IT 314
Software Engineering

Real Time Collaborative Editor

Sprint Details

Dhirubhai Ambani Institute of Information and
Communication Technology
Gandhinagar, Gujarat

Submitted By   Date of Submission
G35   2/12/2024

Sprint 1: Authentication and Project creation:

User Stories :

User Story 1

- **Front:** As a user, I want to log in and out of the system securely so that my documents and projects are protected.
- **Back:**
    - The user must log in with valid credentials to access the system.
    - The user can log out securely to end their session.

User Story 2

- **Front:** As a user, I want to reset my password so that I can recover access to my account.
- **Back:**
    - The user can request a password reset.
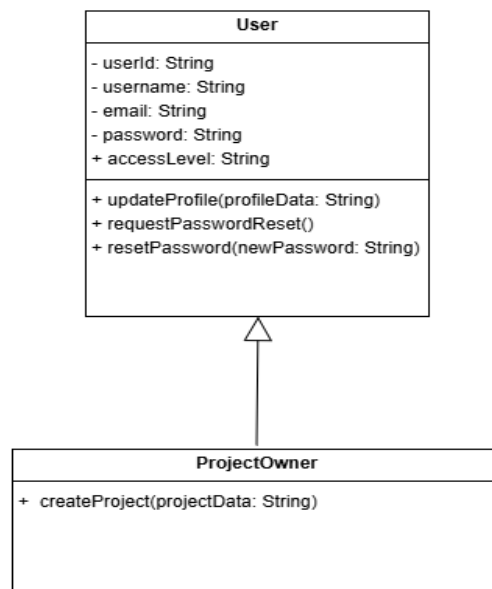   - The user receives instructions to reset their password.

User Story 3

- **Front:** As a user, I want to update my profile so that I can make changes.
- **Back:**
    - The user can access a profile page displaying their personal information (e.g., name, email, role).
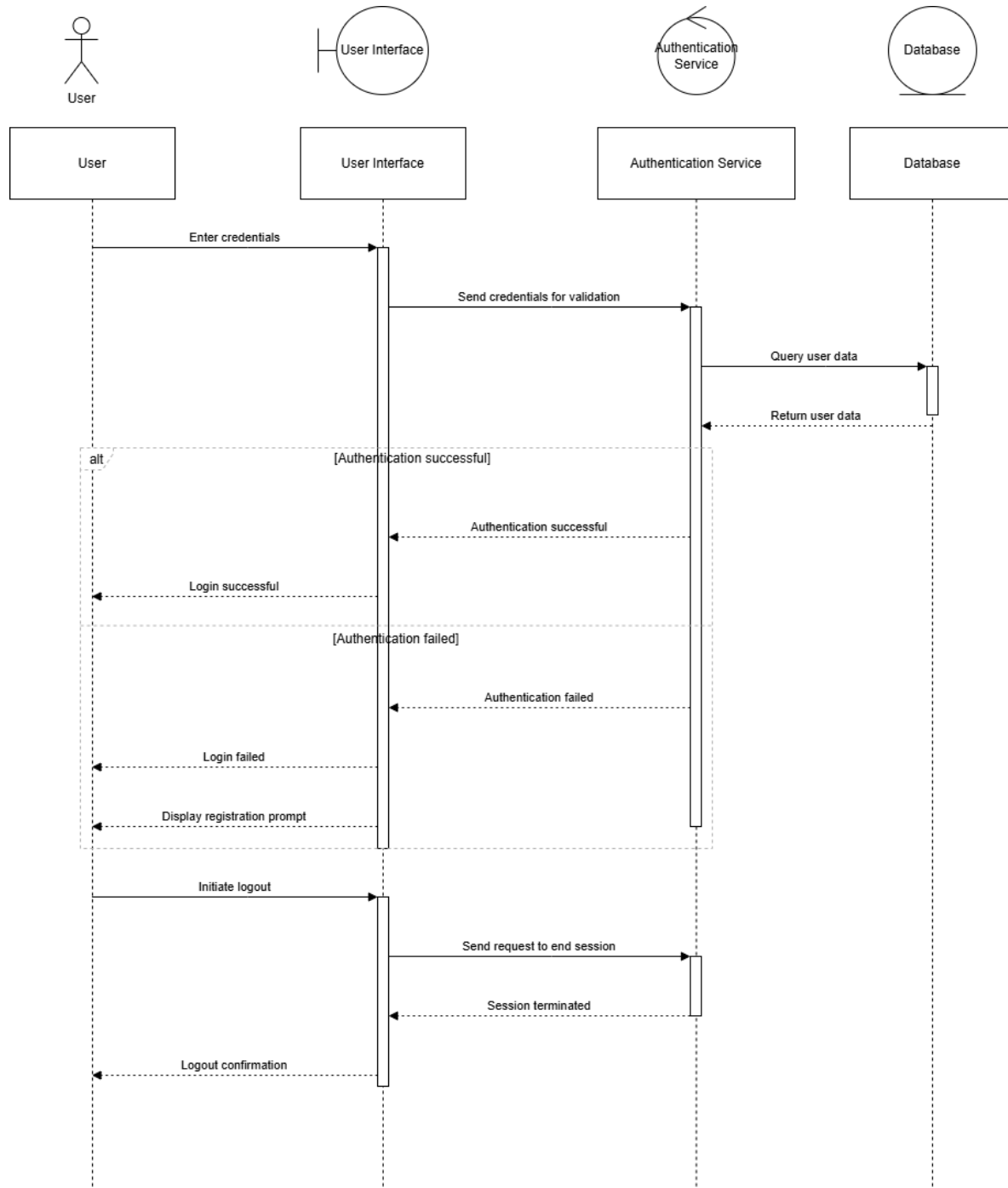   - The user can view their activity history and preferences.

