



**FINAL SEMESTER ASSESSMENT (FSA) B.TECH.  
(CSE)  
VI SEMESTER**

**UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN  
WITH SOFTWARE ENGINEERING LABORATORY**

**PROJECT REPORT  
ON**

**VIDEO STREAMING**

SUBMITTED BY

NAME	SRN
1) Atmik Ajoy	PES1201800189
2) Chethan U Mahindrakar	PES1201801126
3) Dhanya Gowrish	PES1201800965

**JANUARY – MAY 2021**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
RR CAMPUS,  
BENGALURU – 560100, KARNATAKA, INDIA**

<b>TABLE OF CONTENTS</b>		
<b>Sl.No</b>	<b>TOPIC</b>	<b>PAGE No</b>
	ABSTRACT	3
1.	SOFTWARE REQUIREMENTS SPECIFICATION	4
2.	PROJECT PLAN	35
3.	DESIGN DIAGRAMS	43
4.	MODULE DESCRIPTION	52
5.	TEST CASES	54
6.	SCREEN SHOTS OF OUTPUT	64

## **Abstract**

This project is a small scale simulation of the working of a video streaming service. The purpose of this project as a part of the 'Object oriented Analysis and Design of Software Engineering Laboratory' is to demonstrate the various Object oriented coding practices learnt as a part of the course.

The idea of a video streaming platform is to allow users to view and interact with videos. The videos itself are uploaded by other users and are globally available to everyone or a select audience to watch/interact with.

In this project, we essentially simulate a Video Streaming service. The functionality that is implemented in this project is limited but demonstrates the working of a real service. Functionality included are: Watch videos, Interact with videos using Likes and comments, Creating new playlists, Adding/deleting videos from the playlist, Recommendation engine to recommend videos to the user. Most importantly, it allows for a user to create a profile on the platform.

# Chapter 1: Software Requirements Specification

*Software Requirements Specification for Video Streaming*

*Page 2*

## Table of Contents

<b>Introduction</b>	<b>6</b>
Purpose	6
Intended Audience	6
Product Scope	7
References	7
<b>Overall Description</b>	<b>7</b>
Product Perspective	7
Product Functions	8
User Classes and Characteristics	9
Operating Environment	10
Design and Implementation Constraints	10
Assumptions and Dependencies	11
<b>External Interface Requirements</b>	<b>11</b>
User Interfaces	11
Software Interfaces	14
Communications Interfaces	14
<b>Analysis Models</b>	<b>16</b>
Login and sign up use case diagram	16
User dashboard use case diagram	17
Home page use case diagram	18
<b>System Features</b>	<b>18</b>
Registration	18

Login	20
Home Page	21
Video Playback	22
User Dashboard	23
Upload Videos	24
Subscribe Channel	25
Search Video / Channel	26
Like Video	27
Playlist	28
Recommendation Engine	29
<b>Other Non-functional Requirements</b>	<b>30</b>
Performance Requirements	30
Safety Requirements	30
Security Requirements	31
Software Quality Attributes	31
Business Rules	32
<b>Other Requirements</b>	<b>32</b>
<b>Appendix A: Glossary</b>	<b>33</b>
<b>Appendix B: Field Layouts</b>	<b>33</b>
Report Requirements	33
<b>Appendix C: Requirement Traceability Matrix</b>	<b>33</b>

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirement Specifications document is to describe in simple, understandable language and at a functional level, the purpose and features of a video streaming application. This document will describe the scope and the demographic of this application and elaborate on the software and hardware requirements of developing and implementing the application to a large scalable audience and will describe the interfaces and the constraints within which the application will operate. It also describes any and all nonfunctional requirements and other factors that are necessary for the successful operation of the application.

## 1.2 Intended Audience

The document is primarily intended for developers, software architects, testers, marketing staff, project managers and documentation writers. But anyone with prior knowledge in software design and/or development or users wanting to better understand the application and its working can read and understand this document.

This document is divided into 7 sections with sections 3 to 7 intended for developers and software architects. Sections 1,2 and 4 can be understood by anyone who wishes to read about the software and descriptions of its various functionalities.

This Software Requirement Specification contains the following sections in order:

- **Overall description of the product:** This section will describe the context and origin of the product to shed perspective on the product. It will summarize the major functions that the product is expected to perform, identify the various user classes of the product, its operating environment and specify constraints, assumptions and dependencies if any.
- **External interface requirements:** This section elaborates the external interface requirements in detail. It describes the user, software and communication interfaces that are necessary for the functioning of the product.
- **Analysis models:** This section contains all pertinent analysis models. It contains use case diagrams and entity relationship diagrams for all the functionalities that the product provides.
- **System Features:** This section organizes the functional requirements for the product by system features or the major services provided by the product.

- **Other non functional requirements:** This section highlights the performance, safety and security requirements if any and describes the software quality attributes and business rules.
- **Other requirements:** This section will highlight any miscellaneous requirements necessary for the platform's seamless functionality.

### 1.3 Product Scope

The product is designed to function as a video streaming service that is accessible by any user with an internet enabled device. The product will enable uploading, sharing and streaming of videos between users on a global scale, and provide a platform for users to stream a wide variety of content seamlessly. The focus will be on building a safe, monitored platform while still enabling creative freedom to all users. The product will have a simple yet interactive user interface to appeal to all demographics. The goal of this product is to enable simple, easy to use video streaming to users on a global scale. Reliability, security and scalability will be prioritized to ensure a hassle-free experience for users and developers. The product will ensure that the company secures its position as a global leader in the technology and software industry and provide the foundation for successful product releases in the future.

### 1.4 References

- Reference for user classes and characteristics: [Understanding the characteristics of YouTube use and Users](#)
- Reference for use case diagram: [Modelling a context diagram and use case diagram for YouTube](#)
- Reference for working of a video streaming service: [YouTube community](#) ●  
Reference for use case diagrams: [Reuse in use case diagrams](#)

## 2. Overall Description

### 2.1 Product Perspective

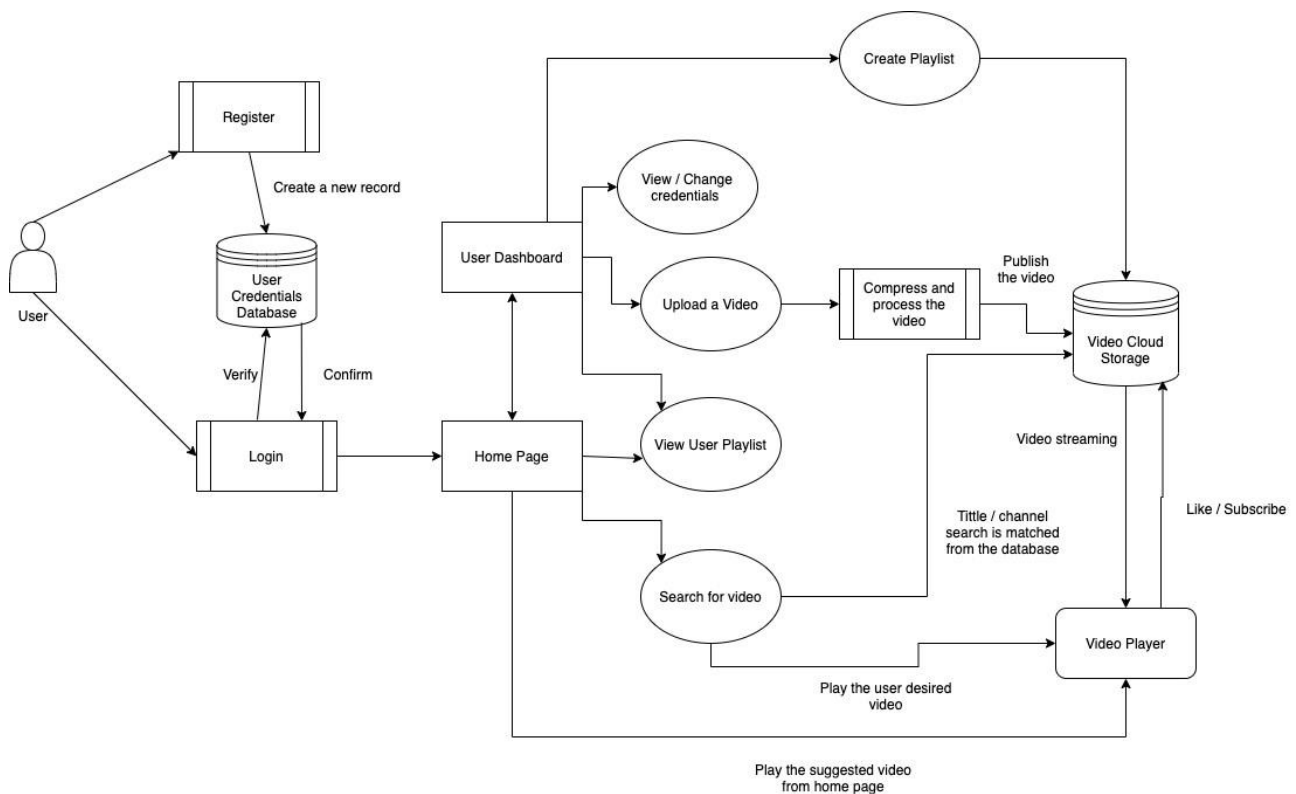
A website based video streaming application which ensures to provide the best service as per the user constraints and preference. The main intention of our website is to provide

users with uninterrupted streaming of video while ensuring minimal consumption of network bandwidth.

Our goal is to build an application that allows users to discover and watch videos, provides a playlist capability for users, can subscribe to channels and acts as a distribution platform for original content.

## 2.2 Product Functions

- Users can register their account.
- Users can login and log out of their account.
- User accounts maintain watch history.
- Users can search for and watch videos.
- Users can upload their own videos.
- Videos playback speed and quality can be configured based on user preference.
- Users can like and share the videos.
- Users can comment on a video.
- Users can subscribe to users/channels.
- Users can create playlists to organize and group videos together.



Top level data flow diagram



## 2.3 User Classes and Characteristics

The intention of this application is to provide a source of video streaming for users. Our application targets users who are above the age of 13 and are interested to watch videos for entertainment or education purposes.

Users can be divided into classes based on various factors such as video category, watch duration, role on the platform, age etc.

Users can be divided based on the category of videos they watch most often. These categories can range from but are not limited to; entertainment, gaming, education, cooking, sports, history and music.

Users can be divided based on watch duration and usage as:

- **Light Users:** These users do not spend more than 20 minutes on the website at any time. Their video choices are mostly limited to shorter duration videos ranging around the 5 minute mark
- **Heavy Users:** These users spend a considerable amount of time on the website watching longer videos that could either be long playlists, streams or other content that can be hours long.

A user can be categorized into one of several roles on the platform :

- **Viewer :** Any user can be a viewer to any video, with the exception of age restricted videos. Users can utilize the interactive interface to search and view videos.
- **Uploader :** Any user can upload video, but the number of videos a user can upload is limited and the user must ensure that it is original content (copyright) and must follow our guidelines. Uploading a video also changes the user's profile to that of a channel, allowing him to gain subscribers and monetize his content.
- **Partnered Content Creators:** These are the uploaders who have been consistently uploading on the platform and sign a contract with the company to be promoted as a content creator and to allow advertisements to be played on their videos. These users gain a percentage of the advertisement revenue.
- **Moderator:** These users are officially designated by the platform to monitor and moderate the platform. These users have limited administrator privileges that allow them to keep any unusual, harmful or illegal activity in check.

There are only 2 types of distinctions on users based on age:

- **Young Users:** These are users above 13 years old (the minimum age required to create an account on the platform) and below 18 years old.
- **Adult Users:** These are users above 18 years old.

The defining characteristic applied on users based on age is video restriction. Young users are allowed to access any content that has not been flagged or labelled as age restricted. This ensures that the platform remains safe for all users irrespective of age.

Characteristics of application provided to all users :

- **Uninterrupted streaming services** : Ensuring mostly ad-free video streaming. Ads if included will be minimal, played at the beginning only and at most 5 seconds in duration.
- **Less network bandwidth consumption** : By creating a lightweight interface while ensuring convenience and ease of use with functionality of video streaming capable of competing with the service provided by other similar top companies.
- **All system features of the platform**: Every user on the platform can search for videos, watch them, like or dislike them, subscribe to channels, share or upload videos and create playlists.

Users can choose if they want to monetize the video by allowing advertisements on their video. Such users must agree to additional guidelines and policies and form a partnership with the platform to be represented as the platform's partner and content creator.

## 2.4 Operating Environment

Our application is web based, thereby making this independent of the operating system. To watch videos, users must have a functioning, popular browser running on the latest version and capable of rendering HTML5/CSS pages and supporting Python/Javascript:

- Newest version of Google Chrome, Firefox, MS Edge, Safari, or Opera
- Internet connection with 500+ Kbps

## 2.5 Design and Implementation Constraints

- Based on the cloud platform chosen for server deployment and depending on the hardware provided, the server might be limited to a number of users accessing the website at one time. Thereby developers have to improve the video stream performance. (Real Time Messaging Protocol [RTMP])
- Limited number of videos can be uploaded depending on storage provided by the cloud platform provider. Thereby developers have to improve the compression of video without losing information.

- The application will be developed in python, thereby developers should be proficient in python, mysql as the backend database management, HTML/CSS for frontend.
- To ensure safe and secure experience for our users, the developer should ensure that the code is not prone to attacks like sql injection and other malicious attacks.

## 2.6 Assumptions and Dependencies

This application is dependent on :

- Web hosting service
- User web browser
- Language used for developing - python
- Main Packages used - flask, requests
- Server - Flask / Apache
- Database used - MySQL

## 3. External Interface Requirements

### 3.1 User Interfaces


- **Home Page (Logged Out):** This view is seen by users who have not logged into the service. It includes a button on the top right for the user to sign in if needed as well as relevant video suggestions based on the user's location, and other such information. If the user chooses to sign in, the user is redirected to the Login page.

