

Document Control System

By

Mukeshkumar Khaniya

Gaurav Keshwarwani

Dhruv Jain

Maheswar Barrenkala

Yutthapong Kawunruan

Pratibha Awasthi

International Technological University

SWE 685 Cloud and Software Architecture

Professor: Qamar Asghar

Spring 2023

Abstract

The architecture document for the document management system describes the high-level design and technical implementation details of the system. The system is a web-based application that provides organizations with a centralized repository for storing and managing documents. The architecture document provides an overview of the system's components, including the user interface, application logic, database, and external integrations.

The system is designed using a modular architecture that allows for easy scalability and maintenance. The system uses a cloud-based infrastructure for hosting, ensuring high availability and scalability. The document management system's architecture is designed with performance in mind, utilizing caching mechanisms and efficient database queries to ensure optimal performance even with large volumes of data.

Overall, the architecture document for the document management system provides a comprehensive overview of the system's design and implementation details, ensuring that the system meets industry standards for reliability, security, and performance.

Table of Content

Abstract	2
Project Plan and Schedule	4
Planning and Analysis (2 weeks)	4
Design (4 weeks)	4
Development (20 weeks)	4
Testing and Quality Assurance (4 weeks)	5
Deployment and Maintenance (2 weeks)	5
Project Deliverables	6
Requirement Specification	8
Functional Requirement	8
Non-Functional Requirement	9
Software Architecture	10
Use Case Diagram	11
Deriving Use Case From Requirements	12
Use Case Details	13
UML Activity Diagram	20
UML Sequence Diagram	21
UML Class Diagram	22
UML Component Diagram	23
References	24

Project Plan and Schedule

Develop a document control system that allows users to create, edit, share, and manage documents in real time. The system will have access control, version control, and backup features to ensure data security and continuity.

Planning and Analysis (2 weeks)

- Define project scope and objectives
- Gather requirements from stakeholders
- Analyze project feasibility and constraints
- Identify risks and mitigation strategies
- Create a project plan and schedule

Design (4 weeks)

- Design system architecture and database schema
- Create wireframes and UI mockups
- Identify necessary third-party tools and APIs
- Create a detailed design document

Development (20 weeks)

- Set up development environment and tools
- Implement user authentication and access control features
- Implement document creation, editing, and sharing features
- Implement version control and backup features
- Integrate third-party tools and APIs
- Perform unit testing and integration testing

- Fix bugs and issues

Testing and Quality Assurance (4 weeks)

- Develop test cases and test plans
- Perform functional, performance, and security testing
- Identify and fix bugs and issues
- Perform user acceptance testing
- Document test results and quality metrics

Deployment and Maintenance (2 weeks)

- Deploy the system on the production server
- Conduct user training and support
- Monitor system performance and security
- Perform bug fixing and maintenance as necessary

Project Deliverables

Project plan and schedule: outline the project scope, objectives, milestones, and timelines. It includes a detailed description of each project phase, including planning, design, development, testing, deployment, and maintenance. The project plan also includes a schedule that outlines the start and end dates of each phase, as well as the duration of each task.

Requirements specification document: This deliverable outlines the requirements of the software system. It includes a detailed description of the functional and non-functional requirements of the system, including access control, document creation, editing, sharing, and management features. The document also includes any constraints, assumptions, or dependencies that may impact the development of the system.

Design document: This deliverable outlines the system architecture, including the database schema, API design, and user interface. The design document includes detailed diagrams and descriptions of the system components and how they interact with each other. The document also includes any third-party tools or APIs that will be integrated into the system.

Source code and database schema: This deliverable includes the actual code and database schema of the software system. The source code includes the front-end and back-end code of the system, including any APIs or third-party tools used. The database schema includes the structure and relationships of the system's database.

User manual and technical documentation: This deliverable includes documentation that explains how to use the software system. The user manual includes instructions on how to perform common tasks in the system, such as creating, editing, and sharing

documents. The technical documentation includes detailed descriptions of the system architecture, database schema, and any third-party tools or APIs used.

Test cases and test results: This deliverable includes the test cases used to test the software system, as well as the results of the testing. The test cases include a description of the test scenario, the steps taken, and the expected and actual results. The test results include a summary of the testing performed, any bugs or issues found, and any remediation that has been taken.

System deployment and maintenance documentation: This deliverable includes documentation that outlines how to deploy the software system to a production environment. It includes instructions on how to install and configure the system components, as well as any third-party tools or APIs used. The maintenance documentation includes instructions on how to monitor and maintain the system, as well as how to perform any necessary updates or upgrades.

