
Software Requirements Specification

for

Video Streaming

Version 1.0 approved

Prepared by:

Danish Mohammed Ebadulla(PES1201800096)

Rahul Raman(PES1201800146)

Joseph Dominic Cherukara(PES1201800328)

PES University

5 February 2021

Table of Contents

Introduction	1
Purpose	1
Intended Audience	1
Product Scope	2
References	2
Overall Description	2
Product Perspective	2
Product Functions	3
User Classes and Characteristics	4
Operating Environment	5
Design and Implementation Constraints	5
Assumptions and Dependencies	6
External Interface Requirements	6
User Interfaces	6
Software Interfaces	9
Communications Interfaces	9
Analysis Models	10
Login and sign up use case diagram	10
User dashboard use case diagram	10
Home page use case diagram	11
System Features	12
Registration	12
Login	13
Home Page	14
Video Playback	15
User Dashboard	16
Upload Videos	17
Subscribe Channel	18
Search Video / Channel	18
Like Video	19
Playlist	20
Other Nonfunctional Requirements	21
Performance Requirements	21

Safety Requirements	21
Security Requirements	22
Software Quality Attributes	22
Business Rules	23
Other Requirements	23
Appendix A: Glossary	23
Appendix B: Field Layouts	23
Report Requirements	24
Appendix C: Requirement Traceability Matrix	25

