SCHOOL OF INFORMATION, COMPUTER AND COMMUNICATION TECHNOLOGY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY

Final Report

QikVid - Video Processing and Distributing Application

Group Members

6222770313	Thanyachanok Rachavongsuk
6222780379	Paphana Yiwsiw
6222782425	Kawiya Pholjaroen
6222790345	Time Kitilimtrakul
6522808210	Levin Kaus

Present to

Dr. Apichon Witayangkurn

DES424 Cloud-based Application Development
Semester 1 Academic Year 2022 Digital Engineering (DE)

Table of Contents

Overview and Background

Current video-sharing platforms like TikTok have closed source code that does not allow the public to check the efficiency or safety of the platform. Due to the nature of these video-sharing platforms, we believe that an open-source approach would be more appropriate as the platform will serve as an online public space for people to share their videos and therefore, should have access to the source code as well.

Furthermore, there have been multiple criticisms about video recommendation algorithms and the invasion of users' privacy. Although this feature can be helpful, there have been multiple instances where companies are found to sell user data to other organizations. In a world where privacy is becoming a priority concern, better platform implementations should be created.

Lastly, these video-sharing platforms tend to have intrusive amounts of advertisements that are usually not related to the user's demand. Our aim is to create a nonprofit free online video sharing platform that people can use in a safe online environment.

Problem Statement

Our goal is to create a free open-source cloud-based application that allows users to upload videos on a cloud server and have access to them later to view, edit, or delete them. This involves establishing efficient video encoding algorithms and cloud server optimization to effectively provide the service to the users.

Key Users and Stakeholders

1. Users
    a. Video Uploader - Content creators and people who love to make short videos
    b. Video Viewer - People who enjoy watching the videos uploaded by other users
2. Admin
    a. Content moderators - Staff who oversee the content on the application.

Functional requirements

- User Management System
  - Admins can view and deactivate user accounts on the application
- Videos get distributed to each user randomly and play one at a time
- Each video has a like counter, that increases by one every time a user clicks on it

Nonfunctional requirements

- User Interface
  - Time-Zone: Bangkok (UTC+7)
  - Input/Output devices: Desktop, Laptop PC, Mobile
  - Handle multiple users simultaneously
  - Web application platform
- Performance
  - Video size limit: no longer than 5 minutes and no more than 250 megabytes.
  - Response time must be less than 5 seconds
- System Interface
  - Inputs (APIs): Media Services API v3 - Microsoft Azure
  - Outputs: MP4
- Quality
  - Runs 24/7 without interruption
- Backup
  - There is at least one video backup location
    - The backup of the videos is the original uploaded video
    - The videos on display are the videos that have been encoded
  - Admins are responsible for backing up server data
- Security
  - Only users who upload the video have access to delete the video
  - Admins can view/delete all videos
  - Access Control - Admin can activate or deactivate each registered user

Programming Language

- Javascript
- HTML
- CSS

Cloud Technology and Components

In this project, we select the cloud provider from Microsoft Azure to implement the cloud technology and resources in our video processing and distributing application.

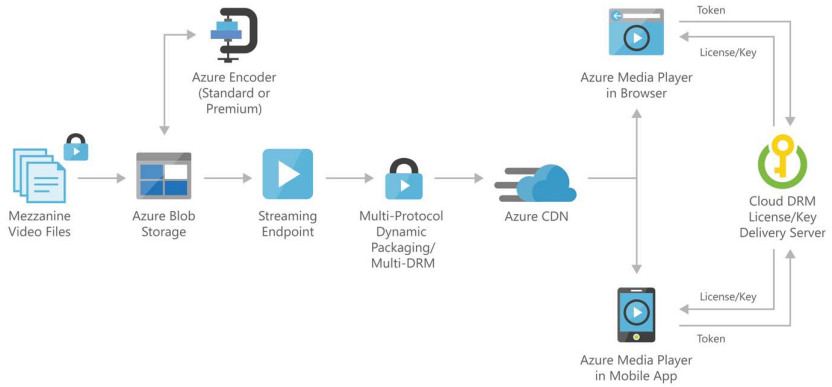
- Web application deployment and User authentication: Azure App Service, Static Web Apps.

  Azure App is a service provided by Microsoft Azure for building and hosting web applications and RESTful API with support for Node.js and other languages. Azure Static Web Apps also capable of hosting web applications with support of automatically configuring CI/CD.
- Databases and storage: Azure Cosmos DB and Azure Blob Storage

  Azure Cosmos DB is used to handle the data generated by users such as likes count, while Azure Blob Storage is used to store large volumes of unstructured data for cloud applications.
- Video streaming services: Azure Media Service Encoder, Streaming Endpoint, and Azure Media Player

  Azure Media Service provides video encoding and streaming services to serve video distributing applications. Azure Media Player is used to provide playback and JavaScript API support.
- Video distributing: Content Delivery Network

  Content Delivery Network or CDN provides a reliable method of delivery of video via a web application platform.



Architecture of the video streaming and distributing application
(Reference: Video-on-demand digital media, Microsoft Azure)