Requirement Specification

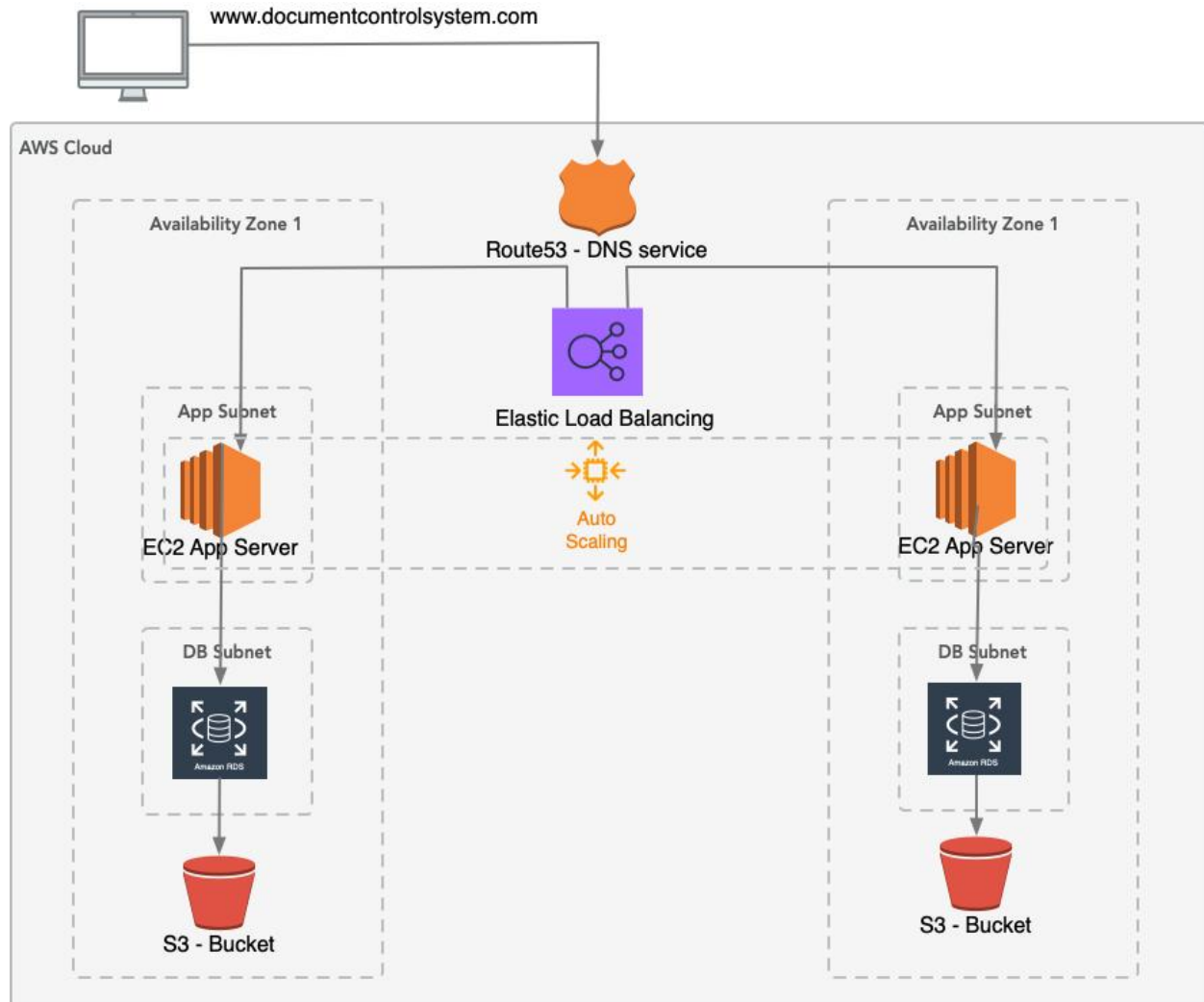
Functional Requirement

#	Functional Requirement	Details
REQ1	Users & Access	Supported user types Admin, Users(Non Admin)
REQ2	Access Control	Access Controls Admin Access to create and delete users.
REQ3	Groups	Provision of Various Groups
REQ4	Roles (For Role based Access)	Create Roles e.g. Roles for Creating document, Read only roles, Roles to update the document
REQ5	Category of Document	Marketing Document,, Sales Document, QE Document, Operational Document.
REQ6	Type of Format Supported	Text based, Spread Sheets
REQ7	Max permissible size	10 kb
REQ8	Versioning	Support of versioning of document to maintain the history of documents
REQ9	Searchability	User should be able to search any document consisting of the word or particular sentence user is looking for
REQ10	Workflow support	System should support the lifecycle of the document like Review, Feedback mechanism, Signoff, expiry, etc.

