

System Analysis:

User Requirements:

- can create new account
- can login to his personal account using the email address and the password
- can search for the author of the text
- can view his own profile
- can update his own profile
- can view the history of his search
- can save some of name authors to favorites
- can delete any of his search

System Requirements:

User:

-can create new account:

User input: email, password, age, username Output: open home page

Situation: show error message if the account is already existing

-can login to his personal account using the email address and the password User input: email, password

Output: open home page

Situation: show error message in case of wrong email or password

-can search for the author of the text User input: text

Output: the name of the author and the accuracy

-can view his own profile System input: user ID Output: view profile

-can update his own profile

User input: email, password, age, username

Output: show message which prompts the data updated successfully

Situation: show error message if the new data's pattern does not match the current pattern. so, data will not be changed.

-can view the history of his search System input: user ID

Output: view the history of his search page

-can save some of the name authors to favorites System input: user ID, author ID

Output: show message which prompts the name of the author saved successfully

-can delete any of his search System input: search ID

Output: show message which prompts the search deleted successfully

Non-Functional Requirements

-Easy to search the information about the author.

-The web page load time should not exceed 1 second.

-Any operation that will be done by any system user will not exceed 1 second.

-Each modification operation requires user password.

-To use the system, it is a must to have an account and be logged in / registered.

- Password encryption to enhance privacy and security.
- Show fake password in password field while login process instead of the one inputted by user.
- There is no way to use fake accounts. As it is required to check email to get the personal login ID.
- Each password must have a pattern to follow (At least 8 characters, a mixture of both uppercase and lowercase letters, a mixture of letters and numbers, inclusion of at least one special character and don't have <, >, ', ").
- Each ID must be unique.
- Each class must have only one instance.

Non-Functional Requirements

- Easy to search the information about the author.
- The web page load time should not exceed 1 second.
- Any operation that will be done by any system user will not exceed 1 second.
- Each modification operation requires user password.
- To use the system, it is a must to have an account and be logged in / registered.
- Password encryption to enhance privacy and security.
- Show fake password in password field while login process instead of the one inputted by user.
- There is no way to use fake accounts. As it is required to check email to get the personal login ID.
- Each password must have a pattern to follow (At least 8 characters, a mixture of both uppercase and lowercase letters,

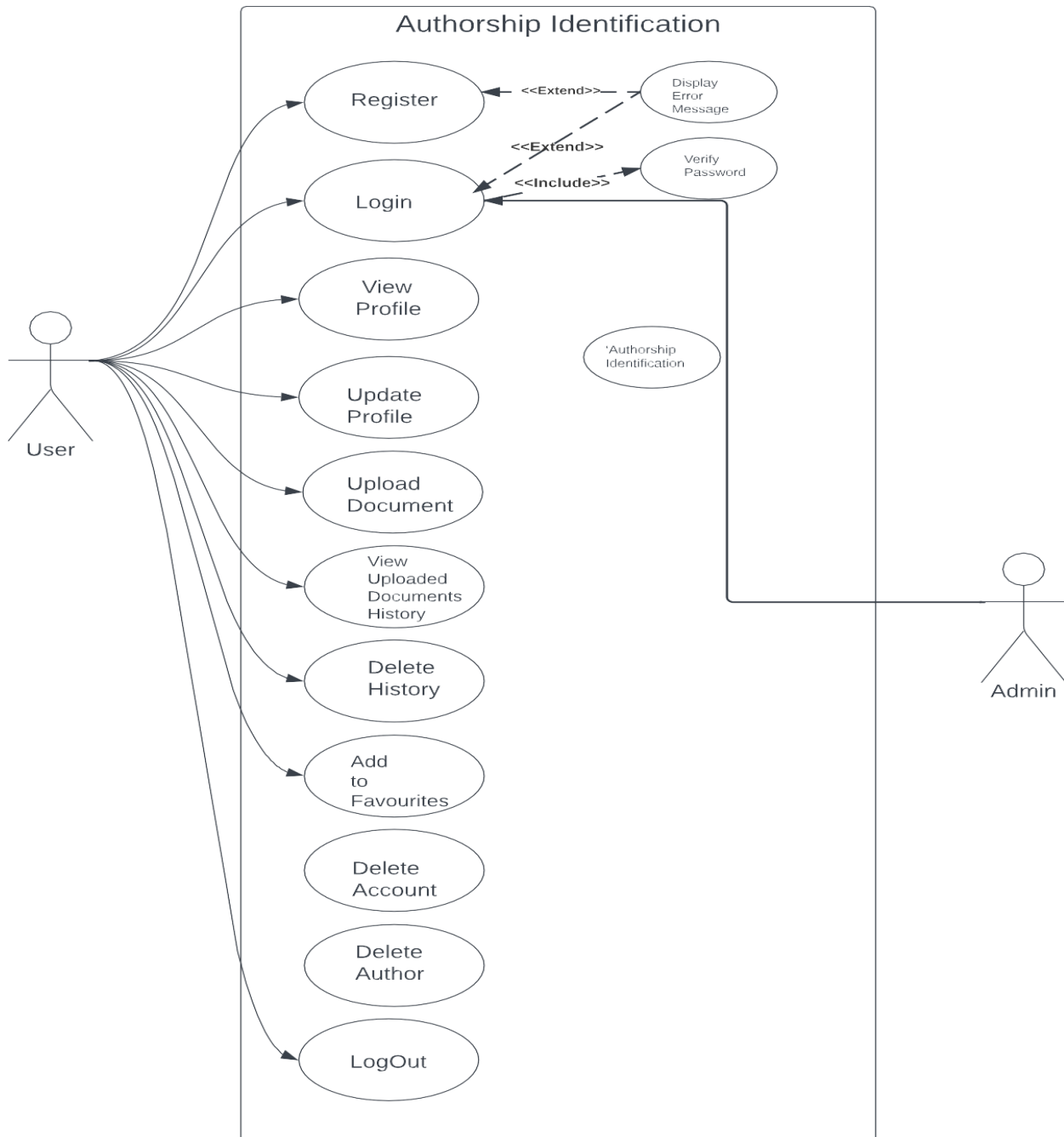
a mixture of letters and numbers, inclusion of at least one special character and don't have <, >, ', ").

