CONTENT MANAGEMENT SYSTEM

Software Construction And Development

BSSE 5TH SEMESTER

REPORT (Assignment 01)

Course Instructor: Sir FAROOQ IQBAL

Group Members:

1. **Syeda Anshrah Gillani (1337-2021) - (Group Leader)**
2. **Umema Mujeeb (2396-2021)**
3. **Maheen Ali (1589-2021)**
4. **Areej Asif (2276-2021)**

18/NOV/2023

i. Vision and Scope for Content Management System (CMS):

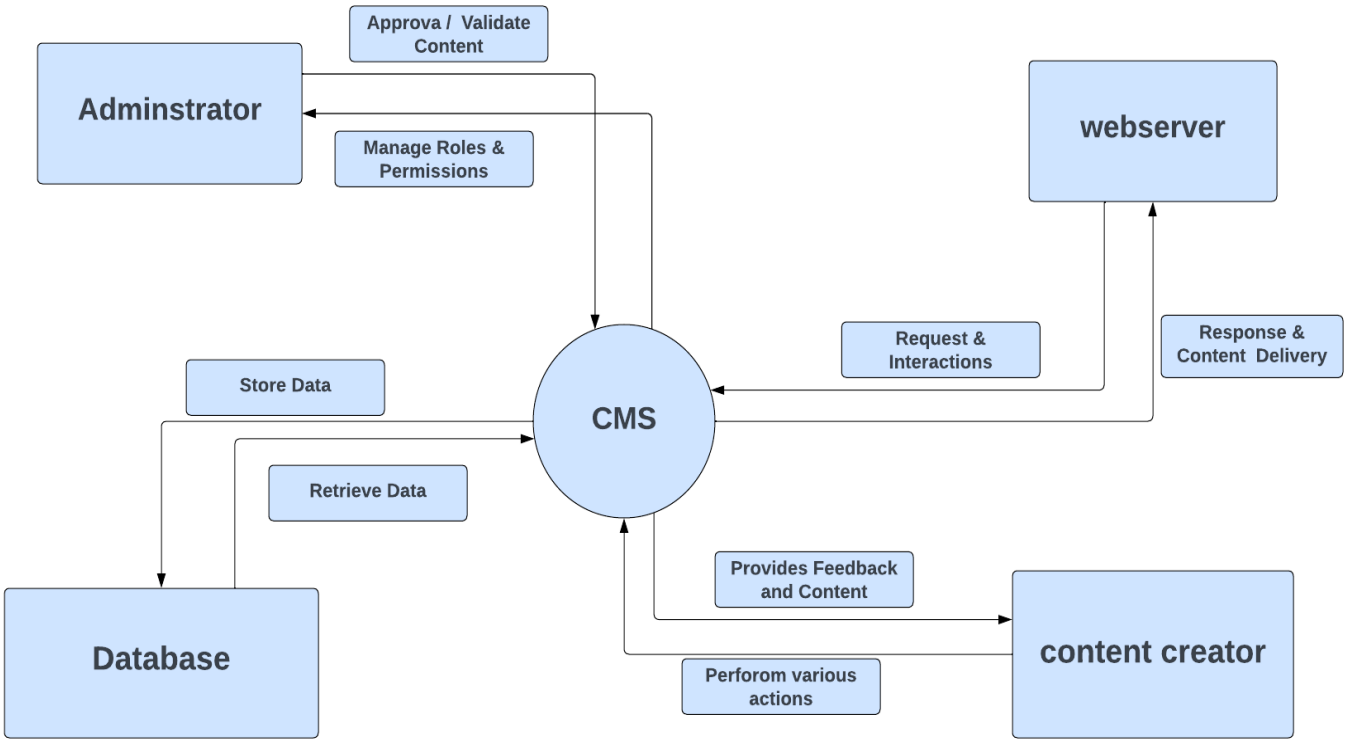
➢ Vision:

To develop a robust Content Management System that empowers users to efficiently create, manage, and publish digital content for websites. The system aims to streamline content workflows, enhance collaboration, and provide a user-friendly interface for content creators and administrators.