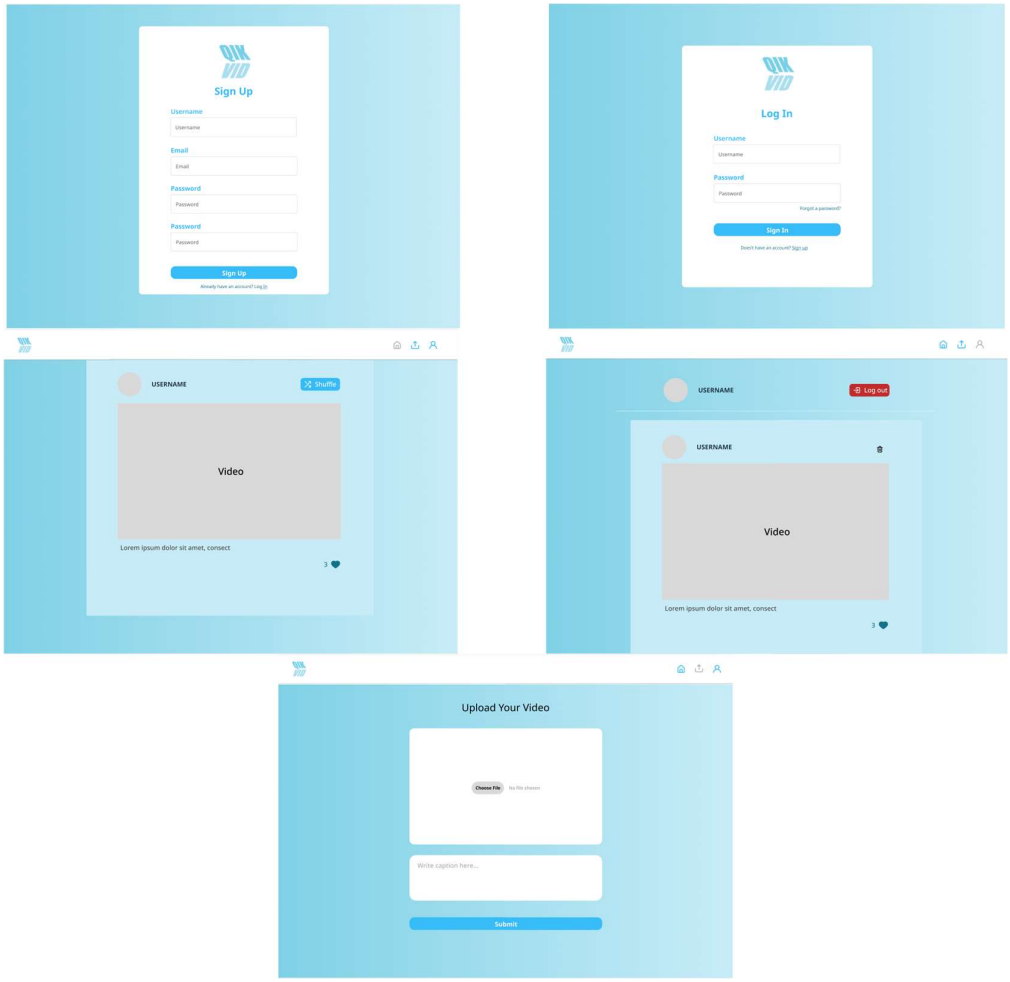
Framework and Tools

- Source and version control
  - Git
  - GitHub
- Front-end Development
  - React
  - Tailwind CSS
- Back-end Development
  - Node.js
  - Express.js
- Databases and Storage
  - Azure Cosmos DB for MongoDB to store user and video data
  - Azure Blob Storage to store uploaded and encoded video
- Software Testing
  - Robot Framework
  - Selenium Library

Prototype

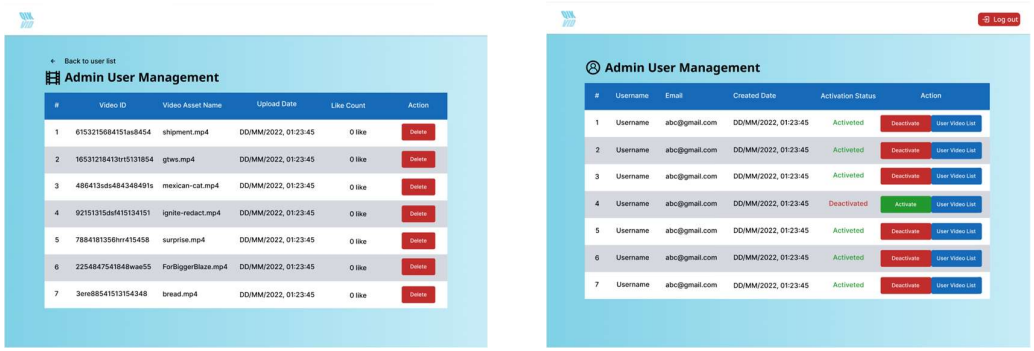
User Application Interface

User interface consists of sign up, log in, video player, user profile, and video upload page.



Admin Application Interface

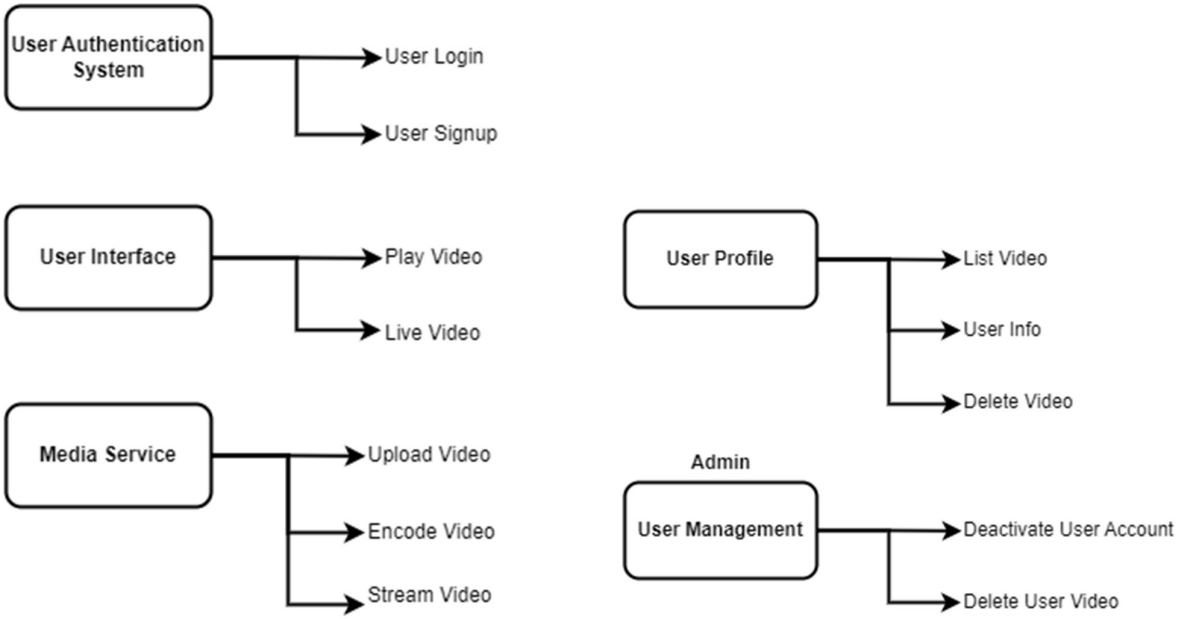
Admin interface consists of user management and video management page.



Software architecture

Subsystem decomposition

The diagram below illustrates the different subsystems implemented into QikVid.



Data Management

Persistent data

- Uploaded & Encoded video files are stored in Azure Blob Storage
- User Information and Video Information are stored in Azure Cosmos DB for MongoDB

Boundary Condition

Initialization

When a user accesses the web app, the application sends a HTTP request (GET /fetchVideo) from the server. This request then retrieves a video which then displays a loading element to the screen.

