Development of a Digital Content Management Application

Application inspired by Notion to facilitate the personal and professional organization of users.

Introduction

1 Project Description

Application to manage digital content efficiently.

2 Justification

Notion stands out for its versatility, usefulness, popularity and simplicity.

3 General Objective

Create an intuitive and functional tool.

Specific Objectives

Page Creation

Allow users to create and customize pages.

Effective Databases

Integrate functionality to create databases.

Permit Management

Provide administrators with access control to spaces.

User Stories

General User

Registration, login, creation and editing of pages.

Administrator

Permission management and usage statistics.

CRC cards

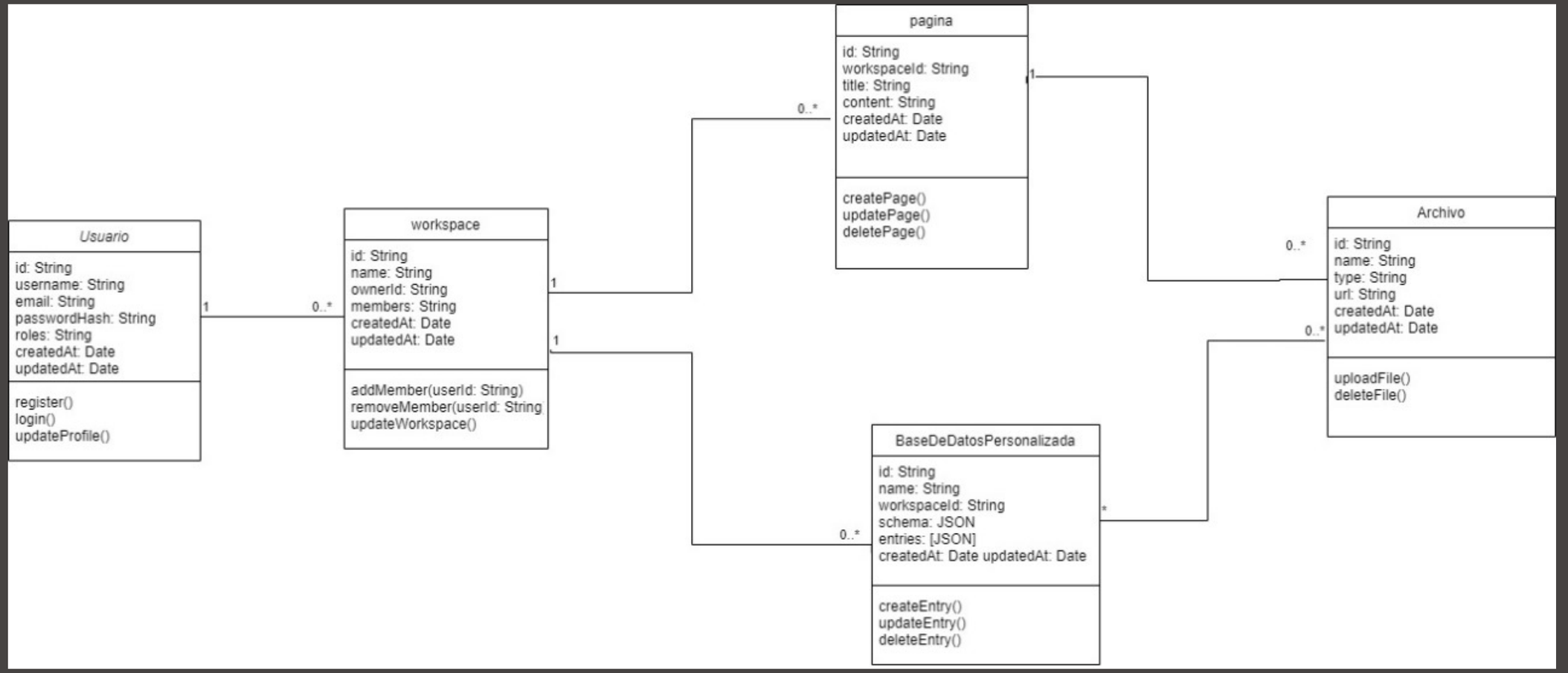
User

Class: User	
Responsibilities: Register a new user (register()). Log in (login()). Update the user's profile (updateProfile()).	**Collaborators:** Collaborates with the Workspace class, as a user can have multiple workspaces.

workspace

Class: Workspace	
Responsibilities: Create, modify, or delete a workspace (updateWorkspace()). Add members to the workspace (addMember()). Remove members from the workspace (removeMember()).	**Collaborators:** Collaborates with the User class, as a user can own or be a member of several workspaces. Collaborates with the Page class, as a workspace can contain multiple pages . Collaborates with the CustomDatabase class, as a workspace can have custom databases. Collaborates with the File class to manage files within the workspace.

