic deployment.

2. Team Expertise and Resources

Team Expertise

- **Front-end Development:** Proficiency in NextJS and TypeScript
- **Back-end Development:** Experience with Express.js
- **Database Administration:** Familiarity with MongoDB and management tools

Development Tools

- **IDE:** Visual Studio Code
- **Version Control:** Git and GitHub for code management and collaboration

3. Technical Demonstrations

- ➢ User authentication with JWT Tokens
- ➢ User validation with ZOD validation
- ➢ Encrypting and Decrypting passwords with bcrypt
- ➢ Implementing voice recognition with react-speech-recognition
- ➢ Implement text-to-speech with Web speech
- ➢ Basic Froala Text editor integrated into the content creation page.

4. System Architecture

- ➢ Use of API gateway for routing and managing API requests.
- ➢ Separate controllers for user management, content management, and media handling

5. Integrated Tools

- ➢ **Firebase:** Firebase for handling the email verification.

6. Nonfunctional Tools

- ➢ Uses Jest framework for testing the code for improved security.
- ➢ Uses bcrypt encryption algorithm for hashing the password for enhanced security.
- ➢ Uses token authentication mechanism. (JWT)

7. Technical Risks and Mitigation

Technical Risks

- ➢ Performance issues with more user load.

Mitigation Measures

- ➢ Frequent performance testing and optimizations

The technical feasibility study for "GyanShristi" CMS confirms that the project is technically feasible. The chosen technology stack aligns with the project requirements, and the team has necessary skills. Key technical challenges have been identified, ensuring the project can proceed successfully.

2.5.3 Operational Feasibility Study

1. Introduction

Operational feasibility evaluates whether the "GyanShristi" CMS can be effectively integrated into the daily operations of schools and colleges, ensuring it meets the needs of teachers, students, administrations or any other users related to this CMS.

2. User Needs and Acceptance

Stakeholder Analysis:

- **Teachers:** Needs an easy-to-use platform for creating and sharing content, including voice recognition for creating content.
- **Students:** Require engaging content, the ability to listen to content, and features to interact with content, like commenting,
- **Administrators:** Needs content approval system for managing the content in a better way.
- **Other Users:** Need to participate in the content creation program of CMS as an author.

User Involvement:

- Conduct surveys and focus groups with teachers and students to gather feedback on the propose features.
- Gather feedback from every category of users to enhance requirements.

Training and Support:

- Provide comprehensive training for the teachers, students, and administrators to ensure they are comfortable with the system.
- Create tutorial videos for other users who are users of the system.
- Provide technical support and user assistance through the help desk.

3. System Integration:

- Analyze the existing content creation and delivery process in different institutions.
- Ensure CMS integrates smoothly with the current system of different institutions.
- Evaluate any necessary data migration from current systems to the new CMS.

4. Operational Impact:

- Assess how the new system will change daily routines for users.
- Implement a change management plan to help users transition to the new system. This plan includes communications, training sessions, and support.

5. Resource Requirement:

- Assess the infrastructure required for CMS such as servers, storage, and network bandwidth.
- Ensure there are backups in one place.

6. Operational Costs:

- Estimated cost provides benefits as per the calculation of ROI.

The operational feasibility indicates that the “GyanShristi” can be effectively integrated into the existing educational environments. With appropriate training, support and change management strategies, the system can meet the needs of teachers, students, administrators, and other users.

2.5.4 Schedule Feasibility Study

Time planning deals with the measures to plan the available time within the time frame. If a project takes more than the time frame, it is likely to get rejected. Utilizing tools like Gantt charts facilitates effective time management by visually organizing tasks, dependencies, and deadlines. This ensures timely completion of project milestones and enhances the project's chances of meeting client expectations within the designated time frame.

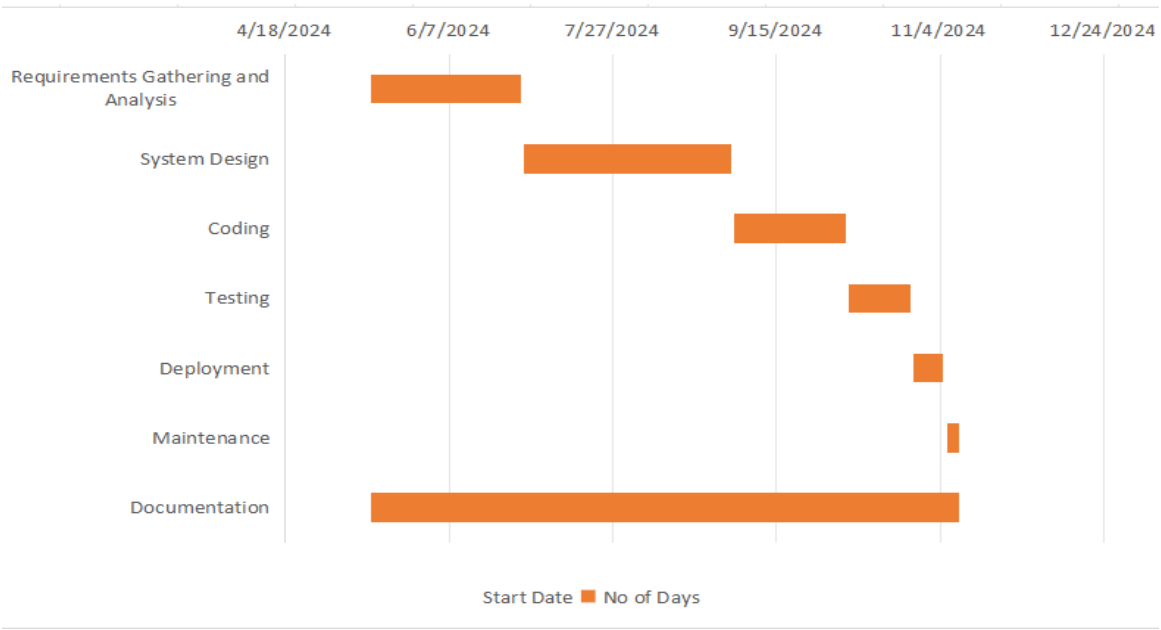

Figure 2.1: Gantt Chart (Project Timeline)

Presenting project phases and their durations in a table clarifies the sequential timeline, aiding in planning, resource allocation, and tracking progress efficiently within a structured format.

Table 2.2: SDLC Phases Duration

Description	**Number of Days**
Requirements Gathering and Analysis	46
System Design	63
Coding	34
Testing	19
Deployment	9
Maintenance	4
Documentation	180

2.6 Structuring System Requirements

Structuring system requirements is crucial in software development and systems engineering, as it ensures clarity and mutual understanding among all stakeholders. This practice reduces ambiguity and miscommunication, streamlines task tracking, and facilitates prioritization.

Moreover, structured requirements aid in early risk identification and management, simplify change processes, and promote system consistency. They also serve as a basis for legal agreements, ensuring all parties understand the deliverables.

In essence, well-structured requirements are fundamental to delivering high-quality systems on time and within scope.

2.6.1 System Flowchart

A system flowchart is a graphical representation of the processes and workflows within a system, illustrating how data flows and is processed. It uses standardized symbols to depict inputs, outputs, processes, decisions, and storage, providing a clear and concise overview of the system's functionality. System flowcharts are used to analyze, design, and document systems, making them easier to understand and troubleshoot.