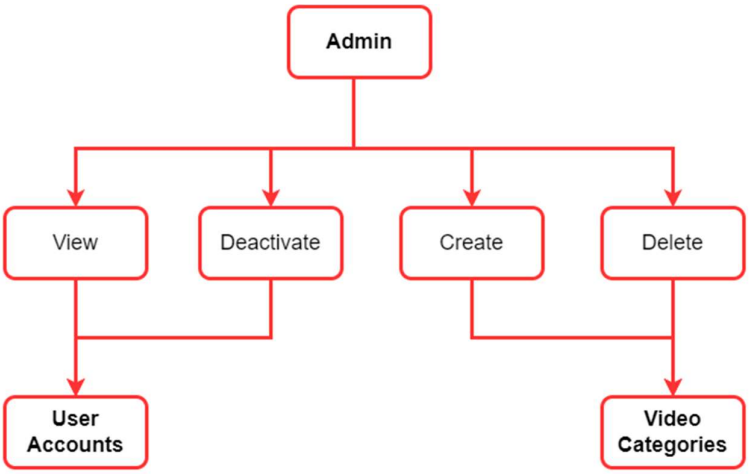
Termination

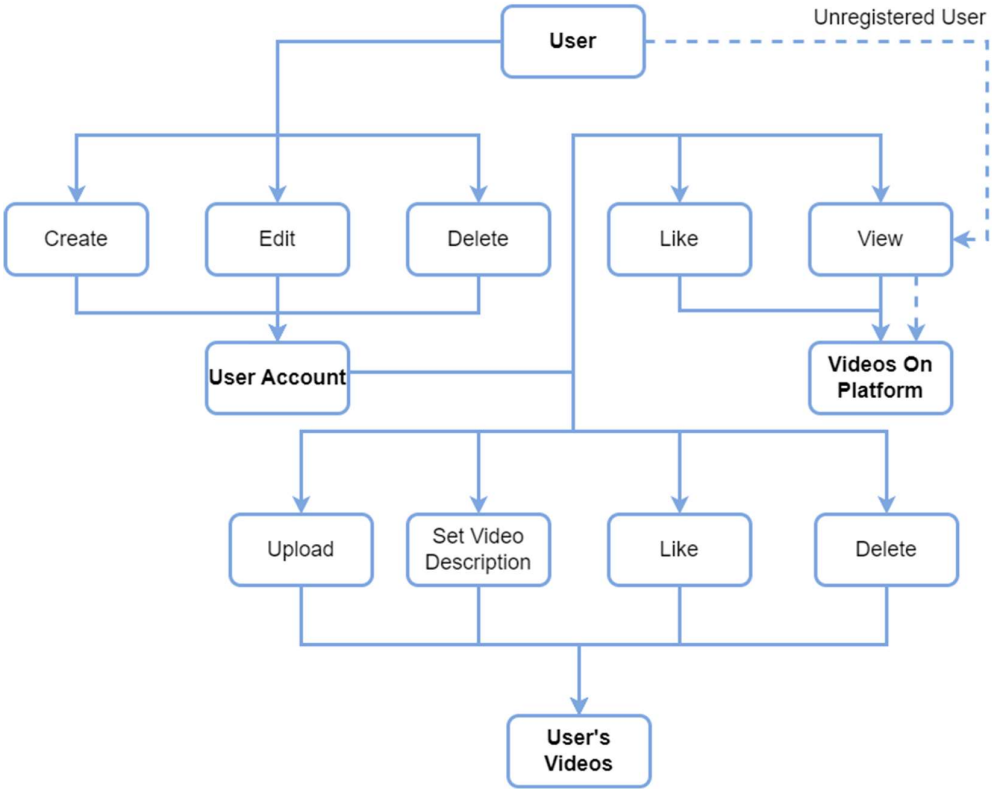
For our system, there is no single subsystem that is allowed to be terminated. Termination only occurs when a User logs out.

Failure

When an error occurs, an alert popup window is displayed on the user interface. If the application fails, the system will try to restart. In addition, a notification will be sent to the system administrator in line with the Azure monitor rule.

Access Control & Security

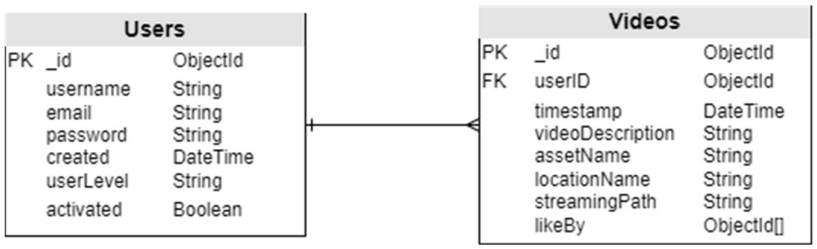




Detailed Design

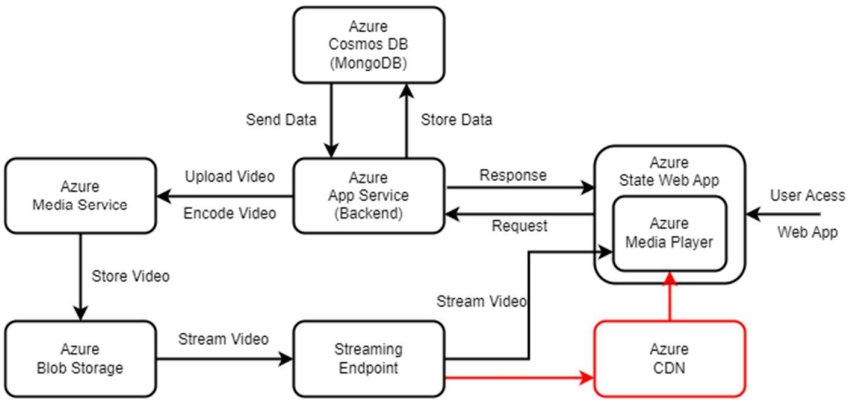
ER diagram

In this section, use mongoDB for our database system which is a non-relational database. The equivalent of an ER diagram in a non-relational database is shown on the following diagram.



- In the Users collection, _id is an ObjectId which is defines userID of each user
- In the Videos collection userID is used as the foreign key to define which user upload that video
- Note that only hashed passwords are stored, not plain text passwords.

