UC 1.4 Hashing Passwords

|  |  |
| --- | --- |
| Name | Hashing Passwords |
| Summary | System securely encrypts user passwords through a hashing process before storing them in the database |
| Rationale | Storing plain-text passwords is a security risk. Hashing passwords ensures that even if there's a data breach, the actual passwords remain confidential and secure. It protects user credentials and reinforces user trust in the platform. |
| Actors | System, User |
| Precondition | UC 1.5 Log in Page |
| Elements | 1. Password    1. String input provided by the user that meets password criteria 2. Hashing algorithm    1. Secure algorithm that the system uses to transform the password 3. Salt    1. Randomly generated data that is used to further increase security |

Ideal Flow:

|  |  |
| --- | --- |
| Step | Action |
| 1 | User provides a password during registration or password change/update process |
| 2 | System generates a salt for the password |
| 3 | System combines the user’s password with the salt by appending it |
| 4 | System processes the salted password through the hashing algorithm |
| 5 | Hashed password and salt are stored in the database |

Alternative Flow 1:

|  |  |
| --- | --- |
| Step | Action |
| 1 | User provides password during login attempt |
| 2 | System retrieves the hashed password and salt associated with the user from the database |
| 3 | System combines the provided password with the salt and hashes it |
| 4 | System compares the provided password with the stored password |
| 5 | If they match, the user is granted access, otherwise system shows an incorrect password error is shown |