User Story 4

- **Front:** As a user, I want to view my profile so that I can see my personal information and settings.
- **Back:**
    - ○ The user can access a profile page displaying their personal information (e.g., name, email, role).
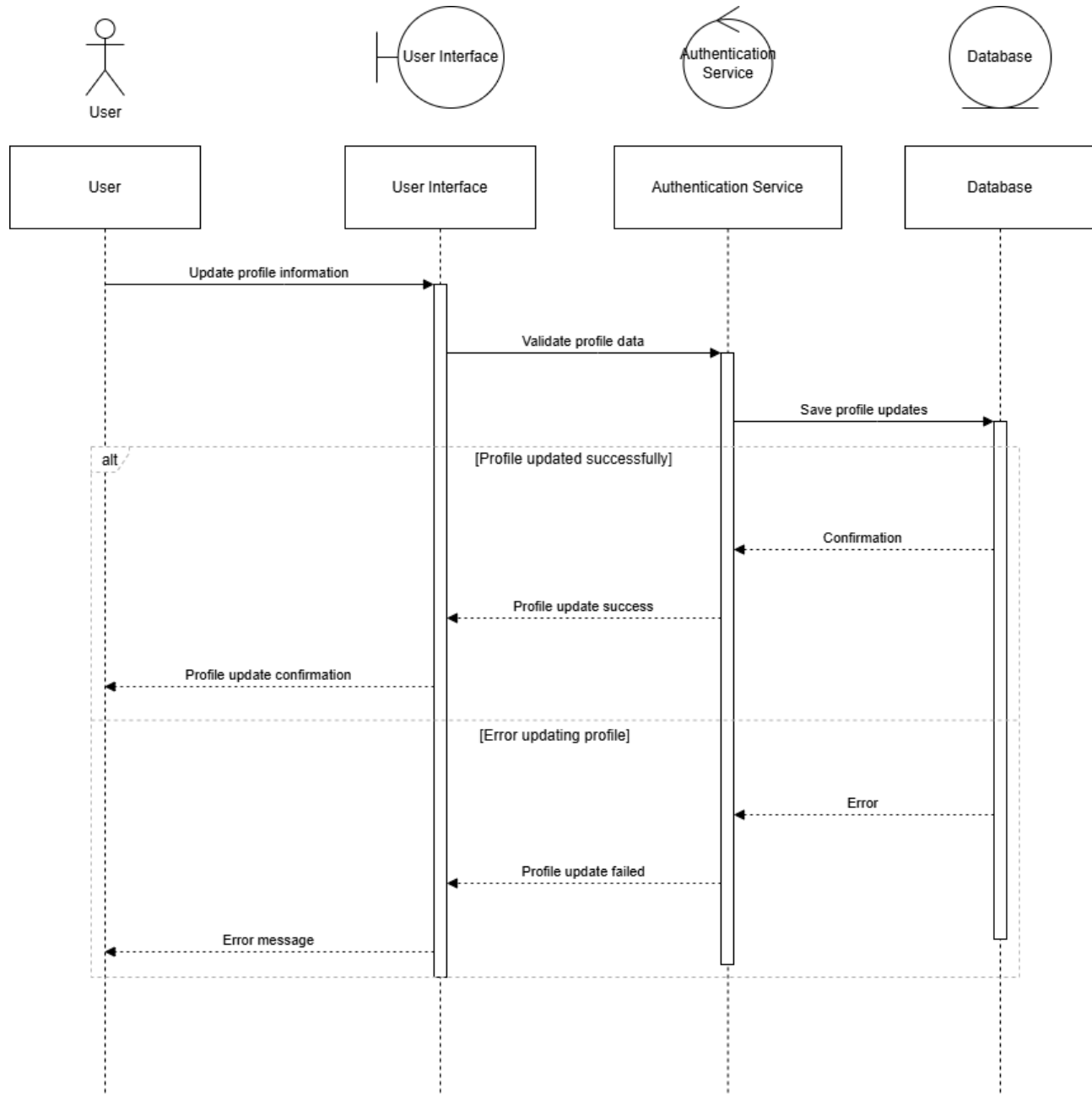   - ○ The user can view their activity history and preferences.
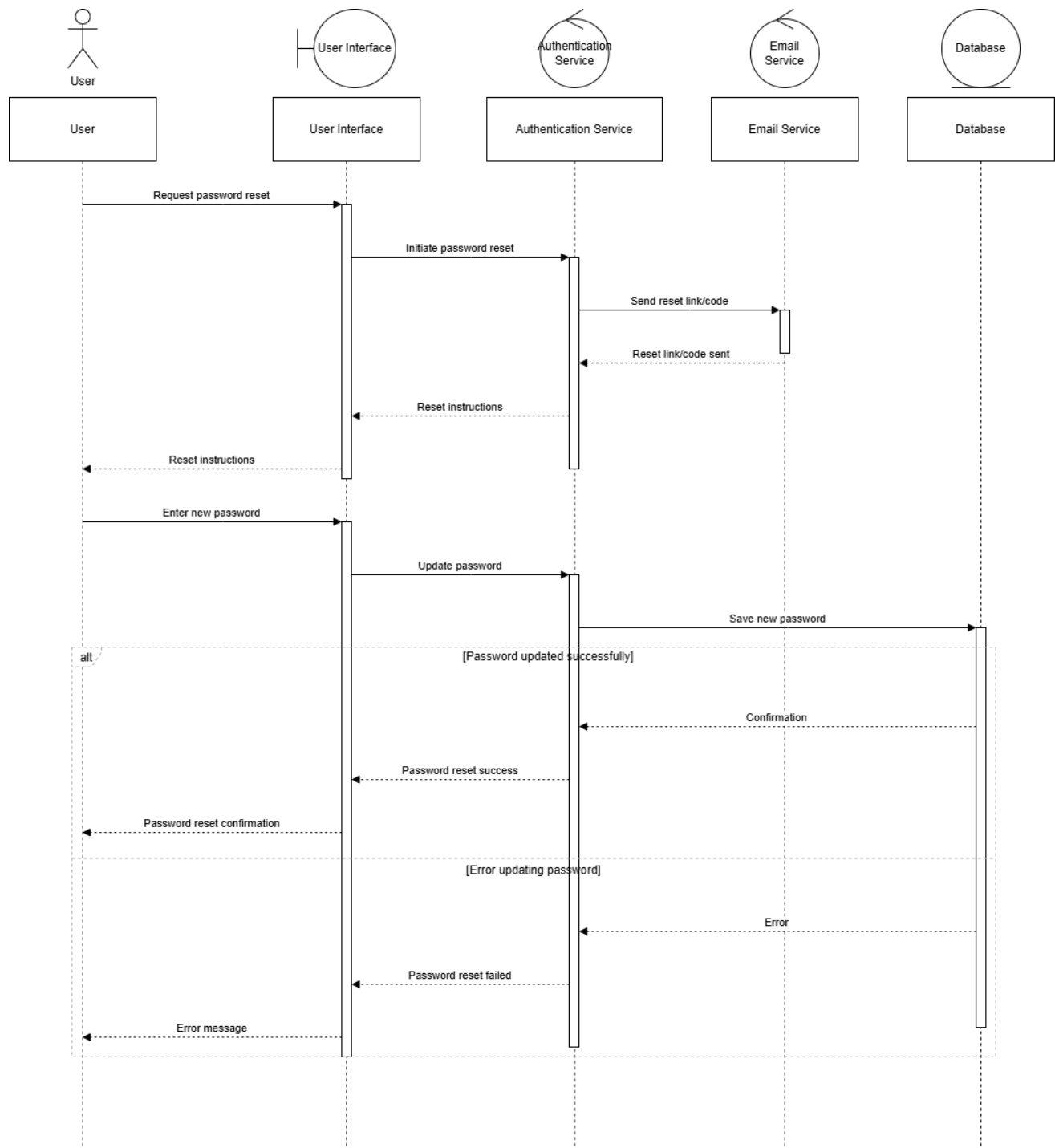
Class Diagram :