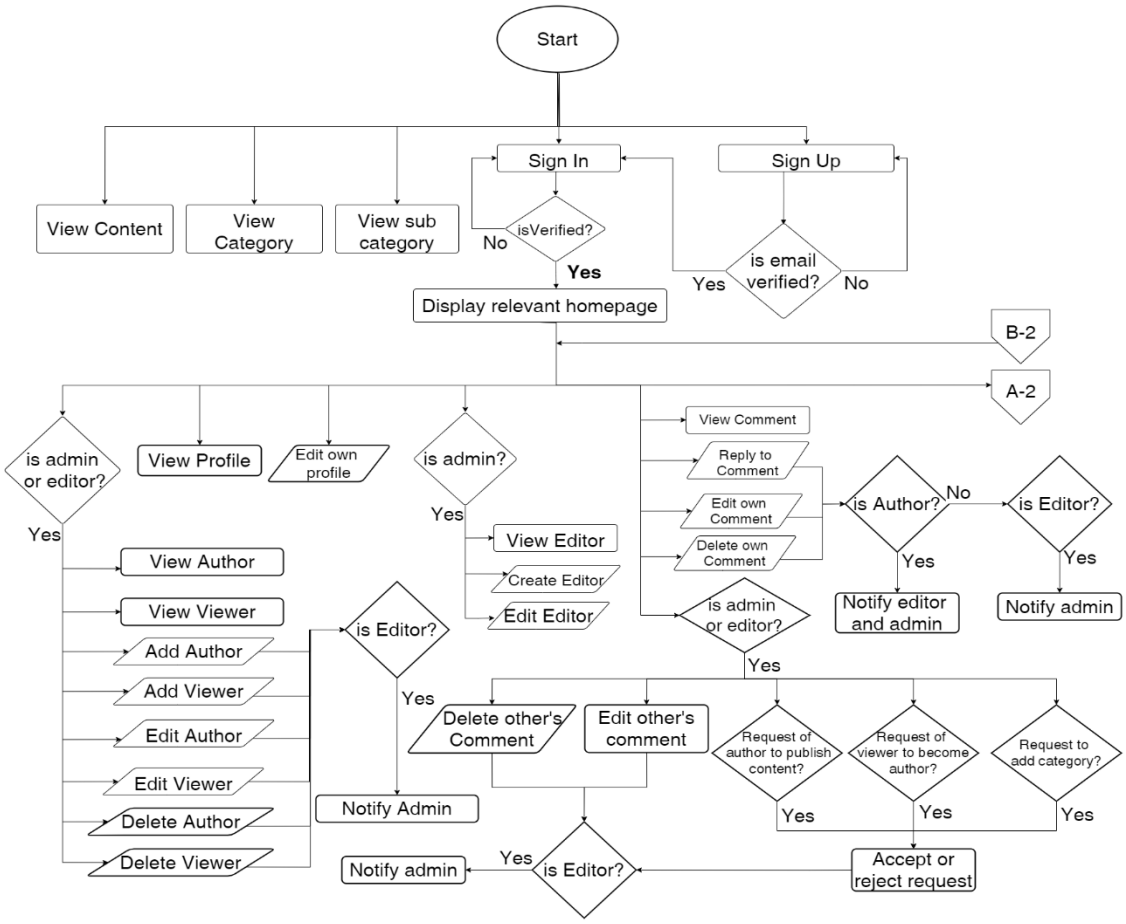

Figure 2.2: System Flowchart 1

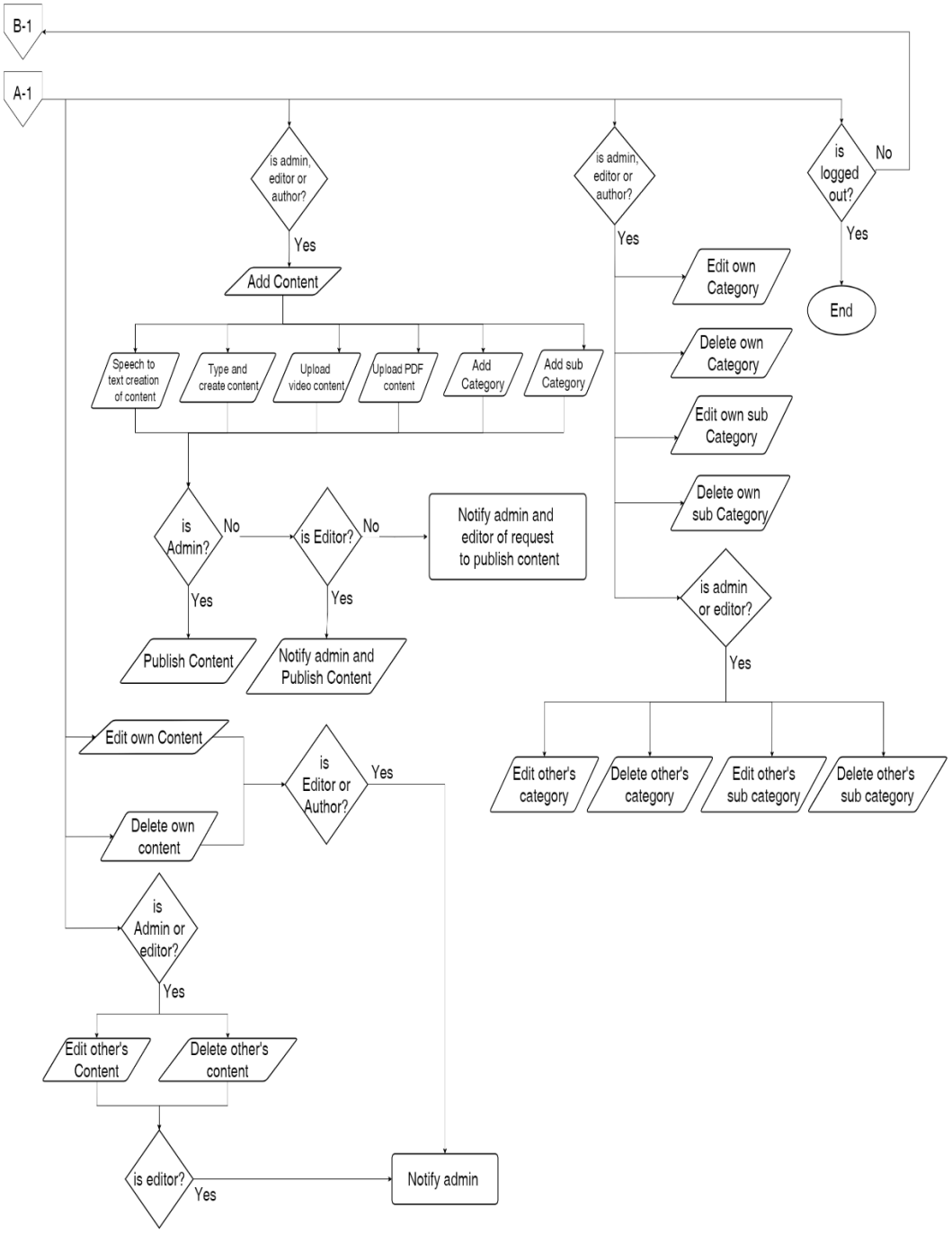

Figure 2.3: System Flowchart 2

2.6.2 Use Case Diagram

A use case diagram is a UML tool that shows the interactions between users (actors) and a system, highlighting key functions (use cases) and their relationships to define system requirements.