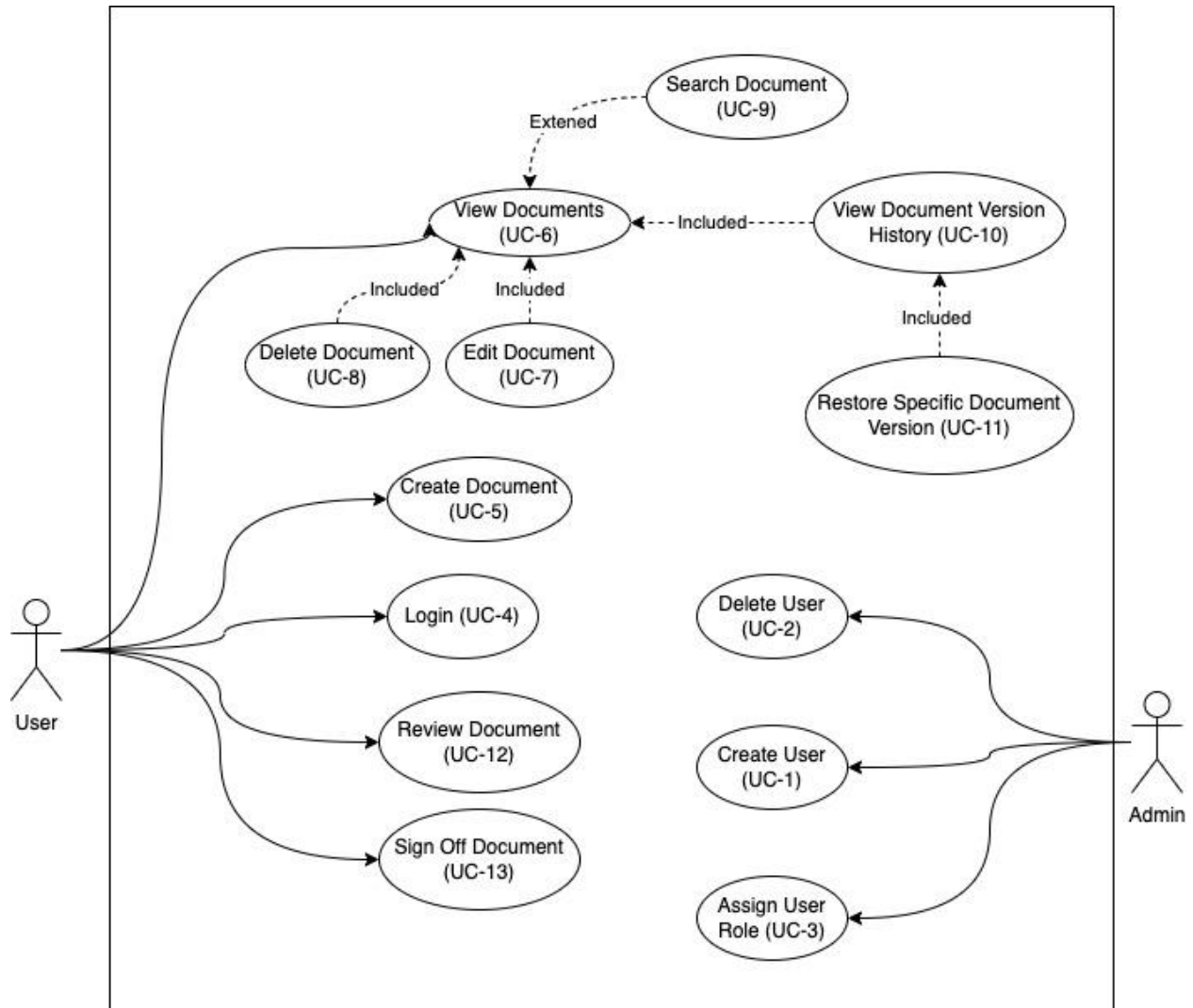
Non-Functional Requirement

#	Non Functional Requirement	Details
1	Platform Support	Cloud based Support. Application will be deployed on cloud
2	Storage	Document can be stored in S3
3	Document Size	Max permissible size 10 kb

