VoD Streaming System
Design Document

Authors:
G.L.H. Binoli Gaweshika

This page intentionally left blank.

Document Revision History

Date	Version	Description
15/11/2020	0.0	Selecting the domain
18/11/2020	0.1	Studied about system architecture of a VoD System
20/11/2020	0.2	Planned unified process for the VoD System
23/11/2020	0.3	Design Use case diagram
26/11/2020	0.4	Design first cut class & complete class diagram
28/11/2020	0.5	Drafting the scenarios
30/11/2020	0.6	Designing Sequence diagram & Collaboration diagram
2/12/2020	0.7	Prepared the Introduction of the document
5/12/2020	0.8	Explained System architecture in the document
6/12/2020	0.9	Drafting System interfaces
7/12/2020	1.0	Finalize the document
9/12/2020	1.0	Added Activity diagram & State diagram
10/12/2020	1.1	Prepared GUI
11/12/2020	1.1.1	Prepared GUI
12/12/2020	1.1.2	Prepared GUI
12/12/2020	2.0	Re-finalized the System Document

This page intentionally left blank.

Abstract

This report includes how VoD Streaming system designed & following unified process. Initially as the first step took the streaming media technology as the starting point, stated the work principle of the streaming media-based VOD system, put forward the design scheme of the structure of streaming media-based VOD system by understanding user requirements. Then using UML diagrams user requirements & baseline architecture is established. As the construction step just implemented the GUI & as future work hoping to develop it further & deploy the system & train, support the users.

Key words- VoD Streaming System, Unified Process, UML diagrams, GUI

Contents

1 Introduction

The demand of video contents has rapidly increased in the past ten years & one of the services that has especially become attractive to the customer is real time VoD because it offers an immediate streaming of large variety of video contents.

The VoD system became an application hotspot along with the development of streaming media technology. Streaming video is content sent on compressed from over the internet & displayed by the viewer in real time. With streaming video or streaming media, a web user doesn't wait to download a file to play it, it can be transmitted & played at any time. The best example is Youtube.

Purpose of this document is to provide design architecture UML diagrams for an online video database management system.

In the upcoming sections of this document, the design of the final system including several detailed diagrams will be described in detail.

1.1 Purpose

Purpose of this document is to provide a clear idea about a VoD streaming system and to provide design architecture, UML diagrams which will shows what are the functions & how should the system to be developed.

1.2 System Overview

VoD is a way of accessing content from online libraries, it' s services allow audience to access videos at their leisure from any compatible device. Some videos may be available to subscribers or account holders only while others are freely distributed to anyone online. Where traditional broadcast channels and media outlet forced consumers to view content on their schedule.

This VoD platform give users access to the content they want when they want it. Which is key in creating successful online viewing experience.

1.3 Design Objectives

Design is mainly based on following requirements & main objective is to fulfill them.

User can watch videos without possessing them, upload videos & earn money etc. Administrator must manage the online video database by adding or deleting videos from it etc.

A major difference in this system is that, user can access the online video database only if he/she has an account & logs in to the system.

Functional requirements

- User can watch videos without possessing them, upload videos & earn money.
- Administrator must manage the online video database by adding or deleting videos from it.
- User authentication procedure.
- Support internet & public users as well as administrators.
- Development should be based on object-oriented architecture.
- Support database, here developers use MySQL server database to save user data.

- Functions with certain degree of complexity such as functions that generate page content from a database.
- Required accessibility, globalization features already implemented.

<u>Non-functional requirements</u>

- *Usability:*
  - ✓ The system is user friendly & self-explanatory.
  - ✓ Since all users are familiar with the general usage of browsers, so no need specific knowledge or training is required.
- *Performance:* The design of the system & selection of software frameworks has been undertaken with performance in mind to ensure that they shall
  - ✓ Accept up to 1000 users accessing the web interface.
  - ✓ Receive up to 6000-10000 requests per day.
  - ✓ Submit up to 3000 verifications per day.
  - ✓ Complete the execution of actions initiated by a user using the system up to 3 seconds.
  - ✓ No downtime.
- Any video uploaded by a user shouldn't be lost.
- *Portability:* Non-portable parts of the system, if applicable, should be isolated & documented.
- *Scalability*: The design of the system and selection of technologies support a scalable solution.
- *Reusability:* A design based on components lends support for reusability.
- *Maintainability:* The components are loosely coupled therefore they should require less maintenance overhead.
- *Modularity:* The System is delivered as software components.