Sequence Diagram : Login & Logout

User | User Interface | Authentication Service | Database

User → User Interface: Enter credentials

User Interface → Authentication Service: Send credentials for validation

Authentication Service → Database: Query user data

Database ⇢ Authentication Service: Return user data

alt

[Authentication successful]

Authentication Service ⇢ User Interface: Authentication successful

User Interface ⇢ User: Login successful

[Authentication failed]

Authentication Service ⇢ User Interface: Authentication failed

User Interface ⇢ User: Login failed

User Interface ⇢ User: Display registration prompt

User → User Interface: Initiate logout

User Interface → Authentication Service: Send request to end session

Authentication Service ⇢ User Interface: Session terminated

User Interface ⇢ User: Logout confirmation

Update User profile :

Reset User Password :

Sprint 2: Project & Document Creation

User stories :

User Story 5

- **Front:** As a user, I want to create projects so that I can manage different tasks and resources.
- **Back:**
  - o The user must be able to create a new project by specifying a name and description.
  - o The user must be able to view and track the progress of the project.
  - o The project must be saved and retrievable from the list.

User Story 6

- **Front:** As a user, I want to create folders so that I can organize my documents efficiently.
- **Back:**
  - o The user can create new folders with custom names.
  - o The user can view folders in a hierarchical structure.

User Story 7

- **Front:** As a user, I want to create documents so that I can store important information.
- **Back:**
  - o The user can create new documents by specifying a title and content.
  - o The user can organize documents into folders or categories.

User Story 8

- **Front:** As a user, I want to see a list of all available projects so that I can select the one I need to work on.
- **Back:**
  - The user can view all projects for which they have access.

User Story 9

- **Front:** As a user, I want to see a list of all files within a project so that I can select the document I need to work on.
- **Back:**
 - The user can view all documents within a selected project.

User Story 10

- **Front:** As a user, I want to view a document so that I can read its content without making changes.
- **Back:**
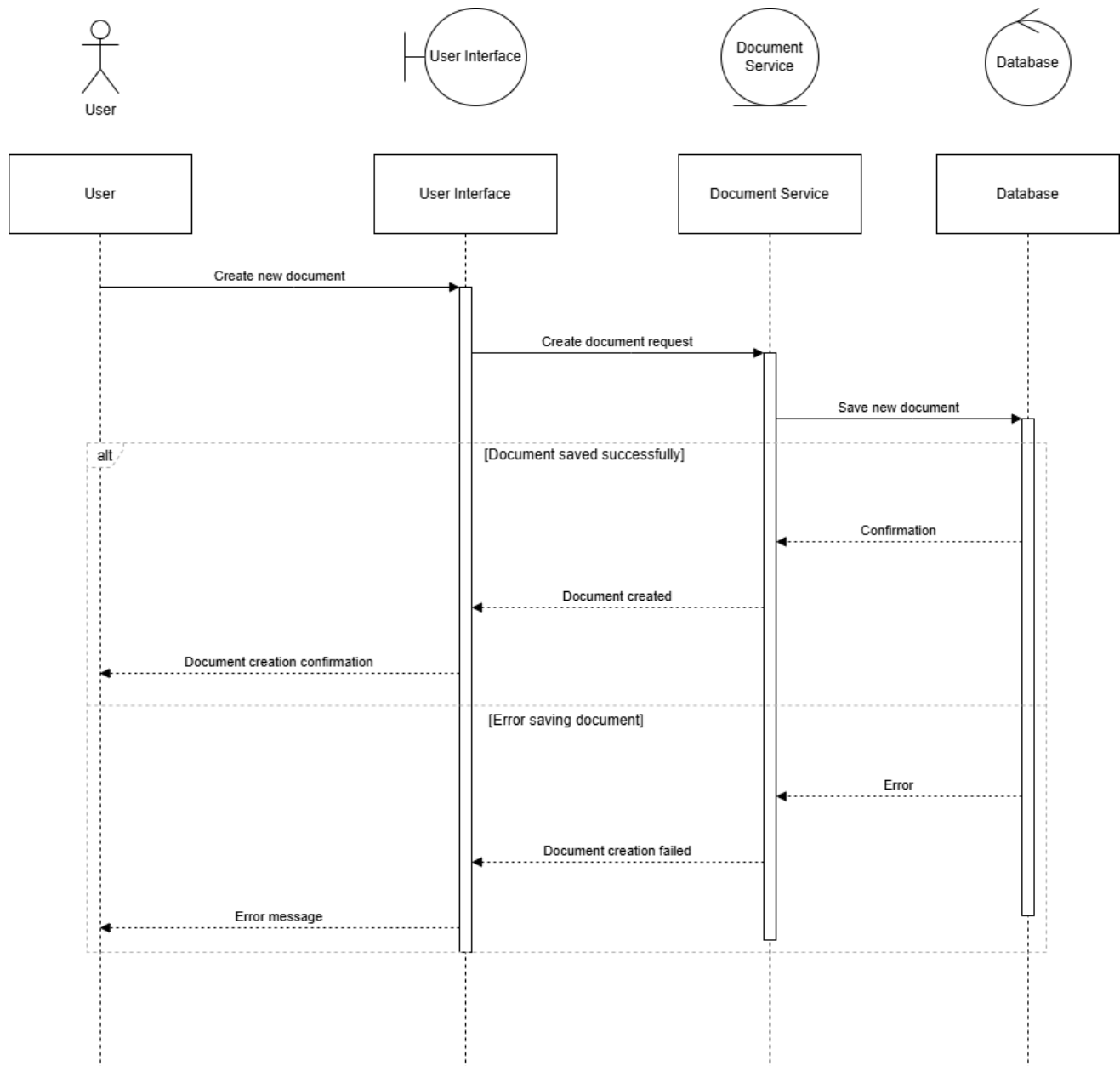 - The user can open and read a document in view-only mode.