- Each ID must be unique.

- Each class must have only one instance.

System Design:

Use Case Diagram



Use Case Scenario:

Use case name		Register
Actors		User
Pre-conditions		User is not logged in to the system and goes to the login page.
Normal flow	Description	1. The System prompts user for registration information, Username, password, etc. 2. The user enters in their information.
	Post-conditions	System verifies information and creates account and is returned to the home page as a logged in user.
Alternative flow and exceptions		1. User clicks submit after entering information system asked for. 2. System displays information with appropriate message to correct invalid information. 3. User re-enters information.
Non-functional requirements		Registration process must be secure and fast.

Use case name		Login
Actors		Admin, User.
Pre-conditions		Login page is opened.
Normal flow	Description	The actor will open home page, and login if the password is verified.
	Post-conditions	The actor can do all his operations.
Alternative flow and exceptions		Fail to login and stay in the Login page.
Non-functional requirements		Login process must be secure and fast.

Use case name		View profile
Actors		User.
Pre-conditions		User must be logged in.
Normal flow	Description	The actor will click the profile button, the profile data page will be opened and can edit or update his/her data and password.
	Post-conditions	Data must be saved in the database if changed.
Alternative flow and exceptions		Fail to logout as an operation is working, still in current page.
Non-functional requirements		Data must be secure.

Use case name		Update profile
Actors		User.
Pre-conditions		User must be logged in.
Normal flow	Description	The actor will click the profile button, the profile data page will be opened and can edit or update his/her data and password.
	Post-conditions	Data must be saved in the database if changed.
Alternative flow and exceptions		Fail to logout as an operation is working, still in current page.
Non-functional requirements		Data must be secure.

Use case name		Upload Document
Actors		User.
Pre-conditions		User must be logged in.
Normal flow	Description	The actor will click the upload pdf button, the pdfs on his own device will be opened so he chooses and it will be uploaded.
	Post-conditions	Data must be saved in the database if changed.
Alternative flow and exceptions		Fail to logout as an operation is working, still in current page.
Non-functional requirements		Data must be secure.

Use case name		View Uploaded Documents History
Actors		User.
Pre-conditions		User must be logged in.
Normal flow	Description	The actor will click the View History button, the page including all the uploaded pdfs will be opened.
	Post-conditions	Data must be saved in the database if changed.
Alternative flow and exceptions		Fail to logout as an operation is working, still in current page.
Non-functional requirements		Data must be secure.

Use case name		Delete Documents History
Actors		User.
Pre-conditions		User must be logged in.
Normal flow	Description	The actor will click the Delete History button, All the history will be deleted ,the page including all the uploaded pdfs will be opened and shown empty.
	Post-conditions	Data must be saved in the database if changed.
Alternative flow and exceptions		Fail to logout as an operation is working, still in current page.
Non-functional requirements		Data must be secure.

Use case name		View Uploaded Documents History
Actors		User.
Pre-conditions		The Uploaded Documents History page must be opened.
Normal flow	Description	The actor will click the Star button, the page including all the added to favorites will be opened.
	Post-conditions	Data must be saved in the database if changed.
Alternative flow and exceptions		Fail to logout as an operation is working, still in current page.
Non-functional requirements		Data must be secure.

Use case name		Delete Author And His Books
Actors		Admin
Pre-conditions		Admin logged in successfully.
Normal flow	Description	After logging in, the admin dashboard will be shown and (s)he can select the operation (s)he wants from the buttons which are already exist in the dashboard and do the required operation.
	Post-conditions	The changes must be saved in the database and the admin will return to the dashboard to be able to select another operation to do or logout.
Alternative flow and exceptions		Operation failed, show error message.
Non-functional requirements		All operations' processes must be secure and fast.

Use case name		Delete User Account
Actors		Admin
Pre-conditions		Admin logged in successfully.
Normal flow	Description	After logging in, the admin dashboard will be shown and (s)he can select the operation (s)he wants from the buttons which are already exist in the dashboard and do the required operation.
	Post-conditions	The changes must be saved in the database and the admin will return to the dashboard to be able to select another operation to do or logout.
Alternative flow and exceptions		Operation failed, show error message.
Non-functional requirements		All operations' processes must be secure and fast.

Use case name		Logout
Actors		Admin, User.
Pre-conditions		Actor's dashboard is opened.
Normal flow	Description	The actor will click the logout button, and the home page will be opened instead of current one.
	Post-conditions	The actor cannot do any operation until (s)he login again.
Alternative flow and exceptions		Fail to logout as an operation is working, still in current page.
Non-functional requirements		Logout process must be secure and data are safe.

Class Diagram