1.4 References

https://www.youtube.com/

https://www.dacast.com/blog/vod-streaming/

1.5 Definitions, Acronyms, and Abbreviations

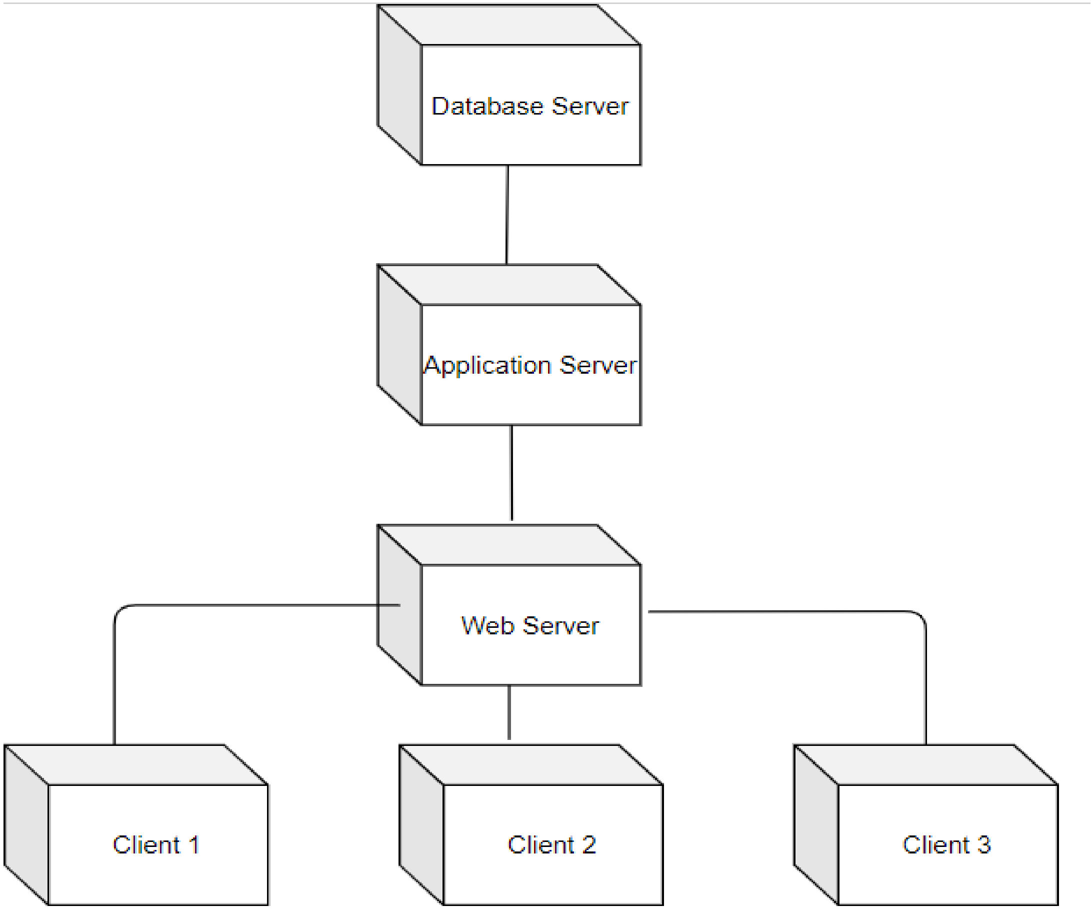
VoD – Video on demand
Streaming- The activity of listening to or watching sound or video directly from the internet

2 System Architecture

2.1 *System Software Architecture*

The design approach used in this system is Object Oriented design Architecture. The proposed architecture of the system is Client- Server. Draw.io has been used as the drawing tool of UML diagrams.

2.2 *System Hardware Architecture*



2.2.1 Figure

This diagram shows the nodes on which the components of a system execute. It represents the hardware topology of the system.