Software Architecture



Use Case Diagram



Deriving Use Case From Requirements

Actor	Actor's Goal	Use Case Name
Admin	Create a new user account, so that new users can be added to the system.	Create User (UC-1)
Admin	Delete a user account, so that users who no longer need access to the system can be removed.	Delete User (UC-2)
Admin	Assign roles to users, so that users have the appropriate level of access to the system.	Assign User Role (UC-3)
User	Log in to the system, so that I can access my documents and perform actions within the system.	Login (UC-4)
User	Create a new document, so that I can store information in the system.	Create Document (UC-5)
User	View a list of documents, so that I can see all the documents I have access to.	View Documents (UC-6)
User	Edit a document, so that I can modify its content.	Edit Document (UC-7)
User	Delete a document, so that I can remove it from the system.	Delete Document (UC-8)
User	Search for a specific document by keyword or sentence, so that I can find the document I need quickly and easily.	Search Document (UC-9)
User	View the version history of a document, so that I can see how the document has changed over time.	View Document Version History (UC-10)
User	Restore a previous version of a document, so that I can undo any unwanted changes.	Restore Specific Document Version (UC-11)
User	Send a document for review, so that other users can provide feedback and suggest changes.	Review Document (UC-12)
User	Sign off on a document, so that I can indicate that the document is complete and ready for use.	Sign Off Document (UC-13)

Use Case Details

Use Case UC-1	Create User
Related Requirements	REQ1, REQ2
Initiating Actor	Admin
Actor's Goal	To create a new user account and grant access to the system.
Preconditions	Admin is logged into the system. Admin has the necessary privileges to create a new user account.
Postconditions	A new user account is created and added to the system. A new user is notified of their login credentials.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. Admin selects the option to create a new user account. 2. The system prompts the admin to enter the necessary details for the new user, such as username, email address, and password. 3. The system notifies the new user of their login credentials via email. 	

Use Case UC-2	Delete User
Related Requirements	REQ1, REQ2
Initiating Actor	Admin
Actor's Goal	To remove a user's access to the system.
Preconditions	Admin is logged into the system. Admin has the necessary privileges to delete a user account. User to be deleted has been identified.
Postconditions	The user account is removed from the system.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. Admin selects the option to delete a user account. 2. The system presents a list of users for the admin to choose from. 3. Admin selects the user to be deleted. 4. The system prompts the admin to confirm the deletion. 5. Admin confirms the deletion. 6. The system removes the user account from the system. 	

Use Case UC-3	Assign User Role
Related Requirements	REQ1, REQ3
Initiating Actor	Admin
Actor's Goal	To assign roles to users to manage access to the system.
Preconditions	Admin is logged into the system. Admin has the necessary privileges to assign roles. The roles to be assigned have been defined.
Postconditions	The roles are assigned to the selected user.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. Admin selects the option to assign roles to a user. 2. The system presents a list of users for the admin to choose from. 3. Admin selects the user to assign roles to. 4. The system presents a list of available roles for the admin to choose from. 5. Admin selects the roles to be assigned to the user. 6. The system validates the input and assigns the selected roles to the user. 	

Use Case UC-4	Log In
Related Requirements	REQ1, REQ2
Initiating Actor	User
Actor's Goal	To log in to the system and gain access to their documents and other system features.
Preconditions	User has a valid user account in the system. User knows their login credentials.
Postconditions	User is logged in to the system and has access to their documents and other system features.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User navigates to the login page. 2. The system presents a form for the user to enter their login credentials. 3. User enters their username and password and submits the form. 4. The system validates the user's credentials and logs them in. 5. The system redirects the user to their document list. 	

Use Case UC-5	Create Document
Related Requirements	REQ5, REQ6
Initiating Actor	User
Actor's Goal	To create a new document and store information in the system.
Preconditions	User is logged in to the system. User has appropriate access permissions to create a new document.
Postconditions	A new document is created and stored in the system.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User selects the option to create a new document. 2. The system presents a form for the user to enter details about the new document, such as title, category, and file format. 3. User enters the necessary details and submits the form. 4. The system validates the input and creates a new document. 5. The system displays the newly created document to the user. 	

Use Case UC-6	View Documents
Related Requirements	REQ5, REQ9
Initiating Actor	User
Actor's Goal	To view a list of documents they have access to.
Preconditions	User is logged in to the system. User has appropriate access permissions to view documents.
Postconditions	User sees a list of documents they have access to.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User selects the option to view their list of documents. 2. The system presents a list of documents the user has access to, sorted by category or other criteria. 3. User has the option to search for a specific document by entering keywords or phrases. 4. The system presents a filtered list of documents based on the user's search criteria. 	

Use Case UC-7	Edit Document
Related Requirements	REQ4, REQ8
Initiating Actor	User
Actor's Goal	To modify the content of a document.
Preconditions	User is logged in to the system. User has necessary permissions to edit the document.
Postconditions	Document content is updated. Document version is incremented.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User selects the document they want to edit from the list of documents. 2. User modifies the content of the document. 3. User saves the changes. 4. System increments the document version and saves the updated document. 5. System displays a confirmation message indicating that the document has been successfully updated. 	