Class Diagram

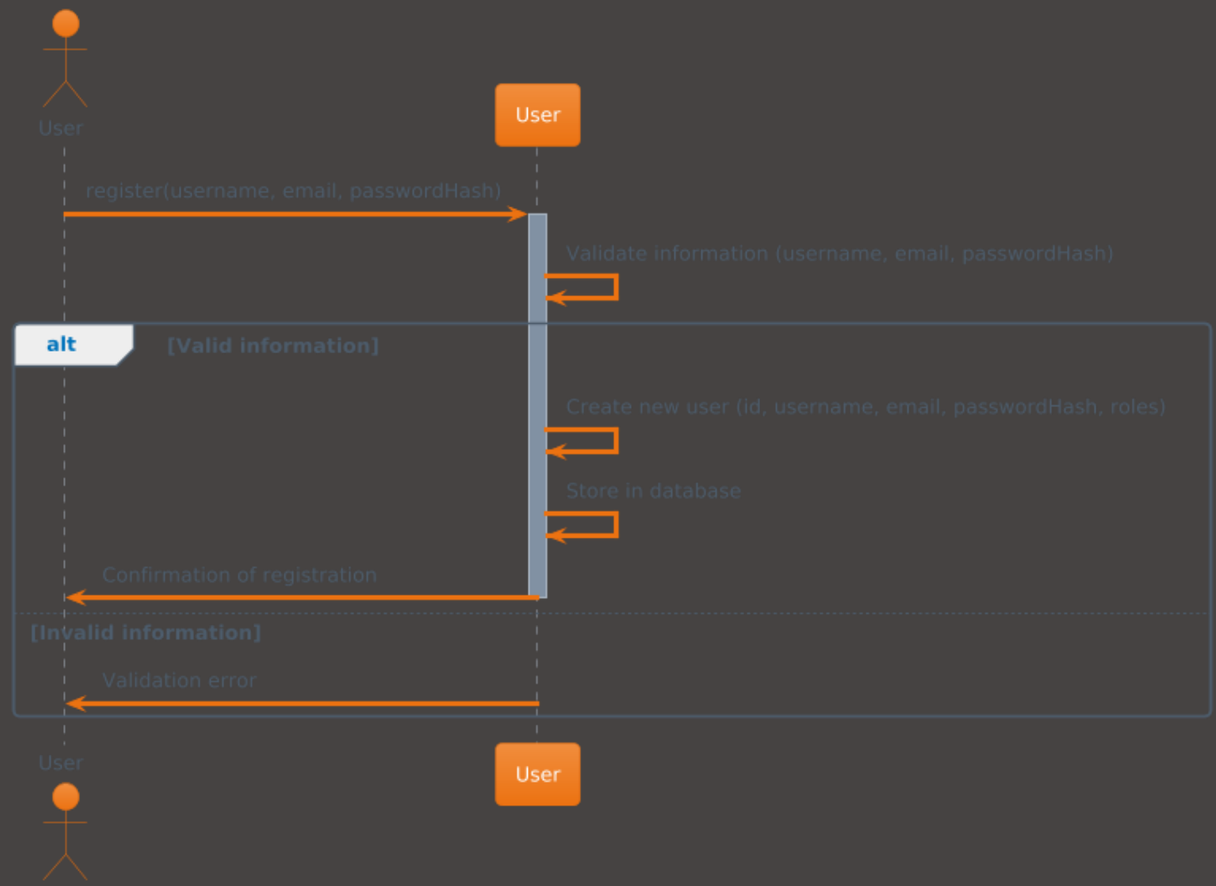


Representation of classes, attributes and relationships.