User Story 11

- **Front:** As a user, I want to add content in my project document.
- **Back:** The user can add text, images, tables, or other supported content to their project document.

Class Diagram :
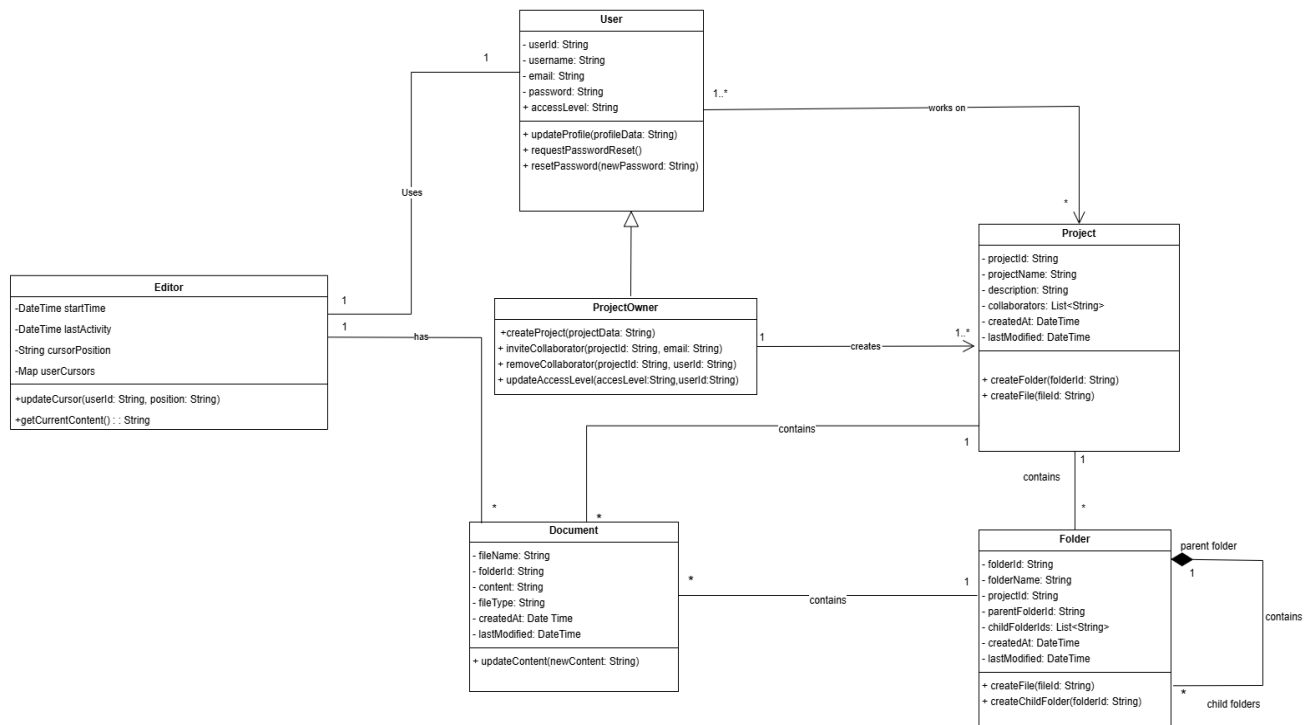
Sequence Diagram :
Document creation:

Sprint 3: Collaboration in real time

User stories :