Main components: Database Server, Application Server, Web Server

Database Server and Application Server respectively correspond to the Online Video Database and the Administrator. Clients are users and Web Server is their corresponding LAN Server.

When a client makes a request, that request initially goes to the Web Server. Then the Web Server realizes the request using the http protocol & send it to the Application Server.

2.3 Constraints and Assumptions

Constraints

The information of all the users including client's information must be stored in the database that is accessible by the online video database management system.

Users can access the system from any compatible device that has internet browsing capabilities & internet connection.

The user must have their correct username & password to enter in to the VoD Streaming System.

Video duplication & video deletion are out of the scope.

Assumption

The user must have sufficient knowledge of accessing of web application.

3 System Interfaces

3.1 User Interfaces

1) **Login Page:** Users can enter User Id and Password to login or can create a new account.

2) **Search Page:** This page is opened after a successful login. Users can enter a search term to search any video. The search results are displayed in the non-header part of the page. Each search result shows name, thumbnail, download button (for free videos), like and dislike icons, number of views etc., corresponding to a video. Also, an upload button is located at the top right corner of the page.

3) **Download Page:** If on the search page, download button corresponding to a video is clicked, a dialog box is displayed asking to set resolution and format. After selection of required resolution and format, this page is opened showing the percentage of download completed etc. User cannot do any work until the download is completed or cancelled. So, network connection must be good.

4) **Watch Page:** If on the search page, name corresponding to a video is clicked, this page is opened if the video is a free video else a dialog box requesting to rent/purchase video is opened. A non-free video is displayed if and only if it is taken for rent or purchased by the user. Unlike YouTube, video is shown full screen only and only play/pause, replay, close etc. buttons are present on this page.

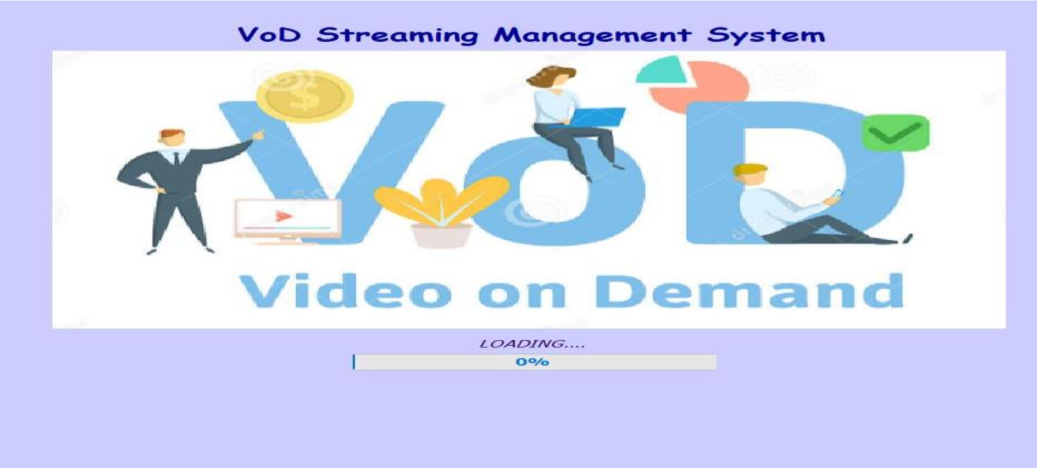
5) **Upload Page:** If on the search page, upload button is clicked, this page is opened and asks to select a video and fill the details required. After selection of video and filling the details, the video is uploaded to the administrator database and gets ready for processing.

6) **Process Page:** After the upload of video into the administrator database from the upload page, this page opens automatically showing the details of processing.

   After 'close button click' on Watch Page or 'completion or cancellation of download' in Download Page or 'completion of processing' in Process Page, Search Page is automatically displayed. *Sign In Page, Edit Profile Page etc,.* are similar to YouTube in this system.