Revision History

Name	Date	Reason For Changes	Version

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

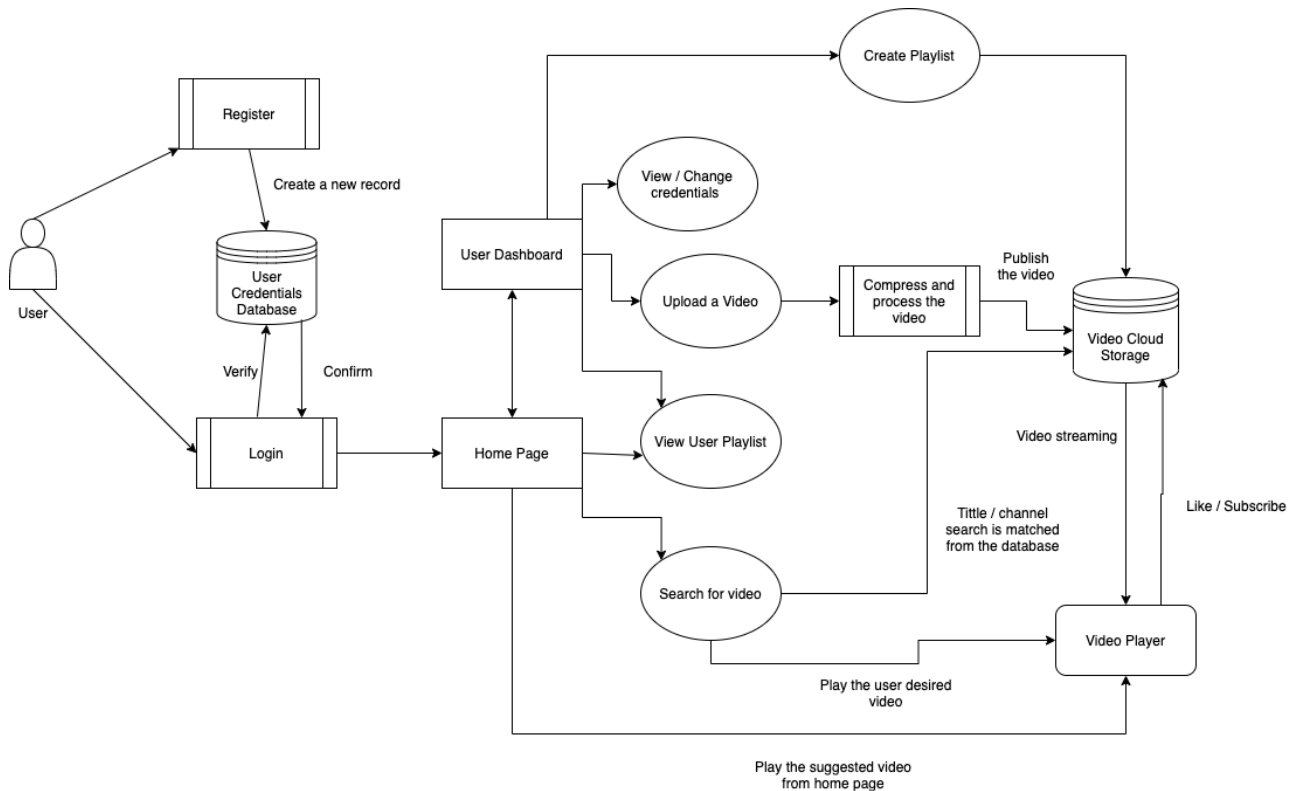
2.1 Product Perspective

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

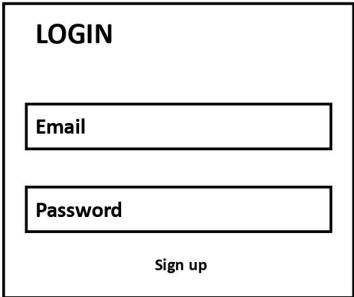
3.1 User Interfaces

- **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.

The wireframe illustrates the layout of the Home Page for a user who is not logged in. At the top left is a hamburger menu icon. To its right is a search bar with the placeholder text 'Search' and a magnifying glass icon. Further right is a 'Sign in' button. Below these elements is a grid of 12 rectangular boxes, arranged in 3 rows and 4 columns, representing video suggestions or content tiles.