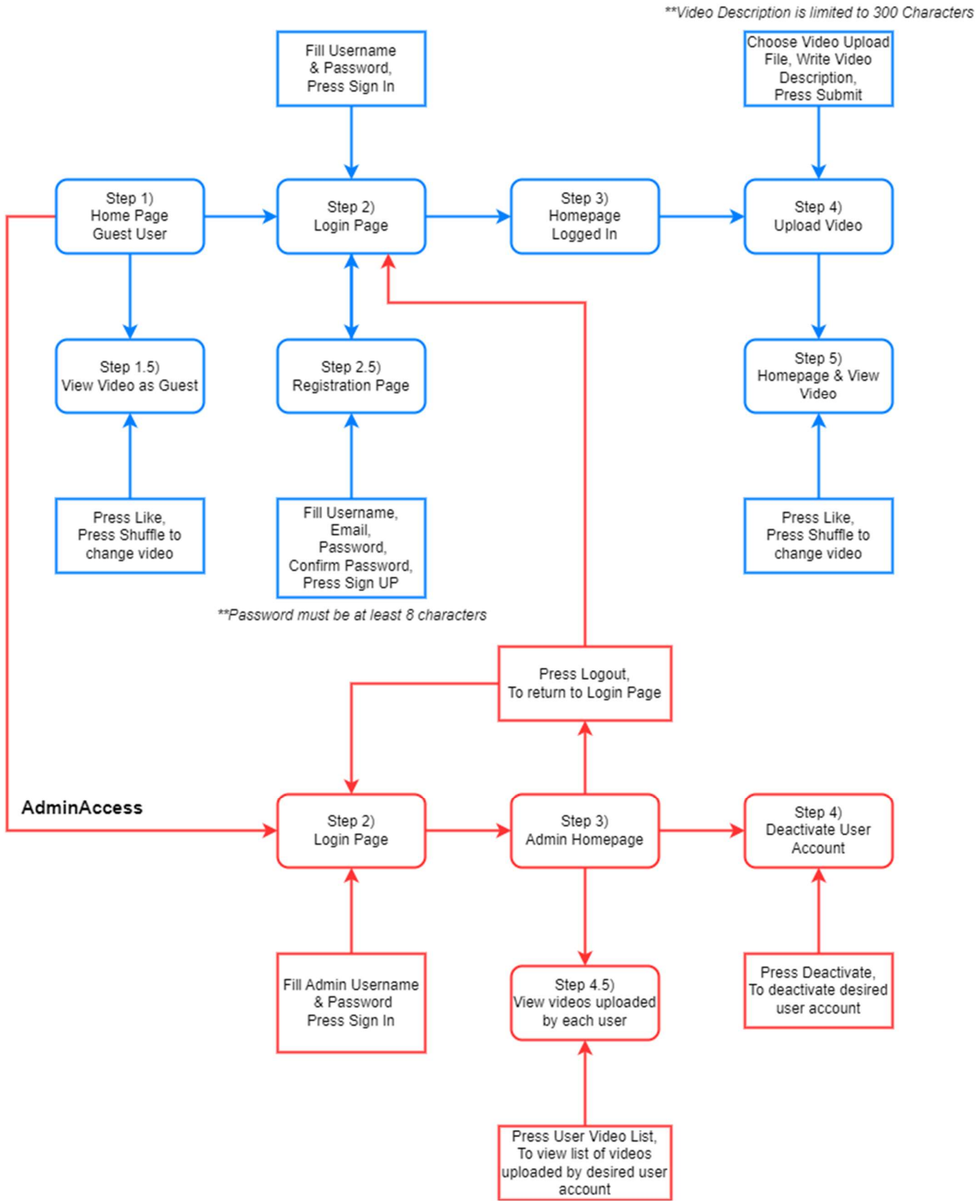
System Architecture Diagram with Cloud



The figure above is an illustration of Qikvid's cloud system architecture. The azure static web app is used to host QikVid user applications. The request from the user application is sent to the API which is hosted on Azure App Service. When a user requests the data, such as user information or video information, the data is stored and retrieved from Azure CosmosDB for MongoDB. When a user uploads the video file, the original video file is stored on Azure Blob Storage and the video encoding job is created on Azure Media service. The encoded video file is then stored in Azure Blob Storage. When the user requests to play video, the streaming endpoint is used to deliver the video from Blob storage to Azure Media Player on the user application.

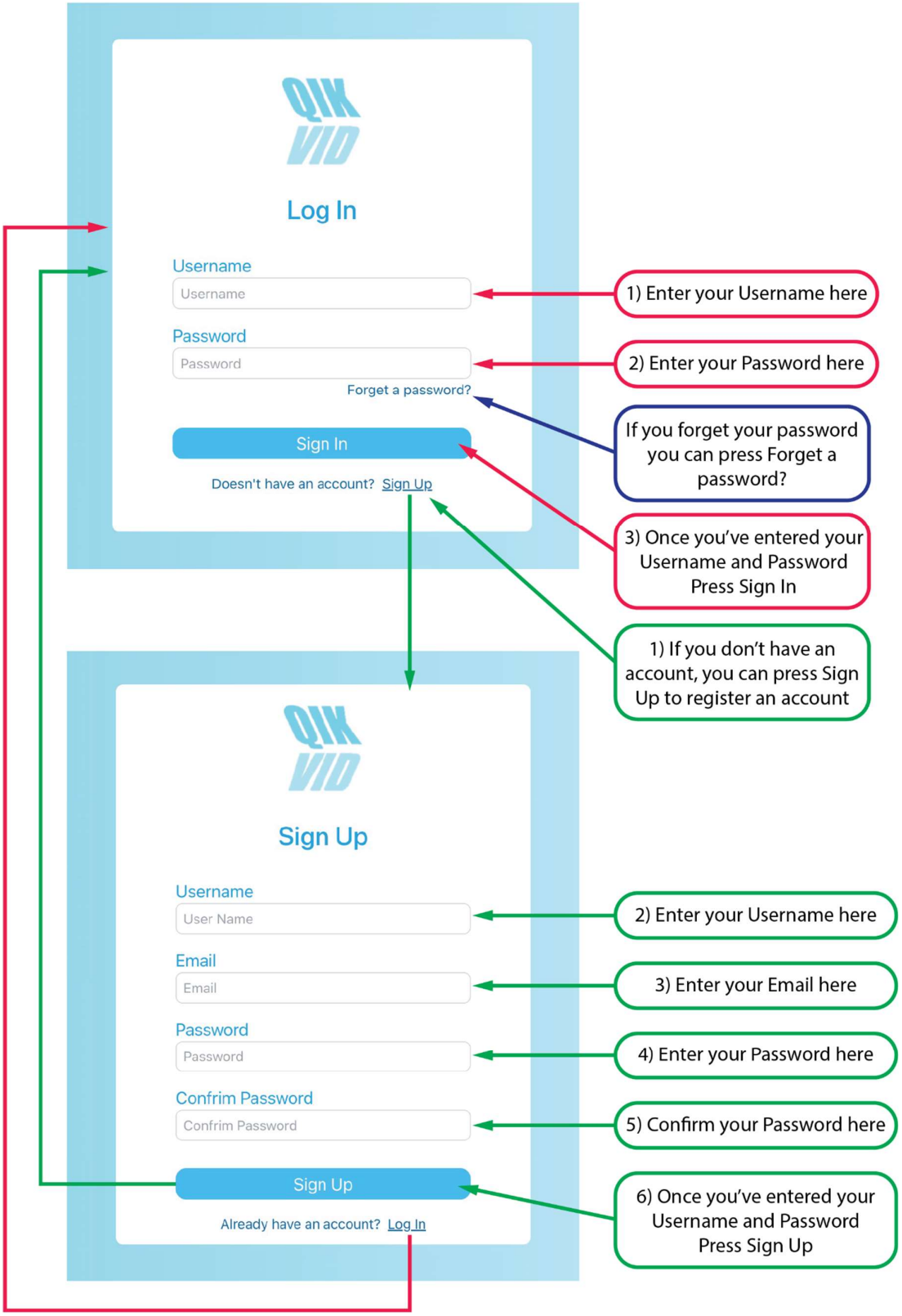
This process can be improved by implementing Azure CDN to deliver content from streaming endpoints using a content delivery network (CDN). However, due to the limitation of Azure Student Subscription which prevents the use of Azure CDN, the CDN is not implemented.