➢ Scope:

The CMS will include features for content creation, editing, version control, user management, and publishing. It will support various content types such as articles, images, and videos. The system will prioritize scalability, security, and ease of use, catering to the needs of both small businesses and large enterprises.

Context Diagram:



Context Diagram Components:

1. CMS System:

Description: Represents the Content Management System itself. This is the central component of the system responsible for managing content creation, editing, and publication.

Functionality: Coordinates the overall functioning of the CMS, including user interactions, content workflows, and system configuration.

2. Users:

Description: Represents individuals who interact with the CMS. Users could include content creators, editors, and other stakeholders.

Functionality: Engage with the CMS to create, edit, and manage content. Users may have different roles and permissions.

3. Content Creators:

Description: Specific category of users responsible for generating new content within the CMS.

Functionality: Create and submit content items for publication, initiate workflows, and collaborate with other users in the content creation process.

4. Administrators:

Description: Users with elevated privileges for system configuration and management.

Functionality: Configure system settings, manage user roles and permissions, oversee content workflows, and ensure the overall security and functionality of the CMS.

5. External Systems (e.g., Database, Web Server):

Description: Represents systems external to the CMS but essential for its operation.

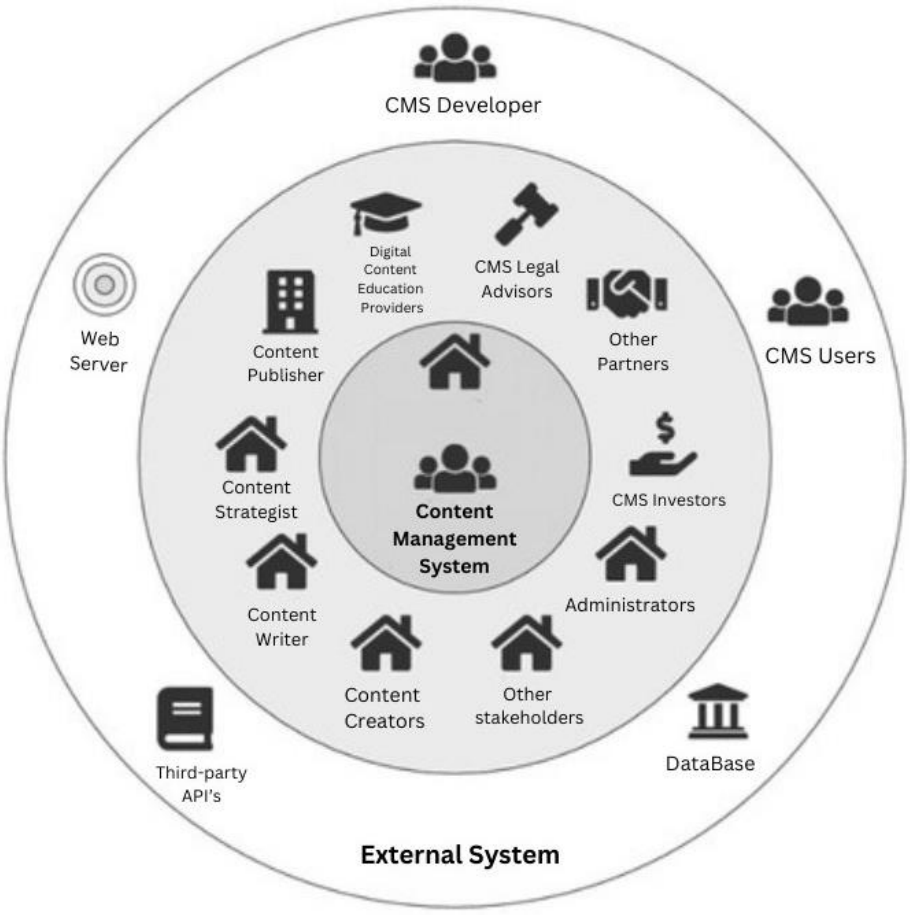
Functionality:

Database: Stores and retrieves content data, user information, and system configurations.