A wireframe of a website header and main content area. The header contains a hamburger menu icon on the left, a search bar with the placeholder text "Search" and a magnifying glass icon on the right, and a "Sign in" button on the far right. Below the header is a grid of 12 empty rectangular boxes arranged in 3 rows and 4 columns.

- **Login page:** Here the user is prompted to enter their login details (Email ID, password). Once the user's credentials are verified, the user is redirected to the service's homepage. The user also has the option to create an account, if he/she wishes to do so .

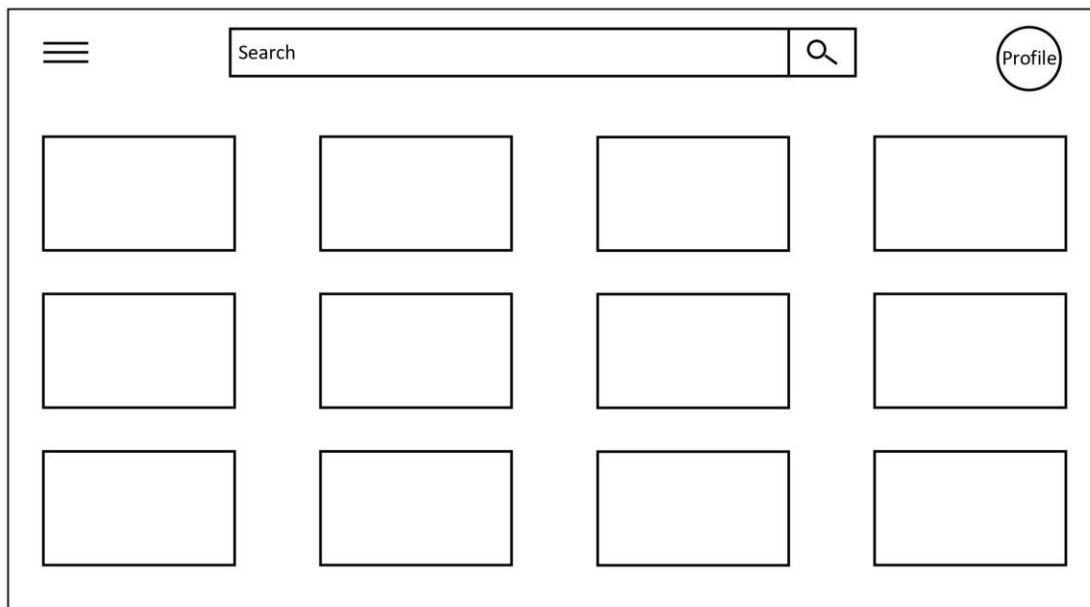
A wireframe of a login page. It features a central box with the title "LOGIN" at the top. Below the title are two input fields: the first is labeled "Email" and the second is labeled "Password". At the bottom of the box is a "Sign up" button.

- **Sign up page:** The user is redirected to this page from the login page if they do not have an account registered. Here the user is prompted to enter relevant user information such as name, email id, password, etc. necessary to create the user's account and store it in the service's servers.



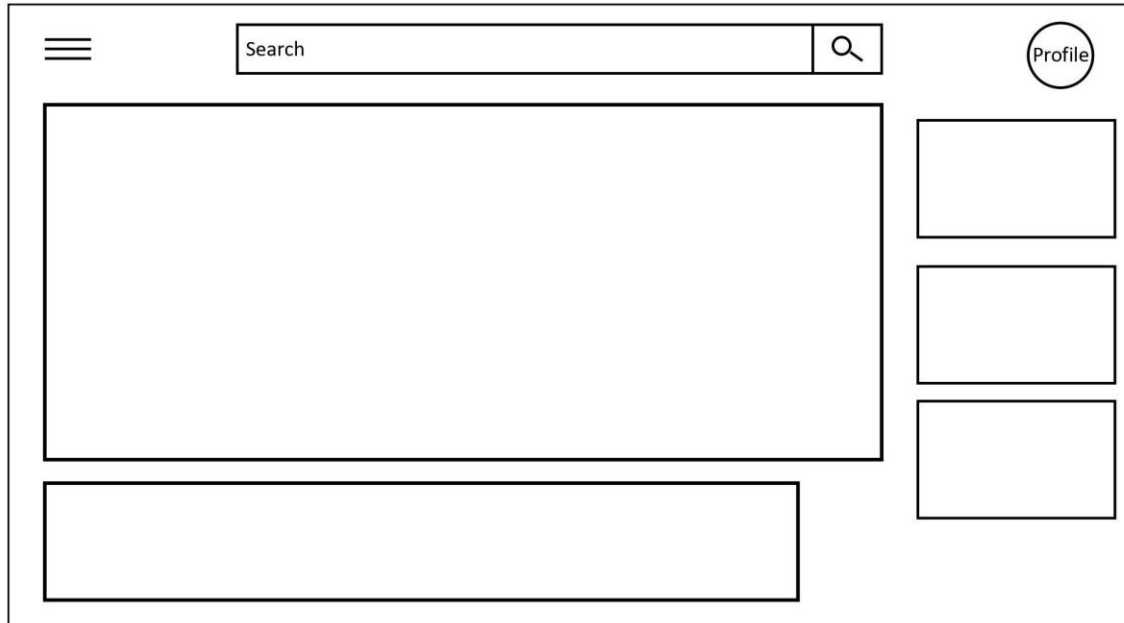
A sign-up form titled "Sign Up" is centered within a large rectangular frame. The form contains five input fields stacked vertically, each with a label inside: "First Name", "Last Name", "Email", "Password", and "Confirm Password".

- **Home Page(Logged In):** This view is seen by users who have logged in. The user is now provided with video suggestions that are personalised to suit the users preferences based on their viewing patterns and subscriptions. The user is also provided with a menu to view their playlists, subscriptions, and so on.



The home page layout for a logged-in user is shown within a rectangular frame. At the top, there is a navigation bar containing a hamburger menu icon on the left, a search bar with the placeholder text "Search" and a magnifying glass icon on the right, and a circular profile icon labeled "Profile" on the far right. Below the navigation bar, the main content area is a grid of 12 empty rectangular boxes arranged in 3 rows and 4 columns, intended for video suggestions.

- **Video playback page:** This page is where the user is taken when a video is clicked on. It streams the video at the user's desired quality and playback speed. It also provides other videos to watch which are suggested based on the user's viewing patterns.



## 3.2 Software Interfaces

- **User Interface:** Users can access the web application through any up-to-date browser.
  - The web application can be accessed through any web-browser. The interface will be developed using common frameworks such as HTML, CSS, React, Flask.
  - The user receives the HTML objects efficiently with the help of requests module and ensures secure communication.
  - The user interface is mainly built with React js for faster rendering, reusable components and flexible.
  - Users can also upload video to our application and needs to only provide the best quality, the lower quality will be generated from our application backend server.
- **Back-end server:** The server will run on flask (Werkzeug) / apache. The server interacts with the user interface and performs the following services:
  - The server will be deployed on an online cloud platform such as AWS / GCP.

- Ensures a secure and confidential connection while the user is providing his credentials for registration / verification by encrypting the connection. Asymmetric encryption method is used for encryption with the help of cryptography, base64 modules in python.
- Provides buffered playback of videos that the user requests (using TCP and HTTPS)
- Interacts with the user database to store vital user information using the pymysql module.
- After an user uploads a video, the video is replicated to lower quality for later streaming purpose and is compressed and stored in the cloud storage. For this functionality, API such as transcoder might be used.
- Online cloud storage such as dropbox will be used for storing the videos and another separate database acting as a transaction log for monitoring and accessing these videos.
- **Database:** Stores videos as well as user information. The back-end server updates the database based on the user's input to the user interface.

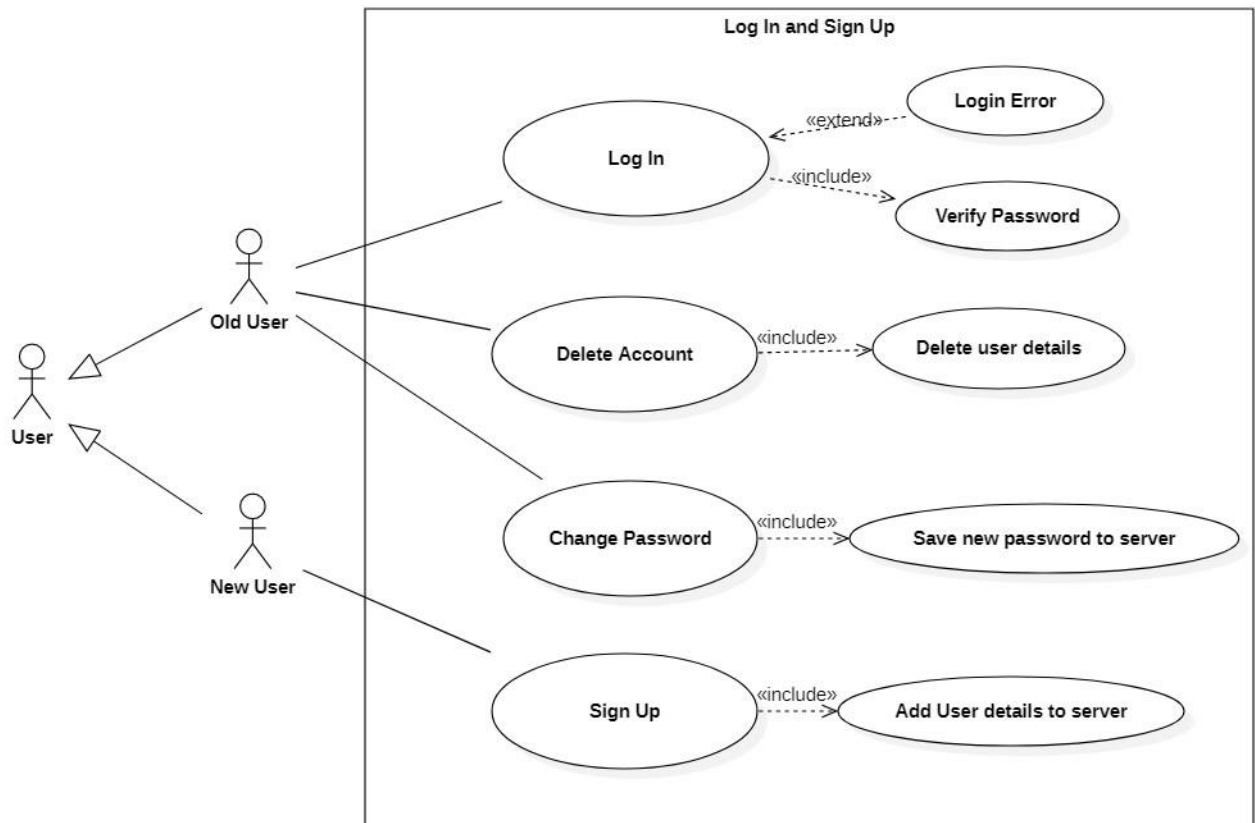
### 3.3 Communications Interfaces

HTTP and HTTPS protocols will be used to communicate between the client and the server. All communication between the software interfaces will be done through encrypted

protocols such as SSL/TLS, as sensitive login information is sometimes sent across and this would need to be encrypted.