User Story 12

- **Front:** As a user, I want to add users or give access to my project so that we can work together on it.
- **Back:**
  - o The owner can add users via email/username.
  - o Permissions (view or edit) can be set for each user.
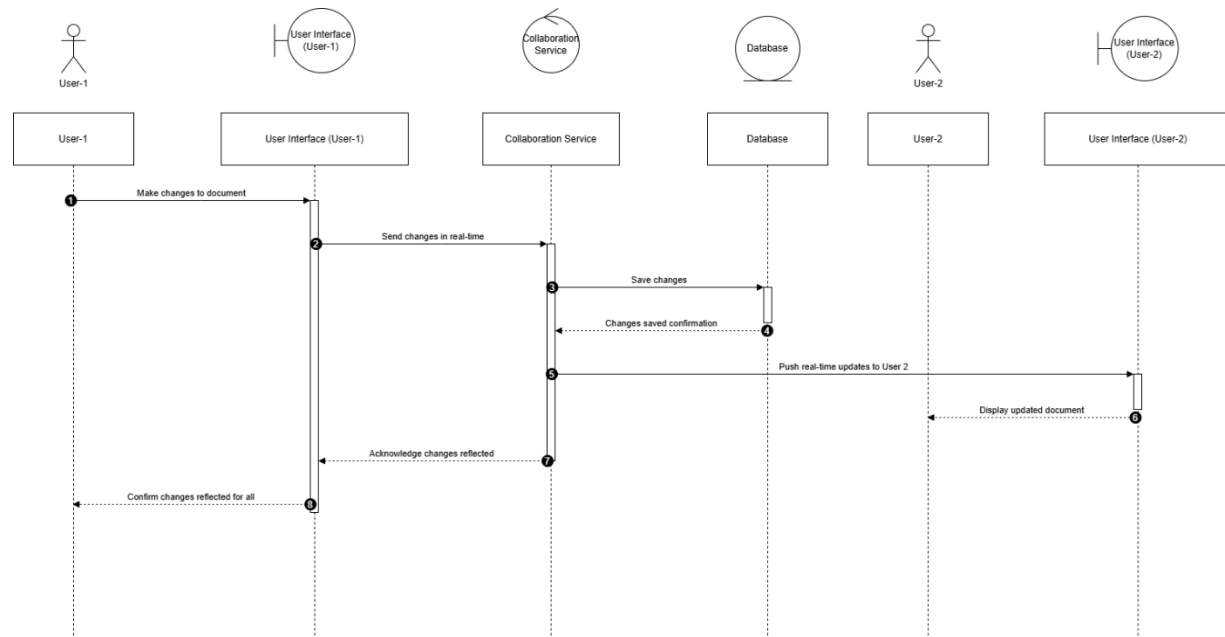
User Story 13

- **Front:** As a user, I want to collaborate with others in real-time so that changes are reflected immediately for all users.
- **Back:**
  - o All changes are updated in real-time for all collaborators.
  - o Conflicts are managed automatically.

User Story 14

- **Front:** As a user, I want to see who else is working on a document in real-time so that I know who is collaborating with me.
- **Back:**
  - o The user can view a list of all other users currently editing or viewing the document.

Class Diagram :

Sequence Diagrams :
Real time collaboration:

Sprint 4: Comment/ Download Document

User Story 15

- **Front:** As a user, I want to download a document so that I can have an offline copy for my records.
- **Back:**
  - o The user can download a document in the format of their choice.

User Story 16

- **Front:** As a user, I want to rename an existing document so that its title better reflects its content.
- **Back:**
  - o The user can change the name of a document.
  - o The new name is saved and reflected in all associated views.

User Story 17

- **Front:** As a user, I want to rename a project so that its name accurately reflects the work being done.
- **Back:**
  - o The user can rename a project.
  - o The updated project name is displayed in all associated views.

User Story 18

- **Front:** As a user, I want to delete a document so that I can remove outdated or unnecessary content.
- **Back:**
    - The user can delete a document.
    - The document is permanently removed from the system, or optionally moved to a trash/recycle bin.

User Story 19

- **Front:** As a user, I want to delete a project so that I can remove an entire set of documents that are no longer needed.
- **Back:**
    - The user can delete a project, along with all documents within it.
    - The project and its contents are removed from the system, or optionally moved to a trash.