Web Server: Serves the CMS to users, handling requests and responses for web-based interactions.

Eco System Map:

- Maps out the ecosystem in which the CMS operates.
- Identifies external systems, services, and their sinteractions with the CMS.



ii. Domain Analysis:

➢ Problem Definition:

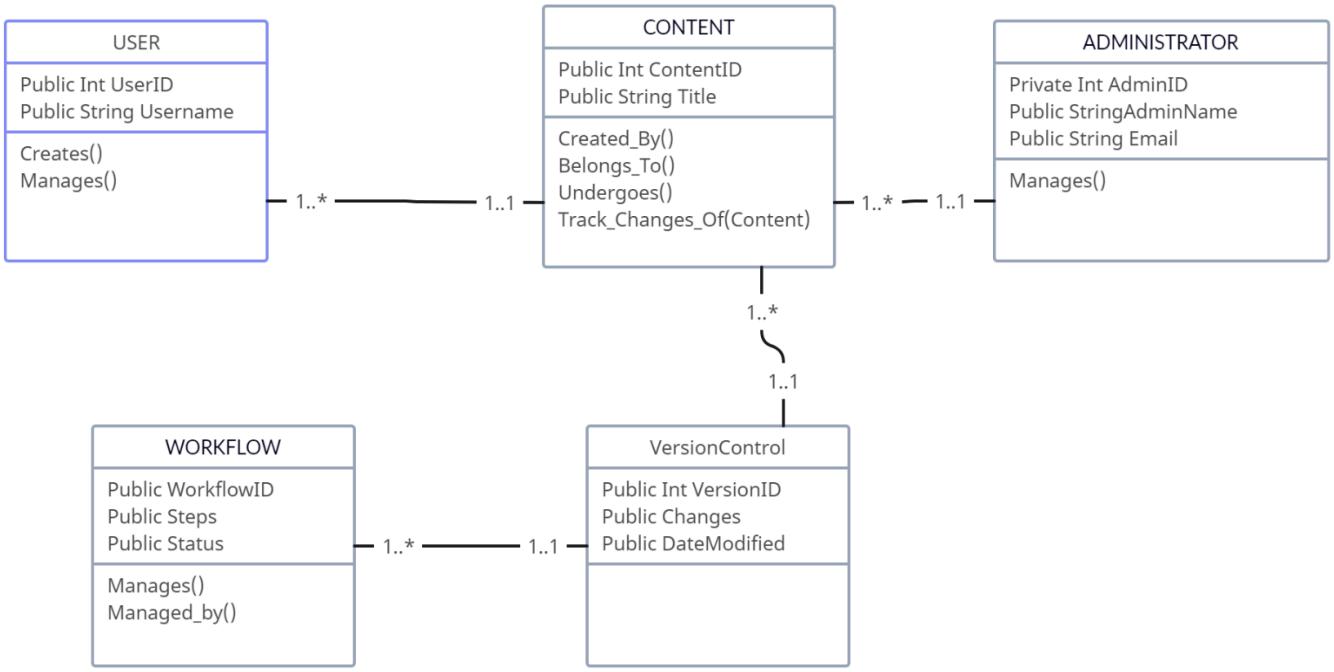
The problem is to create a CMS that addresses the challenges of content creation, management, and publication, ensuring efficiency, collaboration, and security.

➢ Dictionary of Terms:

1. **Content:** Digital information such as text, images, and videos.
2. **User:** Individuals interacting with the CMS.
3. **Administrator:** User with elevated privileges for system configuration.
4. **Version Control:** System for tracking and managing content changes.
5. **Workflow:** Defined sequence of steps for content creation, review, and publication.