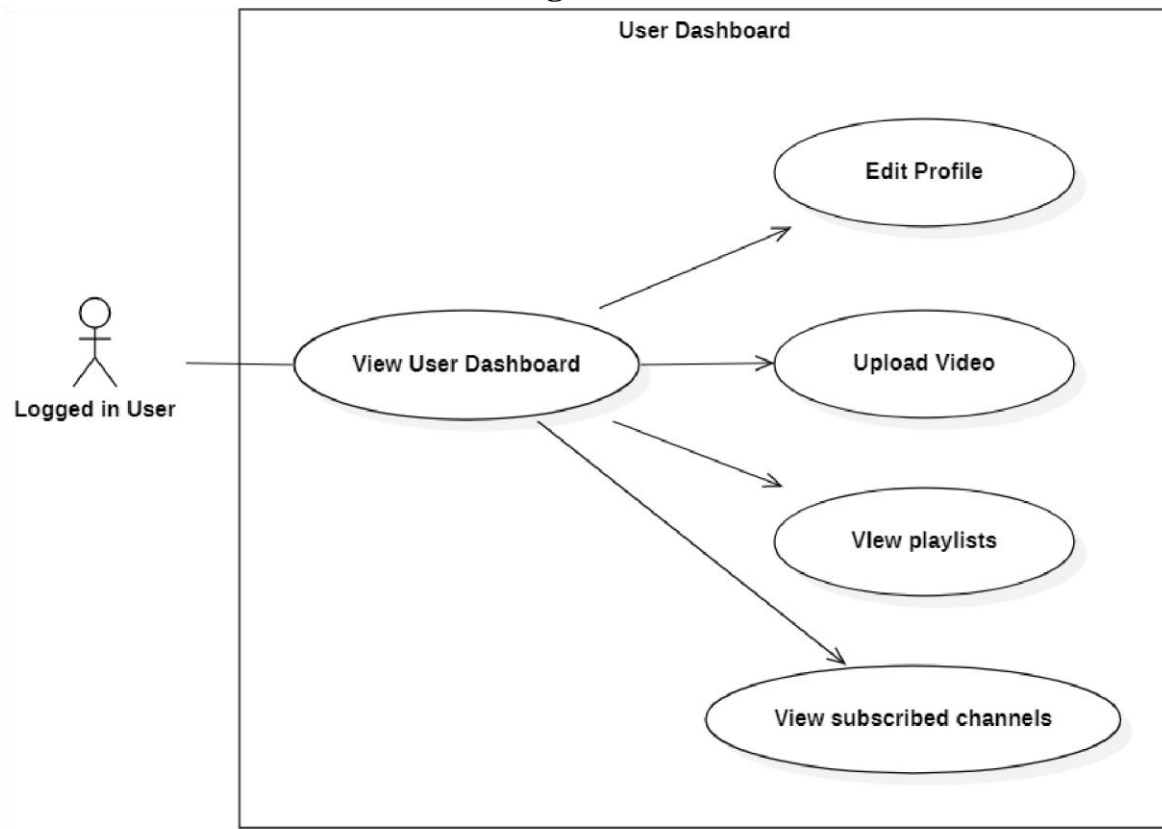
## 4. Analysis Models

### 4.1 Login and sign up use case diagram

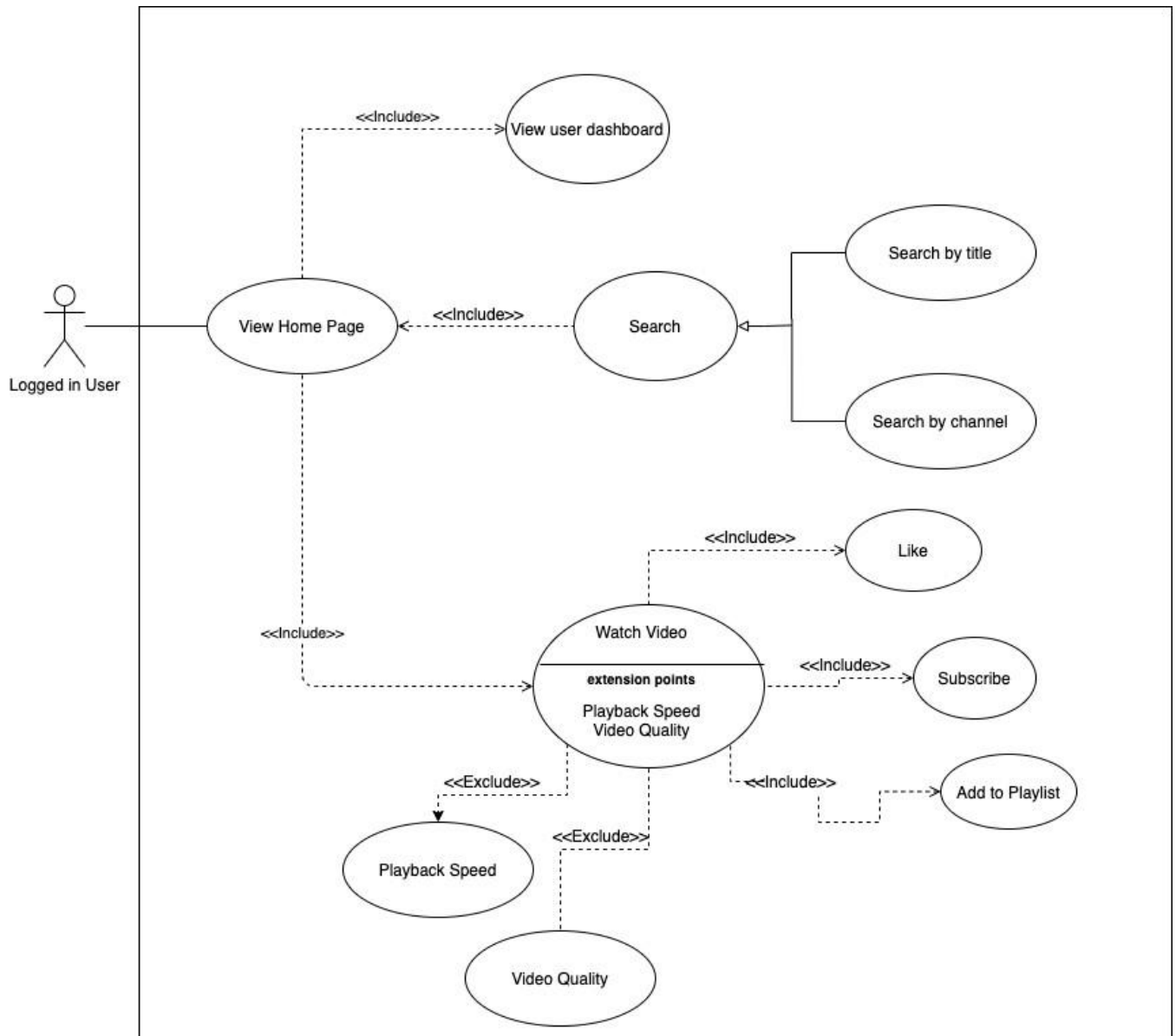




## 4.2 User dashboard use case diagram



### 4.3 Home page use case diagram



## 5. System Features

### 5.1 Registration

#### 5.1.1 Description and Priority

Through registration a user can now be identified in our application ecosystem, and along with this now users have the ability to create their own playlist, follow channels, comment, like, subscribe, upload content and become a channel themselves. Users who are just viewing anonymously do not have

much priority, but for creators having an account is necessary. Since these details accepted from the users are verified from the database there is risk of a potential security breach and hence this page is tightly secured.

- Benefit - 7
- Risk - 8
- Priority - 9

### **5.1.2 Stimulus/Response Sequences**

- Stimulus
  - The user must enter his email-id of his desired domain.
  - The user must enter a unique name, which has not been used before in our application.
  - The user has to enter a password and re-enter to confirm it.
- Response
  - If the user has provided the necessity information and is verified, then the user is taken to the homepage.
  - An entry is made into the database for this user, any future modifications/ preferences will be recorded accordingly.
  - If the credentials were invalid then those required columns need to be refilled again for verification.

### **5.1.3 Functional Requirements**

- REQ-FE\_1 : A front end sign up page, where the user can enter his credentials securely.
- REQ-BE\_1 : A backend program to ensure that the credentials entered are safe(protected from malicious attacks such as sql injection) and satisfy the given conditions. If all conditions are met, then create an account, add a record to the database and redirect the user to the home page. But if the credentials are invalid then redirect the user back to the login page specified with the invalid credentials.
- REQ-BE\_2 : Permission from user to store cookies, for saving credentials and enhancing the user experience with the website.

- REQ\_FE\_2: The home page which users enter for the first time after creating an account.

## 5.2 Login

### 5.2.2 Description and Priority

If a user has already created an account in our application, then the user can login from any device to gain access to his/her account for their personalized homepage and other features (as mentioned in the register section).

- Benefit - 7
- Risk - 8
- Priority - 9

### 5.2.2 Stimulus/Response Sequences

- Stimulus
  - The user must enter his email-id/ name of his desired domain.
  - The user has to enter his password.
  - These details can be directly accessed from the cookies (if available)
- Response
  - The email-id and password are cross checked from the database.
  - If the credentials are valid, then the user is directed to the home page.
  - If the email-id, password did not match in the database then the user is redirected back to the login screen.

### 5.2.3 Functional Requirements

- REQ-FE\_3 : A front end login page, where the user can enter his credentials securely.
- REQ-BE\_4 : A backend program to ensure that the credentials entered are safe (protection from malicious attacks such as sql injection) and then these details are cross checked with the database. If valid then the user is directed to the home page else the user is redirected back to the login page.
- REQ\_FE\_5: The personalized home page as per user preferences

## 5.3 Home Page

### 5.3.1 Description and Priority

This is the main page of our video streaming application. Every user will have a personalised website, if the user has created an account then the home page has video suggestions based on previously watched videos and videos of the user subscribed channels. From this page users have access to several features provided by our application.

- Benefit - 9
- Risk - 5
- Priority - 9

### 5.3.2 Stimulus/Response Sequences

- Stimulus
  - The page has a search bar, to discover new videos and channels.
  - The page has a playlist folder to show all the playlist created by the current user.
  - The page has thumbnails for all videos suggested and its respective link for streaming.
- Response
  - After searching for the video / channel, all the channels and video matched to the search text are displayed.
  - Through the playlist section, the user can view their previously created playlist or can also play their previously made playlist.
  - By clicking on any video thumbnails are streamed on a new web page.

### 5.3.3 Functional Requirements

- REQ\_FE\_5: The personalized home page as per user preferences
- REQ\_FE\_6 : A web page to display / create new playlist.
- REQ\_BE\_5 : A backend to search for the given video.
- REQ\_BE\_6 : Personalise the home feed of each user based on the user activity
- REQ\_FE\_7 : A new webpage to display the video streaming.

- REQ\_FE\_8 : To display the search results.

## 5.4 Video Playback

### 5.4.1 Description and Priority

The videos are played in a custom video player of size aspect ratio 16:9. For having an uninterrupted video session, no ads are played during the video, the video quality is configured based on the user network connection and can also be configured based on user preference. Even the playback speed can be configured.

- Benefit - 9
- Risk - 5
- Priority - 9

### 5.4.2 Stimulus/Response Sequences

- Stimulus
  - The user can play/pause the video.
  - Users can configure the video quality.
  - Users can configure the video playback speed.
  - Homepage button
  - Users can like the video
  - Users can subscribe to the channel.
- Response
  - The quality of the video ranges from 144p to 1080p.
  - The playback speed ranges from 0.5 to 2 with a 0.25 step size.
  - Clicking on the home page button redirects back to the user home page.
  - If the user has an account, only then the like is considered.
  - Only if the user has an account, can that user subscribe to other channels.

### 5.4.3 Functional Requirements

- REQ\_FE\_7 : A new webpage to display the video streaming.

- REQ\_FE\_9 : Video player with the configurable options.

REQ\_BE\_7 : Configure the video settings as per user preference.

- REQ\_BE\_9 : Add / remove the channels and update the user database.

## **5.5 User Dashboard**

### **5.5.1 Description and Priority**

In this section the user can view his account details, modify his/her credentials, upload videos to their channel, view their subscribed channels and view their created playlist.

- Benefit - 7
- Risk - 5
- Priority - 7

### **5.5.2 Stimulus/Response Sequences**

- Stimulus
  - Users can change credentials such as user name or password.
  - Users can upload videos to their channel, by clicking on the upload video button.
  - Users can view / create their own playlist.
  - Users can view the channels they are subscribed to.
- Response
  - Users for changing credentials are redirected to a new page and ensured that the user is authorised to change the credentials.
  - Users when clicking on the upload button are redirected to the upload video section.
  - Users are redirected to the playlist web page.
  - Users are displayed with the list of channels they have subscribed to,

### **5.5.3 Functional Requirements**

- REQ\_FE\_10 : A new webpage to display the user dashboard.
- REQ\_BE\_8 : User verification and changing credentials.



- REQ\_BE\_9 : To view the user subscribed channels, by connecting to the user database in the server backend.
- REQ\_FE\_6 : A web page to display / create new playlist.

## **5.6 Upload Videos**

### **5.6.1 Description and Priority**

Once users have logged into their account, then users can upload videos to their channel and these videos are globally available to any other user who has searched or subscribed to the channel. The video is uploaded only once, but then the videos are processed on our server to ensure the video with multiple quality versions so that viewers can watch their own preferred quality(to reduce video buffering). These videos are codec compressed on our server side to ensure to decrease storage usage and efficient streaming. A status bar is present to indicate the percentage of video uploaded.

- Benefit- 9
- Risk - 8
- Priority - 9

### **5.6.2 Stimulus/Response Sequences**

- Stimulus
  - Uploader must provide the title of the video
  - Uploader can also provide a description of the video..
- Response
  - The video title is checked in the database to ensure that the title is unique.
  - The video is uploaded to the cloud storage and now the server can stream the video to anyone who searched for this video

### **5.6.3 Functional Requirements**

- REQ\_FE\_11 : A web page for uploading video interface.

- REQ\_BE\_10 : The title of the video must be unique and is cross checked from the database.
- REQ\_BE\_11 : If all details are valid, then the video is stored inside the cloud storage.
- REQ\_BE\_12 : The video is compressed, and processed to all different types of quality ranging from 144p to 1080p.

## **5.7 Subscribe Channel**

### **5.7.1 Description and Priority**

Users can subscribe to channels of their choice and by subscribing to a channel, the user's home page will show up any new video posted by the subscribed channels thereby personalising the home page.

- Benefit - 7
- Risk - 3
- Priority - 5

### **5.7.2 Stimulus/Response Sequences**

- Stimulus
  - Users must click on the subscribe button found below any video of their preferred channel or found on the channel dashboard.
- Response
  - Only if the user has logged in his/her account, then the user can subscribe / unsubscribe and with the button click the user subscription list is updated.

### **5.7.3 Functional Requirements**

- REQ\_BE\_9 : Add / remove the channels and update the user database.
- REQ\_BE\_9 : To view the user subscribed channels, by connecting to the user database in the server backend.

## **5.8 Search Video / Channel**

### **5.8.1 Description and Priority**

A search bar will be present on the top of every web page and with this users can search for their preferred video or channel via searching for the title.

- Benefit - 9
- Risk - 8
- Priority - 8

### **5.8.2 Stimulus/Response Sequences**

- Stimulus
  - Search bar on top of the web page for searching video / channel.
- Response
  - Once the user enters the channel / video title the results are by string matching from the database.

### **5.8.3 Functional Requirements**

- REQ\_BE\_13 : Search video from the database based on the title provided by the user.

## **5.9 Like Video**

### **5.9.1 Description and Priority**

Users can like a video to increase the popularity of the video and by liking videos the users home page will be further personalised based on his liked videos.

- Benefit - 7
- Risk - 1
- Priority - 4

### **5.9.2 Stimulus/Response Sequences**

- Stimulus
  - The like button is present below every video.
- Response
  - Only users who have logged into their account can like the video and the video like counter is incremented.

### **5.9.3 Functional Requirements**

- REQ\_BE\_14 : Incrementing the like counter in the database.  
REQ\_BE\_15 : Keeping a track of user liked video and personalizing the home page accordingly.

## **5.10 Playlist**

### **5.10.1 Description and Priority**

Once users have logged into their account, users can create their own playlist. Videos can be added into the playlist, by clicking on the add button below a video and later these collections of videos can be viewed in the playlist section.

- Benefit - 6
- Risk - 3
- Priority - 4

### **5.10.2 Stimulus/Response Sequences**

- Stimulus
  - User clicks on the add to playlist button.
  - User clicks on the view playlist column.
- Response
  - The current video gets added to the playlist.
  - The user can view all videos under their playlist section.

### **5.10.3 Functional Requirements**

- REQ\_FE\_6 : A web page to display / create new playlist.
- REQ\_BE\_16 : Add / remove videos from the playlist.

## **5.11 Recommendation Engine**

### **5.11.1 Description and Priority**

Once the users have logged into their account, they will have a Recommendation engine that recommends videos based on tags associated With the video.