Use Case UC-8	Delete Document
Related Requirements	REQ4
Initiating Actor	User
Actor's Goal	To delete a document from the system.
Preconditions	User is logged in to the system. User has necessary permissions to delete the document
Postconditions	Document is removed from the system.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User selects the document they want to delete from the list of documents. 2. System prompts the user to confirm the deletion of the document. 3. User confirms the deletion. 4. System removes the document from the system. 5. System displays a confirmation message indicating that the document has been successfully deleted. 	

Use Case UC-9	Search Document
Related Requirements	REQ9
Initiating Actor	User
Actor's Goal	Users can search for documents by keyword or sentence.
Preconditions	User is logged in to the system.
Postconditions	System displays a list of documents that match the search criteria.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User enters a keyword or sentence in the search bar. 2. System searches for documents that match the search criteria. 3. System displays a list of documents that match the search criteria. 	

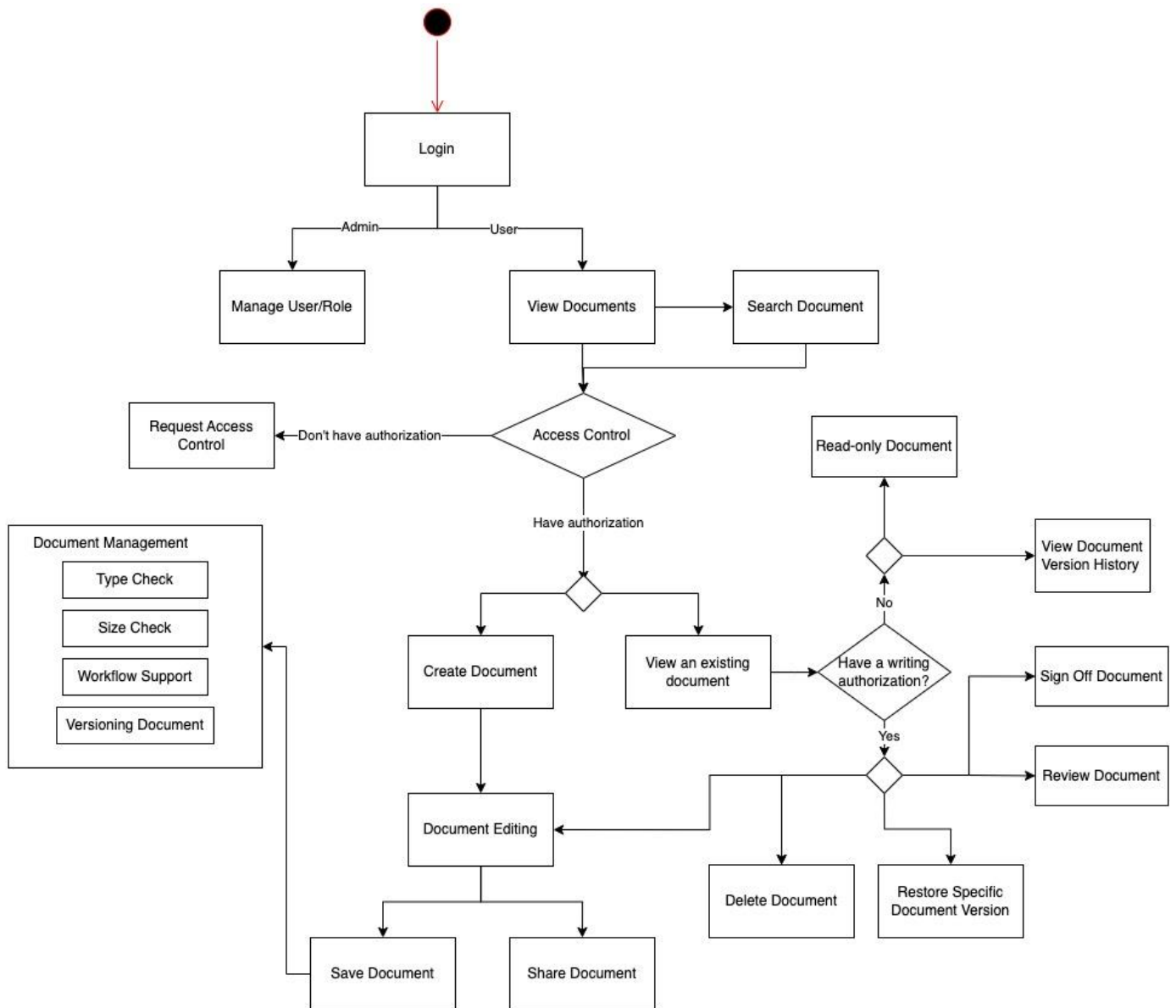
Use Case UC-10	View Document Version History
Related Requirements	REQ8
Initiating Actor	User
Actor's Goal	To view the version history of a specific document.
Preconditions	User is logged in to the system. Document must have at least one version.
Postconditions	User can view the version history of the document.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User selects the desired document to view its version history. 2. System displays a list of all versions of the selected document. 3. User selects a specific version to view. 	