➢ Problem Domain Class Diagram:

Identifies key entities in the problem domain and their relationships, e.g., User, Content, Workflow.



ii. Subsystems or Packages:

1. User Management Subsystem:

Function: Manages user authentication, authorization, and profiles.

Significance: Ensures secure access to the CMS.

2. Content Creation and Editing Subsystem:

Function: Facilitates the creation, editing, and versioning of content.

Significance: Enhances the efficiency of content creators.

3. Workflow and Approval Subsystem:

Function: Implements workflows for content review and approval.

Significance: Ensures a systematic process before content publication.

4. Publishing Subsystem:

Function: Manages the publication and visibility of content.

Significance: Controls the timing and accessibility of published content.

➢ Subsystem Communication:

"Emphasizing minimal coupling and maximum cohesion."

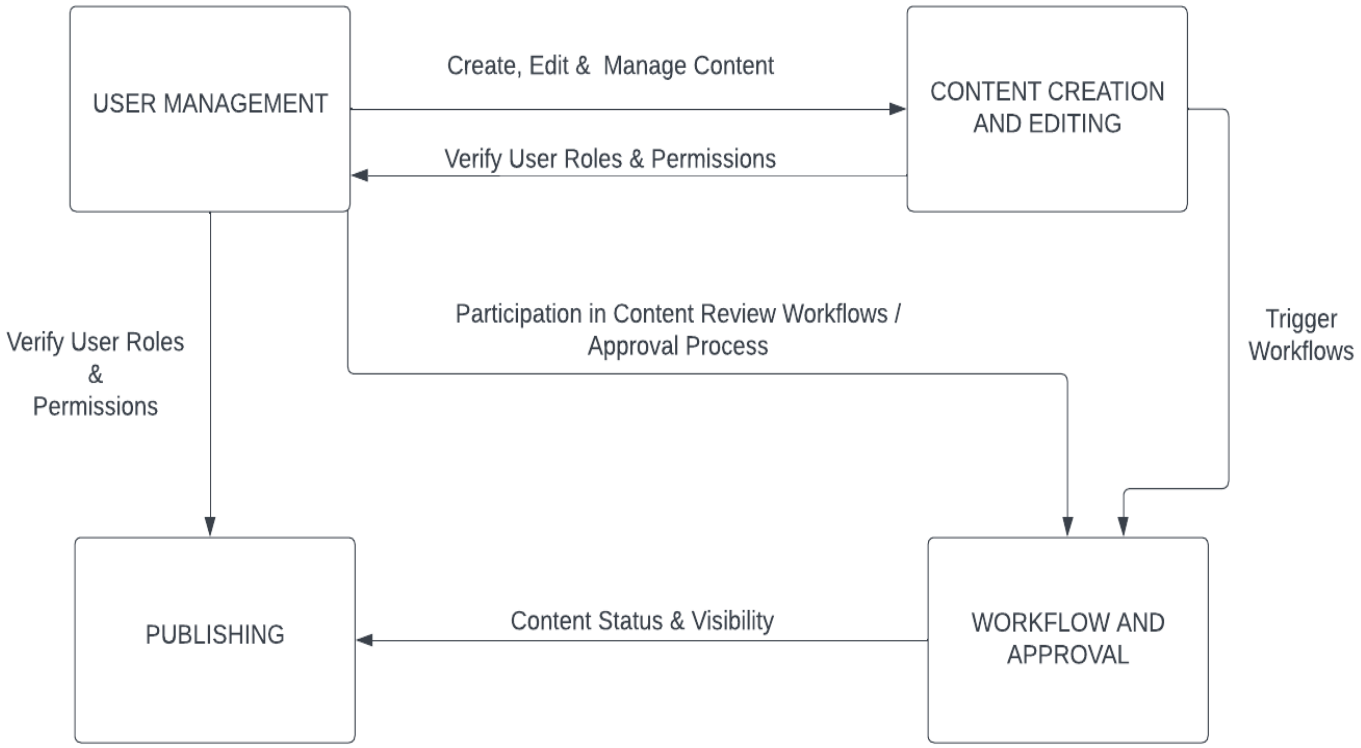
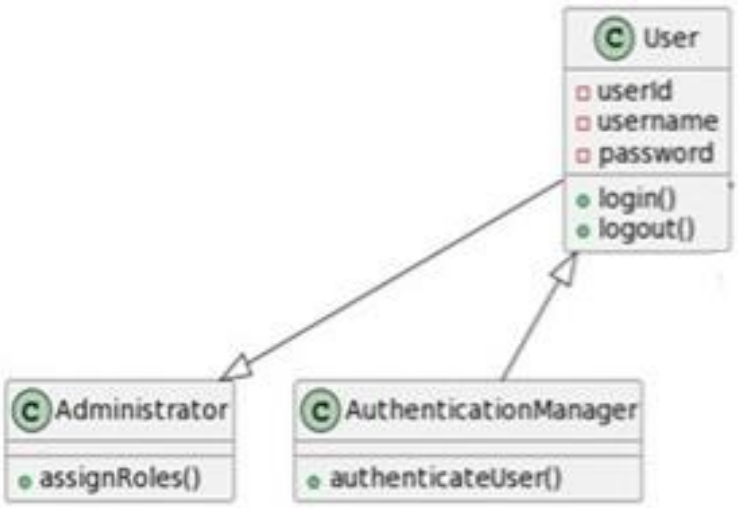