Class Diagram:
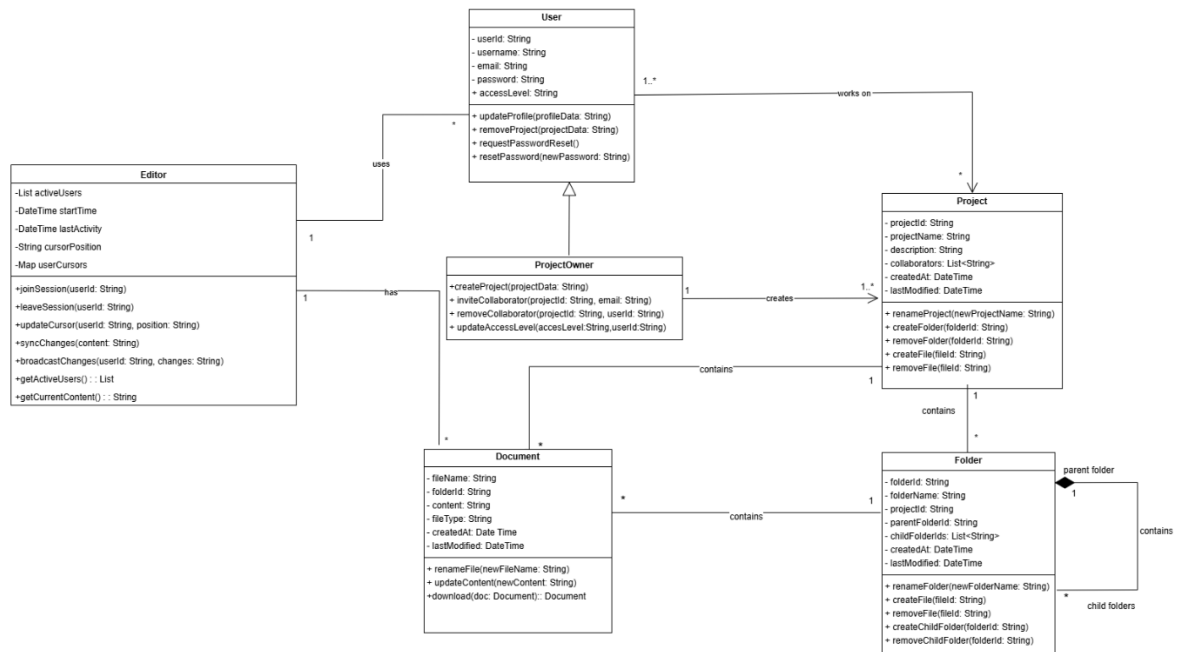
Sequence Diagram :
Download Document :
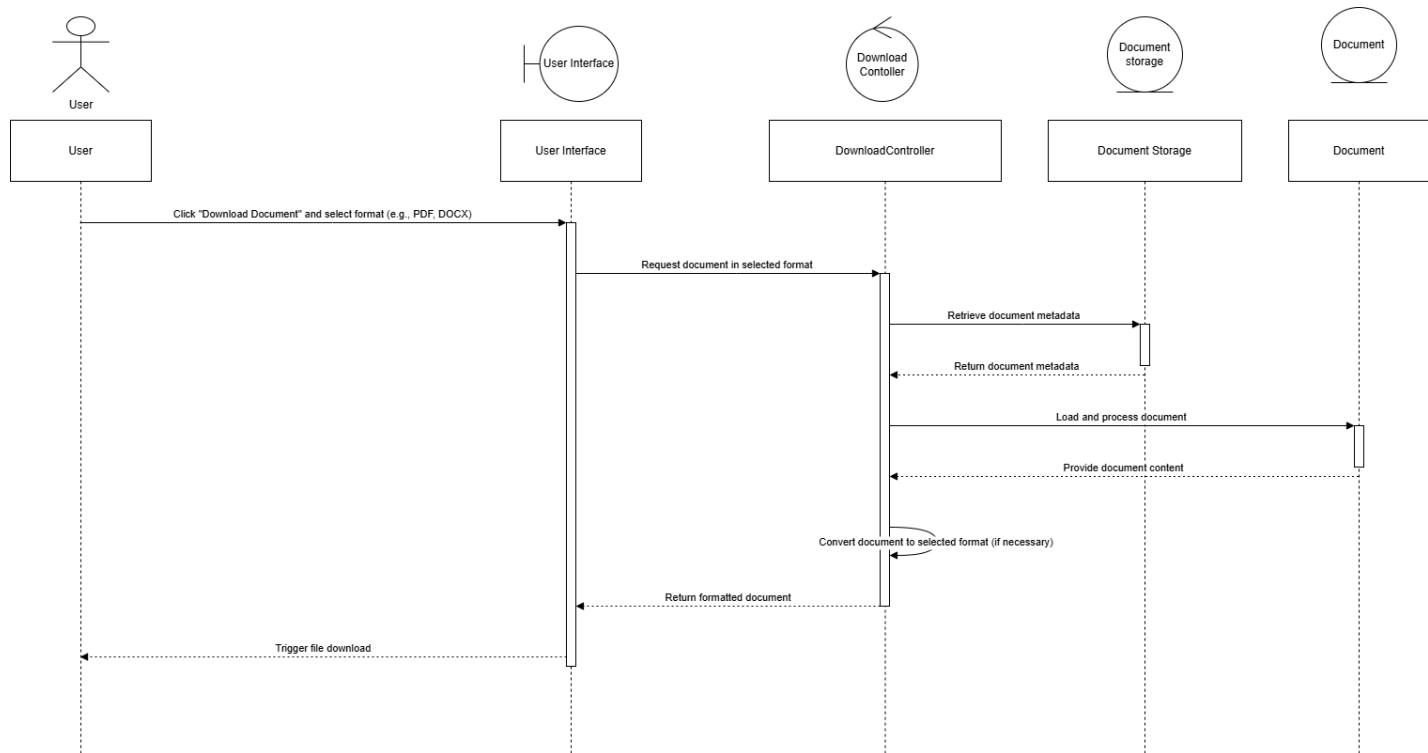
Delete Document :
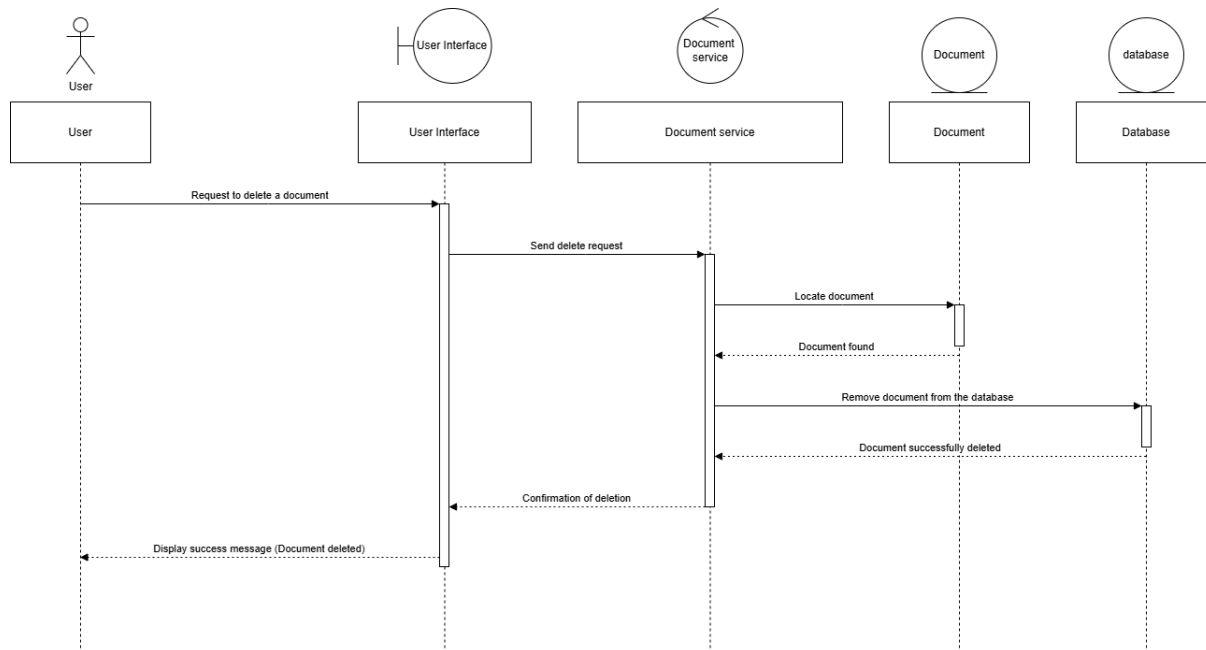
Rename Document :
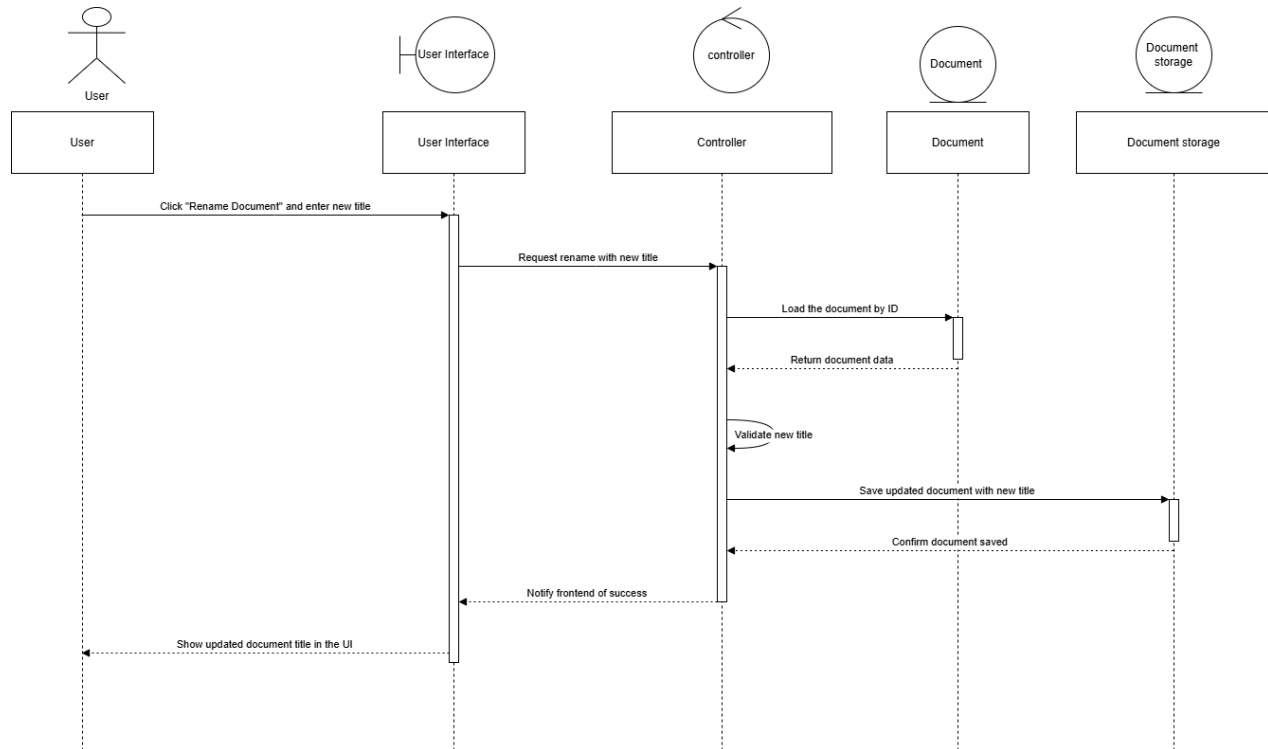
Sprint 5: Chat & Document History & run programs:

User Story 20

- **Front:** As a user, I want to view the history of changes made to a document so that I can track edits and updates over time.
- **Back:**
    - o The user can view a log of all changes made to the document, including the time of the change and the user who made it.

User Story 21

- **Front:** As a user, I want to send and receive text messages in real-time so that I can communicate instantly with others.
- **Back:**
  - The user can send and receive text messages within a chat.
  - Messages are delivered and displayed instantly in the chat window.

User Story 22

- **Front:** As a user, I want to run my (java, c etc)program files on editor.
- **Back:**
  - User should be able to compile and run the program directly from the editor.
  - Display errors or output of the program within the editor.
  - Support for selecting or setting the programming language for accurate execution.

Class Diagram :

Chat

-string message

-DateTime sentAt

+sendMessage()

+deleteMessage()

+getChatHistory()

User

- userId: String
- username: String
- email: String
- password: String
+ accessLevel: String

+ updateProfile(profileData: String)
+ removeProject(projectData: String)
+ requestPasswordReset()
+ resetPassword(newPassword: String)

Editor

-List activeUsers

-DateTime startTime

-DateTime lastActivity

-String cursorPosition

-Map userCursors

+joinSession(userId: String)

+leaveSession(userId: String)

+updateCursor(userId: String, position: String)

+syncChanges(content: String)

+broadcastChanges(userId: String, changes: String)