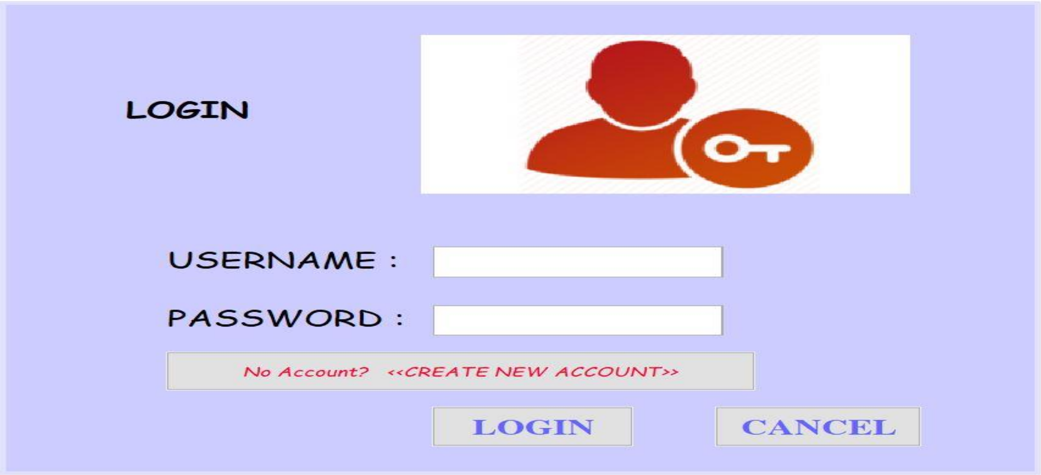
GUI

WELCOME PAGE



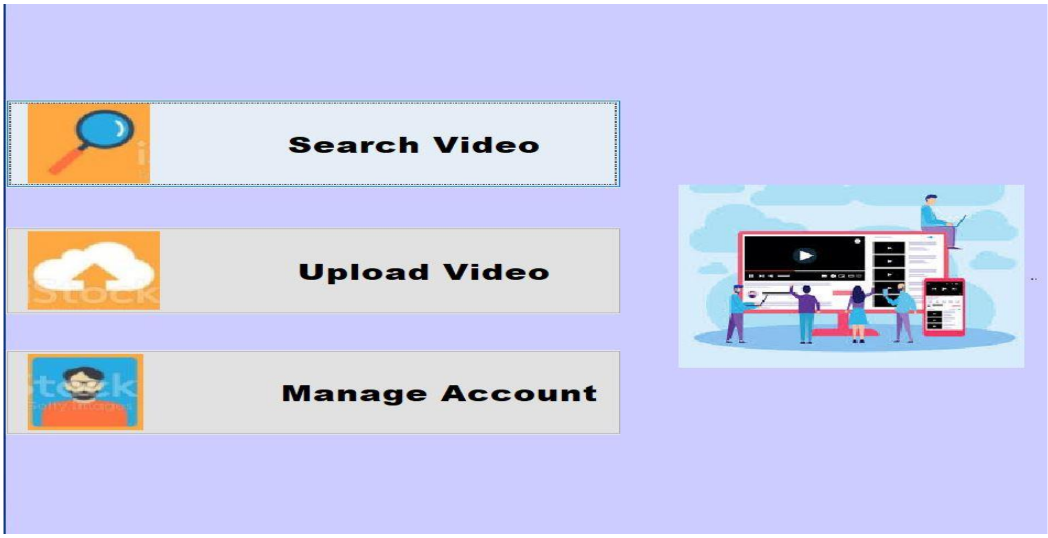
3.1.1 Figure

LOGIN PAGE



3.1.2 Figure

UPLOAD PAGE



3.1.3 Figure

4 UML Diagrams

4.1 Use Case Diagram

- Requirement Specification.

It shows the Actors, Use Cases and their Relationships in a system. Only one Use Case Diagram exists for a system. In this online video database management system, actors and use cases are as follows.