- Benefit - 7
- Risk - 5
- Priority – 6

### **5.11.2 Stimulus/Response Sequences**

- Stimulus
  - User Log's in
  - User clicks on recommend videos button
- Response
  - Videos get recommended to the user based on tags.
  -

### **5.11.3 Functional Requirements**

- REQ\_FE 7: A webpage to accept tag's input and display recommended videos
- REQ\_BE 17: Build a recommendation engine

## **6. Other Nonfunctional Requirements**

### **6.1 Performance Requirements**

Users requirements are :

- A device capable of playing videos from any up-to-date browser at 144p.
- For uninterrupted buffer free video streaming, a minimum of 500kbps network download speed is required.
- Videos being uploaded must follow these constraints:
  - File format: AVI, ASF, Quicktime, Windows Media, MP4 or MPEG
  - Video Codec: H.264, MPEG-2 or MPEG-4
  - Audio Codec: AAC-LC or MP3
  - Aspect ratio: 16:9 or 4:3 is recommended
  - Frame rate: 30 FPS
  - Maximum file size: 1 GB
  - Video should not contain violence, any R rated content, hate speech.
  - For advertisements the maximum ad-length is 30 seconds and should be skippable.

Technical requirements are :

- The server must be scalable for use by up to 15000 users simultaneously accessing the service.
- The system must be capable of storing 7000 hours worth of video at 1080p quality. The storage space for these videos must be defined and delivered.
- The streaming system must be designed and delivered in high availability configuration with built in redundancy to ensure smooth operation even if a component fails.
- The streaming system must have the capability to stream at various bandwidths, picture sizes and qualities, as well as allow for different playback speeds (ranging from 0.25x to 3x) depending on the user's preference and internet connection. Videos have to be streamed up to Full-HD (1080p at 30 fps), while 144p video should be considered as the minimum supported format.

### **6.2 Safety Requirements**

If the server crashes or the database gets corrupted due to any malicious attack or through overload, our product has an archival database storage present and a secondary backup server is also present for such scenarios.

### 6.3 Security Requirements

- All user credentials are encrypted and stored in a secure cloud storage.
- The application is protected against various vulnerabilities and attacks.
- User personal data are confidential to other users.

### 6.4 Software Quality Attributes

- **Adaptability** : The application can run on any device as long as the minimum requirements are met, the user interface is adapted according to the screen size of the device to provide the best user interface.
- **Availability** : The application will be available all time, in case of server down or any database corruption issues then all the users get redirected to the server status web page.
- **Correctness** : The application is very intuitive and easy to use, verifies each user and provides the permission accordingly. Users are redirected to appropriate web pages in case of an error.
- **Flexibility** : The whole product is broken down to components, in case of any improvement to one of the features then that updated feature can be added without disturbing the other features.
- **Maintainability** : MySQL is used for the backend database and the flask server takes care of the functionality of the product.
- **Portability** : This product can be accessed from any device which has met the minimum requirement and is OS independent. Developers note : The application runs on flask, thereby any web hosting service which supports flask should be able to handle the product.
- **Reliability** : All of our users credentials are encrypted and stored in a secure cloud storage.
- **Reusability** : There are many more functionalities that can be added to this product, thereby the present features can be re-used in a larger project.
- **Robustness** : The product is ensured to be safe from malicious attacks and in case of any error or any type of malicious attack the database connection might get locked to rectify the error, but the website can still be accessed.
- **Testability** : The application will be tested on a regular interval and these tests will not interrupt the user end experience.

- **Usability** : The interface is very user friendly and intuitive and this product can be accessed from anywhere provided the user has a device which satisfies the minimum requirement and has stable internet connection.

## 6.5 Business Rules

- While creating an account users must give a unique name and a password that satisfies the constraints.
- Users must have an account to like videos and to subscribe to channels.
- Users must be verified before being able to monetize any uploaded videos.
- If a user wants to view / change their credentials, then the user must provide the old credentials for verification purposes.

## 7. Other Requirements

- Must provide the channels with useful statistics such as views, likes, dislikes and other relevant information.
- Hate speech and harassment of other users must be identified and dealt with appropriately.
- Inappropriate content in uploads must also be monitored and dealt with by deleting the video and (or) banning the user for repeated offences.
- Could optionally add a feature where some users are restricted from accessing specific videos based on the user's age, location and other relevant information.

## Appendix A: Glossary

Under System Features functional requirement section :

- REQ\_FE\_<Number> : Indicates that it is a front end component and each component has an unique identification number. These components can be reused in other system features.
- REQ\_BE\_<Number> : Indicates that it is a back end component and each component has an unique identification number. These components can be reused in other system features.



## Appendix B: Field Layouts

### Report Requirements

#### Video Report

User ID  
Email ID  
Password  
Username  
Age  
Phone Number  
Location  
Platform Join Date  
Status  
Upload count

#### User Report

Video title  
Channel name  
Uploader/Channel ID  
Like count  
Comment count  
Video length  
Upload date  
Video codec  
Age restriction  
Metadata

## Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	REQ_1	Registration	Section 5.1	Registration Design	<a href="#">Code file ref</a>	UT-01	ST-01 & ST-02
2	REQ_2	Login	Section 5.2	Login Design	<a href="#">Code file ref</a>	UT-02 & UT-03	ST-01 & ST-02
3	REQ_3	Home Page	Section 5.3	Home page design	<a href="#">Code file ref</a>	-	ST-01 & ST-02
4	REQ_4	Video Playback	Section 5.4	Playback design	<a href="#">Code file ref</a>	UT-06 & UT-07	ST-01 & ST-02
5	REQ_5	User Dashboard	Section 5.5	Dashboard design	<a href="#">Code file ref</a>	UT-10 to UT-12	ST-01 & ST-02

6	REQ_6	Upload Videos	Section 5.6	Upload design	<a href="#">Code file ref</a>	UT-08 & UT-09	ST-01 & ST-02
7	REQ_8	Search Video / Channel	Section 5.8	Search video design	<a href="#">Code file ref</a>	UT-04 & UT-05	ST-01
8	REQ_9	Like Video	Section 5.9	Like video design	<a href="#">Code file ref</a>	IT-02 & IT-03	ST-01 & ST-02
9	REQ_10	Playlist	Section 5.10	Playlist design	<a href="#">Code file ref</a>	UT-17 to UT-21	ST-05 to ST-08
10	REQ_11	Recommendation Home Page	Section 5.11	Home design	<a href="#">Code file ref</a>	UT-10 to UT-16	ST-03 & ST-04

# Chapter 2: Project Plan

## Life cycle followed for execution of the project:

Our project uses the Agile Software Development Lifecycle model. The Agile SDLC model is a combination of iterative and interactive process models focused on rapid delivery of working software products with high adaptability and customer interaction and satisfaction. Since our project has a short time frame, adopting the Agile model will allow for the product to be developed incrementally in multiple iterations consistently ensuring working deliverables with each iteration improving on the functionality of the software product.

Every iteration of the life cycle will stretch over 1-3 weeks and include teams working simultaneously on:

- Planning
- Design and requirements Analysis
- Building and implementation
- Unit and acceptance testing
- Review

Every iteration will produce a working product that can be demonstrated to customers or stakeholders, and the final iteration will produce the completed build holding all the features required by the customer or stakeholder.



The above diagram depicts the iterations over a time frame of 1-2 months. Each iteration lasts around 2-3 weeks and each iterative product launch will ensure addition of new functionality to the product.

Advantages of using the Agile SDLC for this project:

1. Can incorporate versioning in the SRS and ensure the product is flexible and capable of adapting to stakeholders requirements by incorporating additional functionalities and changes during development
2. Face to face communication and continuous inputs from either customers or stakeholders will ensure the developers completely understand the users' requirements and can modify their functionalities accordingly.
3. Gives flexibility to developers and requires little to no planning.
4. Delivers early partial working solutions and enables regular delivery with short time frames
5. Since the product does not involve any complex dependencies, the functionalities can be developed concurrently
6. Since all team members are developers, it overcomes the disadvantage of high individual dependency, as every member can regularly interact with other members and work on development even with minimum documentation

## References:

<https://bikeshsrivastava.blogspot.com/2017/01/part-43what-is-agile-methodology.html>

[https://www.tutorialspoint.com/sdlc/sdlc\\_agile\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm)

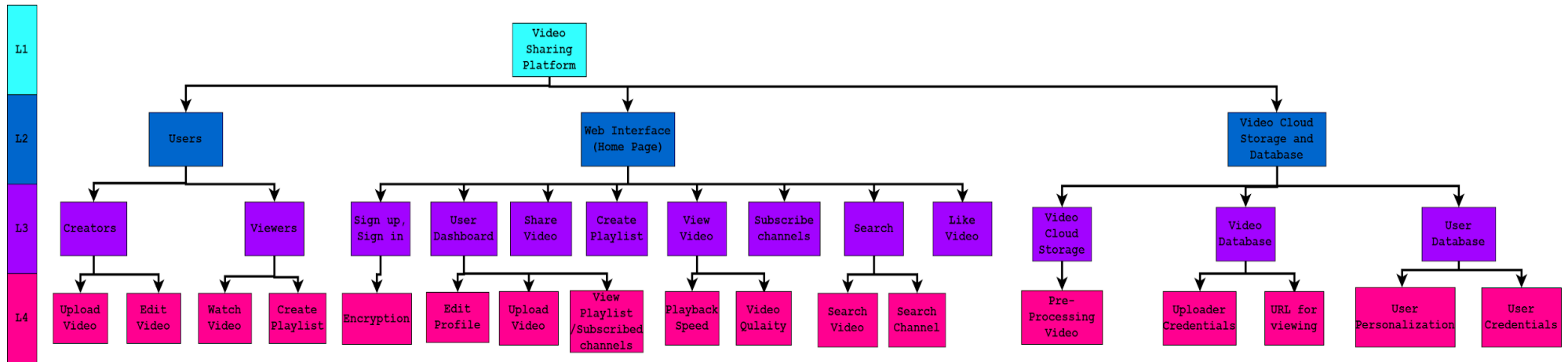
<https://medium.com/@melsatar/software-development-life-cycle-models-and-methodologies-297cfe616a3a>

## Tools used:

- **Planning Tools**
  1. Google Docs
  2. Google Sheets
  3. ToDo Lists
  4. Teamwork
- **Design Tools**
  1. StarUML
  2. Draw.io
  3. Figma
- **Version Control**
  1. Git Version Control with Github
- **Development Tools**
  - Front End: ReactJS, HTML, CSS, Bootstrap

- Back End: Flask, Firebase, Heroku or SQL
  - Development tools: Github and Visual Studio Code
- **Bug Tracking**
  - 1. Github Issues
- **Testing Tools**
  - 1. Selenium Web Driver
  - 2. Flask Security

# WORK BREAKDOWN STRUCTURE



## Deliverables:

During each iteration a component will either be created or further enhanced. After the first iteration, every iteration followed by that will have a **testing phase** for checking the working/security/improvement of the components. This is to ensure that the components till that iteration meet the expectation of the customers. Few of these iterations can be modified based on the customer feedback.

Sl no.	Deliverable	Description	Category
1	<b>Software Requirement Specification</b>	The SRS document outlines the features provided by the software and a rough idea of implementation, provides functional and nonfunctional requirements and also describes the targeted audience, their user interface and minimum user requirements.	<p><b>Build :</b> This document is initially written to lay out the description of the software that is to be developed as well as the intention of the software under development.</p> <p><b>Reuse :</b> Later versions can bring changes based on customer feedback to the document due to our agile life cycle implementation.</p>
2	<b>Project Plan</b>	This document provides the SDLC, outlines the execution procedures along with the specified period to achieve the intended software components.	<p><b>Build :</b> It is documented to guide the project execution and project control.</p> <p><b>Reuse :</b> Due to the agile SDLC, modifications can be made on the execution timeline.</p>
3	<b>Database Initialization and Cloud Storage</b>	Initialize database for storing user credentials, user personalization options (playlist created and subscribed channels) and video ownership details (only uploaders) Cloud Storage for storing the video uploaded by users.	<b>Build :</b> The database will be built to initiate the other functional components implementation.

4	<b>Video Playback, Video Pre-Processing and searching video</b>	A minimalistic frontend page consisting of an embedded video player with the ability to change the video playback speed and quality of the video. Pre-Processing of the video to all the lower available qualities and storing in the cloud storage. Simple video/channel searching via the title name.	<b>Build :</b> The main functionality of our software is video streaming service with features which ensure low network bandwidth consumption.
5	<b>User Details, User Dashboard and Database Integration</b>	To create a personal user account for viewing and uploading videos and keeping a record of all the users in our ecosystem with the database (Iteration 2).	<b>Build :</b> The frontend user interface and the backend for integration with the database. <b>Reuse :</b> User Database.
6	<b>User Personalization</b>	Enables users to subscribe to channels and like videos. The user homepage is personalized accordingly. Users can create their own playlist.	<b>Build :</b> The frontend for the homepage and the backend for homepage suggestion video. <b>Reuse :</b> Channel details and video database used by backend for personalization and for video/channel analytics (like and subscribe count)
7	<b>Advanced Web User Interface</b>	Further Development on the pre-existing user interface for more user friendliness, less network bandwidth consumption and viewed in any device(satisfying the minimal hardware and software requirements)	<b>Reuse :</b> All the existing functional components are used and only the interface is modified.
8	<b>Server Deployment on Web</b>	Now this software can be	<b>Reuse :</b> All the components



	<b>Service (Hosting)</b>	accessible from any device with internet connection.	used till this phase are reused with few changes as per the hosting services.
<b>9</b>	<b>Additional Security Features</b>	To ensure that all user details via communication and storage are safe and secure all details are encrypted using an asymmetric key encryption method along with TLS.	<b>Reuse :</b> The basic security features will be similar to localhost security features.  <b>Build :</b> To ensure that the software is not vulnerable to malicious attacks.