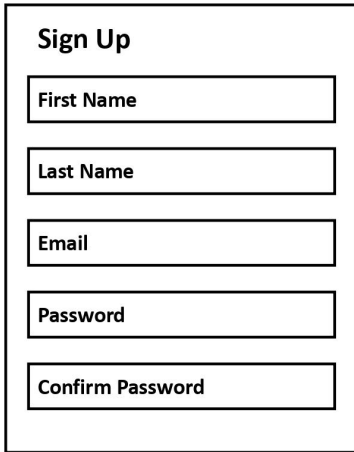
- **Login page:** Here the user is prompted to enter their login details (Email ID, password). Once the user's credentials are verified, the user is redirected to the

service's homepage. The user also has the option to create an account, if he/she wishes to do so .



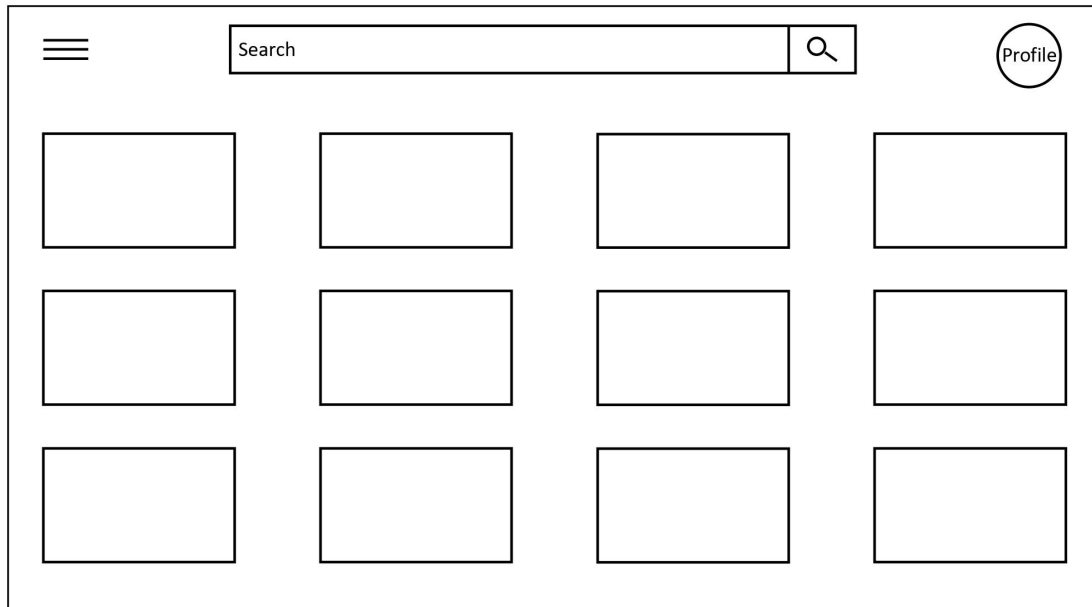
A diagram of a login form. It is a rectangular box with a black border. Inside, at the top, is the word "LOGIN" in bold. Below it are two input fields: the first is labeled "Email" and the second is labeled "Password". At the bottom of the box is a button labeled "Sign up".