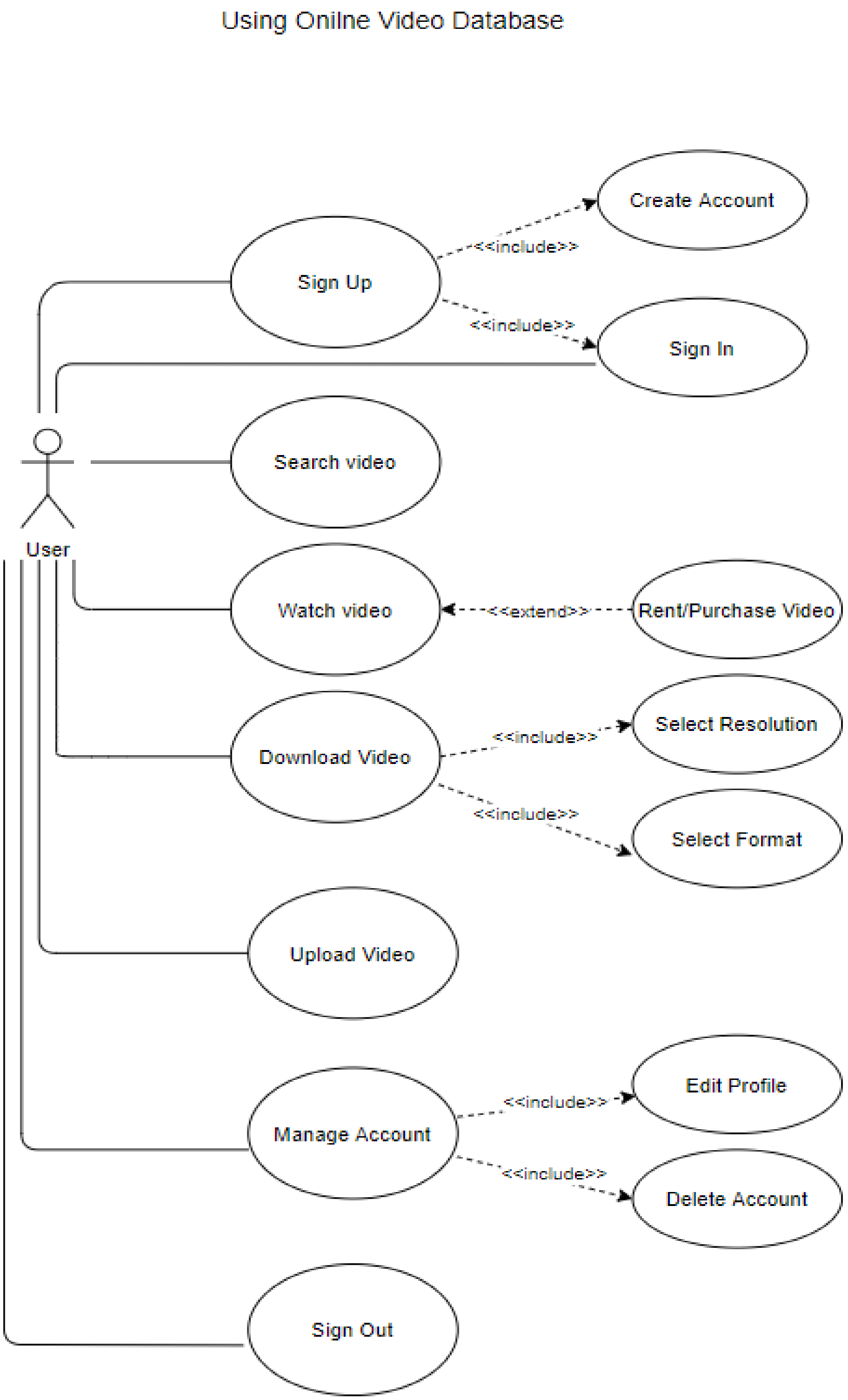
Actors

User: Uses the online video database.
Administrator: Manages the online video database.