### Rough Effort Estimate:

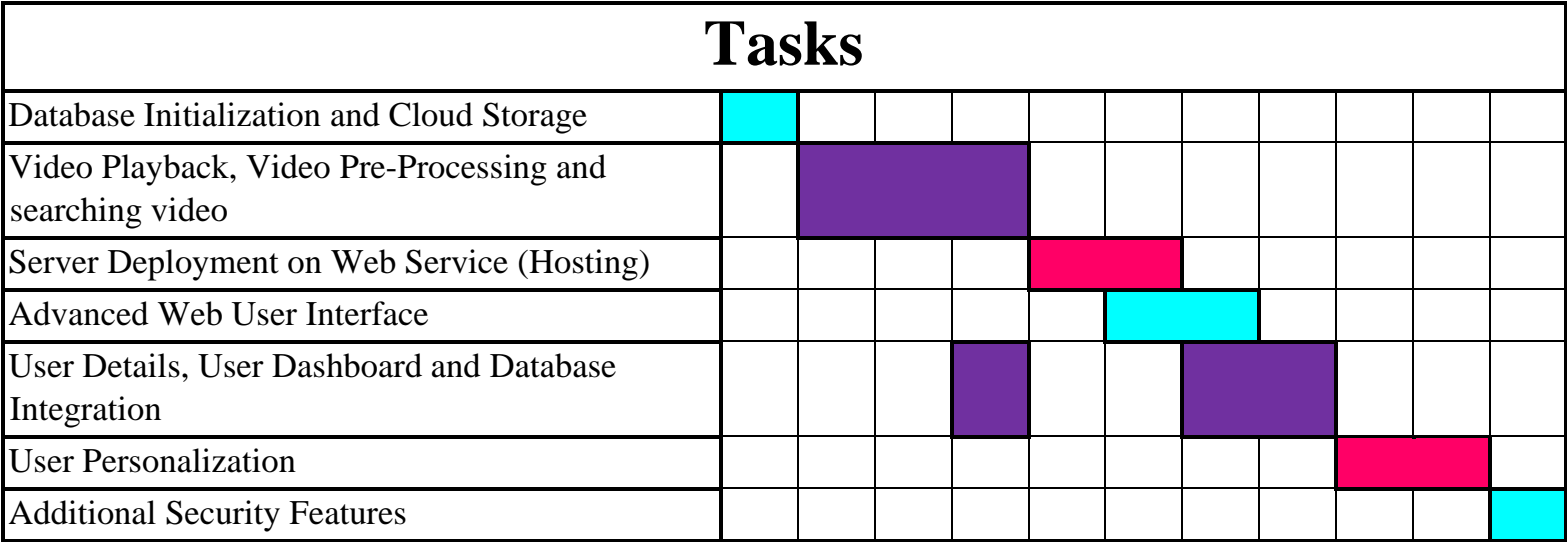
This estimate assumes that there are 3 developers working for a total of 2 months, thus giving us 6 person months of time to execute this project.

The stages for the chosen life cycle are:

Tasks	Person Months
Software Requirement Specification	0.5
Project Plan	0.5
Database Initialization and Cloud Storage	1
Video Playback, Video Pre-Processing and searching video	1
User Personalization	1.5
Advanced Web User Interface	0.5

Server Deployment on Web Service (Hosting)	0.25
Additional Security Features	0.75
<b>Total</b>	<b>6</b>

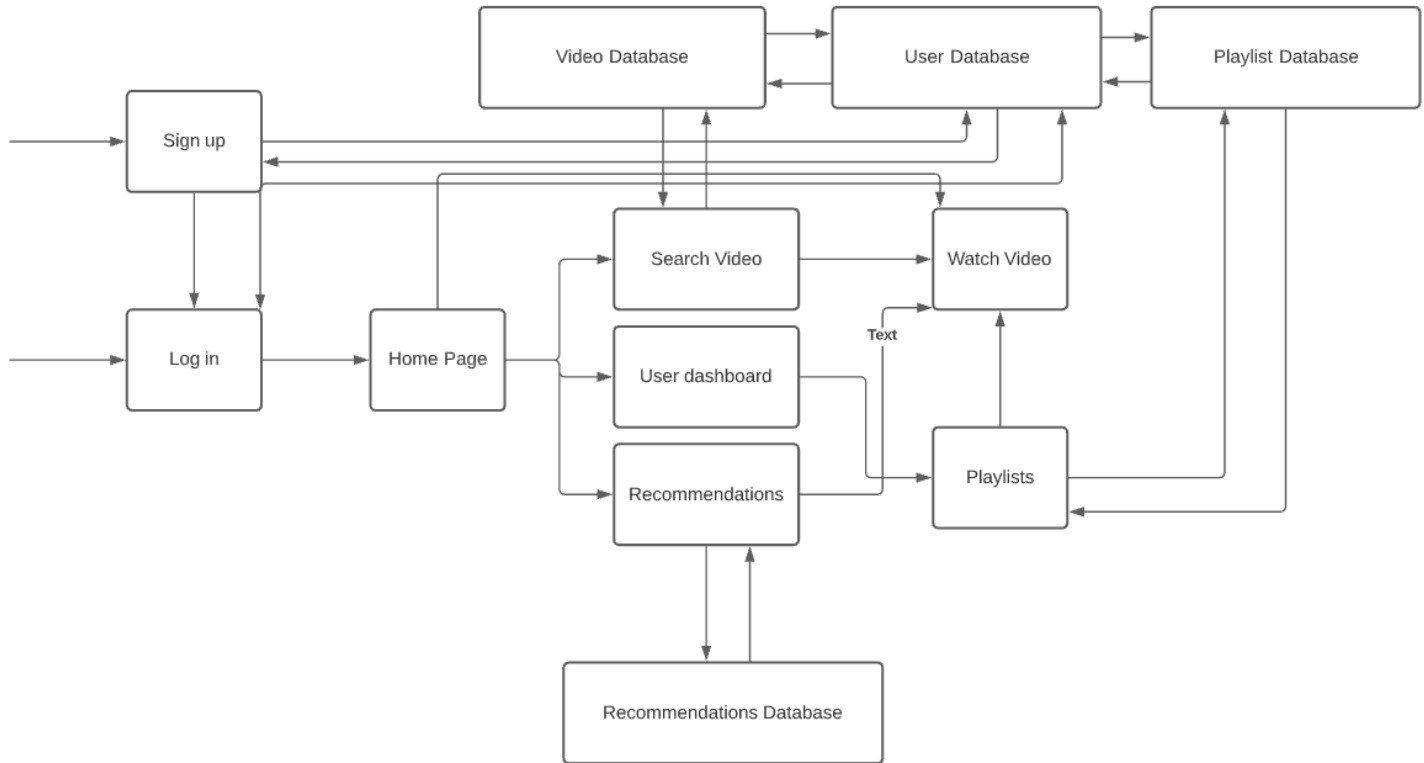
Gantt Chart:



Phase Leader	
Member 1	█
Member 2	█
Member 3	█
	█

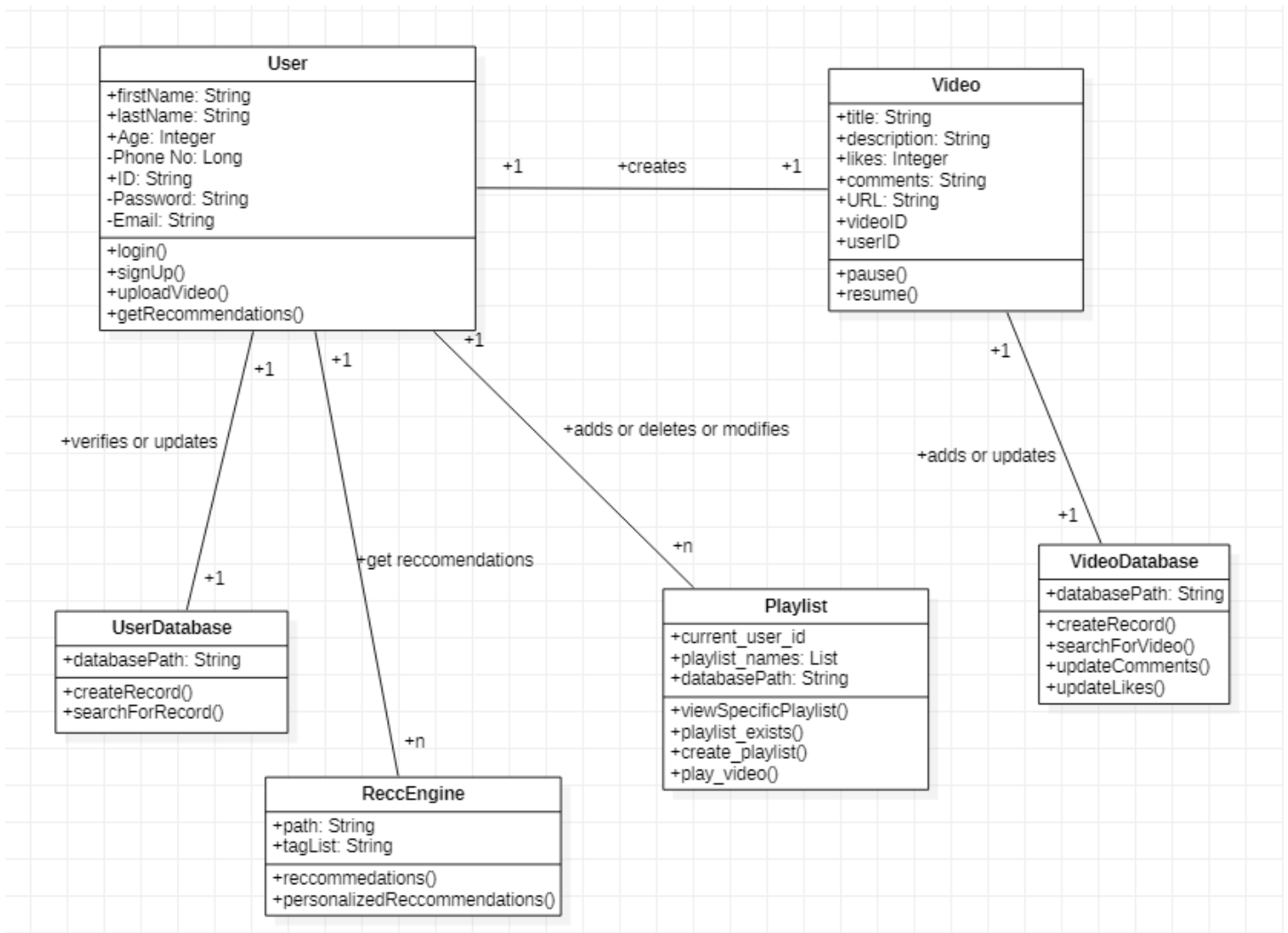
# Chapter 3: Design Diagrams

## 1. High Level System Architecture



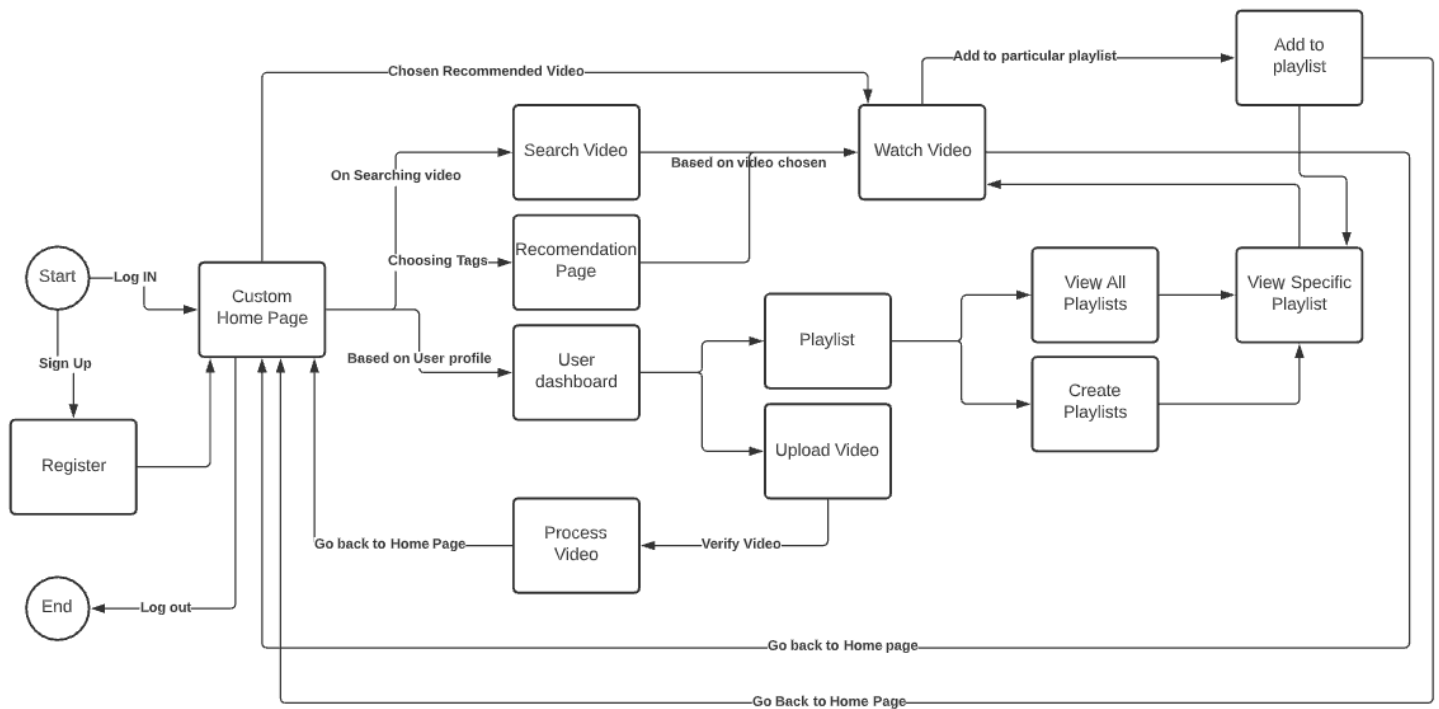
The diagram represents a general overview of the architecture of the video streaming platform with a few chosen use cases. As can be seen in the diagram, there are four databases each with a unique purpose. The video database holds video information and the User database stores user information. Playlist database maps users to the videos they like/have added into their playlist and the recommendations database has metadata information of every video that helps in recommendation. The remaining components show the general interaction with each other.