Workflow of the system

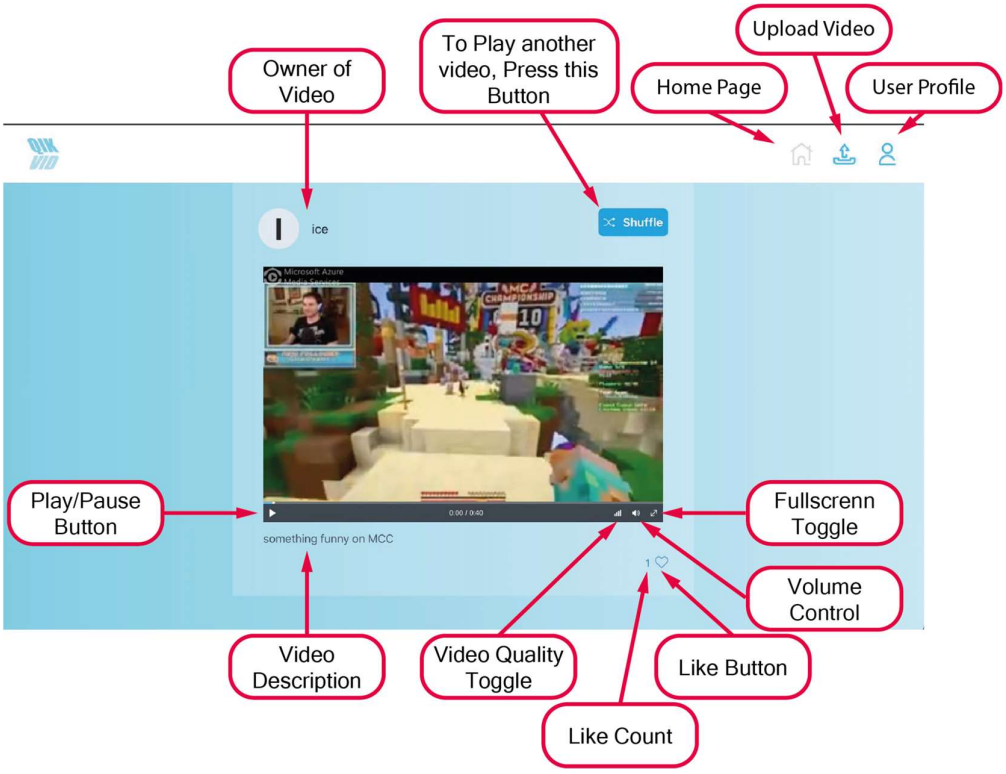


User Manual

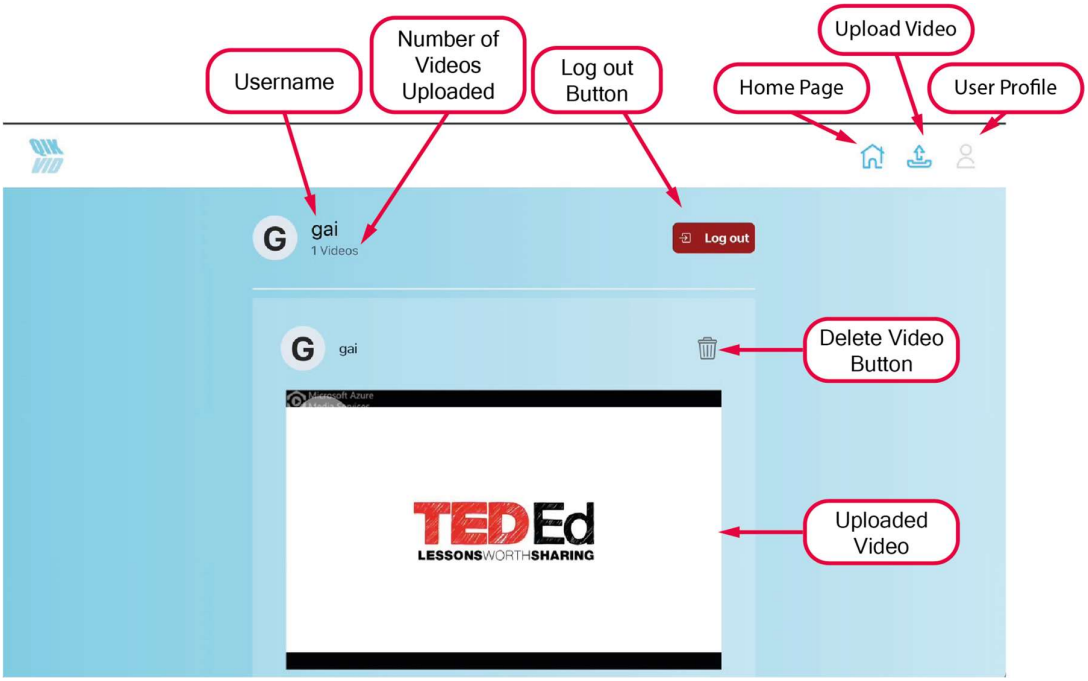
Login Page



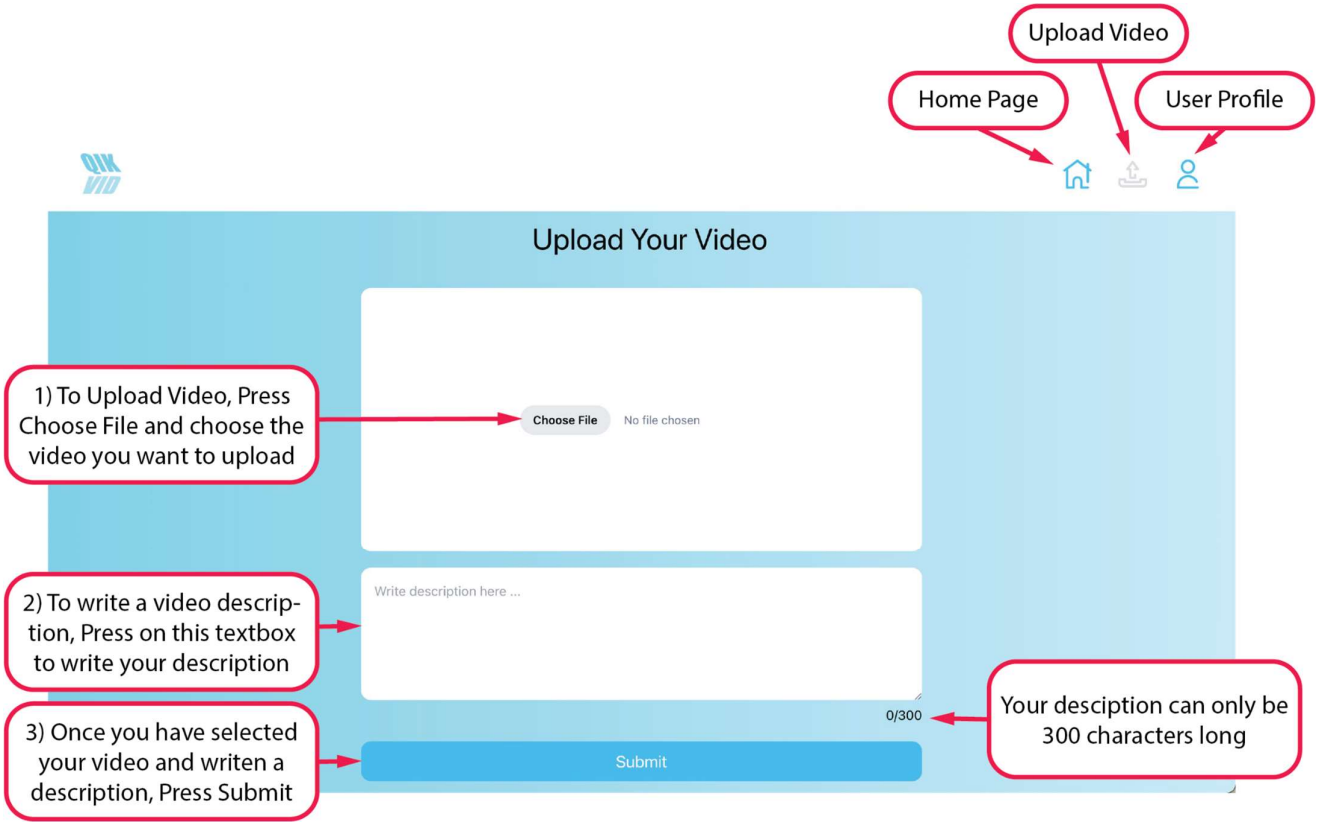
Home Page



User Profile Page



Upload Video Page



Admin - User Management Page

#	Username	Email	Created Date	Activation Status	Action	
1	fuio	fuio@g.com	13/11/2022, 22:06:19	Activated	Deactivate	User Video List
2	test1	test1@gmail.com	13/11/2022, 21:22:23	Activated	Deactivate	User Video List
3	ice	ice@gmail.com	13/11/2022, 21:02:53	Activated	Deactivate	User Video List
4	levin	levin@levin.com	13/11/2022, 18:28:31	Activated	Deactivate	User Video List
5	rarbash	guy@gmail.com	13/11/2022, 01:59:58	Activated	Deactivate	User Video List
6	p8	p8@gmail.com	13/11/2022, 01:14:15	Activated	Deactivate	User Video List
7	wtwtw	aaa@gmail.com	12/11/2022, 16:36:10	Activated	Deactivate	User Video List
8	q	q@gmail.com	12/11/2022, 07:00:00	Deactivated	Activate	User Video List
9	1	1@1.co	12/11/2022, 07:00:00	Activated	Deactivate	User Video List
10	p7	p7@gmail.com	12/11/2022, 07:00:00	Activated	Deactivate	User Video List