- **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.

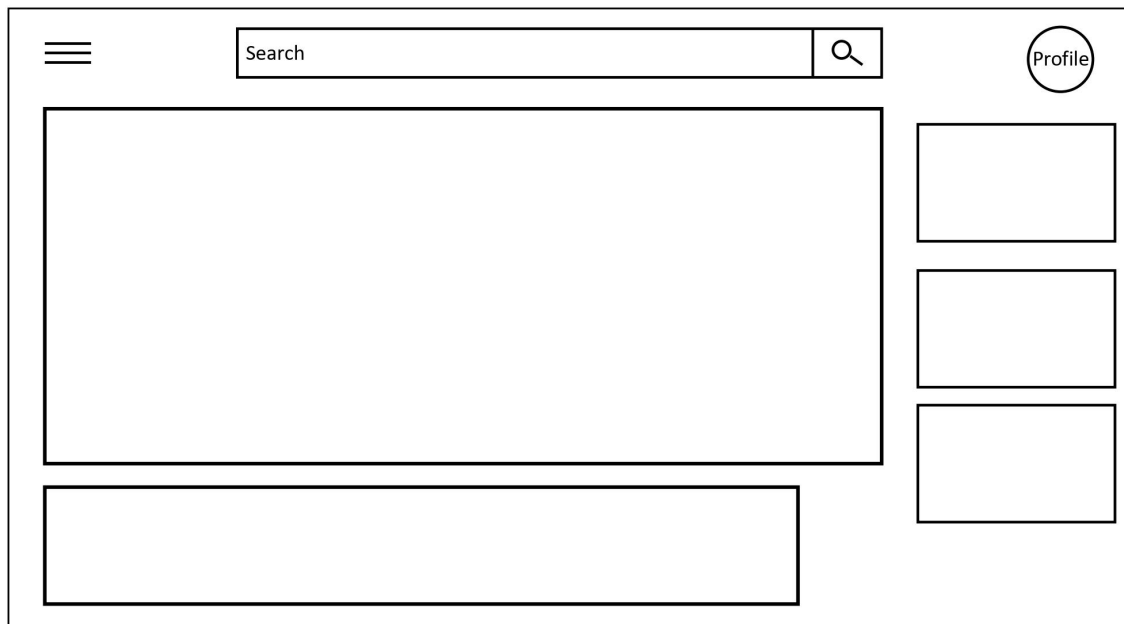


A diagram of a sign up form. It is a rectangular box with a black border. Inside, at the top, is the text "Sign Up" in bold. Below it are five input fields stacked vertically: "First Name", "Last Name", "Email", "Password", and "Confirm Password".

- **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.



- **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.