Figure 2.4: Use Case Diagram of Admin

Figure 2.5: Use Case Diagram of Editor

Figure 2.6: Use Case Diagram of Author

Figure 2.7: Use Case Diagram of Viewer

Table 2.3: Use Case Template of Login

Use Case Id:	1.0
Use Case Name:	Login
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Author, Editor, Viewer
Description:	This use case allows users to login into the system and perform different functions according to the user roles. All the users have to enter their unique username and password.
Precondition:	Users have to validate their account.
Post-condition:	System displays the relevant homepage.
Normal Courses:	● Users enter the username, password and choose the user role. ● System validates the username, password and user role. ● System verifies the username, password and user role. ● System displays the relevant homepage. ● The use case ends.
Alternative Courses:	1. Upon missing username, password or user role: ● System prompts for empty username, password, or user role. ● The use case resumes from the previous step. 2. Upon entering less than 6-character password: ● System prompts a message to enter a password having more than 6 characters. 3. Upon entering more than 32-character password: ● System prompts a message to enter a password having less than 32 characters.
Exceptions:	Users may not login.
Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	Users must login through a registered and verified account.

Special Requirement:	None
Assumptions:	Nil

Table 2.4: Use Case Template of Add User

Use Case Id:	2.0
Use Case Name:	Add User
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor
Description:	This use case allows the admin to add editors into the system by filling up their details. Whereas both editor and admin can add the author into the system.
Precondition:	Admin or Editor are authenticated and given permission to add users according to their role.
Post-condition:	● The added user is registered and accessible to the system. ● User roles are assigned correctly.
Normal Courses:	● Admin or editor log into the system. ● They click on the user on the navigation bar and click on the add user. ● They enter the required details and click on the add button. ● System validates the data. ● The user is added to the system according to their role.
Alternative Courses:	Upon missing any detail, the system prompts for necessary information.
Exceptions:	System failure prevents the add user process.
Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	Newly added users must have a unique username and password.

Special Requirement:	None
Assumptions:	The system is operational and accessible during add user processes.

Table 2.5: Use Case Template of Update User

Use Case Id:	3.0
Use Case Name:	Update User
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor
Description:	This use case allows the admin to update any information of the editor. Whereas both admin and editor can update the information of the author and viewer.
Precondition:	Admin or editor are authenticated and given permission to update users according to their role.
Post-condition:	The user's information is successfully updated.
Normal Courses:	● Admin or editor log into the system. ● They click on the user on the navigation bar and click the update button of the required user. ● Update the required information. ● System validates and saves the updated information.
Alternative Courses:	Upon missing any detail, the system prompts for necessary information.
Exceptions:	Nil
Includes:	None
Priority:	Medium
Frequency of use:	24 hours a day
Business Rule:	Updated information must adhere to specified data format or validation rules.

Special Requirement:	None
Assumptions:	The system is operational and accessible during user information update processes.

Table 2.6: Use Case Template of Delete User

Use Case Id:	4.0
Use Case Name:	Delete User
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor
Description:	This use case allows the admin to delete editor, viewer or author. Whereas the editor can only delete the author.
Precondition:	Admin or editor are authenticated and given permission to delete users according to their role.
Post-condition:	The deleted user's information is removed from the system.
Normal Courses:	● Admin or editor logs into the system. ● They click on the user on the navigation bar and click the delete button of the required user. ● System Prompts a confirmation box. ● They confirm the action from the confirmation box. ● The user is successfully deleted from the system.
Alternative Courses:	None
Exceptions:	System failure prevents user deletion.
Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	Deletion should be confirmed to avoid accidental removal of users.

Special Requirement:	None
Assumptions:	The system is operational and accessible during user deletion processes.

Table 2.7: Use Case Template of View User

Use Case Id:	5.0
Use Case Name:	View User
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor
Description:	This use case allows the admin and editor to view all the information about the editor, viewer and author.
Precondition:	Admin or Editor have to gain access to their account.
Post-condition:	System displays the information about users according to the user's role and access level.
Normal Courses:	● Admin, editor or author logs into the system. ● They click on the user from the navigation bar. ● Admins view all the information about both editors and authors by clicking on the category editor and author. ● Editors view all the information about the authors. ● Authors view basic details of other authors.
Alternative Courses:	Upon not getting required information, the system displays an error.
Exceptions:	System failure prevents displaying user information.
Includes:	None
Priority:	Medium
Frequency of use:	24 hours a day
Business Rule:	Access to user information is restricted based on user roles and permissions.

Special Requirement:	None
Assumptions:	The system is operational and accessible during the user viewing processes.

Table 2.8: Use Case Template of Approve Viewer

Use Case Id:	6.0
Use Case Name:	Approve Viewer
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor
Description:	This use case allows the admin and editor to approve the request sent by the viewer to be an author.
Precondition:	Admin and editor get notification of pending requests to become an author.
Post-condition:	● The viewer's status is updated to an author upon approval. ● System records the approval action and updated user roles accordingly.
Normal Courses:	● Admin or Editor logs into the system. ● They click on the notification from the navigation bar and access the pending requests. ● Accept or reject the request. ● Viewer's status is updated to an author if the request is accepted.
Alternative Courses:	If the request has already been approved, display a relevant message.
Exceptions:	System failure prevents the approval process.
Includes:	None
Priority:	Medium
Frequency of use:	24 hours a day
Business Rule:	Viewer requests need approval to become authors.
Special Requirement:	Only admins and editors have the authority to approve viewer requests to be an author.

Assumptions:	The system is operational and accessible during the approval processes.

Table 2.9: Use Case Template of Add Content

Use Case Id:	7.0
Use Case Name:	Add Content
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor, Author
Description:	This use case allows admin, editor and author to add content like posts, videos and PDFs to the system.
Precondition:	● The user initiating the action is authenticated and has necessary permissions to add content. ● Suitable storage or space is available to accommodate the new content.
Post-condition:	The new content is successfully added to the system and accessible to users based on access rights.
Normal Courses:	● Admin, Editor or Author logs into the system. ● They click on the content from the navigation bar. ● They click on the type of content (post, video, PDF) and click on the add content button inside the relevant content. ● Creates and submits the content for the approval. ● Content is uploaded after approval.
Alternative Courses:	● If the content upload fails, provide appropriate error messages and retry options. ● If there's insufficient storage space, prompt for additional space or notify administrators.
Exceptions:	System failure prevents adding content to the system.
Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	Uploaded content must adhere to specified formats and guidelines.

Special Requirement:	None
Assumptions:	The system is operational and accessible during the content addition process.

Table 2.10: Use Case Template of Update Content

Use Case Id:	8.0
Use Case Name:	Update Content
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Author, Editor
Description:	This use case allows admin, editor and author to update their own content. Admin has the authority to update other's content too. Whereas the editor has the authority to update the author's content.
Precondition:	● The user initiating the action is authenticated and has necessary permissions to update content. ● The content being updated exists in the system and is accessible to the user.
Post-condition:	The updated content is successfully modified in the system.
Normal Courses:	● Admin, Editor or Author logs into the system. ● They click on the content from the navigation bar. ● They click on the type of content (post, video, PDF) from the category. ● Click on the update button on the required content. ● Submits the content for approval. ● Content is updated after approval.
Alternative Courses:	If the content update fails, provide appropriate error messages and retry options.
Exceptions:	System failure prevents updating content.
Includes:	None
Priority:	High

Frequency of use:	24 hours a day
Business Rule:	Updated content must adhere to specified formats and guidelines.
Special Requirement:	Users must have to login.
Assumptions:	The system is operational and accessible during the content update process.