#	Video ID	Video Asset Name	Upload Date	Like Count	Action
1	6371039a338d0e09f91a27f3	shipment.mp4	13/11/2022, 21:47:54	0 like	Delete
2	636ff071e01bd24c239fddc3	gtws.mp4	13/11/2022, 02:13:53	0 like	Delete
3	636fd4b965cd1e1ec5e4edbe	mexican-cat.mp4	13/11/2022, 00:15:37	0 like	Delete
4	636f6b5d9a6aa512dce2834f	ignite-redact.mp4	12/11/2022, 16:36:10	0 like	Delete
5	636f6b859a6aa512dce28350	surprise.mp4	12/11/2022, 16:36:10	0 like	Delete
6	636f6c4f9a6aa512dce28352	ForBiggerBlaze.mp4	12/11/2022, 16:36:10	0 like	Delete
7	636f69d19a6aa512dce2834e	bread.mp4	12/11/2022, 16:35:09	0 like	Delete

Testing report

The testing report for QikVid application is generated using Robot framework. Testing scripts are created according to the sequence of user's action. Tested functions are user login and logout, user signup, play and like video, and upload and delete video.

Robot Test

Login/Logout - Test Script

```
*** Settings ***
Library    Selenium2Library
Library    XML
*** Variables ***
${BROWSER}    chrome
${URL}    http://localhost:3000
${USERNAME}    ice
${PASSWORD}    12345678
${DELAY}    1.25
*** Test Cases ***
1. Open Browser
   Open Browser    ${URL}    ${BROWSER}
options=add_experimental_option("excludeSwitches", ["enable-logging"])
   Maximize Browser Window
   Set Selenium Speed    ${DELAY}
2. Go to Log In Page
   Click Element    id=nav-user-icon
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Log In
3. Enter Log In Account
   Input Text    id=login-username-input    ${USERNAME}
   Input Password    id=login-password-input    ${PASSWORD}
   Click Button    id=login-signin-btn
4. Go To Profile Page
   Click Element    xpath=//*[@id="nav-user-icon"]
   ${LOGIN_USERNAME}    Get Text    xpath=//h2
   Should Contain    ${LOGIN_USERNAME}    ${USERNAME}
   ${LOGOUT_BUTTON}    Get Text    xpath=//*[@id="user-logout-btn"]/div/p
   Should Contain    ${LOGOUT_BUTTON}    Log out
5. Log out
   Click Element    id=user-logout-btn
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Log In
6. Close Browser
   Close Browser
```

Login/Logout - Test Results

LOG

Testlogin Report

Generated
20221118 16:54:15 UTC+07:00
7 seconds ago

Summary Information

Status:	All tests passed
Start Time:	20221118 16:53:32.424
End Time:	20221118 16:54:15.180
Elapsed Time:	00:00:42.756
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:00:42	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Testlogin	6	6	0	0	00:00:43	

Test Details

All | Tags | Suites | Search

Suite:	Testlogin
Status:	6 tests total, 6 passed, 0 failed, 0 skipped
Start / End Time:	20221118 16:53:32.424 / 20221118 16:54:15.180
Elapsed Time:	00:00:42.756
Log File:	log.html#s1

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
Testlogin. 1. Open Browser			PASS		00:00:06.625	20221118 16:53:32.989 20221118 16:53:39.614
Testlogin. 2. Go to Log In Page			PASS		00:00:05.155	20221118 16:53:39.617 20221118 16:53:44.772
Testlogin. 3. Enter Log In Account			PASS		00:00:14.175	20221118 16:53:44.775 20221118 16:53:58.950
Testlogin. 4. Go To Profile Page			PASS		00:00:07.664	20221118 16:53:58.954 20221118 16:54:06.618
Testlogin. 5. Log out			PASS		00:00:05.110	20221118 16:54:06.620 20221118 16:54:11.730
Testlogin. 6. Close Browser			PASS		00:00:03.445	20221118 16:54:11.733 20221118 16:54:15.178

Signup - Test Script

```
*** Settings ***
Library    SeleniumLibrary
Library    XML
*** Variables ***
${BROWSER}    chrome
${URL}    http://localhost:3000
${USERNAME}    testing2                    # Please change the username every time
you run the script
${PASSWORD}    testing123
${EMAIL}    testing_user2@gmail.com        # Please change the email every time you
run the script
${DELAY}    0.5
*** Test Cases ***
1. Open Browser
   Open Browser    ${URL}    ${BROWSER}
options=add_experimental_option("excludeSwitches", ["enable-logging"])
   Maximize Browser Window
   Set Screenshot Directory    ./testing/screenshot
   Set Selenium Speed    ${DELAY}
2. Go to Log In Page
   Click Element    id=nav-user-icon
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Log In
3. Go to Sign Up Page
   Click Element    xpath=//*[@id="login-signup-hyperlink"]
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Sign Up
4. Fill in Sign Up Form
   Input Text    id=signup-username-input    ${USERNAME}
   Input Text    id=signup-email-input    ${EMAIL}
   Input Password    id=signup-password-input    ${PASSWORD}
   Input Password    id=signup-confrim-password-input    ${PASSWORD}
   Click Button    id=signup-signup-btn
5. Check Alert Text Message
   Alert Should Be Present    timeout=10 s    text=Sign up successfully
6. Check User Name
   Input Text    id=login-username-input    ${USERNAME}
   Input Password    id=login-password-input    ${PASSWORD}
   Click Button    id=login-signin-btn
7. Go To Profile Page
   Click Element    xpath=//*[@id="nav-user-icon"]
   ${LOGIN_USERNAME}    Get Text    xpath=//h2
   Should Contain    ${LOGIN_USERNAME}    ${USERNAME}
   ${LOGOUT_BUTTON}    Get Text    xpath=//*[@id="user-logout-btn"]/div/p
   Should Contain    ${LOGOUT_BUTTON}    Log out
8. Log Out
   Click Element    xpath=//*[@id="user-logout-btn"]
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Log In
9. Close Browser
   Close Browser
```

Signup - Test Results

LOG

Summary Information

Status:	All tests passed
Start Time:	20221118 17:28:30.663
End Time:	20221118 17:29:06.099
Elapsed Time:	00:00:35.436
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	9	9	0	0	00:00:35	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Testsignup	9	9	0	0	00:00:35	

Test Details

All | Tags | Suites | Search

Suite:	Testsignup
Status:	9 tests total, 9 passed, 0 failed, 0 skipped
Start / End Time:	20221118 17:28:30.663 / 20221118 17:29:06.099
Elapsed Time:	00:00:35.436
Log File:	log.html#s1