uml Diagrams

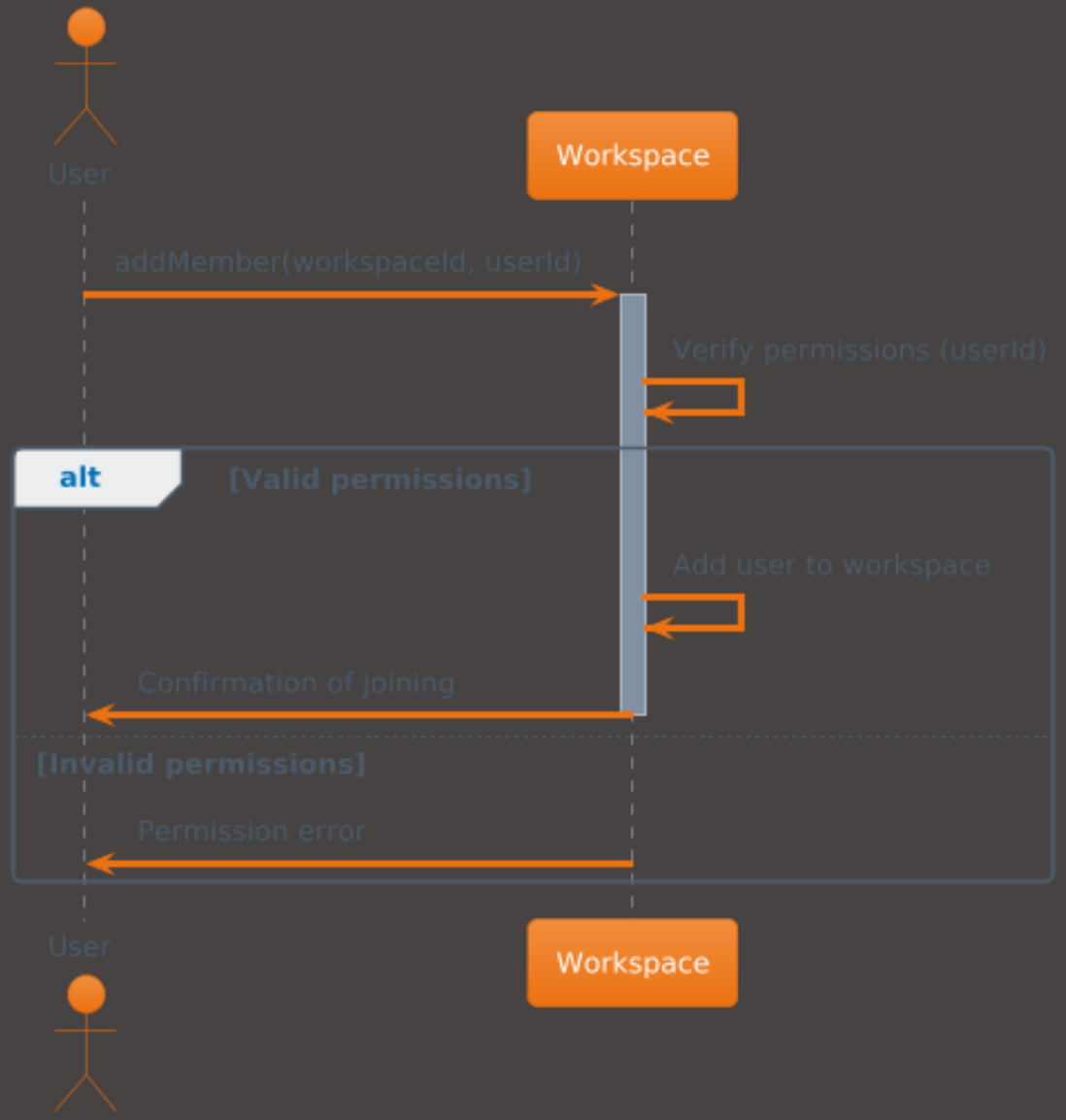
Sequence diagrams

Register user



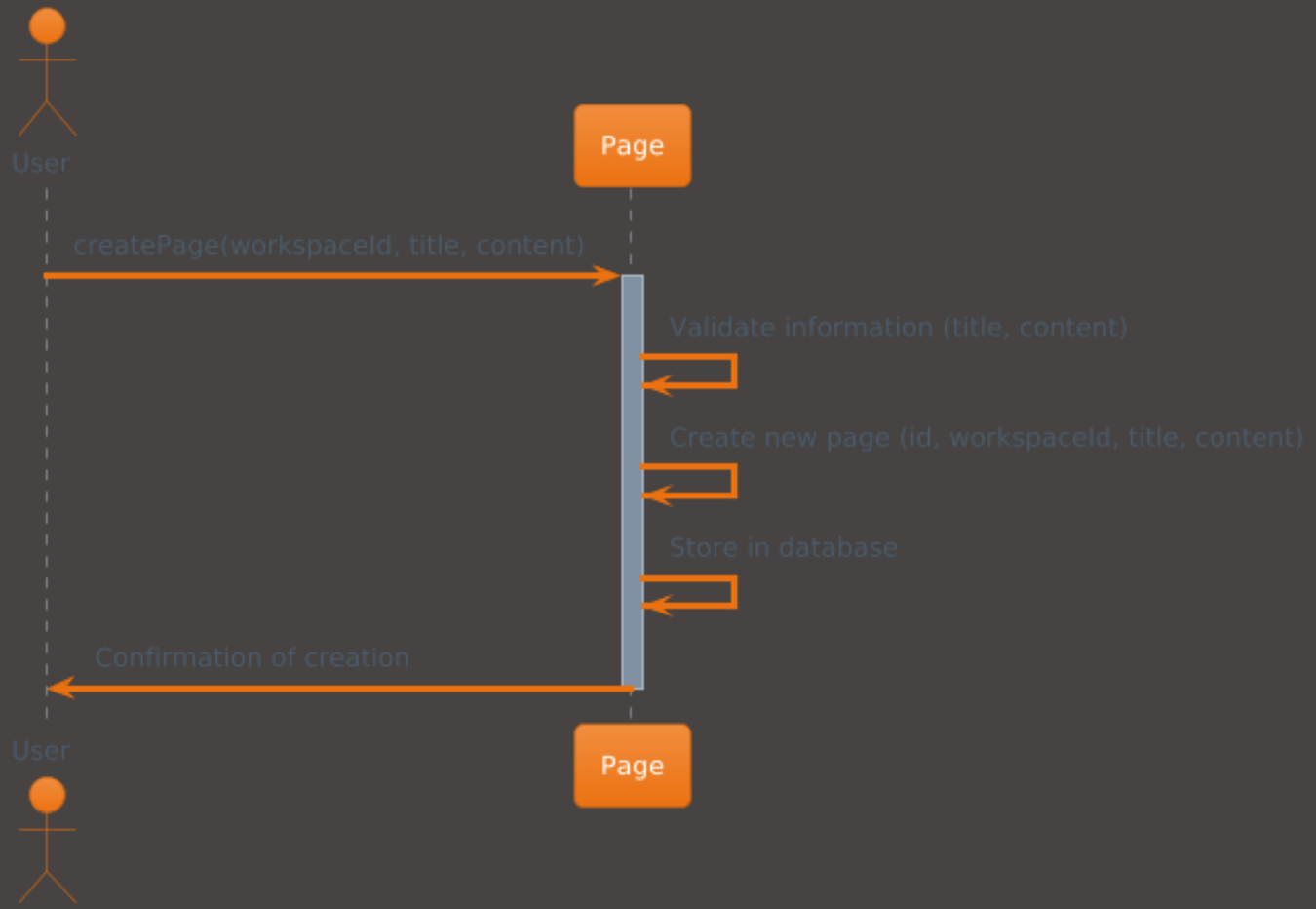
uml Diagrams

Enter the workspace



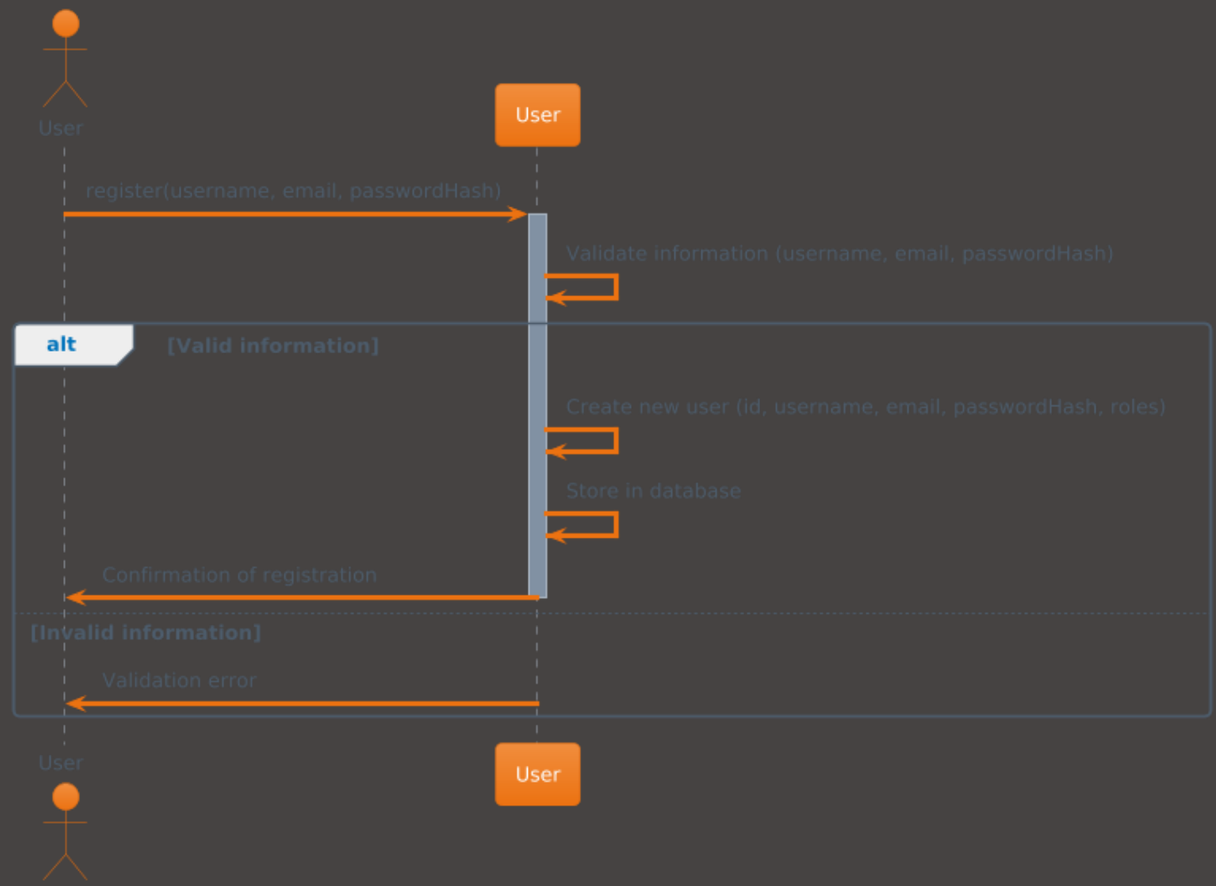
uml Diagrams

Create page



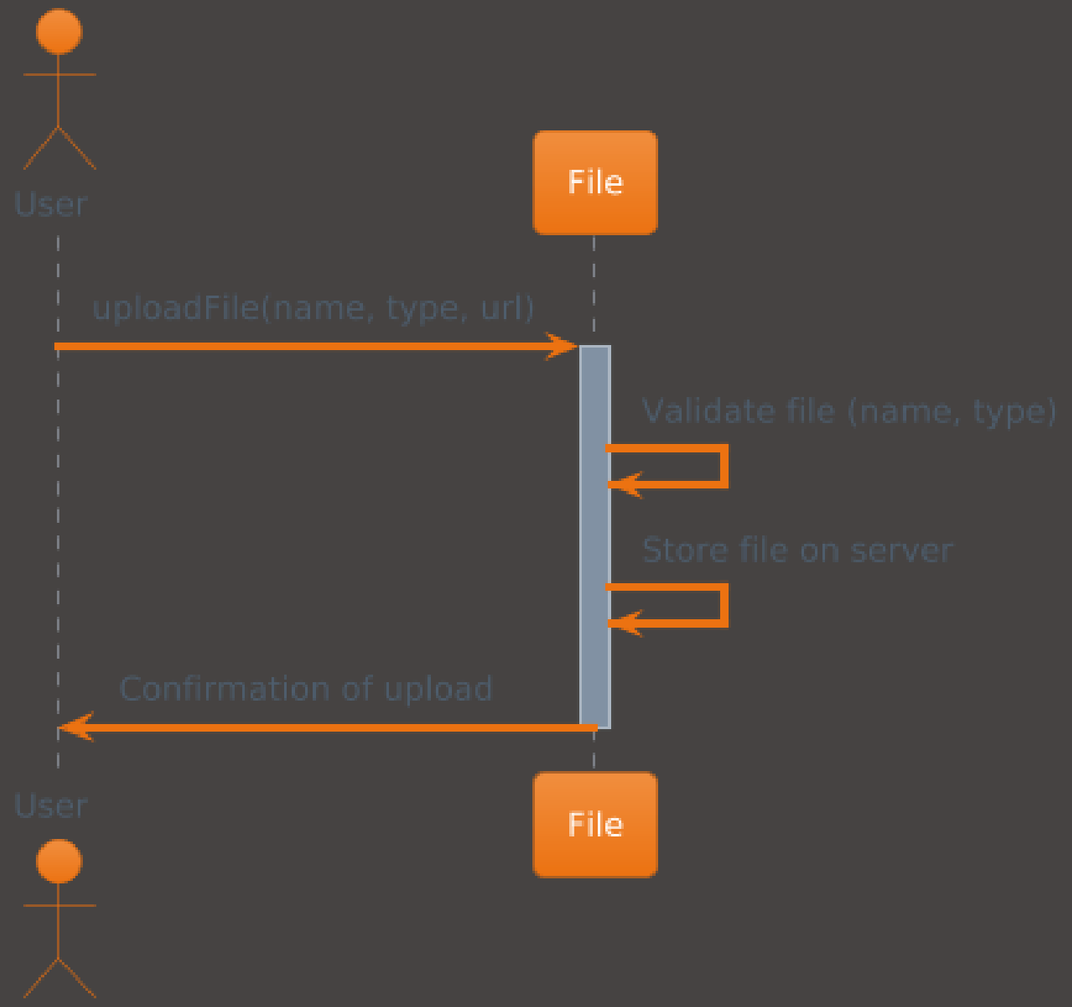
uml Diagrams

Register user



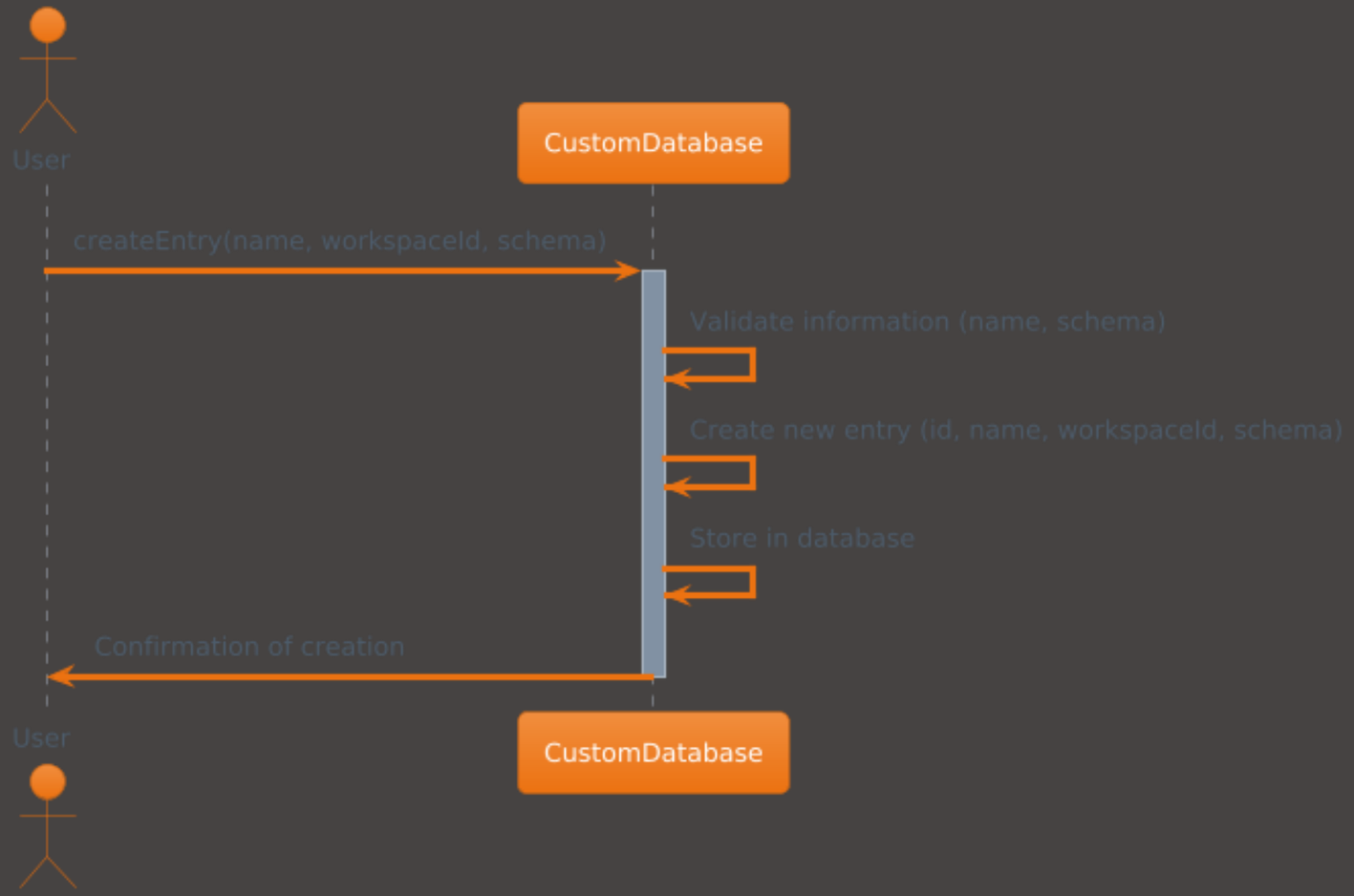
uml Diagrams

Upload file



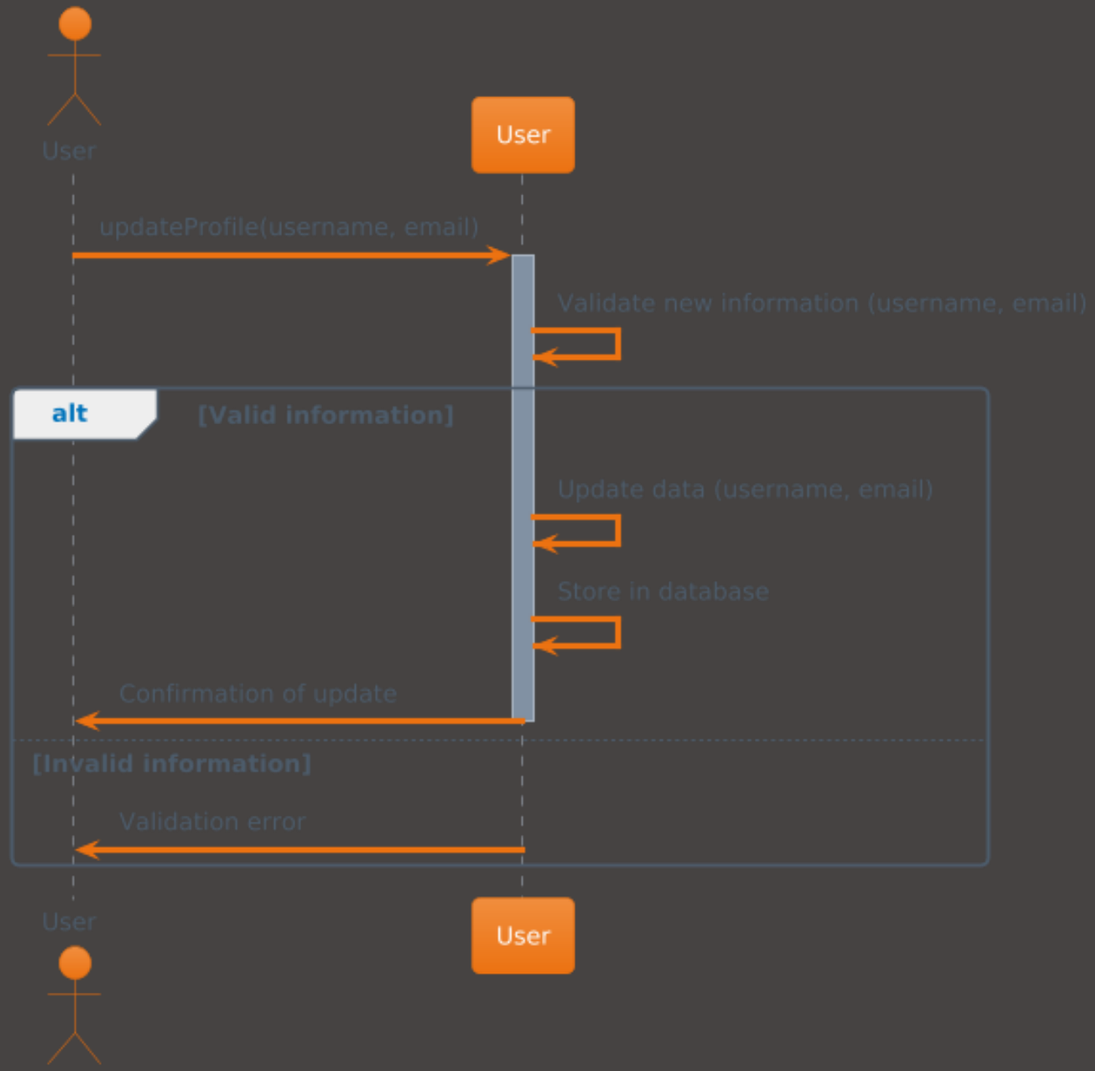
uml Diagrams

Create database



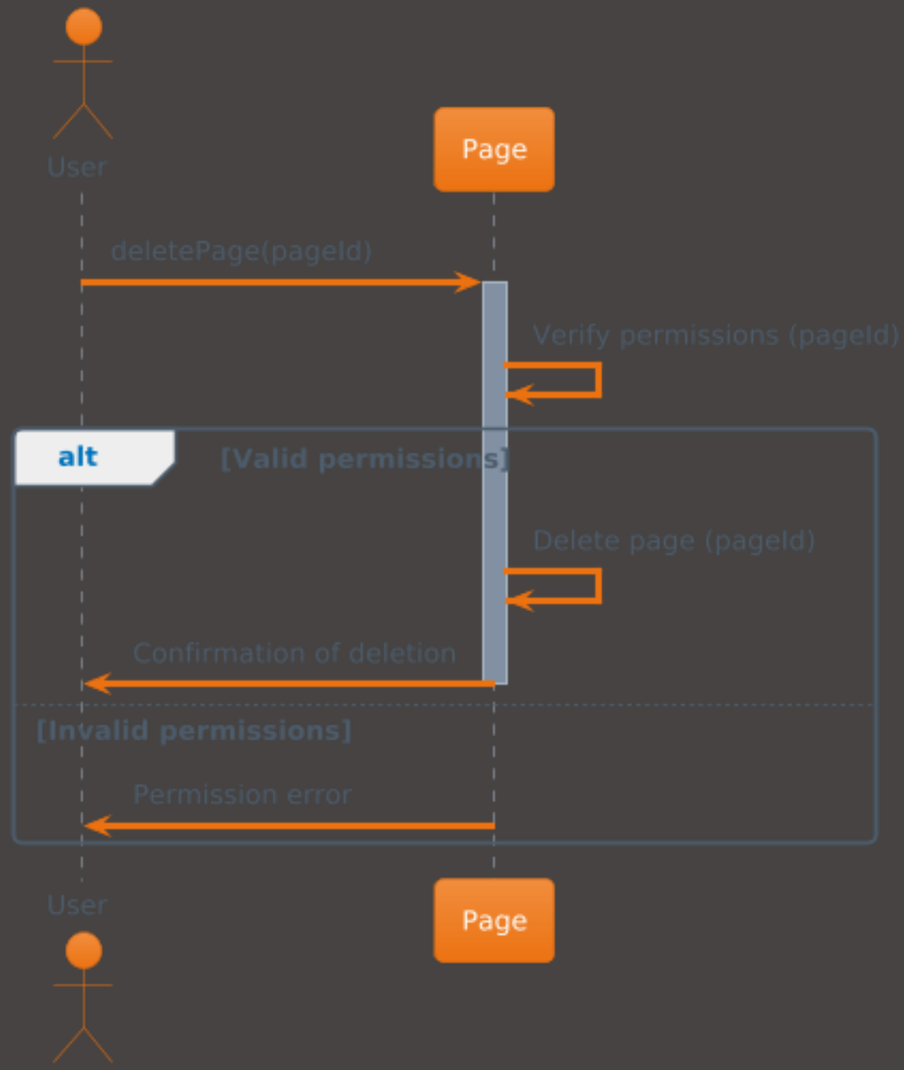
uml Diagrams

Update user profile



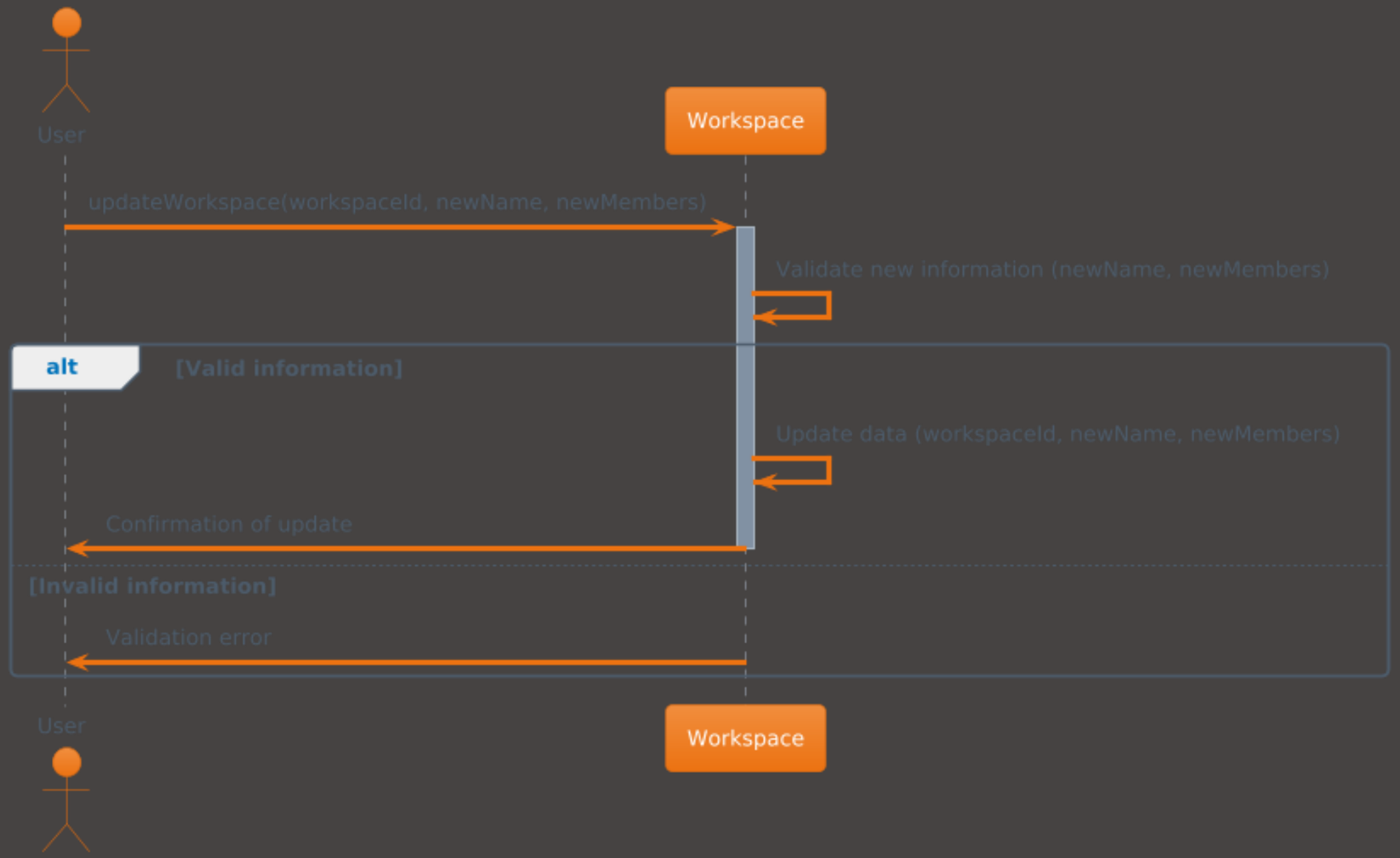