Table 2.11: Use Case Template of Delete Content

Use Case Id:	9.0
Use Case Name:	Delete Content
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor, Author
Description:	This use case allows the admin and editor to delete content. Whereas authors can only delete their own content.
Precondition:	● The user initiating the action is authenticated and has the necessary permissions to delete content. ● The content being deleted exists in the system and is accessible to the user.
Post-condition:	The deleted content is removed from the system and no longer accessible to the users.
Normal Courses:	● Admin, Editor or Author logs into the system. ● They click on the content from the navigation bar. ● They click on the type of content (post, video, PDF) from the category. ● Click the delete button of the required content. ● System Prompts a confirmation box. ● They confirm the action from the confirmation box. ● System removes the content from the system.
Alternative Courses:	If the content deletion fails, provide appropriate error messages and retry options.
Exceptions:	● System failure prevents deleting content.

	• Lack of necessary permissions prevents deleting content.
Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	Deletion should be confirmed to avoid accidental removal of content.
Special Requirement:	User must have to login
Assumptions:	The system is operational and accessible during content deletion processes.

Table 2.12: Use Case Template of Approve Content

Use Case Id:	10.0
Use Case Name:	Approve Content
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor
Description:	This use case allows the admin, and editor to approve the content uploaded by the authors. Whereas content uploaded by admin and editors does not need any approval.
Precondition:	• The user initiating the action is authenticated and has the necessary permissions. • There exist notifications about pending content uploaded by authors requiring approval.
Post-condition:	The approved content is marked as verified and becomes accessible to users.
Normal Courses:	• Admin or Editor logs into the system. • They click on the notification and access the pending content uploads. • Selects and approves the content, marking it as verified.
Alternative Courses:	• If the content approval fails or encounters issues, provide appropriate error messages and retry options.

	• If there's no pending content for approval, display a relevant message.
Exceptions:	System failure prevents the approval of content.
Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	Content uploaded by authors requires approval before being accessible to users.
Special Requirement:	Content uploaded by Admins and Editors does not need approval.
Assumptions:	The system is operational and accessible during the content approval process.

Table 2.13: Use Case Template of View Content

Use Case Id:	11.0
Use Case Name:	View Content
Created By:	Seezan Shrestha Last updated by:
Date Created:	January 3,2024 Date last updated:
Actor(s):	Admin, Editor, Author, Viewer
Description:	This use case allows all the users to view the content according to its types, such as videos, posts and PDFs.
Precondition:	The requested content exists in the system and is accessible based on user roles and permissions.
Post-condition:	The requested content is displayed to the user based on their access level.
Normal Courses:	• User logs into the system. • Choose the content according to the category. • The system presents the available content of the chosen type of the user.
Alternative Courses:	If the required content is not found, display a relevant message or suggest similar content.
Exceptions:	System failure prevents displaying content.

Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	None
Special Requirement:	None
Assumptions:	The system is operational and accessible during the content viewing process.

Table 2.14: Use Case Template of Add Comment

Use Case Id:	12.0
Use Case Name:	Add Comment
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author, viewer
Description:	This Use Case allows admin, editor, registered viewer and author to add comments to the contents.
Precondition:	The user must be logged into the system.
Post-condition:	The comment is successfully added to the selected content.
Normal Courses:	● User logs into the system. ● User navigates to the content they want to comment on. ● User clicks on the "Add Comment" button. ● User enters the comment and submits. ● System associates the comment with the selected content.
Alternative Courses:	If the user cancels the comment, the system will remain the same .
Exceptions:	If the user's session expires, they are prompted to log in again.
Includes:	None
Priority:	Medium

Frequency of use:	Daily
Business Rule:	Comments cannot exceed 500 characters.
Special Requirement:	The system should support plain text for comments.
Assumptions:	Users have the necessary permissions to add comments to the content.

Table 2.15: Use Case Template of View Comment

Use Case Id:	13.0
Use Case Name:	View Comment
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author, viewer
Description:	This use case allows the user to view comments to the contents.
Precondition:	None
Post-condition:	The user successfully views comments on the selected content.
Normal Courses:	● User logs into the system. ● User navigates to the content they want to view comments on. ● System retrieves and displays existing comments for the selected content.
Alternative Courses:	If there are no comments for the selected content, the system displays a message indicating “no comments are available”.
Exceptions:	If there is an issue retrieving comments, the system displays an error message.
Includes:	None
Priority:	Medium
Frequency of use:	Daily
Business Rule:	None

Special Requirement:	The system should support scroll down if there are a large number of comments.
Assumptions:	None

Table 2.16: Use Case Template of Delete Comment

Use Case Id:	14.0
Use Case Name:	Delete Comment
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author, viewer
Description:	This use case allows the user to delete their comments to the contents.
Precondition:	● The user must be logged into the system. ● The user must have previously added the comment.
Post-condition:	The user successfully deletes their comment on their selected content.
Normal Courses:	● User logs into the system. ● Users navigate to the content with their existing comment. ● User clicks on the “Delete” button next to their comment. ● System prompts the user for confirmation. ● User confirms the deletion. ● System removes the comment from the content.
Alternative Courses:	If the user cancels the delete action , the comments remain unchanged.
Exceptions:	If there is an issue saving the deleting comment, the system displays an error message.
Includes:	None
Priority:	High
Frequency of use:	24 hours a day
Business Rule:	● Users can only delete their own comments.

	• Admin can delete other users' comments.
Special Requirement:	The system should maintain a record of deleted comments for audit purposes.
Assumptions:	Users have the necessary permissions to delete their comments on the content.

Table 2.17: Use Case Template of Logout

Use Case Id:	15.0
Use Case Name:	Logout
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author, viewer
Description:	This use case allows the user to logout of the system.
Precondition:	The user must be currently logged into the system.
Post-condition:	The user is successfully logged out of the system.
Normal Courses:	• User clicks on the "Logout" button or navigates to the logout option. • System logs the user out and redirects to the login page. • Session variables and user data are cleared.
Alternative Courses:	If the user cancels the logout action, they remain logged in.
Exceptions:	None
Includes:	None
Priority:	Medium
Frequency of use:	Daily
Business Rule:	None
Special Requirement:	The system should display a confirmation message before logging the user out.

Assumptions:	Users have the necessary permissions to log out, and their session data is successfully cleared upon logout.

Table 2.18: Use Case Template of Register

Use Case Id:	16.0
Use Case Name:	Register
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Viewer
Description:	This use case allows viewer to register
Precondition:	The viewer must not be already registered in the system.
Post-condition:	The viewer is successfully registered in the system.
Normal Courses:	● Viewer accesses the registration page. ● Viewer clicks on the "Register" button. ● System presents the registration form to the viewer. ● Viewer fills in the required information (e.g., username, email,password). ● Viewer submits the registration form. ● System validates the information and creates a new viewer account. ● System logs the viewer into the system.
Alternative Courses:	If the viewer cancels the registration, they are returned to the homepage.
Exceptions:	If there is an issue during registration, the system displays an error message.
Includes:	None
Priority:	High
Frequency of use:	One-time (per viewer)
Business Rule:	Username and email address must be unique within the system.

Special Requirement:	The system should send a confirmation email to the viewer upon successful registration.
Assumptions:	● The viewer provides valid and unique information during registration. ● The system admin is responsible for handling account verification and activation.