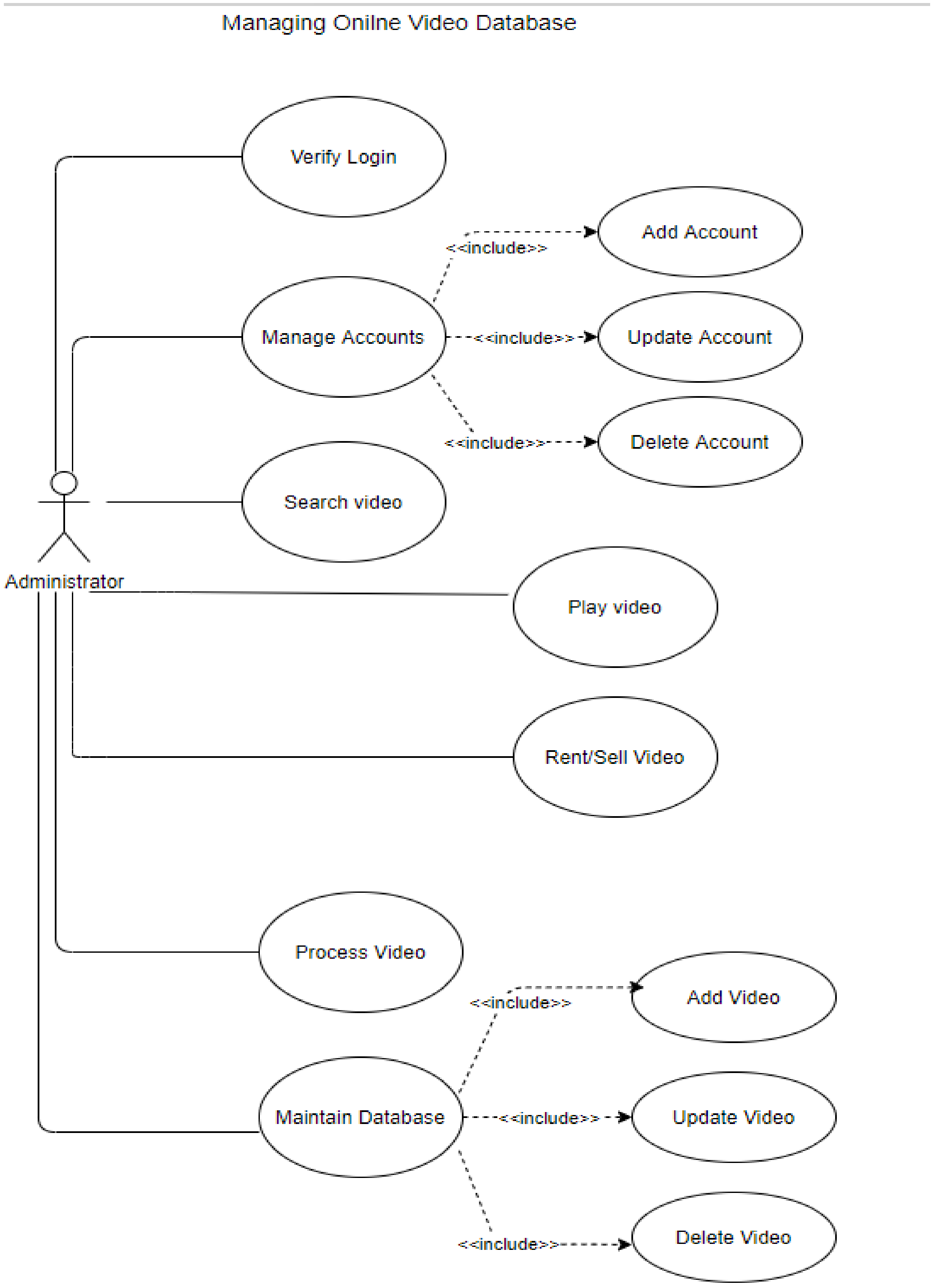
Use Cases

- Sign Up: User creates a new account.
- Sign In: User logs in to the system.
- Search Video: User enters a search term.
- Watch Video: User watches a video.
- Download Video: User downloads a video.
- Upload Video: User uploads a video.
- Manage Account: User performs edit profile or delete account.
- Sign Out: User logs out from the system.
- Verify Login: Administrator verifies login details of users.
- Manage Accounts: Administrator manages user accounts.
- Search Video: Administrator searches for a video in the database.
- Play Video: Administrator streams a video from the database.
- Rent/Sell Video: Administrator manages money transactions.
- Process Video: Administrator processes video uploaded by users.
- Maintain Database: Administrator maintains video database.

