Use Case: Authenticate Actor

## Brief Description

The goal of this use case is to authenticate an actor based on the username and password entered, which is checked against a database and if successfully validated, the actor will be able to access the website. When logged in, the actor will have access according to his or her credential.

# Included Use Cases

N/A

# Preconditions

An account exists for the actor.

# Flow of Events

## Basic Flow – The actor successfully logs into the website and can access it according to his or her access level

1. This use case begins when the user accesses the login page.
2. The actor enters in a username and password (see data requirement).
3. The system checks to make sure no fields are empty.
4. The Authentication Service checks the username and password against the database.
5. The Authentication Service checks what permissions are available to the user (see data requirement).
6. The validation is successful and the user can access the website according to his or her access level.
7. The use case ends.

## Alternative Flows

None

## Exception Flows

E1: There is missing information in the fields.

1. This flow begins when the actor has left one or more fields blank.
2. The Authentication Service checks to see that all fields contain data.
3. The actor cannot access the website.
4. The use case ends.

E2: The actor enters in the wrong login and/or password.

1. This flow begins when the actor enters in the wrong username and/or password. Reference: basic flow step 2.
2. The Authentication Service checks the username and password against a database to make sure the actor exists.
3. The Authentication Service notifies the actor that it cannot find him or her in the database.
4. The actor cannot access the system.
5. The system prompts the actor to re-enter the username and password.
6. The use case ends.

Basic Flow for 4.1.4

|  |  |
| --- | --- |
| **Field Name** | **Notes** |
| Student access | On the website, the Student is given access to:   1. View live stream 2. View photos 3. Log out |
| Administrator access | On the website, the Administrator is given access to:   1. View live stream 2. View photos 3. Log out 4. Create class password 5. Change photo capture increment 6. Delete student account 7. View sign-in log 8. Modify photo database |

# Requirements

### Security Requirements

1. The password shall never be viewable upon sign-in.
2. An actor shall only have one type of access, either student or administrator level, never both.
3. Students shall only be given student level access.
4. Administrators shall be given administrator level access.

### Scalability Requirements

1. Multiple actors shall be able to log in and be verified simultaneously.

### User Interface Requirements

1. The home page will only have two text boxes. One to type in the username and the other one to type in the password.
2. The home page will have a link/button to create an account if the actor does not have a valid account.

# Post-conditions

1. The system sends the actor to the home page of the website after the Basic Flow. Access of the website is given according to the access level.
2. The system stays on the login page of the website after the Exception Flow. The actor may re-enter the username and password.

# Notes & Open Issues

Q1: Should there be an option to show the password in case the actor wants to see what has been typed?

Q2: Should there be a limit of retries to log in the system?

A2: There are no limits of retries

Q3: Will there by a success message or should the actor be redirected to the home page?

A3: No success message, just redirect to the home page.

# Out of Scope (Future Functionality)

None

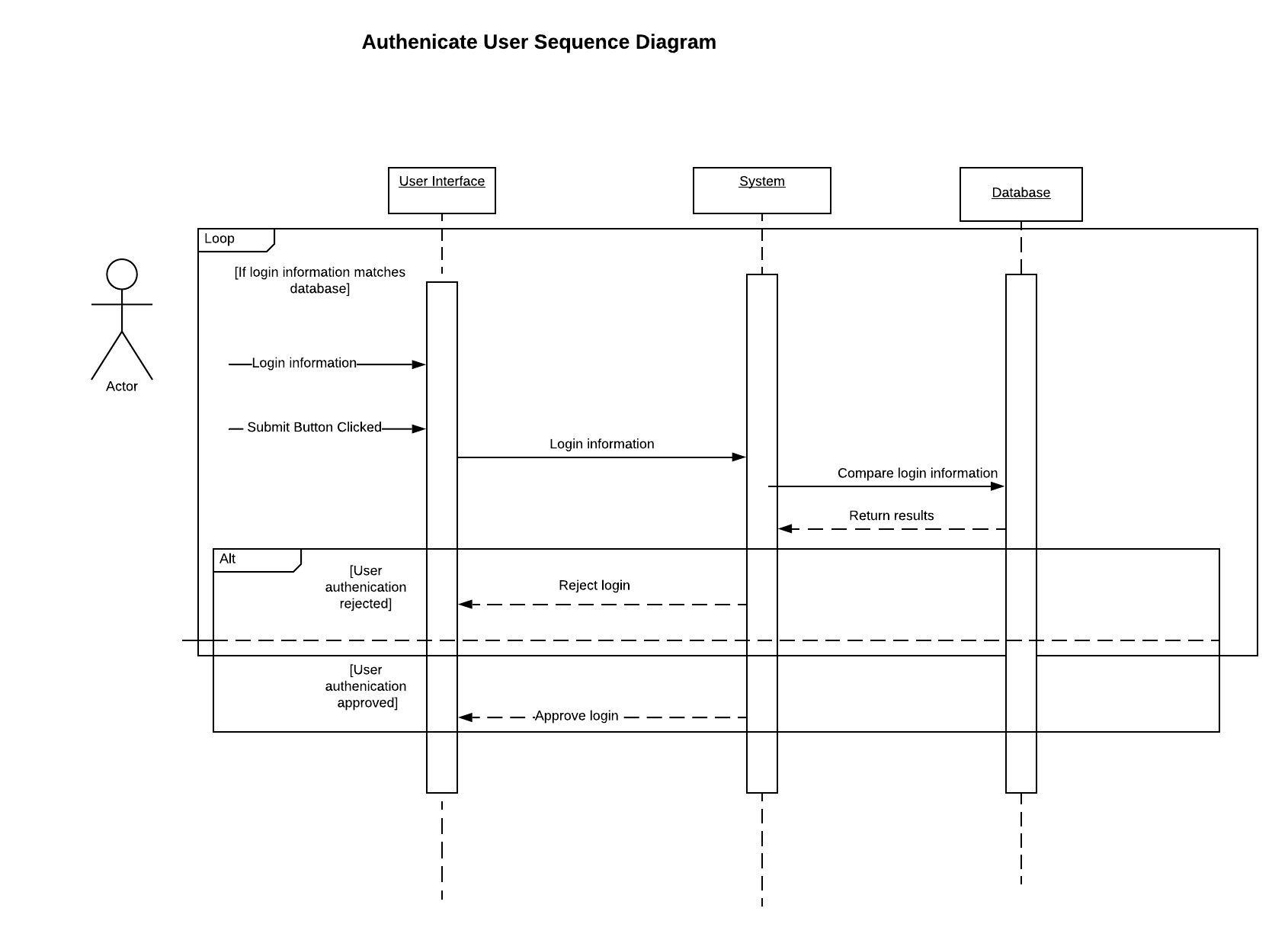
# Appendix A – Data Definition for Login information required

| **Field Name** | **Description** | **Type** | **Valid Values** | **Default** | **Business Rules** |
| --- | --- | --- | --- | --- | --- |
| Username | User name | Text | N/A | N/A | Unique user name |
| Password | Password | bcrypt hash | N/A | N/A | Hash code |

# Appendix B: State diagram

# 

# Appendix C: Sequence Diagram

****