Table 2.19: Use Case Template of Apply as Author

Use Case Id:	17.0
Use Case Name:	Apply as author
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Viewer
Description:	This use case allows the viewer to apply as author.
Precondition:	● The user must be a registered viewer in the system. ● The viewer is logged into the system.
Post-condition:	The system receives the application for authorship from the viewer.
Normal Courses:	● Viewer logs into the system. ● Viewer navigates to the “Apply as Author” section. ● Viewer clicks on the “Apply as Author” button. ● System prompts the viewer to provide necessary details for the author application. ● Viewer fills in the required information (e.g., bio, writing sample) and submits the application. ● System records the author's application for review.
Alternative Courses:	If the viewer cancels the application, they are returned to the homepage.
Exceptions:	If there is an issue submitting the application, the system displays an error.
Includes:	None
Priority:	High

Frequency of use:	Infrequent
Business Rule:	The system admin will review and approve author applications.
Special Requirement:	The system should send a confirmation email to the viewer upon successful submission of the author application.
Assumptions:	● The viewer has the intention and necessary qualifications to become an author. ● The system admin or designated authority is responsible for reviewing and approving author applications.

Table 2.20: Use Case Template of Add Category

Use Case Id:	18.0
Use Case Name:	Add Category
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author
Description:	This use case allows the admin, editor and author to add categories for content.
Precondition:	● The user must be logged into the system. ● The user must have the necessary permissions to add a category.
Post-condition:	The new category is successfully added to the system.
Normal Courses:	● User logs into the system. ● User navigates to the “Add Category” section. ● User clicks on the “Add Category” button. ● System requests the user to provide details for the new category (e.g., name,description). ● User fills in the required information and submits the form. ● System validates and adds the new category to the system.
Alternative Courses:	If the user cancels adding a new category, there will be no changes.
Exceptions:	If there is an issue during category addition (e.g.,duplicate category name), the system displays an error message.

Includes:	None
Priority:	Medium
Frequency of use:	Occasional
Business Rule:	Category names must be unique within the system.
Special Requirement:	The system should support the hierarchical structure of categories (parents-child relationships).
Assumptions:	● Users have the necessary permissions to add categories. ● The system supports the concept of categories for organizing content.

Table 2.21: Use Case Template of Update Category

Use Case Id:	19.0
Use Case Name:	Update Category
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author
Description:	This use case allows the admin, editor and author to update categories.
Precondition:	● The user must be logged into the system. ● The user must have the necessary permissions to update a category. ● There must be existing categories in the system.
Post-condition:	The selected category is successfully updated in the system.
Normal Courses:	● User logs into the system. ● User navigates to the “Update Category” section. ● User selects the category they want to update ● User clicks on the “Update Category” button for the selected category. ● System loads the existing information for the category into an editable form. ● User modifies the category details (e.g., name, description) and submit the form. ● System validates and updates the category information.

Alternative Courses:	If the user cancels the update action, the system retains the original category details.
Exceptions:	If there is an issue during category update (e.g., invalid data), the system displays an error message.
Includes:	None
Priority:	Medium
Frequency of use:	Occasional
Business Rule:	Category names must be unique within the system.
Special Requirement:	The system should maintain consistency in the hierarchical structure of categories.
Assumptions:	● Users have the necessary permissions to update categories. ● There are existing categories in the system to be updated.

Table 2.22: Use Case Template of Delete Category

Use Case Id:	20.0
Use Case Name:	Delete Category
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author
Description:	This use case allows the admin, editor and author to delete categories.
Precondition:	● The user must be logged into the system. ● The user must have the necessary permissions to delete a category. ● There must be existing categories in the system.
Post-condition:	The selected category is successfully deleted in the system.
Normal Courses:	● User logs into the system. ● User navigates to the “Delete Category” section. ● User selects the category they want to delete. ● User clicks on the “Delete Category” button for the selected category. ● System prompts the user for confirmation to delete the category. ● User confirms the deletion.

	• System removes the selected category and associated content.
Alternative Courses:	If the user cancels the delete action, the category remains unchanged.
Exceptions:	If there is an issue during category deletion (e.g., category has associated content), the system displays an error message.
Includes:	None
Priority:	High
Frequency of use:	Occasional
Business Rule:	Categories with associated content cannot be deleted directly;users must first move or delete the associated content.
Special Requirement:	The system should provide an option to reassign or move content before deleting a category.
Assumptions:	• Users have the necessary permissions to delete categories. • There are existing categories in the system to be deleted.

Table 2.23: Use Case Template of View Category

Use Case Id:	21.0
Use Case Name:	View Category
Created By:	Usha Gurung Last updated by:
Date Created:	January 4,2024 Date last updated:
Actor(s):	Admin, Editor, Author, Viewer
Description:	This use case allows users to view categories.
Precondition:	The user must be logged into the system.
Post-condition:	The user successfully views the list of categories.
Normal Courses:	• User logs into the system. • User navigates to the “View Categories” section. • System retrieves and displays a list of existing categories.
Alternative Courses:	None

Exceptions:	If there is an issue retrieving categories, the system displays an error message.
Includes:	None
Priority:	Medium
Frequency of use:	Daily
Business Rule:	Categories are organized hierarchically, displaying parent and child relationships.
Special Requirement:	The system should support filtering or sorting options for an enhancing viewing experience.
Assumptions:	● Users have the necessary permissions to view categories. ● Categories are organized in a hierarchical structure.

2.6.2 ER Diagram

An ER (Entity-Relationship) Diagram is a graphical representation of entities, their attributes, and the relationships between them, used in database design. It helps visualize the data structure, making it easier to understand how data is organized and connected. Entities are represented as rectangles, attributes as ovals, and relationships as diamonds, with lines connecting them to indicate associations. ER diagrams are essential for creating efficient, logical, and well-structured databases by identifying primary keys, foreign keys, and cardinality constraints, ensuring a clear blueprint for implementation.