## 2. Class Diagram



The diagram represents the implementation of the video streaming service. The above diagram shows the different classes and the classes they are associated with. It also shows the relationship between the classes. The user class serves as the most important for storing and accessing used information. It is from here that the User database, Recommendation engine and playlist's are accessed. It also is associated with the video class as a User can upload videos and these videos after verification are put into the Video database.

### 3. State Diagram

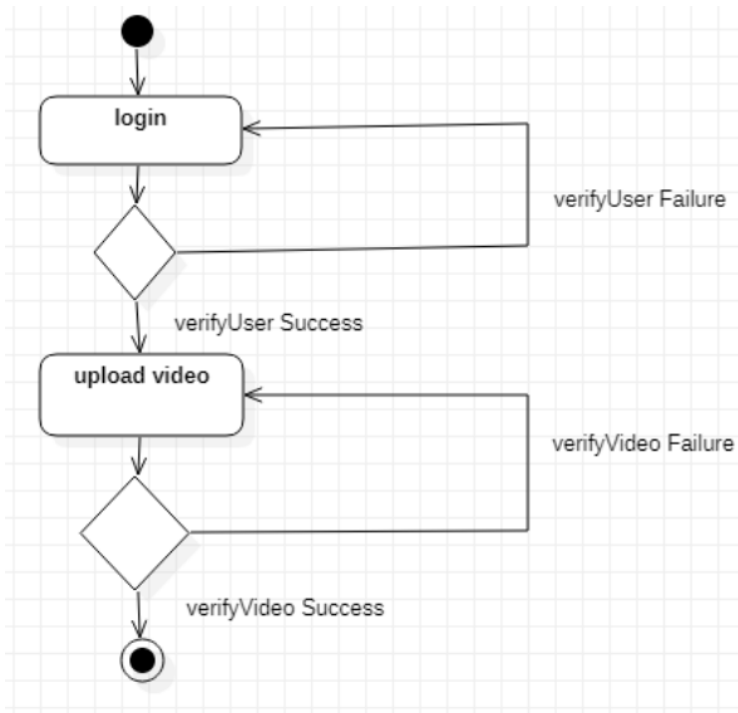


This diagram indicates the different states and the path taken internally by our system to reach that state. This diagram is essentially a more complex version of the High level system architecture and indicates a more thorough interaction with the different components of our system. As can be seen, upon logging in(New user has to sign up and then log in), the user is taken to the home page with a bunch of recommended videos. He/she can either view these videos, search for a specific video or go to his dashboard. At his dashboard, he/she can View/add playlists or upload a new video. The diagram shows all the interactions and the different states the streaming service can be in at any given point in time.

## 4. Activity diagrams

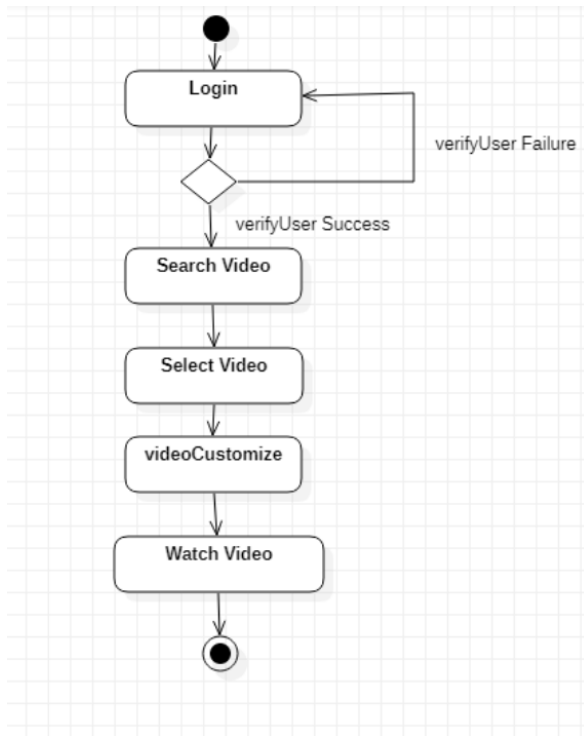
### ● Upload Video

The below diagram represents the activity diagram for the specific case of uploading a video. It shows the control flow of a system when a user tries to upload a video. In order to upload a video, a user must be logged in. Only in this case, will the user be allowed to upload the video. Once the user chooses to upload the video, the video is processed and in case the upload onto database fails, the user is prompted an alert.



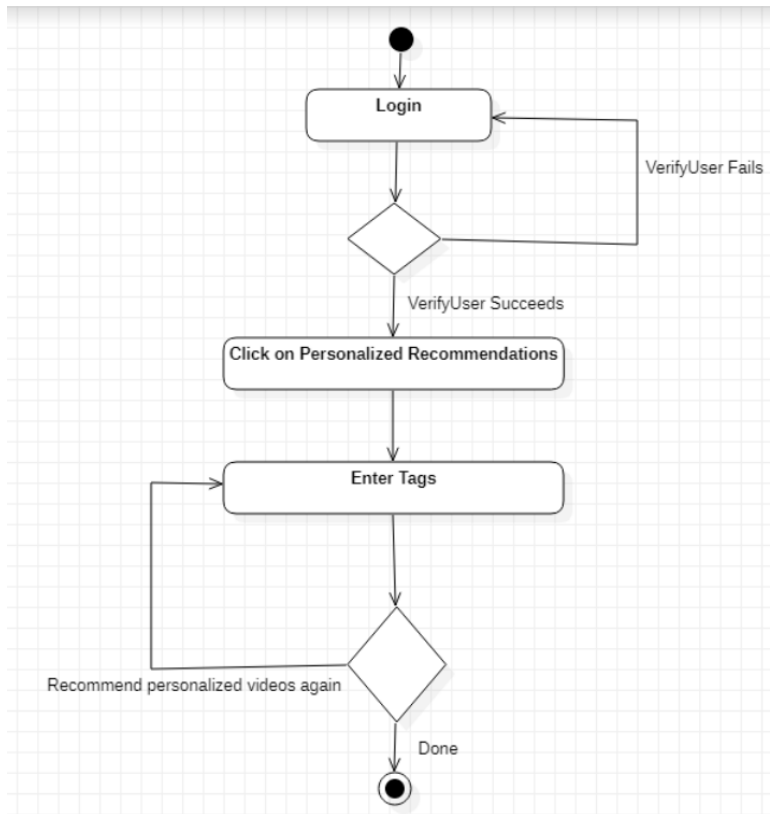
- **Watch Video**

The below diagram represents the control flow of the system when a user is trying to watch a video. As in the necessity in our implementation, the user needs to login first to be able to view any videos. Once he does so, he can Search for the video he wants to watch.



- **Recommendation engine**

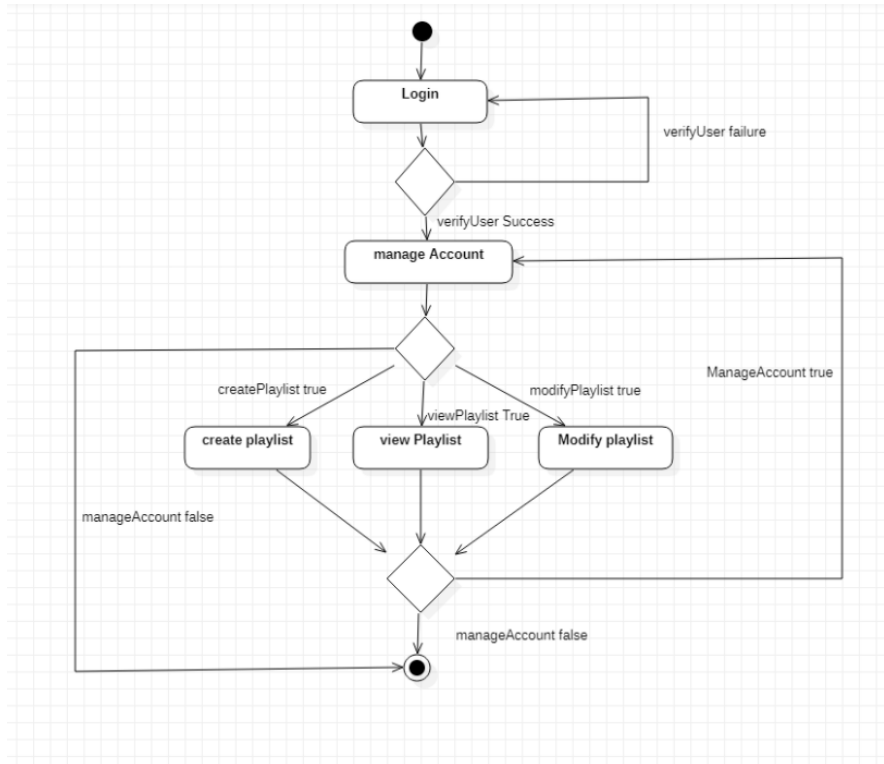
As is a requirement, the user needs to first login to be able to use the recommendation engine. He/she can then view videos that are recommended to him on his/her home page or choose from a bunch of custom tags to watch videos that are recommended based on the chosen tags. This process can be repeated multiple times until the user can finally chose what he/she wants to watch.



- **Manage Playlist**

The below activity diagram shows the control flow of using playlists. As is to all previous tasks, the user must have logged in. At this point, the user can choose between creating a new playlist, viewing a playlist and modifying the contents of the playlist (Modifying the content includes adding and deleting videos from a playlist). The user can choose among these choices multiple times as long as he/she desires.

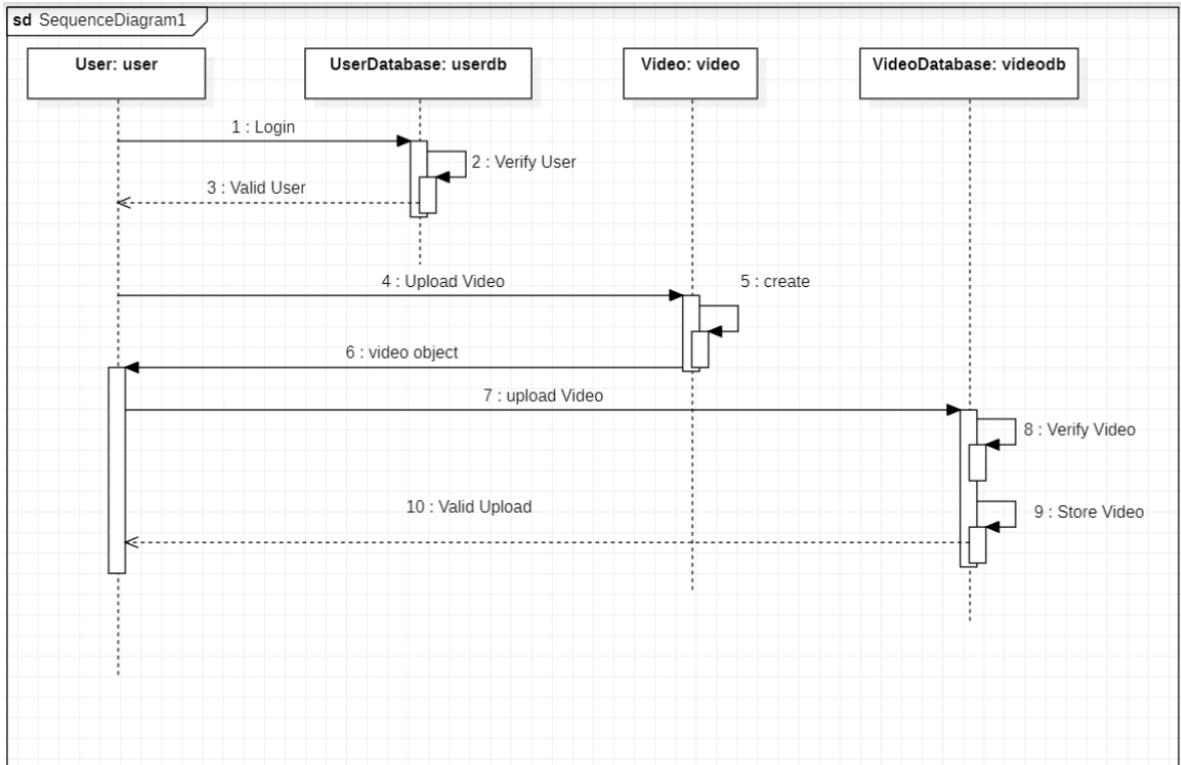




## 5. Sequence Diagram

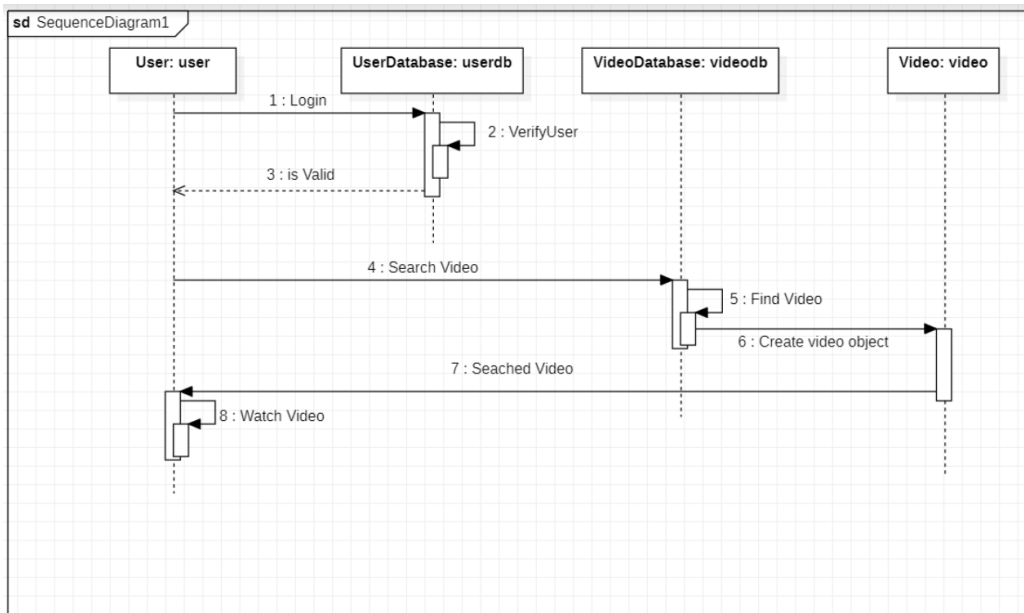
- **Upload Video**

The diagram shows the interactions between the different components of the System when the user chooses to upload a new video. First, the user needs to login and his/her credentials are checked in the user database. In order to upload a video, a user choses the video he wants to upload. The video is processed and if successful it is stored in the video database otherwise the user is notified of the failure.