+getActiveUsers() : : List

+getCurrentContent() : : String

ProjectOwner

+createProject(projectData: String)
+ inviteCollaborator(projectId: String, email: String)
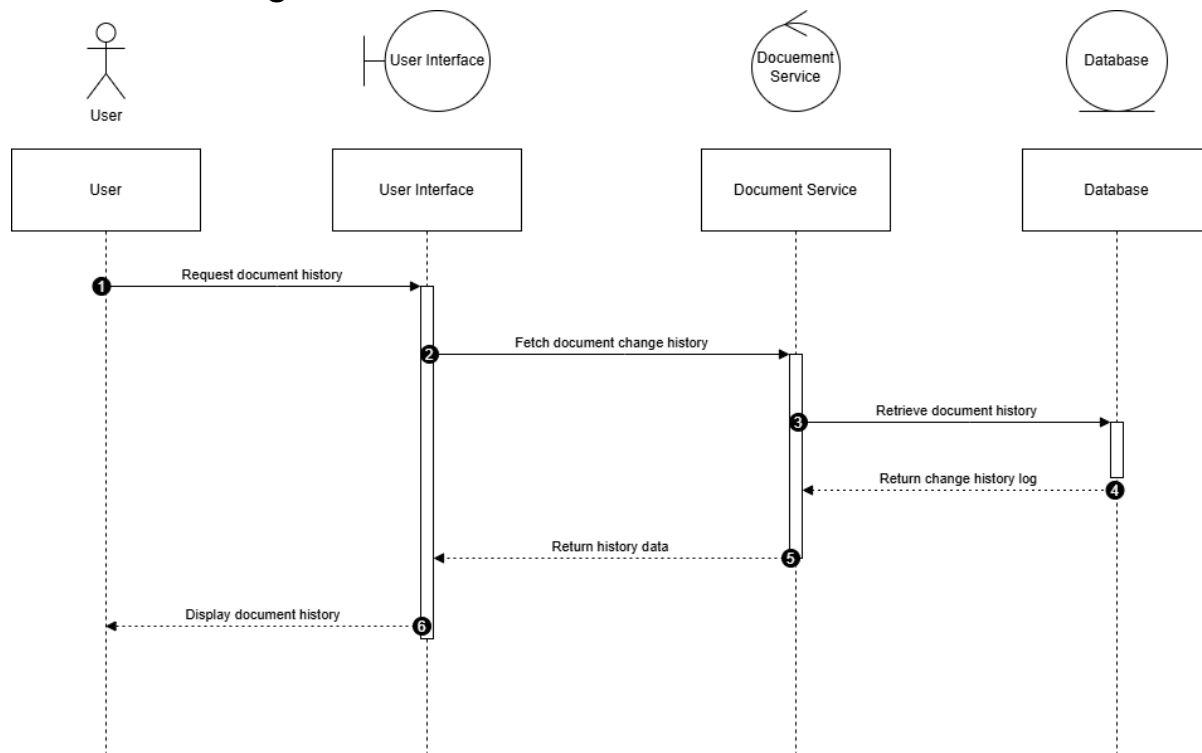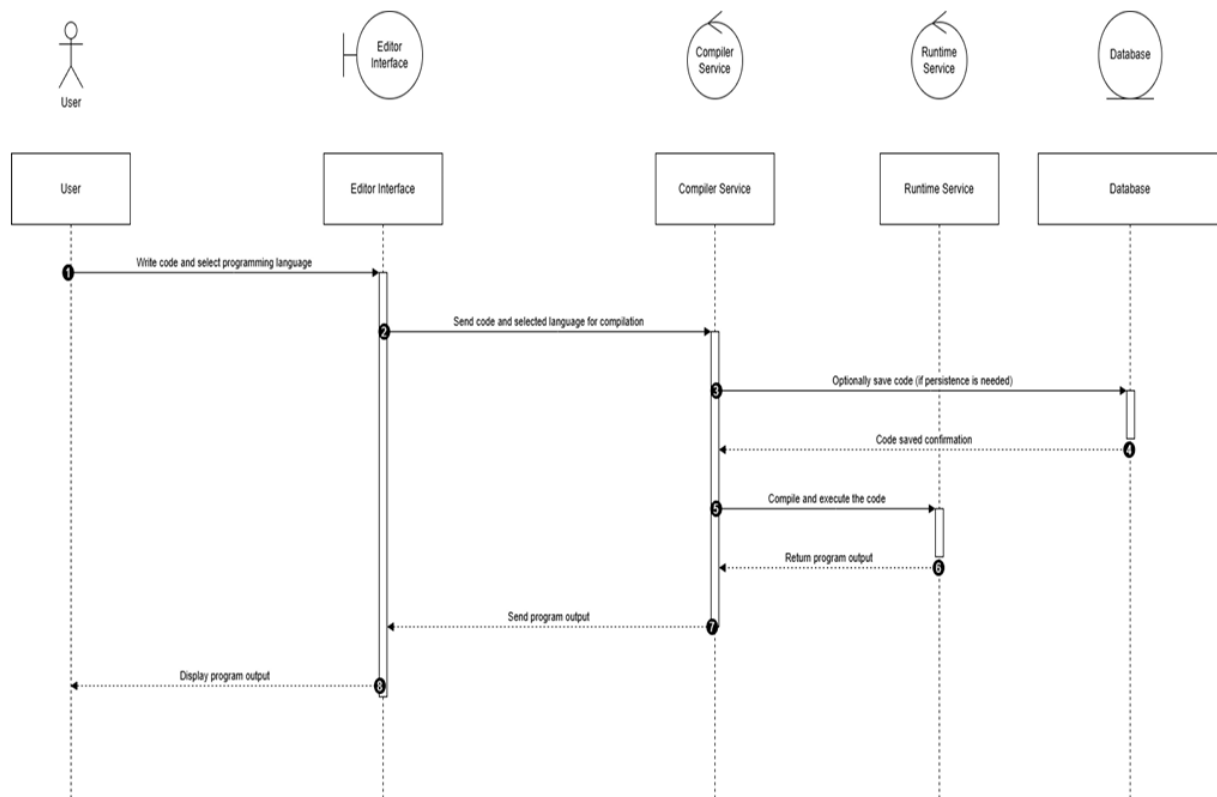+ removeCollaborator(projectId: String, userId: String)
+ updateAccessLevel(accesLevel:String,userId:String)

Project

- projectId: String
- projectName: String
- description: String
- collaborators: List<String>
- createdAt: DateTime
- lastModified: DateTime

+ renameProject(newProjectName: String)
+ createFolder(folderId: String)
+ removeFolder(folderId: String)
+ createFile(fileId: String)
+ removeFile(fileId: String)

Document

- fileName: String
- folderId: String
- content: String
- fileType: String
- createdAt: Date Time
- lastModified: DateTime

+ renameFile(newFileName: String)
+ updateContent(newContent: String)
+download(doc: Document):: Document
+runcode(doc:Document)

Folder

- folderId: String
- folderName: String
- projectId: String
- parentFolderId: String
- childFolderIds: List<String>
- createdAt: DateTime
- lastModified: DateTime

+ renameFolder(newFolderName: String)
+ createFile(fileId: String)
+ removeFile(fileId: String)
+ createChildFolder(folderId: String)
+ removeChildFolder(folderId: String)

Document Logs

-List<String> changes

-DateTime modifiedAt

-string change

+displayChanges(List<String> changes)

uses

works on

has

creates

contains

contains

contains

contains

Maintains

parent folder

child folders

Sequence Diagram :

Document Logs:

Run Code :

Chat :