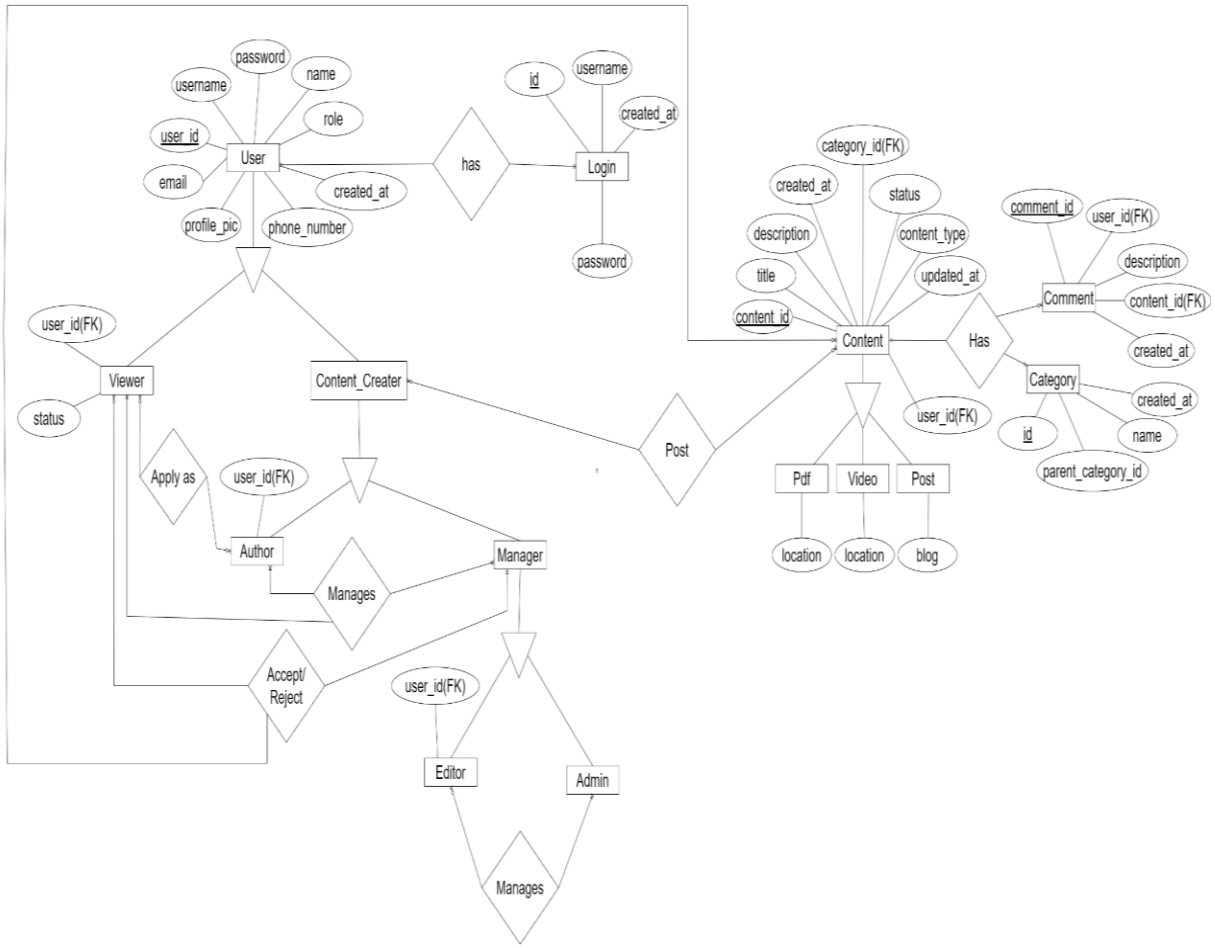

Figure 2.8: Entity Relationship Diagram

2.6.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a visual representation of how data moves through a system, showing processes, data stores, and external entities to illustrate the system's functional flow.

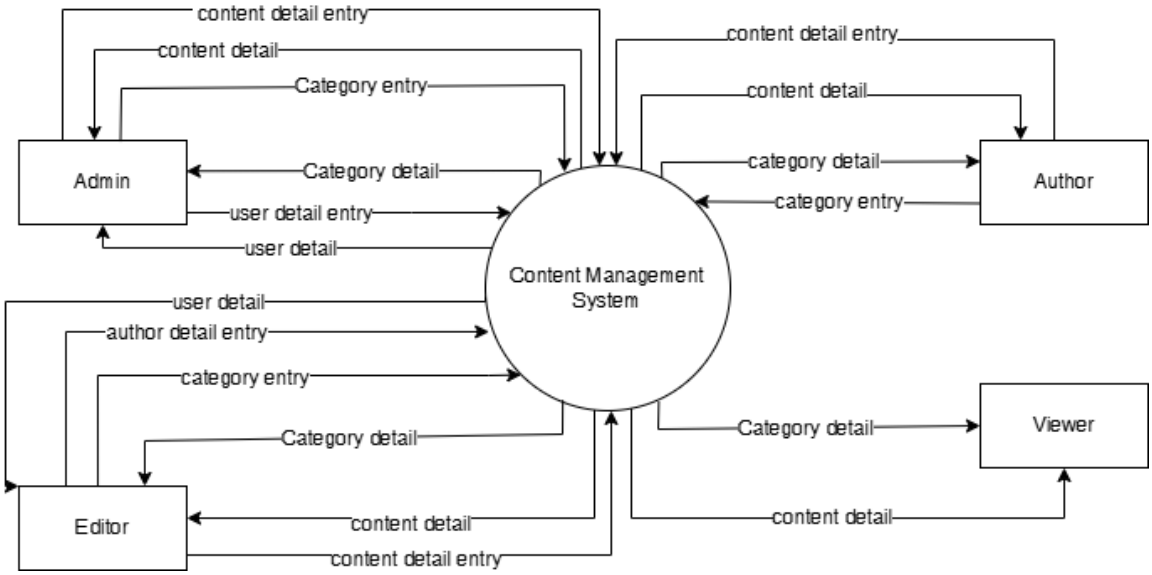

Figure 2.9: DFD Level 0

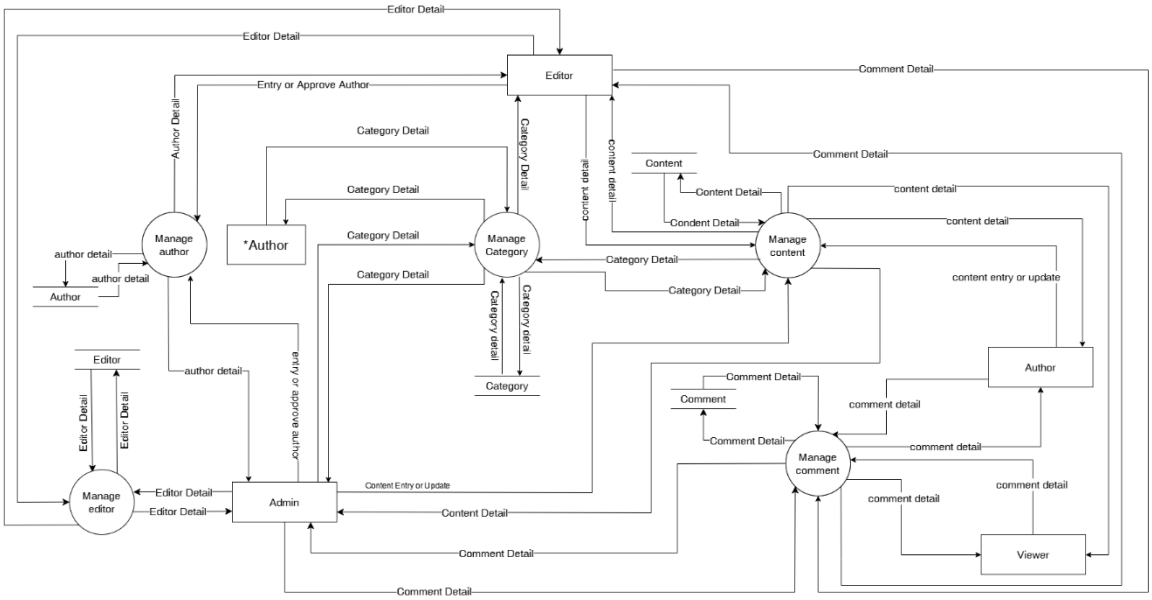

Figure 2.10: DFD Level 1

2.6.6 Activity Diagram

An activity diagram is a UML tool that visually represents the workflow or processes in a system, showing the sequence of activities, decision points, and parallel actions.