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
Testsignup.1. Open Browser			PASS		00:00:04.910	20221118 17:28:31.140 20221118 17:28:36.050
Testsignup.2. Go to Log In Page			PASS		00:00:02.194	20221118 17:28:36.052 20221118 17:28:38.246
Testsignup.3. Go to Sign Up Page			PASS		00:00:02.140	20221118 17:28:38.250 20221118 17:28:40.390
Testsignup.4. Fill in Sign Up Form			PASS		00:00:09.380	20221118 17:28:40.394 20221118 17:28:49.774
Testsignup.5. Check Alert Text Message			PASS		00:00:02.017	20221118 17:28:49.777 20221118 17:28:51.794
Testsignup.6. Check User Name			PASS		00:00:05.894	20221118 17:28:51.797 20221118 17:28:57.691
Testsignup.7. Go To Profile Page			PASS		00:00:03.170	20221118 17:28:57.696 20221118 17:29:00.866
Testsignup.8. Log Out			PASS		00:00:02.111	20221118 17:29:00.868 20221118 17:29:02.979
Testsignup.9. Close Browser			PASS		00:00:03.113	20221118 17:29:02.981 20221118 17:29:06.094

Play and Like Video - Test Script

```
*** Settings ***
Library    SeleniumLibrary
Library    XML
*** Variables ***
${BROWSER}    chrome
${URL}    http://localhost:3000
${USERNAME}    ice
${PASSWORD}    12345678
${DELAY}    0.5
*** Test Cases ***
1. Open Browser
   Open Browser    ${URL}    ${BROWSER}
options=add_experimental_option("excludeSwitches", ["enable-logging"])
   Maximize Browser Window
   Set Screenshot Directory    ./testing/screenshot
   Set Selenium Speed    ${DELAY}
2. Go to Log In Page
   Click Element    id=nav-user-icon
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Log In
3. Enter Log In Account
   Input Text    id=login-username-input    ${USERNAME}
   Input Password    id=login-password-input    ${PASSWORD}
   Click Button    id=login-signin-btn
4. Click Play Video
   Sleep    5s
   Click Element    xpath=//*[@id="videoPlayer"]/div[5]
5. Click Shuffle to Play Next Video
   Click Element    xpath=//*[@id="home-shuffle-btn"]
   Sleep   5s
   Click Element    xpath=//*[@id="videoPlayer"]/div[5]
6. Click Like Video
   ${PREV_LIKE_COUNT}    Get Text
xpath=//*[@id="root"]/div/div/div[3]/div/div/div[3]/h4
   ${PREV_LIKE_COUNT}    Convert To Integer    ${PREV_LIKE_COUNT}
   # Log To Console    ${PREV_LIKE_COUNT}
   Click Element    xpath=//*[@id="videocontainer-heart-icon"]
   Sleep    3s
   ${AFTER_LIKE_COUNT}    Get Text
xpath=//*[@id="root"]/div/div/div[3]/div/div/div[3]/h4
   ${AFTER_LIKE_COUNT}    Convert To Integer    ${AFTER_LIKE_COUNT}
   # Log To Console    ${AFTER_LIKE_COUNT}
   ${DIFF}=    Set Variable    ${${AFTER_LIKE_COUNT} - ${PREV_LIKE_COUNT}}
   Should Be Equal As Integers    ${DIFF}    1
7. Click Unlike Video
```

```
   ${PREV_LIKE_COUNT}    Get Text
xpath=//*[@id="root"]/div/div/div[3]/div/div/div[3]/h4
   ${PREV_LIKE_COUNT}    Convert To Integer    ${PREV_LIKE_COUNT}
   # Log To Console    ${PREV_LIKE_COUNT}
   Click Element    xpath=//*[@id="videocontainer-heart-icon"]
   Sleep    3s
   ${AFTER_LIKE_COUNT}    Get Text
xpath=//*[@id="root"]/div/div/div[3]/div/div/div[3]/h4
   ${AFTER_LIKE_COUNT}    Convert To Integer    ${AFTER_LIKE_COUNT}
   # Log To Console    ${AFTER_LIKE_COUNT}
   ${DIFF}=    Set Variable    ${${PREV_LIKE_COUNT} - ${AFTER_LIKE_COUNT}}
   Should Be Equal As Integers    ${DIFF}    1
8. Close Browser
   Close Browser
```

Play and Like Video - Test Results

Testplayandlike Report

Generated
20221118 18:10:58 UTC+07:00
1 minute 37 seconds ago

LOG

Summary Information

Status:	All tests passed
Start Time:	20221118 18:10:15.335
End Time:	20221118 18:10:57.403
Elapsed Time:	00:00:42.068
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	8	8	0	0	00:00:42	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Testplayandlike	8	8	0	0	00:00:42	

Test Details

All | Tags | Suites | Search

Status:	8 tests total, 8 passed, 0 failed, 0 skipped
Total Time:	00:00:41.505

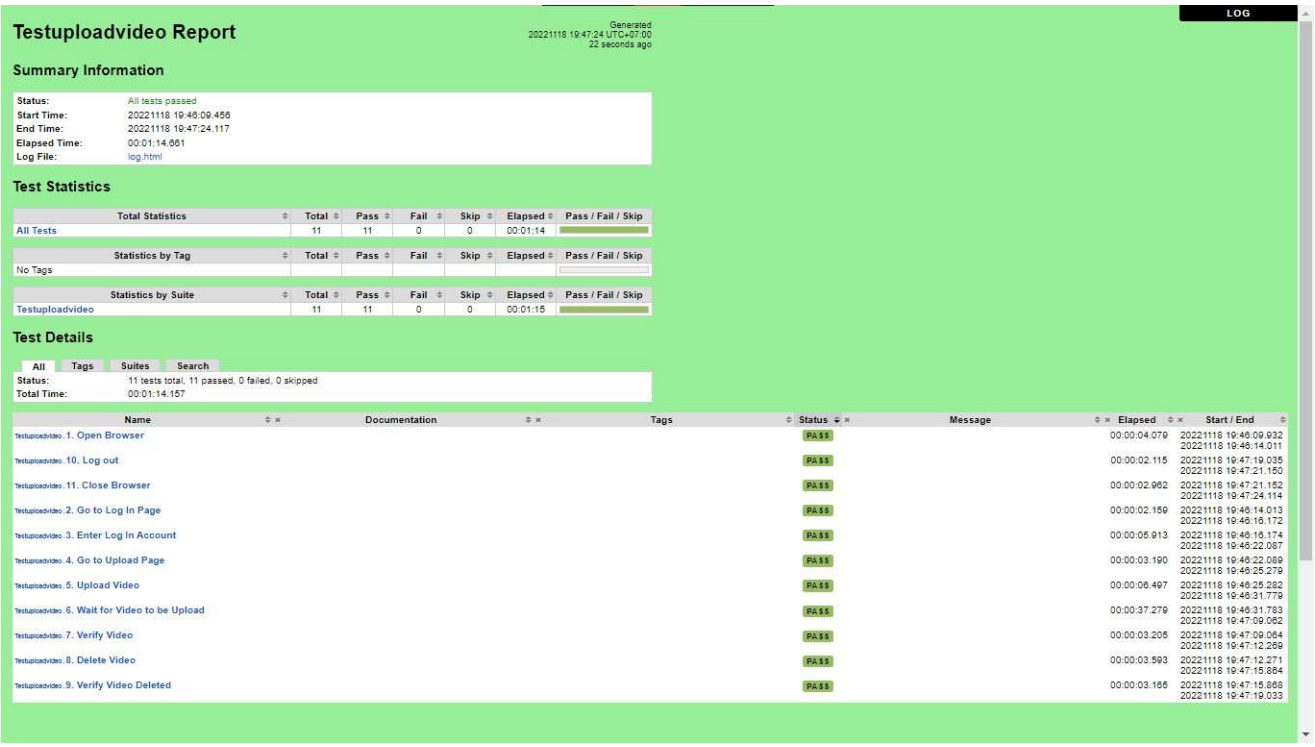
Name	Documentation	Tags	Status	Message	Elapsed	Start / End
Testplayandlike . 1. Open Browser			PASS		00:00:04.818	20221118 18:10:15.878 20221118 18:10:20.696
Testplayandlike . 2. Go to Log In Page			PASS		00:00:02.179	20221118 18:10:20.699 20221118 18:10:22.878
Testplayandlike . 3. Enter Log In Account			PASS		00:00:06.030	20221118 18:10:22.880 20221118 18:10:28.910
Testplayandlike . 4. Click Play Video			PASS		00:00:06.113	20221118 18:10:28.913 20221118 18:10:35.026
Testplayandlike . 5. Click Shuffle to Play Next Video			PASS		00:00:07.149	20221118 18:10:35.028 20221118 18:10:42.177
Testplayandlike . 6. Click Like Video			PASS		00:00:06.196	20221118 18:10:42.180 20221118 18:10:48.376
Testplayandlike . 7. Click Unlike Video			PASS		00:00:06.142	20221118 18:10:48.378 20221118 18:10:54.520
Testplayandlike . 8. Close Browser			PASS		00:00:02.878	20221118 18:10:54.522 20221118 18:10:57.400