Fig . BLOCK DAIGRAM

iii. OO Analysis - Class Diagrams for Each Subsystem:

1. User Management Subsystem Class Diagram:



(1) User Class:

Attributes:

- Private: UserID, UserName, Email

Methods:

- Public: createUser()
- Public: editUserDetails()
- Private: authenticateUser()

(2) Administrator Class:

Attributes:

- Private: AdminID, AdminName, Email

Methods:

- Public: configureSystem()
- Public: manageUserRoles()

- Private: authenticateAdministrator()

(3) AuthenticationManager Class:

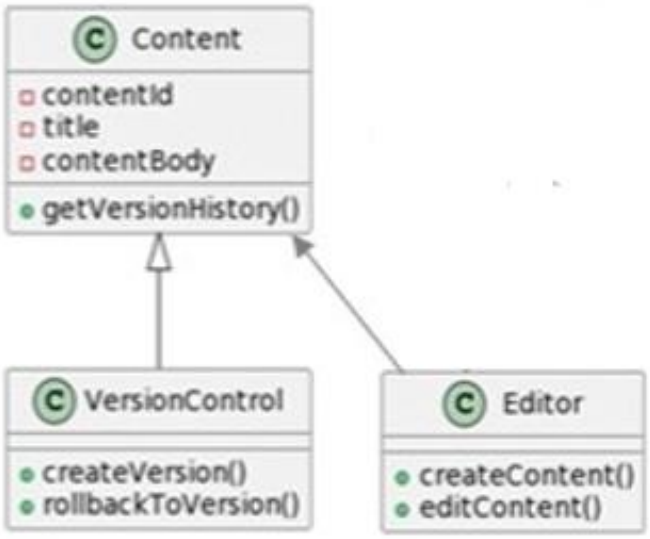
Attributes:

- None

Methods:

- Public: authenticate(User user)
- Private: validateCredentials(String username, String password)

2. Content Creation and Editing Subsystem Class Diagram:



(1) Content Class:

Attributes:

- Private: ContentID, Title, DateCreated

Methods:

- Public: createContent(User creator, String title)
- Public: editContent(User editor, String newTitle)
- Private: applyVersionControl()

(2) Editor Class:

Attributes:

- Private: EditorID, EditorName

Methods:

- Public: editContent(Content content, String newTitle)
- Public: reviewContent(Content content)
- Private: notifyAuthor()

(3) VersionControl Class:

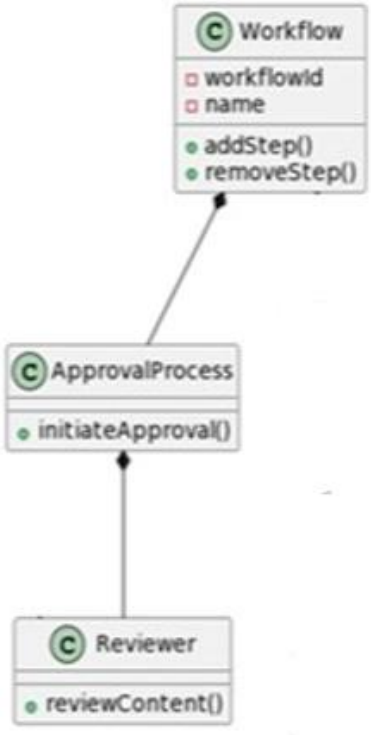
Attributes:

- Private: VersionID, Changes, DateModified

Methods:

- Public: trackChanges(Content content)
- Private: saveVersion()

3. Workflow and Approval Subsystem Class Diagram:



(1) Workflow Class:

Attributes:

- Private: WorkflowID, Steps, Status

Methods:

- Public: initiateWorkflow(Content content)
- Public: progressWorkflow()
- Private: notifyReviewers()

(2) ApprovalProcess Class:

Attributes:

- Private: ApprovalProcessID, Approvers, Status

Methods:

- Public: approveContent(Content content)
- Public: rejectContent(Content content)
- Private: notifyAuthor()

(3) Reviewer Class:

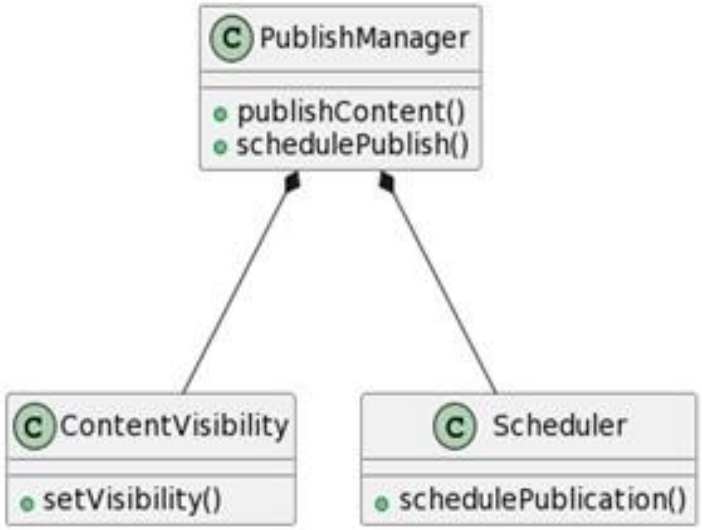
Attributes:

- Private: ReviewerID, ReviewerName

Methods:

- Public: reviewContent(Content content)
- Private: notifyOutcome()

4. Publishing Subsystem Class Diagram:



(1) PublishManager Class:

Attributes:

Private: PublishManagerID, PublishManagerName

Methods:

Public: publishContent(Content content)

Public: schedulePublication(Content content, Date publishDate)

Private: notifySubscribers()

(2) ContentVisibility Class:

Attributes:

Private: ContentVisibilityID, VisibilitySettings

Methods:

Public: setVisibility(Content content, User user)

Private: applyVisibilitySettings()

(3) Scheduler Class:

Attributes:

Private: SchedulerID, ScheduleSettings

Methods:

Public: scheduleTask(Date date, Task task)

Private: executeTask(Task task)

Conclusion:

These class diagrams detail the entities and relationships within each subsystem, emphasizing the principles of object-oriented analysis and design. **The goal is to create modular, cohesive, and maintainable subsystems that collectively form the Content Management System**

THE END