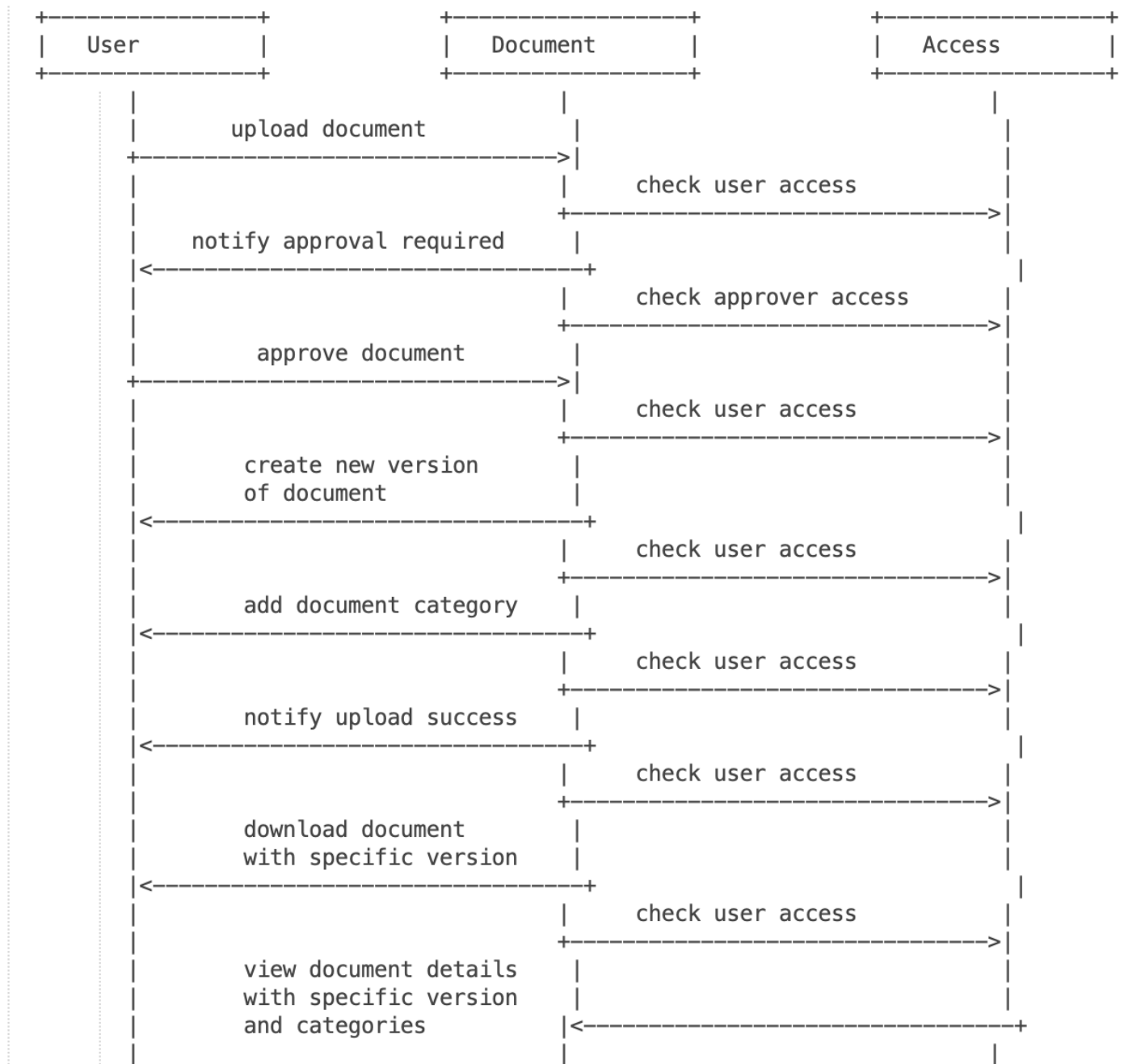
Use Case UC-11	Restore Specific Document Version
Related Requirements	REQ8
Initiating Actor	User
Actor's Goal	To undo any unwanted changes made to a document by restoring a previous version.
Preconditions	User is logged in to the system. User must have appropriate access permissions to the document. Document must have at least one previous version.
Postconditions	The previous version of the document is restored. The current version of the document is updated to reflect the restored version.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User opens the document for which they want to restore a previous version. 2. User accesses the version history for the document. 3. User selects the desired version to restore. 4. System restores the selected version and updates the current version of the document to reflect the restored version. 5. System notifies the user that the document has been successfully restored. 	