Upload Video and Delete Video - Test Script

```
*** Settings ***
Library    SeleniumLibrary
Library    XML
Library    OperatingSystem
*** Variables ***
${BROWSER}    chrome
${URL}    http://localhost:3000
${USERNAME}    time            # Change this to your username
${PASSWORD}    12345678        # Change this to your password
${VIDEO}    D:/SIIT-Y4-Work/DES424/DES424-Term-Project/testing/file/Azure_Bumper.mp4
# Change this to your video file path
${VIDEO_DESCRIPTION}    This is a test script for video upload to QikVid App
# Change this to your video description
${DELAY}    0.5
*** Test Cases ***
1. Open Browser
   Open Browser    ${URL}    ${BROWSER}
options=add_experimental_option("excludeSwitches", ["enable-logging"])
   Maximize Browser Window
   Set Screenshot Directory    ./testing/screenshot
   Set Selenium Speed    ${DELAY}
2. Go to Log In Page
   Click Element    id=nav-user-icon
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Log In
3. Enter Log In Account
   Input Text    id=login-username-input    ${USERNAME}
   Input Password    id=login-password-input    ${PASSWORD}
   Click Button    id=login-signin-btn
4. Go to Upload Page
   Wait Until Element Is Visible    xpath=//*[@id="nav-upload-icon"]
   Click Element    id=nav-upload-icon
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Upload Your Video
5. Upload Video
   Choose File    id=upload-choosefile-btn    ${VIDEO}
   ${VIDEO_NAME}    Get Text    xpath=//*[@id="upload-choosefile-btn"]
   Input Text    xpath=//*[@id="upload-description-input"]    ${VIDEO_DESCRIPTION}
   Click Button    id=upload-submit-btn
6. Wait for Video to be Upload
   Alert Should Be Present    timeout=120 s    text=Video Uploaded Successfully
7. Verify Video
   Click Element    xpath=//*[@id="nav-user-icon"]
   ${LOGIN_USERNAME}    Get Text    xpath=//h2
   Should Contain    ${LOGIN_USERNAME}    ${USERNAME}
   ${EXPECTED_VIDEO}    Get Text
```

```
xpath=//span[contains(text(),'${VIDEO_DESCRIPTION}')]
   # Log To Console    ${EXPECTED_VIDEO}
   Should Contain    ${EXPECTED_VIDEO}    ${VIDEO_DESCRIPTION}
8. Delete Video
   Click Element    xpath=//*[@id="videocontainer-bin-icon"]
   Alert Should Be Present    timeout=120 s    text=Delete video successfully
9. Verify Video Deleted
   Wait Until Element Is Visible    xpath=//*[@id="nav-user-icon"]
   Click Element    xpath=//*[@id="nav-user-icon"]
   ${VIDEO_DELETED}    Get Text    xpath=//span
   Should Not Contain    ${VIDEO_DELETED}    ${VIDEO_DESCRIPTION}
10. Log out
   Click Element    id=user-logout-btn
   ${PAGE_TITLE}    Get Text    xpath=//h1
   Should Contain    ${PAGE_TITLE}    Log In
11. Close Browser
   Close Browser
```

Upload Video and Delete Video - Test Results



Testuploadvideo Report

Generated
20221118 19:47:24 UTC+07:00
22 seconds ago

LOG

Summary Information

Status:	All tests passed
Start Time:	20221118 19:46:09.456
End Time:	20221118 19:47:24.117
Elapsed Time:	00:01:14.661
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	11	11	0	0	00:01:14	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Testuploadvideo	11	11	0	0	00:01:15	

Test Details

All | Tags | Suites | Search

Status:	11 tests total, 11 passed, 0 failed, 0 skipped
Total Time:	00:01:14.157

Name	Documentation	Tags	Status	Message	Elapsed	Start / End
Testuploadvideo . 1. Open Browser			PASS		00:00:04.079	20221118 19:46:09.932 20221118 19:46:14.011
Testuploadvideo . 10. Log out			PASS		00:00:02.115	20221118 19:47:19.035 20221118 19:47:21.150
Testuploadvideo . 11. Close Browser			PASS		00:00:02.962	20221118 19:47:21.152 20221118 19:47:24.114
Testuploadvideo . 2. Go to Log In Page			PASS		00:00:02.159	20221118 19:46:14.013 20221118 19:46:16.172
Testuploadvideo . 3. Enter Log In Account			PASS		00:00:05.913	20221118 19:46:16.174 20221118 19:46:22.087
Testuploadvideo . 4. Go to Upload Page			PASS		00:00:03.190	20221118 19:46:22.089 20221118 19:46:25.279
Testuploadvideo . 5. Upload Video			PASS		00:00:06.497	20221118 19:46:25.282 20221118 19:46:31.779
Testuploadvideo . 6. Wait for Video to be Upload			PASS		00:00:37.279	20221118 19:46:31.783 20221118 19:47:09.062
Testuploadvideo . 7. Verify Video			PASS		00:00:03.205	20221118 19:47:09.064 20221118 19:47:12.269
Testuploadvideo . 8. Delete Video			PASS		00:00:03.593	20221118 19:47:12.271 20221118 19:47:15.864
Testuploadvideo . 9. Verify Video Deleted			PASS		00:00:03.166	20221118 19:47:15.866 20221118 19:47:19.033

Links and Repository

Github	https://github.com/waterthatfrozen/DES424-Term-Project
Jira	https://paphana.atlassian.net/jira/software/projects/DTP/boards/1/roadmap?shared=&atlOrigin=eyJpIjoiNjQzZDhhOTQxZGY3NDdhMGE5ODQzZDAzYjMzY2E1YmYiLCJwIjoiaiJ9
QikVid on Azure App Services	https://qikvid.azurewebsites.net/login
QikVid on Azure Static Web App	https://yellow-island-01d23da00.2.azurestaticapps.net/

End of report.