uml Diagrams

Delete page



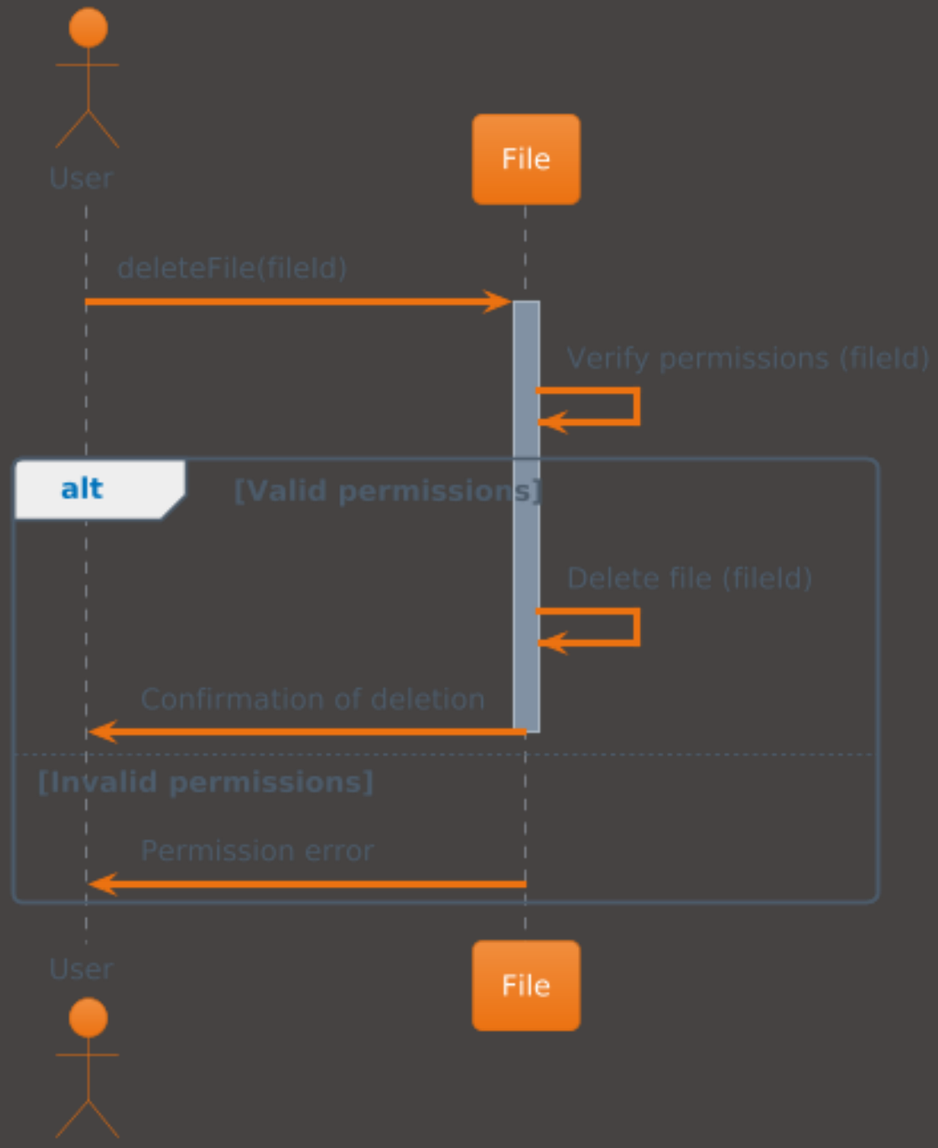
uml Diagrams

Update workspace



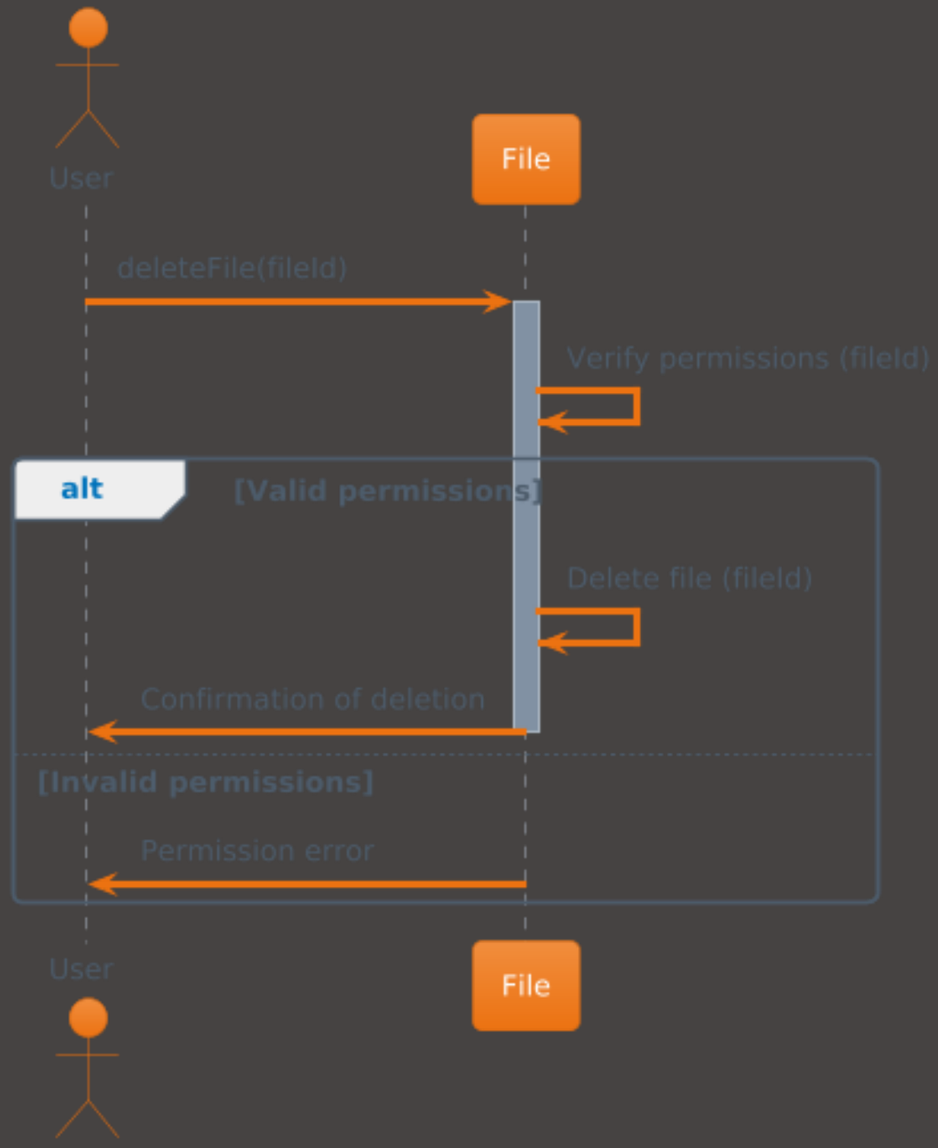
uml Diagrams

Delete file



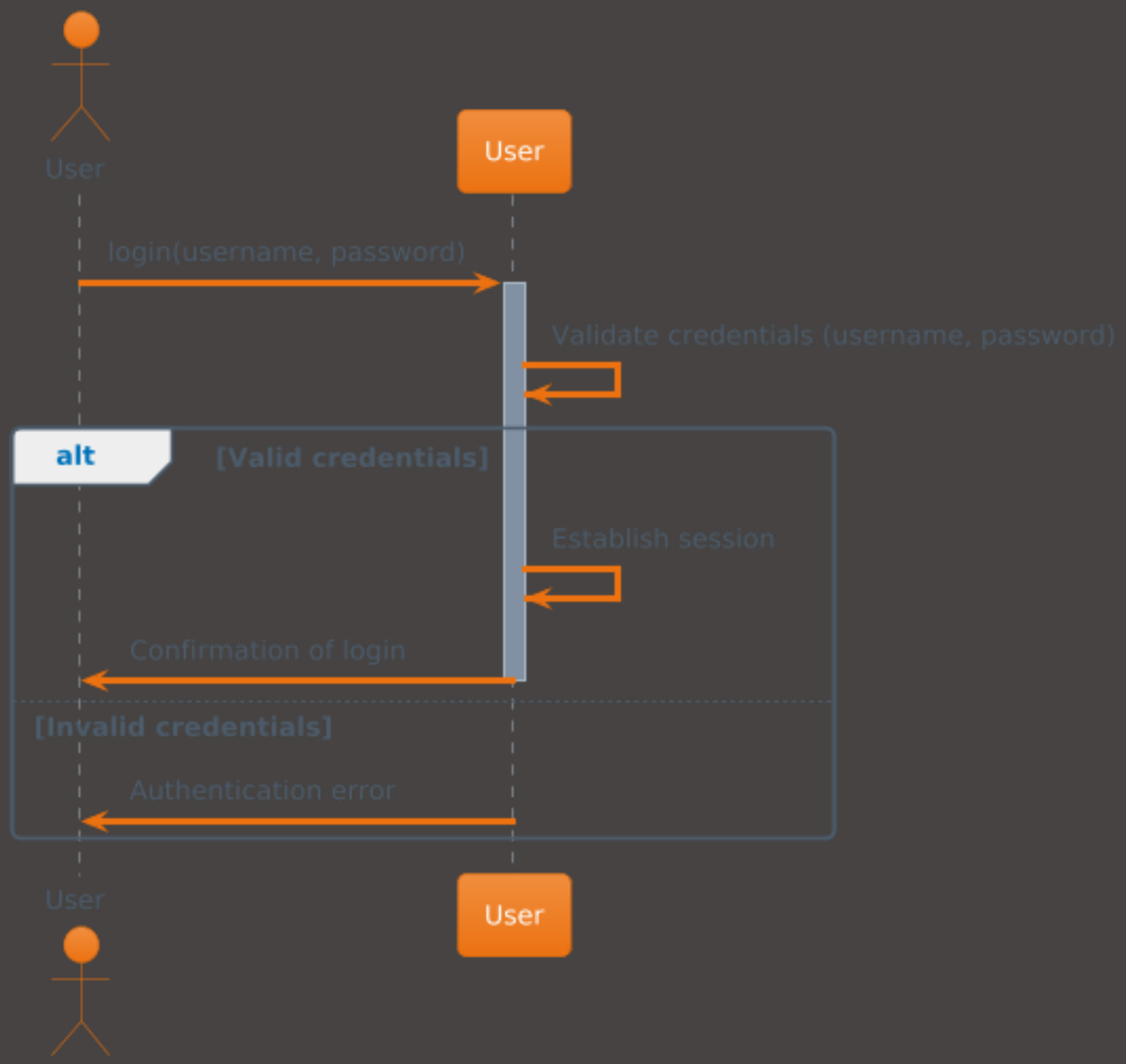
uml Diagrams

Update database

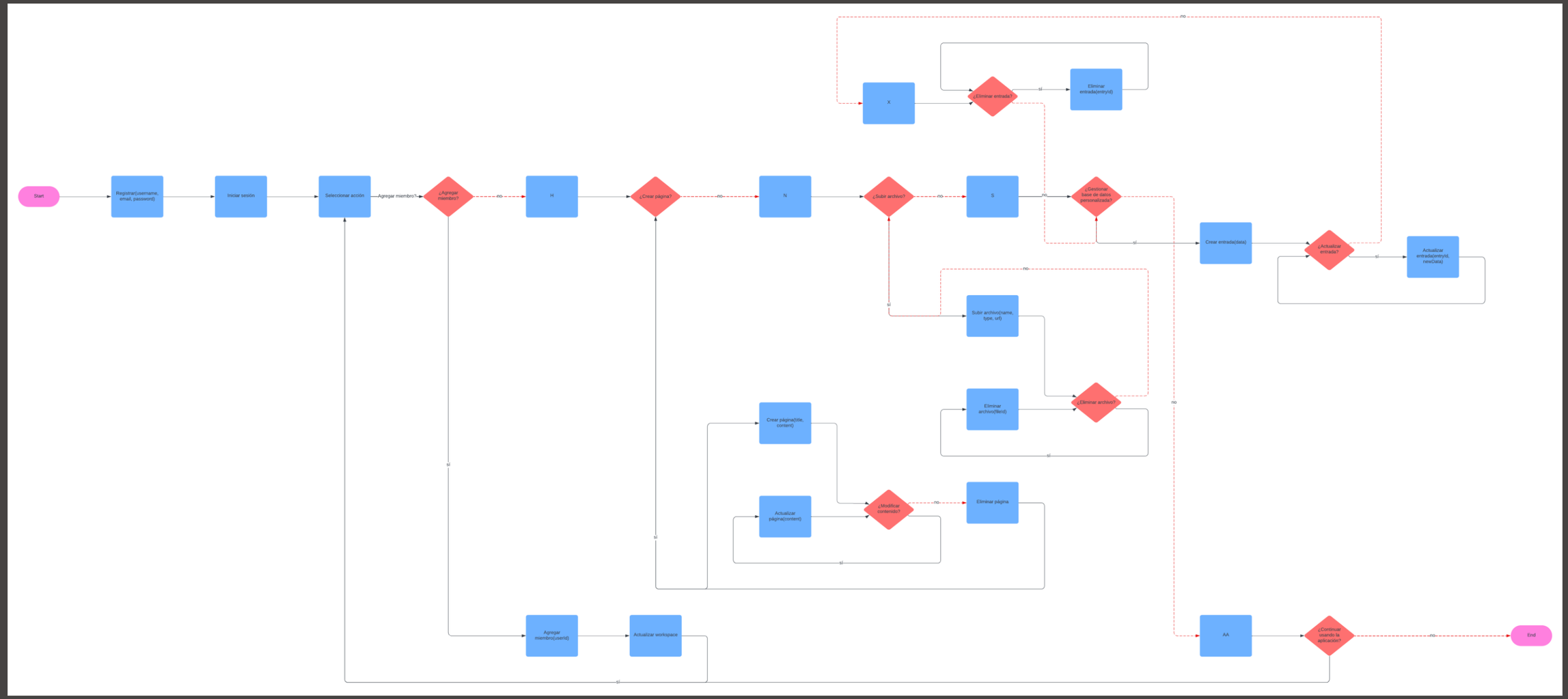


uml Diagrams

Login



Activity diagram



Minimum Features

1

Registration and login

User registration and secure login.

2

Pages and Search

Creation, editing and search of information.

3

Permissions and Statistics

Permission management and statistics display.