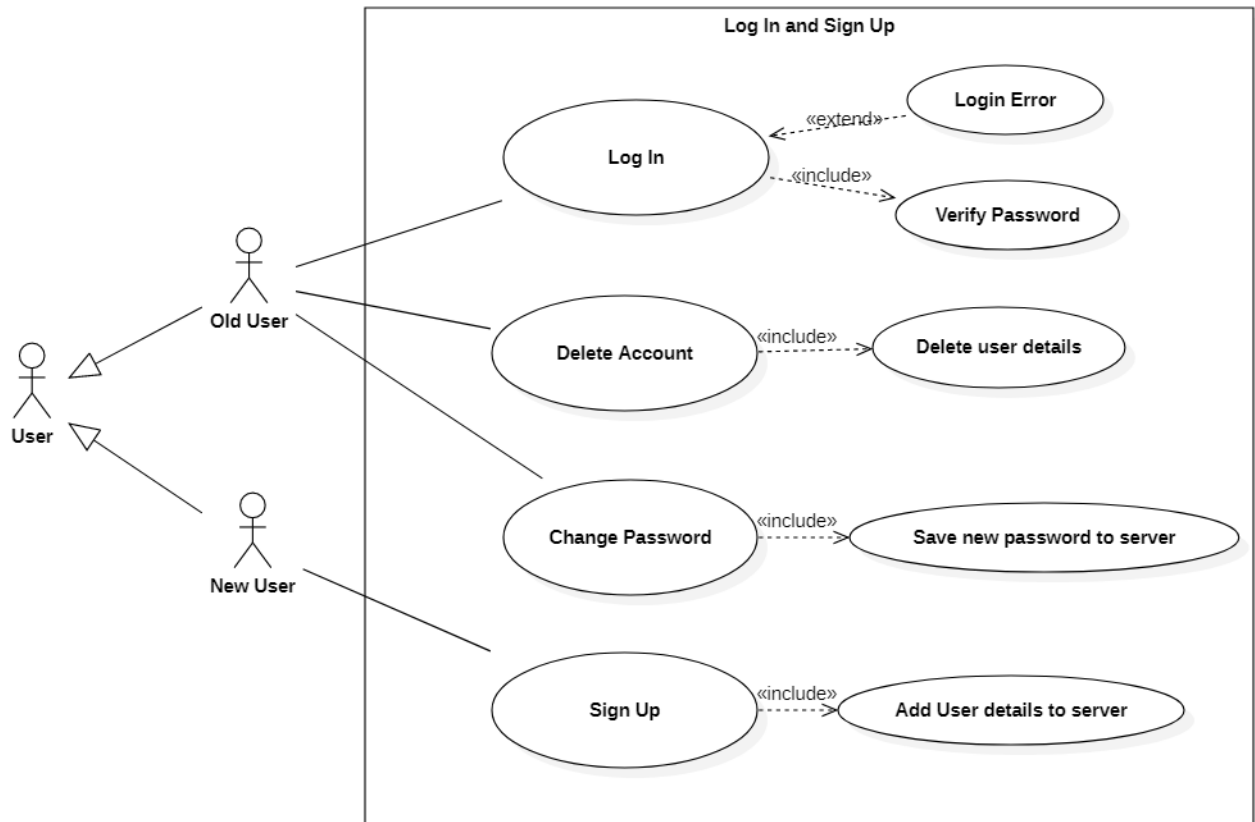
3.3 Communications Interfaces

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

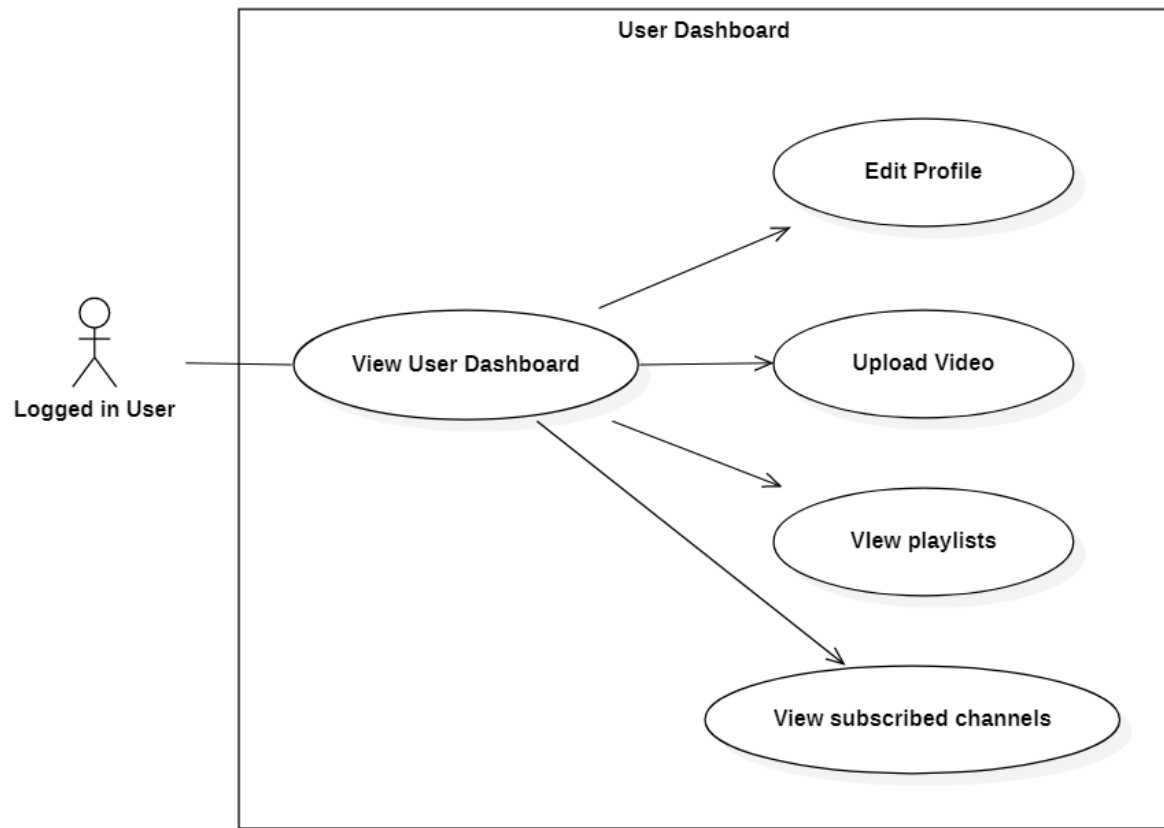
protocols such as SSL/TLS, as sensitive login information is sometimes sent across and this would need to be encrypted.