## • Watch Video

The diagram shows the interaction with the different components of the system when the user tries to watch a video. As is a requirement, the user needs to be logged in. In this process, his/her credentials are checked in the User database. The user can now search for the video based on its title and if the video exists in the Video database, the video can be watched otherwise the user is notified that such a video does not exist.

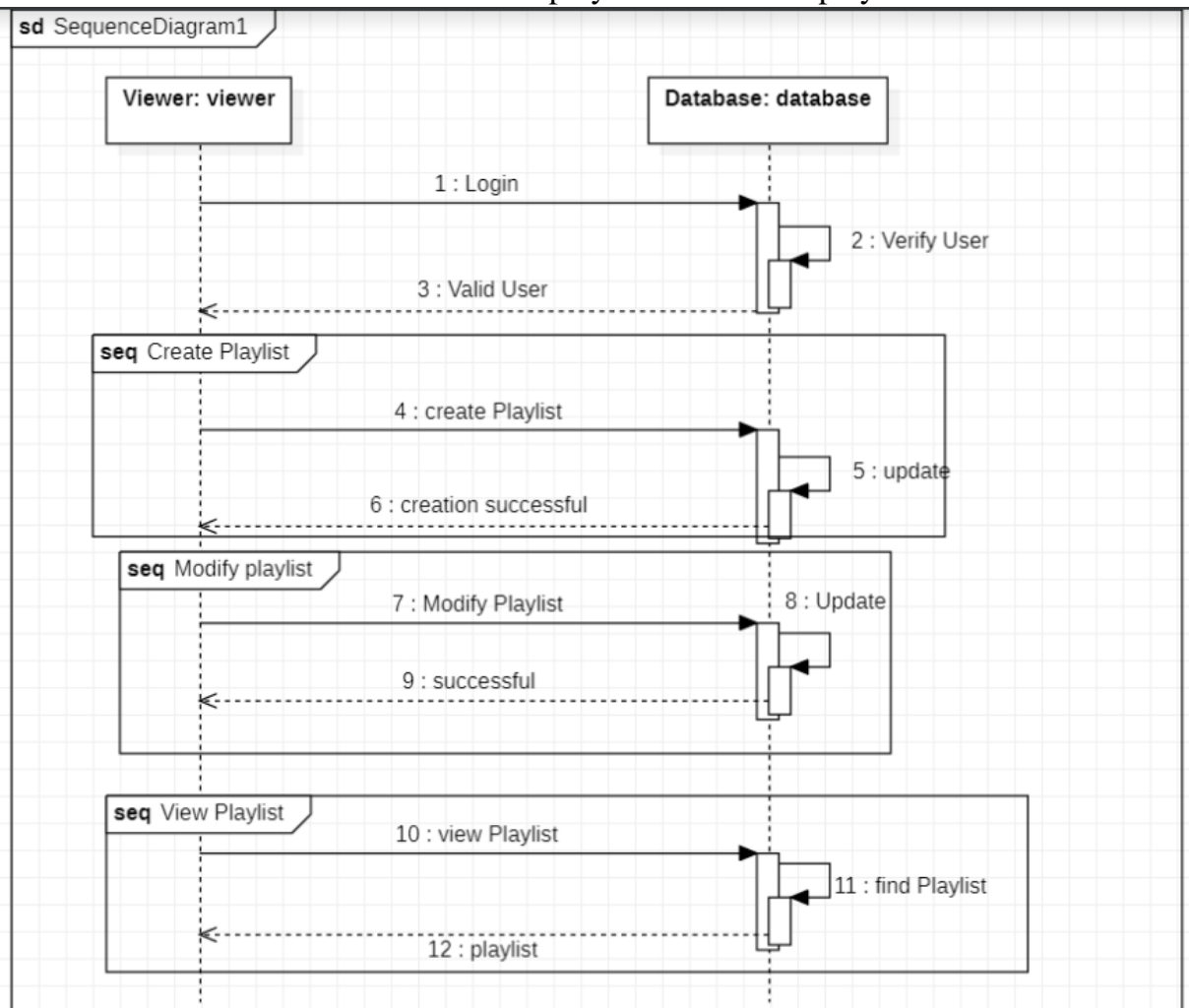


- **Manage Playlist**

The below diagram shows the interaction with the different components when the user is dealing with playlists. A standard requirement is that the user needs to be logged in. When a user wants to create a new playlist, a check is made to see if such a playlist already exists. If not, the database is updated. Else, the user is notified.

When the user wants to upload a new video into a specific playlist, a check is made to see if the playlist already has the video. If it already exists, the user is notified; if not the database is updated.

The user can also search and choose his playlist in the view playlist section.



# Chapter 4: Module Description

## 1) Login/Signup

The most important feature implemented in the login/signup feature. The user cannot interact with the system at all unless he/she has created an account. In order to sign up, the user needs to enter his full name, email id, set up a password, enter his/her phone number and age. The email and password are the most important fields as these are used while logging in also. The personal information of a user is written into the User database in a json format.

All fields must be entered during the signup phase. The credentials entered should strictly adhere to the specific format of the field they are being entered into.

The login feature allows for the development of several other features within the system that include playlist management, uploading a video and personalized recommendations as well.

## 2) Recommendation Engine

An important component of the system is the ability for it to recommend videos. The recommendation itself is implemented in two phases. In the Home page, the user is recommended a bunch of videos that are likely to draw his attention. He/she can choose a video from these recommended videos and watch them. These recommendations in the Home page are dynamic and change every time. A point to note here is that since the dataset that the recommendation engine works on is extremely small, sometimes the same videos might get recommended; but the system works perfectly as a proof of concept.

The second approach for implementing the recommendation engine includes selection of tags. The user can choose from a bunch of tags that he/she is interested in watching and based on this information, the recommendation engine provides relevant videos that the user can choose from.

## 3) Search/Interact with videos

The most important components of the video streaming service is to be able to watch the videos. This can be done with a few ways. Aside from having the recommendation engine recommend videos for the user to watch, the user can search for a specific video based on the title. If such a video exists, the user can watch this video. Additionally, the user can interact with this video in a number of ways. For instance, the user is capable of liking and commenting on the video. Liking and commenting on the video reflect globally and can be seen across all the users of the streaming service. Additionally, the user can choose to add this to his/her playlist.

#### **4) Managing Playlist**

Playlists are of significant importance to the user experience on any giving streaming service. In this system, the user is capable of creating a number of unique playlists. To each of these playlists, the user is capable of adding any video he would like to add. An important thing to note here is that if a particular video already exists within the playlist, the user is alerted about this and asked if he/she would like to remove the video from the playlist.

At any given point in time, the user can access all of his playlists and the content of each of these playlists. He can directly chose to play the video from within the playlist.

It is also important to note that while creating a playlist, the playlist's name must be unique to that particular user's account. Same playlist names will not be allowed.

# Chapter 5: Test Cases

## Test Cases - login/signup/viewing video/searching video/uploading:

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-01	User login	To test the login functionality	Access to the Welcome Page	1. Navigate to the Welcome Page 2. Click on "Login" button 3. Enter "email" and "password"	Email: dhanya@gmail.com. Password : dhans123	Login should be successful and automatic redirection to Home Page	Login successful and automatic redirection to homepage	Pass
UT-02	User sign up	To test the signup functionality	Access to the Welcome Page	1. Navigate to the Welcome Page 2. Click on "Signup" button 3. Enter "email", "password", "first name", "lastname", "age" and "phone no."	Email: <a href="mailto:chethan@gmail.com">chethan@gmail.com</a>  Password: chethu123  firstName: Chethan  lastName: Mahindrakar  age: 20 phone no: 9988998765	Sign up should be successful with a "Account successfully created. Login to use the application!" success message displayed	Sign up successful with Account successfully created. Login to use the application!" success message displayed	Pass
UT-03	User sign up	To test the signup functionality	Access to the Welcome Page	1. Navigate to the Welcome Page 2. Click on "Signup" button 3. Enter "email", "password", "first name", "lastname", "age" and "phone no."	Email: <a href="mailto:chethan@gmail.com">chethan@gmail.com</a>  Password: chethu123  firstName: Chethan  lastName: Mahindrakar  age: 20 phone no: 9988998765	Sign up should be unsuccessful with a "A user with the same email address exists!" message	Sign up is unsuccessful with a "A user with the same email address exists!" message	Pass

IT-01	Search Video and Watch Video	To test the search video functionality integrated with watch video functionality	User must have logged in. Access the Home Page	1.Navigate to the Home Page 2. Click on“Search Video” button 3. Search for thevideo by entering the video title in the search text box 4. Click “Search”button	Search field: “Funny Puppy”	Search should be successful and the user should be directed to the Watch Video page with all the information of the searched	Search should be successful and the user should be directed to the Watch Video page with all the information of the	Pass
-------	------------------------------	--	--	---	-----------------------------	--	---	------

						video i.e <b>Video, Title, Description, Likes, Comments, Channel</b>	searched video i.e Video, Title, Description, Likes, Comments, Channel	
UT-04	Search Video	To test the search video functionality	User must have logged in. Access the Home Page	1.Navigate to the Home Page 2. Click on“Search Video” button 3. Search for thevideo by entering the video title in the search text box	Search field: “Monkey Video”	Search should be unsuccessful. An error message of “No video of this title” must be displayed”	Search is unsuccessful. Error message of “No video of this title” is displayed	Pass
UT-05	Search Video	To test the search video functionality on invalid input	User must have logged in. Access the Home Page	1.Navigate to the Home Page 2. Click on “Search Video” button	Search field: empty	Search should be unsuccessful and an error message “Enter the search field” should be displayed	Search is unsuccessful and the error message “Enter the search field is displayed”	Pass

UT-06	Watch Video	To test the play video functionality	User must have logged in. Be on the Watch Video Page	1. Navigate to Watch Video Page	1. Log in as “ <a href="mailto:dhanya@gmail.com">dhanya@gmail.com</a> ” 2. Navigate to Watch Video page for “Funny Puppy” video 3. Click on “Play Video” button	The video must be played in the video player.	The video is played in the video player.	Pass
IT-02	Watch Video	To test the likes functionality for a user who has not liked the video before	User must have logged in. Be on the Watch Video Page	1. Navigate to Watch Video Page	1. Click on “Like” button	Like counter should be incremented in database and displayed	The <b>like</b> counter is incremented and displayed	Pass
IT-03	Watch Video and Video Database	To test the likes functionality for a user who has already liked the video	User must have logged in. Be on the watch	1. Navigate to Watch Video page	1. Log in as “ <a href="mailto:dhanya@gmail.com">dhanya@gmail.com</a> ” 2. Navigate to Watch Video page for	Like counter should not be incremented	The like counter is not incremented as the user has already liked the	Pass

			video page		“Funny Puppy” video  3. Click on “Like” button		video before	
--	--	--	------------	--	--	--	--------------	--



IT-04	Watch Video and Video Database	To test the comments functionality	User must have logged in. Be on the watch video page	1.Navigate to Watch Video page	1. Log in as <a href="mailto:dhanya@gmail.com">"dhanya@gmail.com"</a> 2. Navigate to Watch Video page for "Funny Puppy" video 3. Enter the comment text in the text field 4. Click on "Add Comment" button	Comment list of the video must be appended with the latest comment in the database and must be displayed	The comment list of the video is appended with the latest added comment in the database. The comment is displayed along with the other pre-existing comments for the video	Pass
UT-07	Watch Video	To test comments functionality when invalid input is given	User must have logged in. Be on the watch video page	1.Navigate to Watch Video page	1. Log in as <a href="mailto:dhanya@gmail.com">"dhanya@gmail.com"</a> 2. Navigate to Watch Video page for "Funny Puppy" video 3. Click on "Add Comment"	Comment addition must be unsuccessful and the error message "Enter the comment field" must be displayed	Comment addition is unsuccessful and the error message "Enter the comment field" is displayed	Pass
UT-08	Upload Video	To test the upload video functionality	User must have logged in. Navigate to the Upload Video Page	1. On the home page, click on "View User Dashboard" 2. Click on "Upload Video" button 3. On the Upload Video Page, enter the "title", "description" and upload video file.	Title : "Brain 101" Description:" Video about the brain"	The video should be uploaded successfully and a "Video successfully uploaded" success message is to be displayed	The video is uploaded successfully and a "Video successfully uploaded" success message is displayed.	Pass
UT-09	Upload Video	To test the upload video functionality, when there is already a	User must have logged in. Navigate to the Upload	1. On the home page, click on "View User Dashboard" 2. Click on "Upload Video" button	Title : "Space 101" Description:" Video about the universe"	The video upload should be unsuccessful and the error message "A video	The video upload is unsuccessful and the error message "A video with the	

		video in the		3. On the Upload				
--	--	--------------	--	------------------	--	--	--	--

		database with the same title	Video Page	Video Page, enter the "title", "description" and upload video file.		with the same title already exists" must be displayed	same title already exists" is displayed	
ST-01	End to End Search Video	To login and search for a video and play it	-	1. login with a valid email and password , 2.click on "Search Video" button on the Home Page, 3. enter the title tosearch for in the Search Page, 4. click on the"Play Video" button in the Watch Video page 5.. Click on "Go back to homepage" button 6.click on "Logout"	1. Log in as " <a href="mailto:dhanya@gmail.com">dhanya@gmail.com</a> "  2. Click on "Search Video" 3. Enter "Funny Puppy" in search field 4. CLick on "Play Video" button	Video must successfully plas.After watching, user must successfully log out	Video successfully plays.After r watching, user successfully logs out	Pass
ST-02	End-to-End Upload Video	To login and upload a video	A video to upload	1.Login with valid email&password 2.Click on "View User dashboard" button on Home Page 3. Click on "Upload Video" button 4. Enter videotitle,description and .mp4 video file 4. Click on "Upload" button 5. Click on "Go back to home page" 6.. Click "log out"	1. Log in as " <a href="mailto:dhanya@gmail.com">dhanya@gmail.com</a> "  2. Title: "Space Boy" 3. Description:" Space music video"	Video must successfully upload, user must be able to successfully logs out of the system	Video successfully uploaded and user logs out of system``	Pass