Conclusion

1

Summary

Significant advances in the development of the project.

2

Achievements

Definition of objectives, functionalities and design.

3

Next Steps

Development of minimum functionalities and improvements.

```java
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.Scanner;

public class MainApp {
    public static void main(String[] args) {
        // Using try-with-resources to automatically close the Scanner
        try (Scanner scanner = new Scanner(System.in)) {
            // Welcome message
            System.out.println("Welcome to the console application.");
            // Ask the user if they want to register or log in
            System.out.println("What would you like to do? (1) Register, (2) Login");
            int option = scanner.nextInt();  // Capture the user's choice (1 or 2)
            scanner.nextLine();  // Clear the input buffer after reading the integer

            // Prompt the user to enter their username
            System.out.print("Enter your username: ");
            String username = scanner.nextLine();

            // Prompt the user to enter their password
            System.out.print("Enter your password: ");
            String password = scanner.nextLine();

            // Determine the command to execute based on the chosen option
            String command = option == 1 ? "register" : "login";

            // Build the command to execute the Python script with the given argument
            String[] commandArray = {"python3", "user_management.py", command, usern
            Process process = Runtime.getRuntime().exec(commandArray);  // Execute t

            // Read and print the output from the executed process (the Python script
            BufferedReader reader = new BufferedReader(new InputStreamReader(process.
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);  // Print each line of output from the scr
            }

            // Wait for the process to finish
            process.waitFor();
        } catch (Exception e) {
            // Display an error message if something goes wrong with the process
            System.out.println("Error interacting with the authentication system.");
```