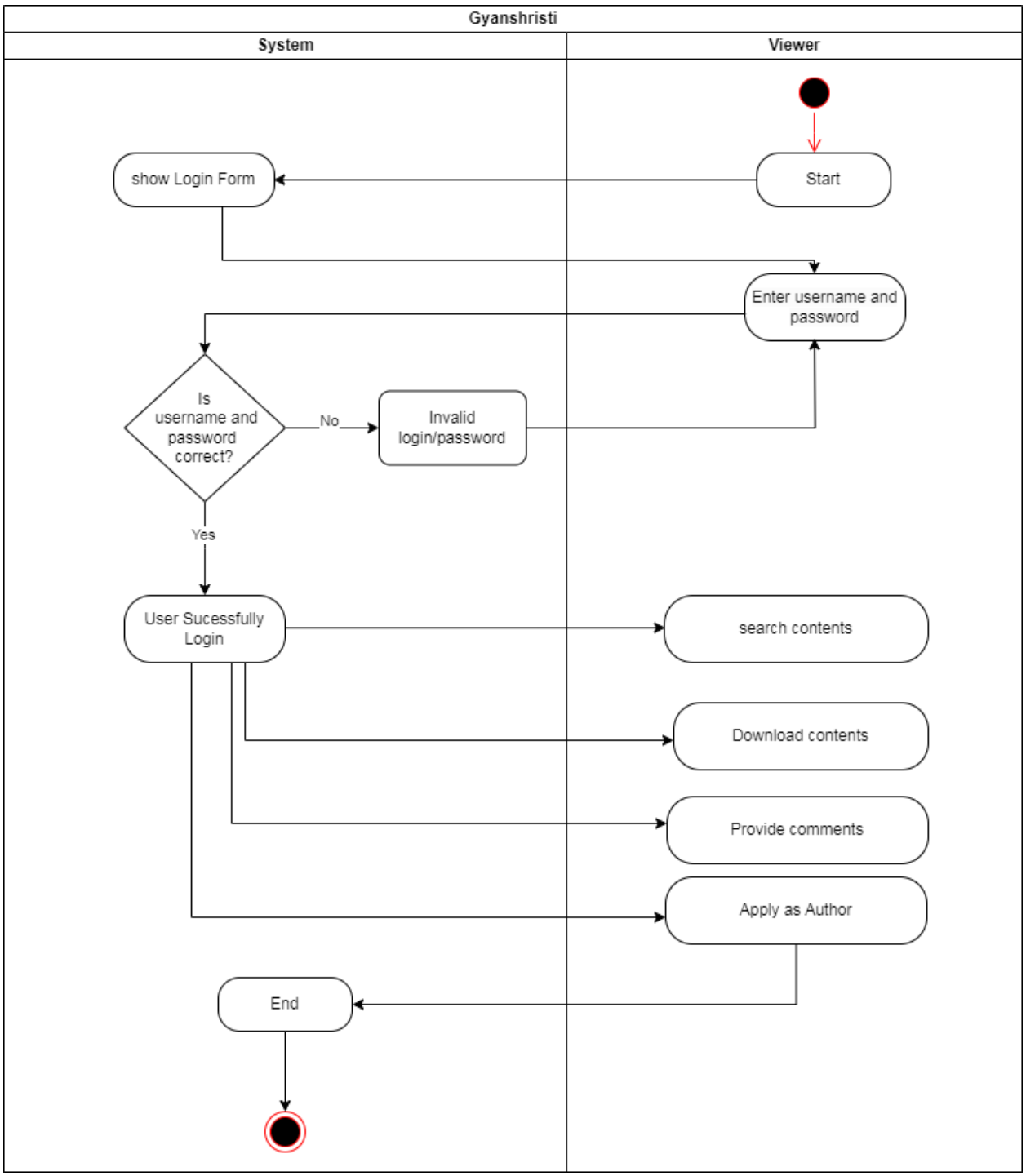

Figure 2.11: Activity Diagram of Viewer

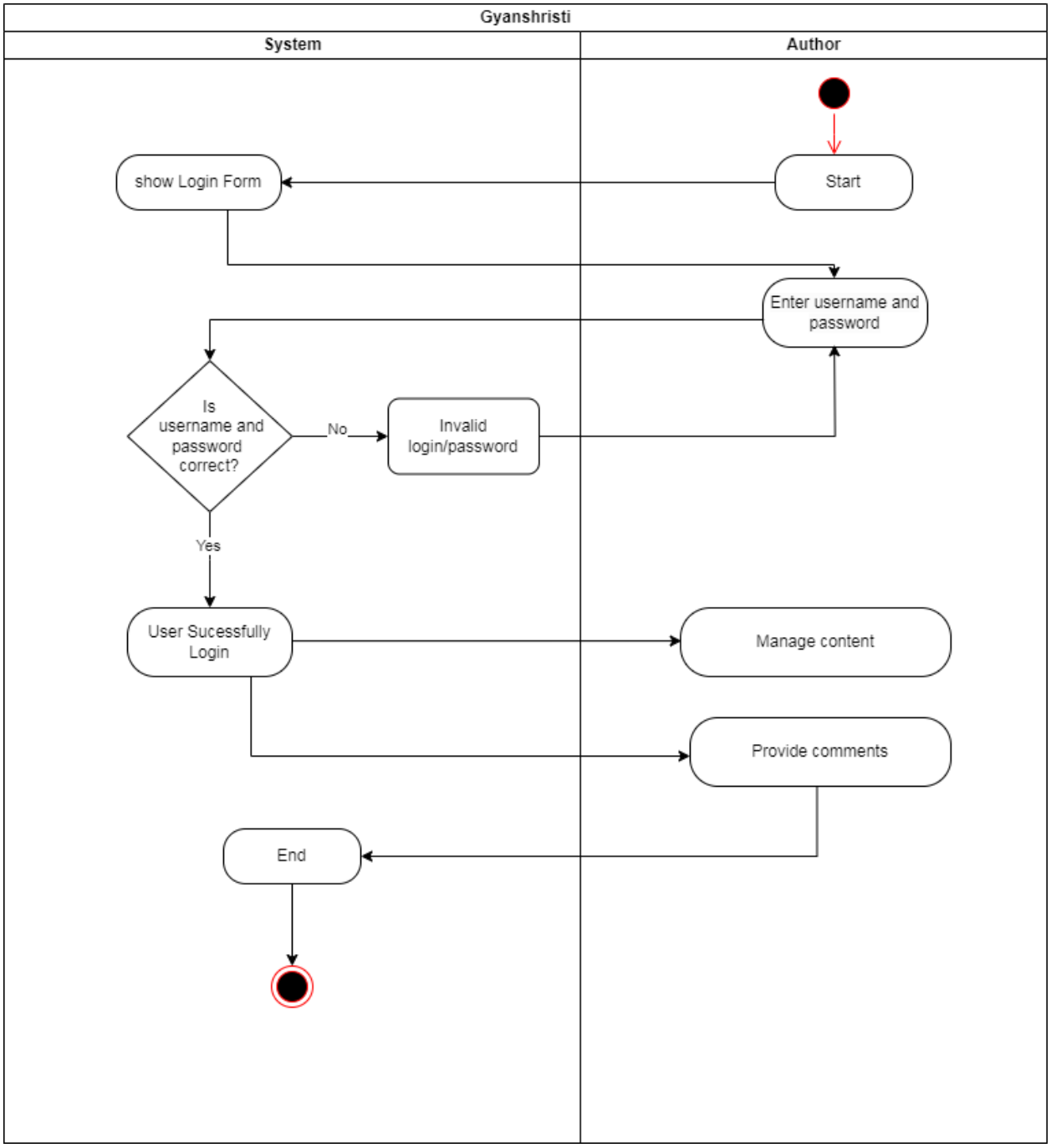

Figure 2.12: Activity Diagram of Author