### Test Cases - Recommend Similar Videos/Homepage:

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-10	Recommending trending videos on Homepage	To test if recommendations are working on the homepage	Must have logged in	-	-	Display 4 popular videos on homepage	Displays 4 popular videos on homepage	Pass
UT-11	Getting to personalized recommendations page	To check user can go to next page on clicking button	Must have logged In	click on Get Personalized Recommendations	-	Display page with ability to personalize recommendations	Displayed page with ability to personalize recommendations	Pass
UT-12	Getting Personalized recommendations	To check if user gets personalized recommendations	Must have logged in and clicked "Get Recommendations" button	Enter the tags (max 2) from displayed list	Any 2 tags from displayed list, seperated by ,	Display 2 videos similar to tags	Displays 2 videos similar to tags	Pass
UT-13	Getting personalized recommendations - pers_recc	To traverse from homepage to end with 2 videos similar to tags entered	Must have logged in	From homepage, click button "Get Personalized recommendations" enter tags from given list enter	Any 2 tags from displayed list separated by ,	Display 2 videos similar to tags	Display 2 videos similar to tags	Pass
IT-05	Playing a recommended Video on Homepage	To play a video recommended to user on homepage	Must have logged in	Click on a picture button of whatever video you want to watch, and then select play video in the viewer	-	Plays the Video	Plays the Video	Pass
IT-06	Playing a personalized recommended video	To play a video, personally recommended based on tags inserted by user	Must have logged in	Click on "Get personalised recommendations" button and enter any 2 tags from given list of possible tags, click on the picture and press play video	Any 2 Tags from the displayed list, seperated by ,	Plays the video	Plays the video	Pass

IT-06	Playing a video by clicking a recommended video	To play a video by clicking the picture displayed (either on homepage or personalised recommendations)	Must have logged in, or must have entered 2 tags from given list	Click on picture, click on play video	-	Plays the Video	Plays the Video	Pass
-------	---	--	--	---------------------------------------	---	-----------------	-----------------	------

UT-14	Going back to dashboard before or after personalised recommendations	To go back to user dashboard once we finish giving out personalised recommendations	Must have logged in and clicked on “get Personalised Recommendations”	Click on “Back to Dashboard”	-	Goes back to homepage	Goes back to homepage	Pass
UT-15	Going to homepage after getting personalised recommendations - from homepage	To start at homepage, get personalised recommendations and go back to homepage	Must have logged in	Click on “Get Personalised Recommendations” button, enter any 2 tags from the given list, click on “get recommendations” and then click the “Back to homepage” button	Any 2 tags from list of given tags	Goes back to homepage safely	Goes back to homepage Safely	Pass
UT-16	Going back to homepage without entering any data in tags entry field	To go back to homepage without entering tags in the entry field	Must have logged in and clicked on “get personalised recommendations” button	Click on “back to homepage”	-	Goes back to homepage safely	Goes back to homepage safely	Pass
ST-03	End to end playing a recommended video on homepage	To login, play a recommended video on homepage	-	Must login by entering valid email password, click on a picture that appears on homepage and click on play video	-	Plays Video	Plays video	Pass

ST-04	End to end playing a personally recommended video	To login, get a personally recommended video and play it	-	Must login by entering valid email password, click on “Get personalized recommendations” Button, enter any 2 tags from given list of valid tags and click “Get recommendations” , clicks the picture of the video they want to watch and then clicks play video	Any 2 tags from given list	Plays video	Plays Video	Pass
-------	---	--	---	---	----------------------------	-------------	-------------	------

### Test Cases - View Playlist, Create playlist, Add/remove from playlist.

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-17	Creating a new playlist with a unique name.	Test to see if a new playlist can be created.	Must have logged in	Log in and navigate to ‘View User Dashboard’. Navigate to ‘Playlists’ and then go into Create Playlist.	Playlist name	New playlist must show in the database	Shows playlist in the database as well as in View Playlist.	Pass
UT-18	Creating a playlist with name that already exists	To see if a playlist can be created if the name is not unique.	Must have been logged in and a playlist with such name should already exist.	Log in and navigate to ‘View User Dashboard’. Navigate to ‘Playlists’ and then go into Create Playlist.	Playlist name	Error message saying playlist exists must show	Error message saying playlist exists must show	Pass
UT-19	Adding a video to an already existing playlist	To add a video into the playlist assuming the video is not in the playlist	Must have logged in and video must not already exist in the playlist	Log in and navigate to a video watch page. Click on the button to add into the playlist and choose the playlist you want to add to.	Playlist name to add into and video name.	Message saying that playlist has been successfully added. Change in the database must be observed.	Data added to the database. Message shows success.	Pass

UT-20	Adding a video to an already existing playlist.	To add a video into the playlist assuming that video already is in the playlist	Must have logged in and video must already exist in the playlist.	Log in and navigate to a video watch page. Click on the button to add into the playlist and choose the playlist you want to add to.	Playlist name to add into and video name	Message saying that playlist already has the video. No changes in the database. Prompt to remove playlist from video.	Data not added to database. Message shows failure and prompts for deleting the video from the playlist is shown	Pass
UT-21	Deleting a video from within a playlist.	To delete a video if it already exists in the playlist.	Must have logged in and video must already exist in playlist.	Log in and navigate to a video watch page. Click on the button to add into the playlist and choose the playlist you want to add to.	Playlist name to add into and video name	Message showing that playlist already has video. Prompt to delete video and message showing successful deletion if	Message showing video already exists in playlist. Button asks for deletion. If chosen, message that says deletion confirmed. Data removed	Pass

						button is clicked.	from database.	
IT-07	Viewing all the playlists of a particular User.	To view the playlists that a user has in his account	Must have logged in and must have created playlists for output to show.	Log in and navigate to User dashboard. From here, navigate to Playlists and view playlists.	Current User information	All user playlists must be shown to the user	All playlists are shown to the user in the form of clickable buttons	Pass
IT-08	Viewing the videos that are a part of the playlist	To view all the videos that a user has put into his playlist.	Must have logged in and must have videos inside his/her playlist	Log in and navigate to User dashboard. Navigate into view playlists from here and click on playlist whose contents need to be seen.	Current User information	All videos within the playlist must be shown to the user	All videos in the playlist are shown to the user in the form of clickable buttons.	Pass
UT-22	Play a video in the playlist.	To watch the video from within a user's playlist.	Must have been logged in and must have video in playlist	Log in and navigate into View playlist. Click the playlist and select the video to watch	Current User information	The video selected must be played	The chosen video is played on a default video playing app.	Pass

ST-05	Navigating to Home page from User dashboard	To navigate back to the home page from the user dashboard.	Must have logged in.	Navigate into the User dashboard page and click on return to Home page.	-	Must be navigated into the home page screen.	Home page screen shows up after clicking the button.	Pass
ST-06	Logging out from the User Dashboard	To log out from the User dashboard screen	Must have logged in	Navigate into the User dashboard page and click on log out.	-	Must be logged out	User is logged out and is taken to the Login or signup page	Pass
ST-07	Navigating to dashboard from the 'All Playlists' screen	To go back to the User dashboard from the 'All playlists' page	Must have logged in	Navigate into user dashboard and select All playlists. Here, click the 'Go back to dashboard' button	-	User must be taken back to User dashboard	User is successfully taken back to his dashboard page	Pass
ST-08	Navigating to Home page from the 'All playlists' screen.	To go back to the Home page from the 'All playlists' page	Must have logged in	Navigate into user dashboard and select all playlists. Here' click the 'go to home page' button.	-	User must be taken back to the Home page	User is successfully taken back to his Home page	Pass

# Chapter 6: Output Screenshots

## 1. The Login page

**Login Page**

Email/Username

Password

LOGIN

[Go back to Welcome Page](#)

**Login Page**

Email/Username

Password

LOGIN

[Go back to Welcome Page](#)

Invalid Login



## 2. The signup page

### Sign Up Page

Email/Username

Password

First Name

Last Name

Age

Phone Number

SIGN UP

Go back to Welcome Page

### Sign Up Page

Email/Username

Password

First Name

Last Name

Age

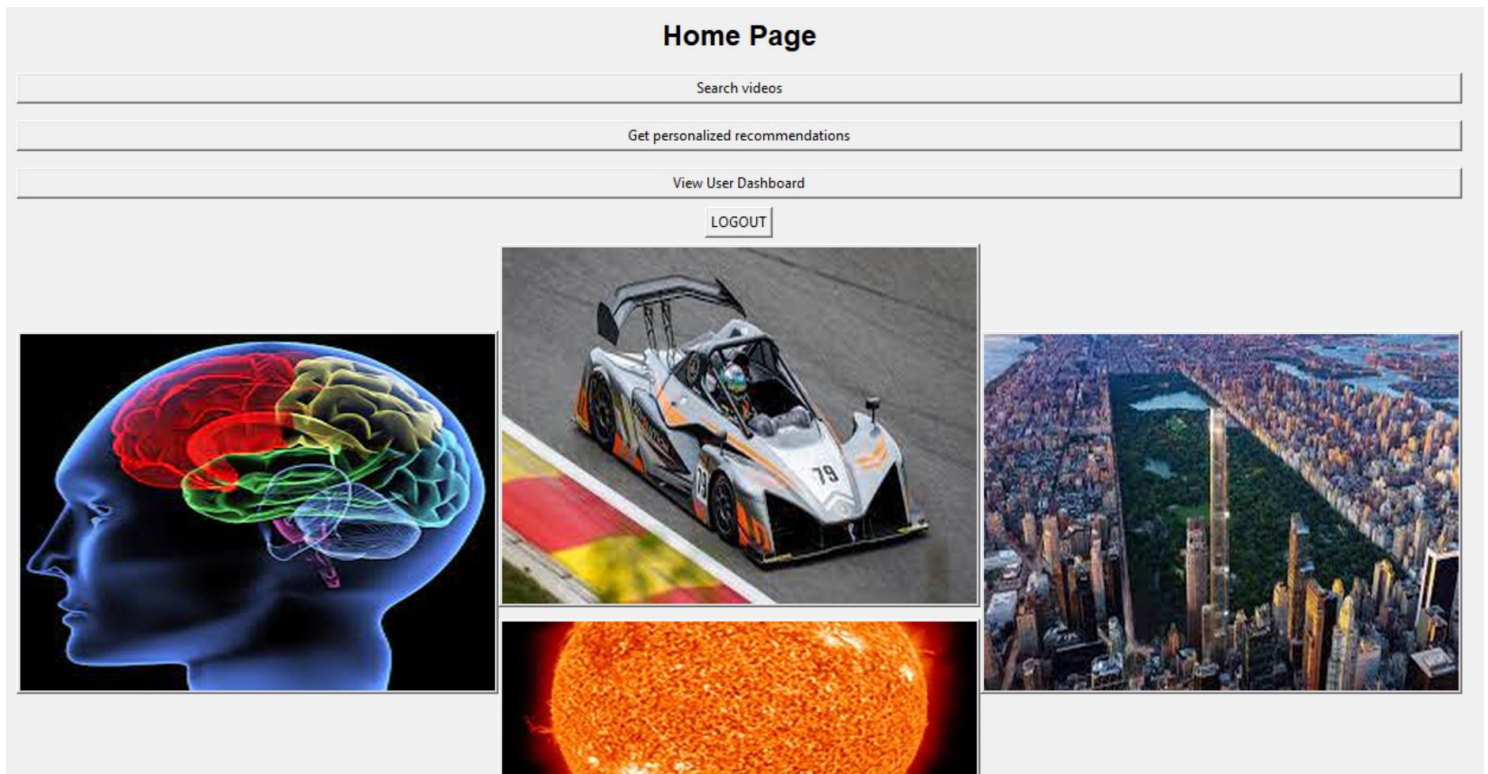
Phone Number

SIGN UP

Go back to Welcome Page

Enter ALL fields

### 3. Home page(Shows recommended videos as thumbnails)



### 4. Search Video Page

**Search Video**

Enter video title

Search

Go back to Home Page

**Search Video**

Enter video title

Search

Go back to Home Page

Enter the search field

No video by this title

## 5. Watch Video

### Watch Video

Title: Sun 101

Description: Take a trip thru the universe

Channel: Dhanya G

Play Video

Add to Playlist

No. of Likes: 1

LIKE

Comments:

Atmik A

upload more videos like this

i love space videos

Jane Doe

Love the video!!

Enter your comment:

ADD COMMENT

Go back to Home Page

## 6. Custom Recommendations



### Choose tags for Personalized recommendations

Please enter of of the following tags (any 2, seperated by ,comma):

- animals
- astronomy
- automobile
- big
- boat
- car
- city
- cute
- food
- funny
- life
- lol
- pets
- planet
- popular
- science
- ship
- smart
- trendy
- trndy
- universe
- warm
- wholesome
- world
- yummy

Get Recommendations!

Go back to HomePage



## 7. View Playlist

### All Playlists

playlist 1

playlist 2

air

Go back to dashboard

Go back to Home Page

## 8. Create Playlist

### Creating new playlist

Enter Playlist name

Playlist added

## 9. Upload video

### Upload Video

Title

Description

Select Video File