4. Analysis Models

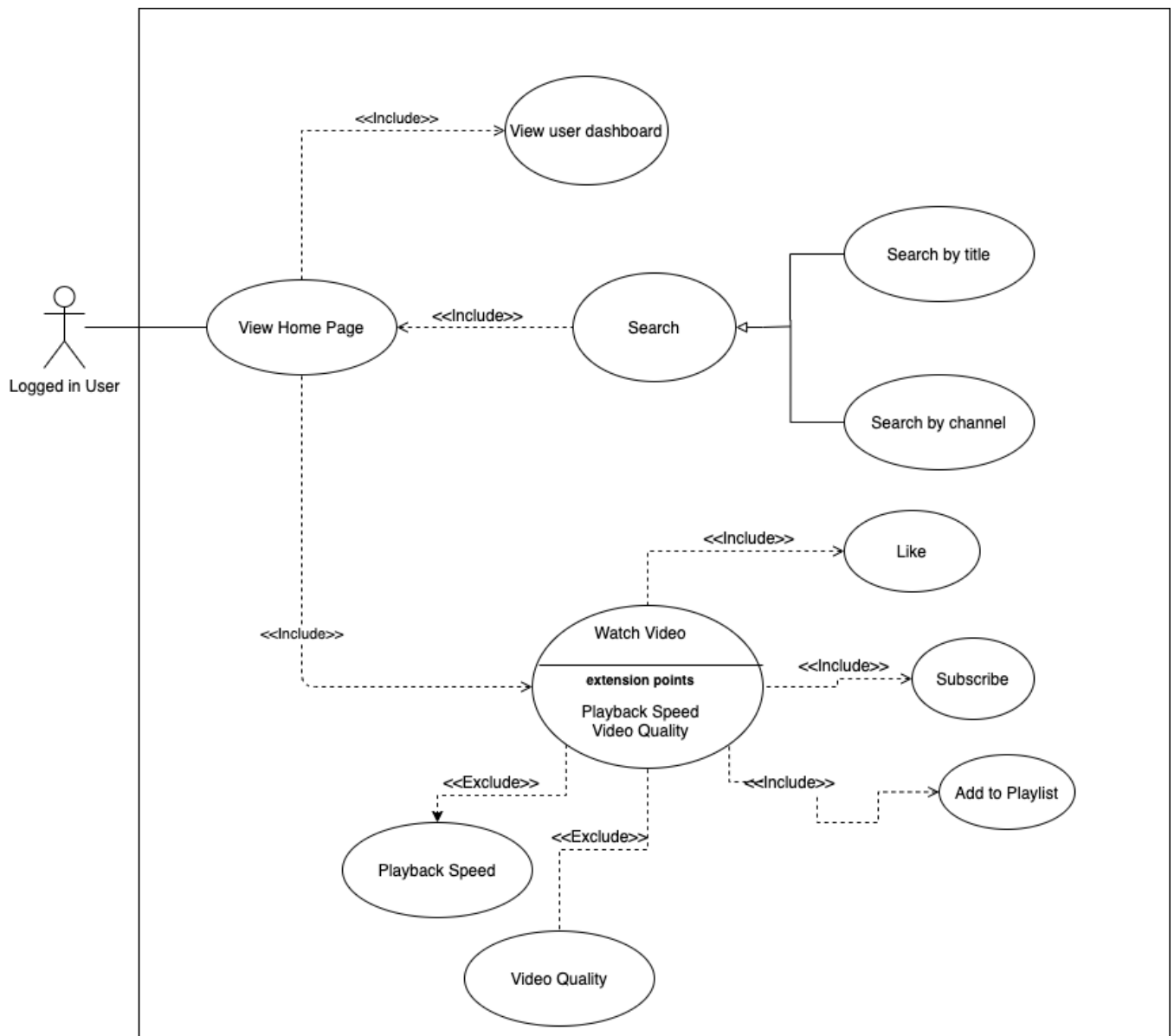
4.1 Login and sign up use case diagram



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.

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

User Credentials				
Field	Length	Data Type	Description	Is Mandatory
User Name	8-30	Alphanumeric	Unique name for each user	Y

Password	8-16	String	For accessing the account	Y
Email ID	8-30	String	For two step verification	Y

Video Database

Field	Length	Data Type	Description	Is Mandatory
Video Title	8-30	String	Used for searching video from the storage	Y
Creator Name	8-16	Alphanumeric	Automatically recorded when the creator publishes	Y
Publish Date	8	date-time	Automatically recorded when the creator publishes	Y
Number of likes	16	integer	Increments when a viewer likes the video	Y

Playlist Database

Field	Length	Data Type	Description	Is Mandatory
User Name	8-30	Alphanumeric	Unique name for each user	Y
Playlist Name	8-30	String	To identify the playlist	Y
Video Title	8-30	String	Video added to the playlist	Y

Subscription Database

Field	Length	Data Type	Description	Is Mandatory
User Name	8-30	Alphanumeric	Unique name for each user	Y
Subscribed Channel	8-30	Alphanumeric	Unique name for each user	Y

Report Requirements

User Report

Video Report

User ID	Video title
Email ID	Channel name
Password	Uploader/Channel ID
Username	Like count
Age	Dislike count
Phone Number	Comment count
Location	Video length
Platform Join Date	Upload quality
Status	Upload date
Total watch time	Video codec
Upload count	Video format
	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					
2	REQ_2	Login					
3	REQ_3	Home Page					
4	REQ_4	Video Playback					
5	REQ_5	User Dashboard					
6	REQ_6	Upload Videos					
7	REQ_7	Subscribe Channel					
8	REQ_8	Search Video / Channel					
9	REQ_9	Like Video					
10	REQ_10	Playlist					