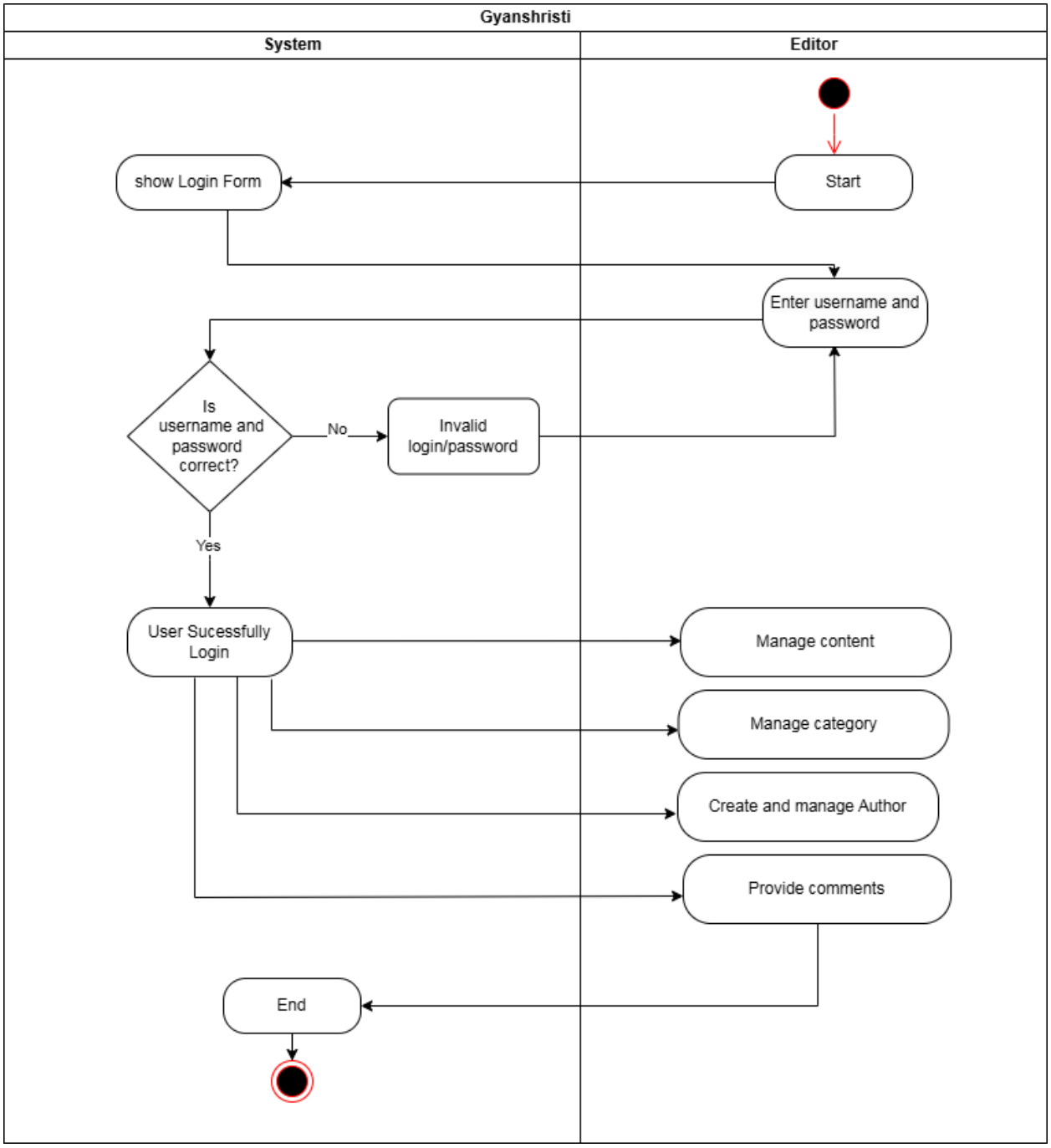

Figure 2.13: Activity Diagram of Editor

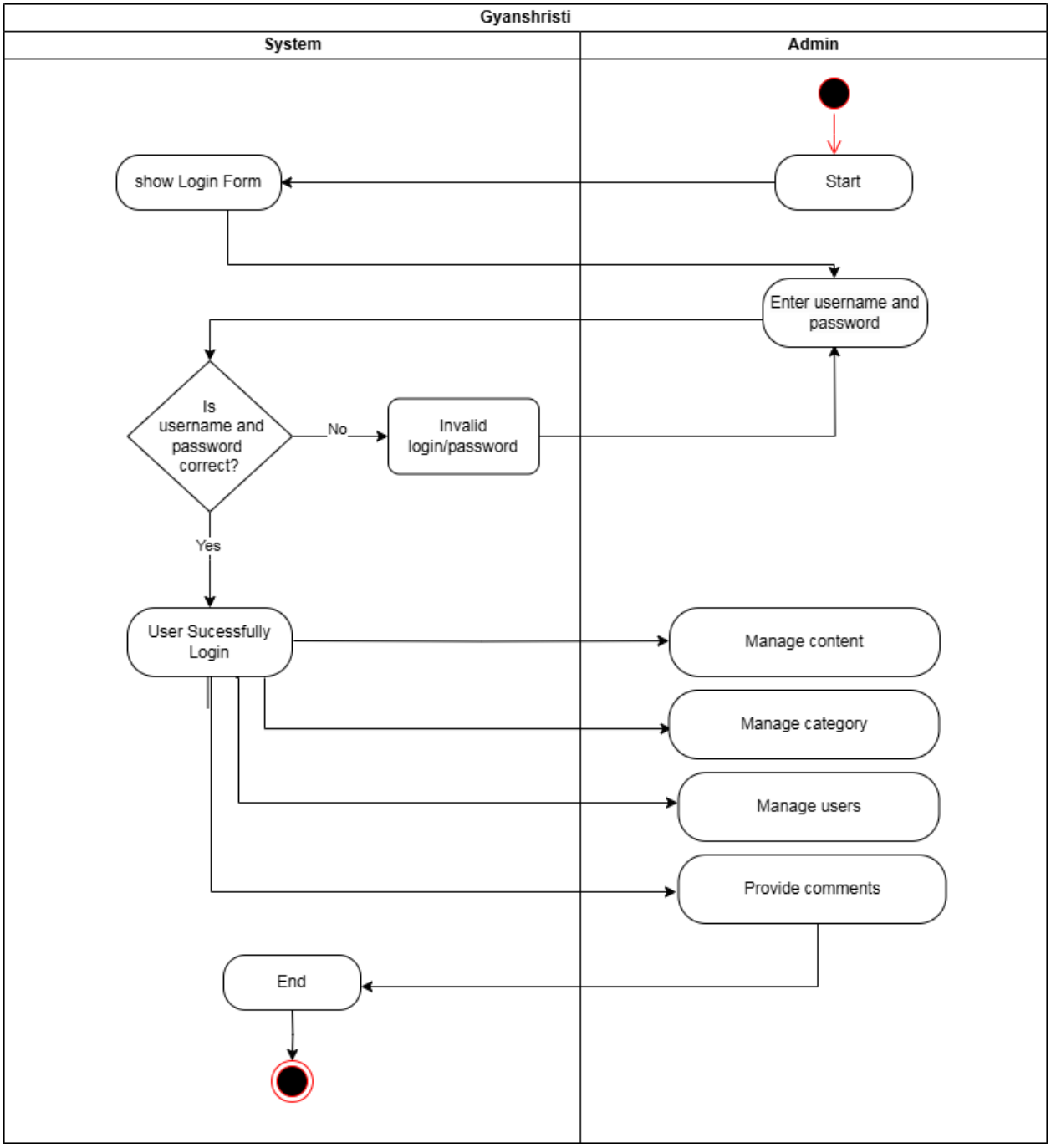

Figure 2.14: Activity Diagram of Admin