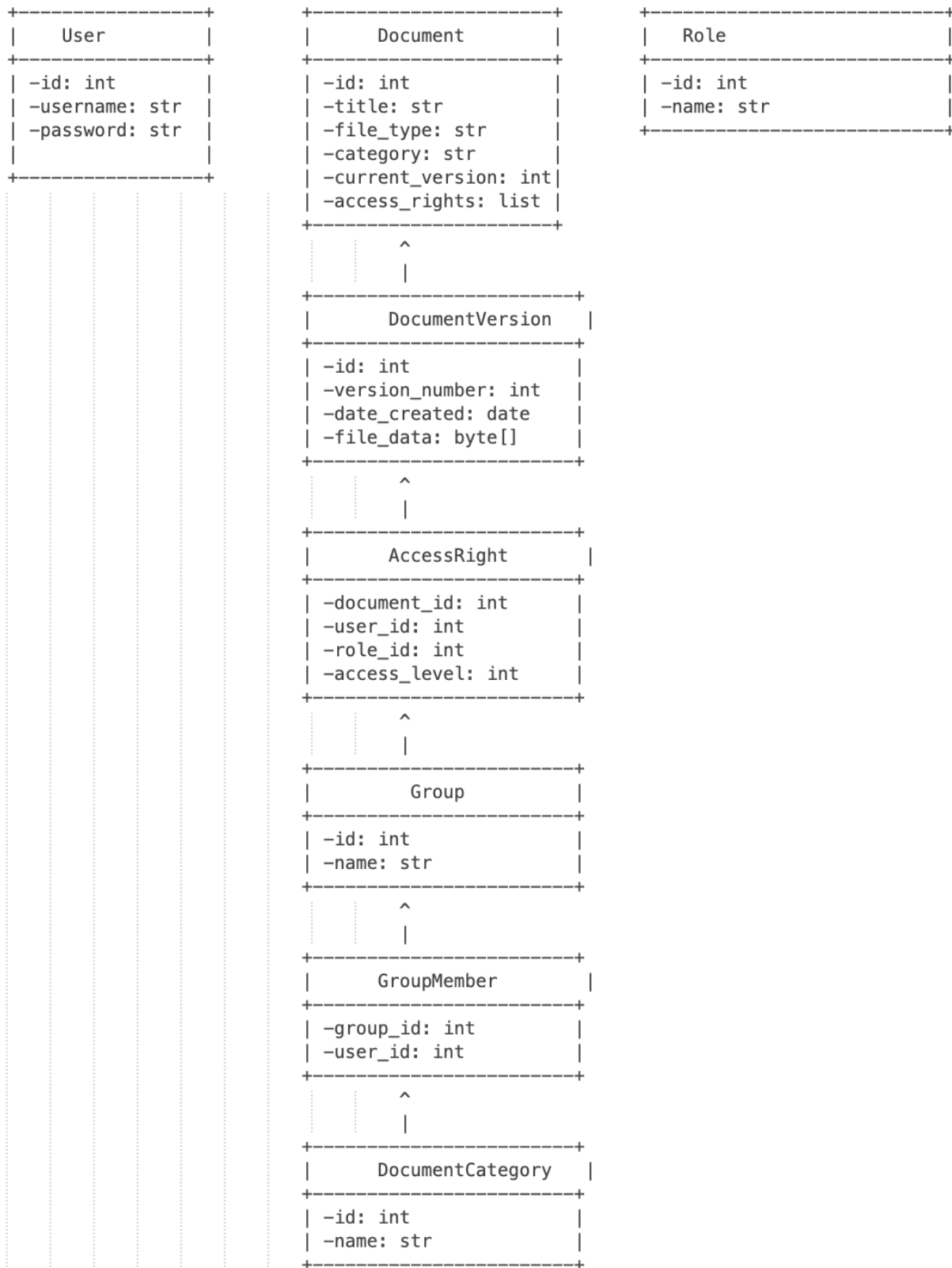
Use Case UC-12	Review Document
Related Requirements	REQ10
Initiating Actor	User
Actor's Goal	To send a document for review so that other users can provide feedback and suggest changes.
Preconditions	User is logged in to the system. User must have appropriate access permissions to the document.
Postconditions	Users designated as reviewers are notified of the new document review request.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none"> 1. User opens the document they want to send for review. 2. User selects the "send for review" option. 3. User designates the users who will be the reviewers for the document. 4. System notifies the designated reviewers of the new review request. 5. Reviewers can access the document and provide feedback. 	

Use Case UC-13	Sign Off Document
Related Requirements	REQ10
Initiating Actor	User
Actor's Goal	To indicate that a document is complete and ready for use.
Preconditions	User is logged in to the system. User must have appropriate access permissions to the document.
Postconditions	Document is marked as "signed off" and users can see that the document has been signed off.
Flow of Events for Main Success Scenario:	
<ol style="list-style-type: none">1. User opens the document they want to sign off.2. User selects the "sign off" option and confirms that they want to sign off on the document.3. System marks the document as "signed off" and other users who have access to the document can see that it has been signed off.	

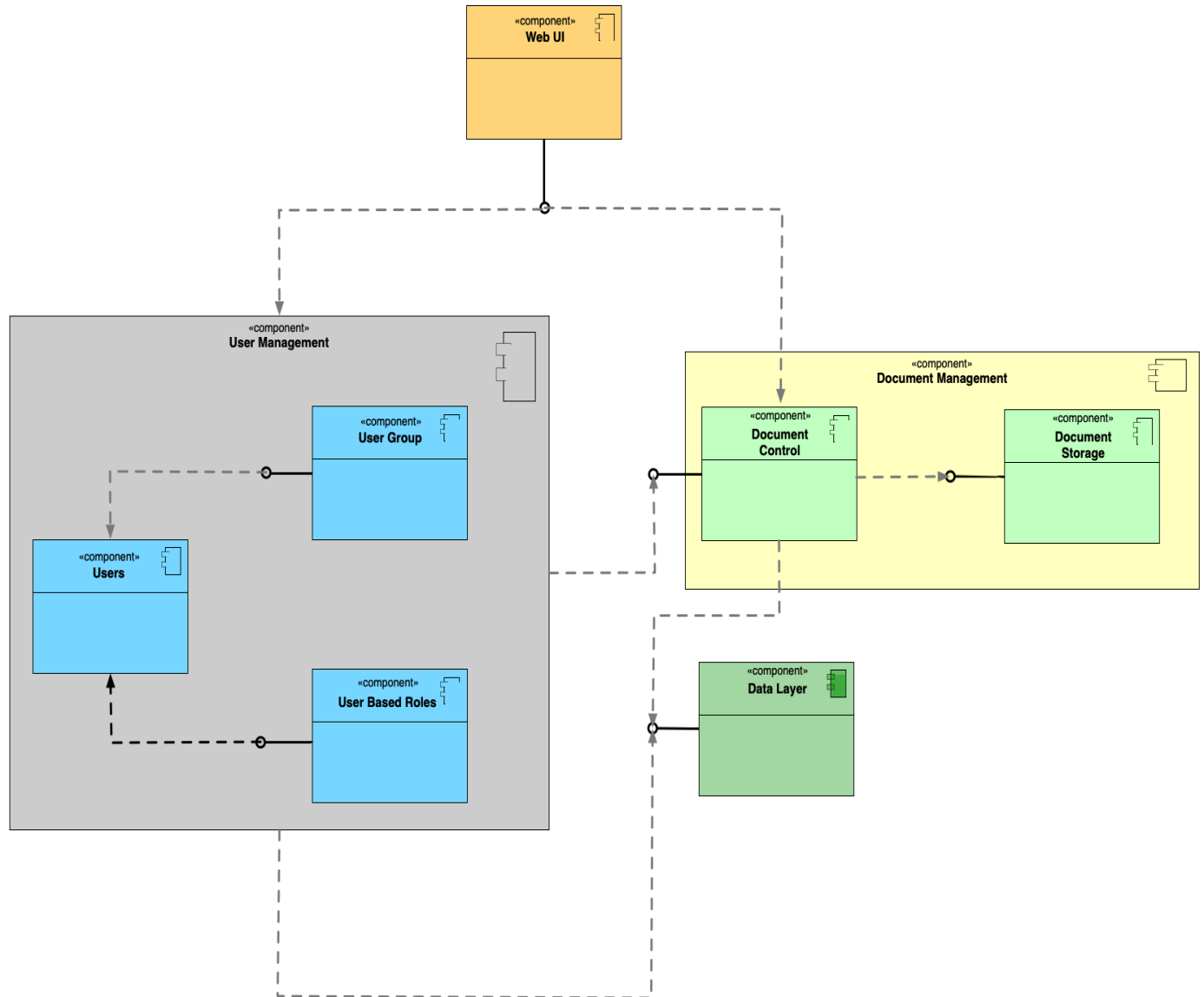
UML Activity Diagram



UML Sequence Diagram

UML Class Diagram

UML Component Diagram



References

Ian Sommerville (2010). *Software Engineering (9th ed.)*. Pearson.

Roger S. Pressman, Bruce Maxim (2014). *Software Engineering: A Practitioner's Approach (8th ed.)*. McGraw Hill.