  All the Include and Extends relationships can be understood from the diagram. As the diagram is large, it is divided into two parts.


Using Onilne Video Database
Create Account
<<include>>
Sign Up
<<include>>
Sign In
Search video
User
Watch video
<<extend>>
Rent/Purchase Video
Select Resolution
<<include>>
Download Video
<<include>>
Select Format
Upload Video
Edit Profile
<<include>>
Manage Account
<<include>>
Delete Account
Sign Out

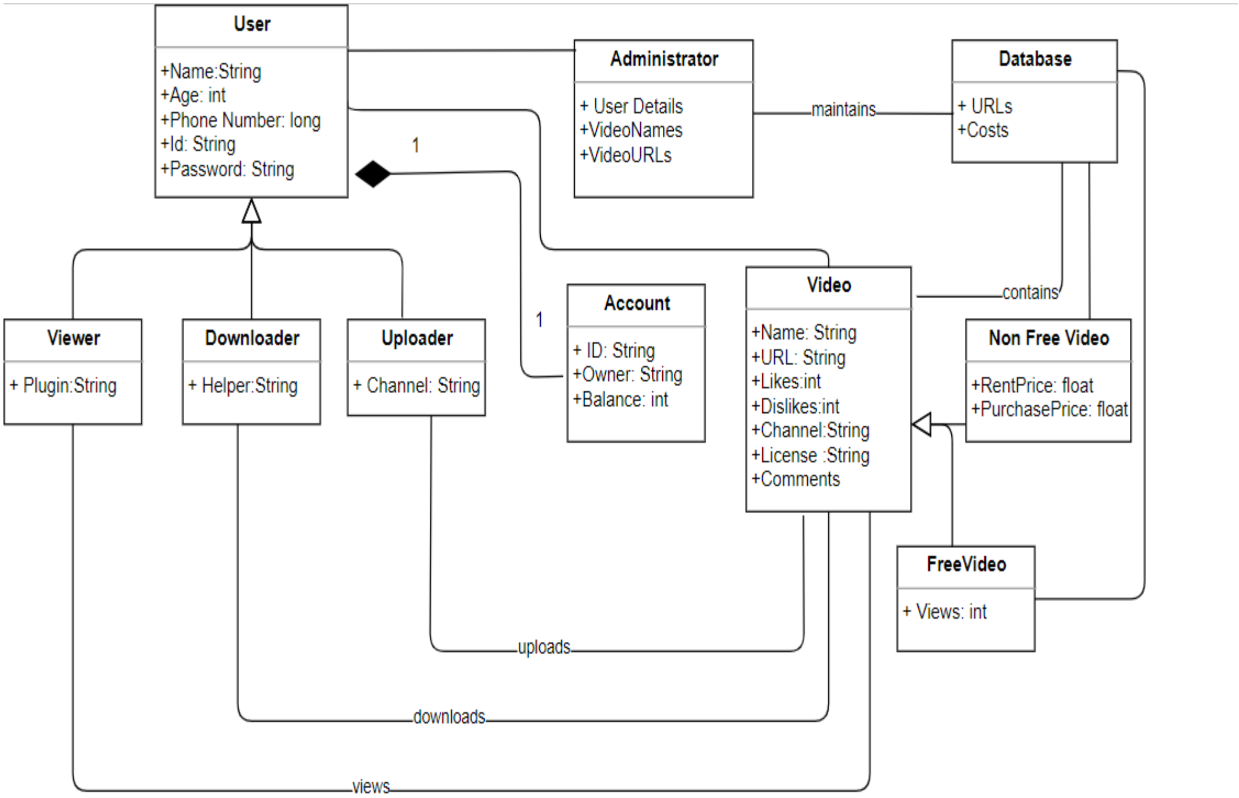
4.1.1.Figure


Managing Onilne Video Database
Verify Login
Add Account
<<include>>
Manage Accounts
<<include>>
Update Account
<<include>>
Delete Account
Search video
Administrator
Play video
Rent/Sell Video
Process Video
Add Video
<<include>>
Maintain Database
<<include>>
Update Video
<<include>>
Delete Video

4.1.2.Figure

4.2 First cut class diagram

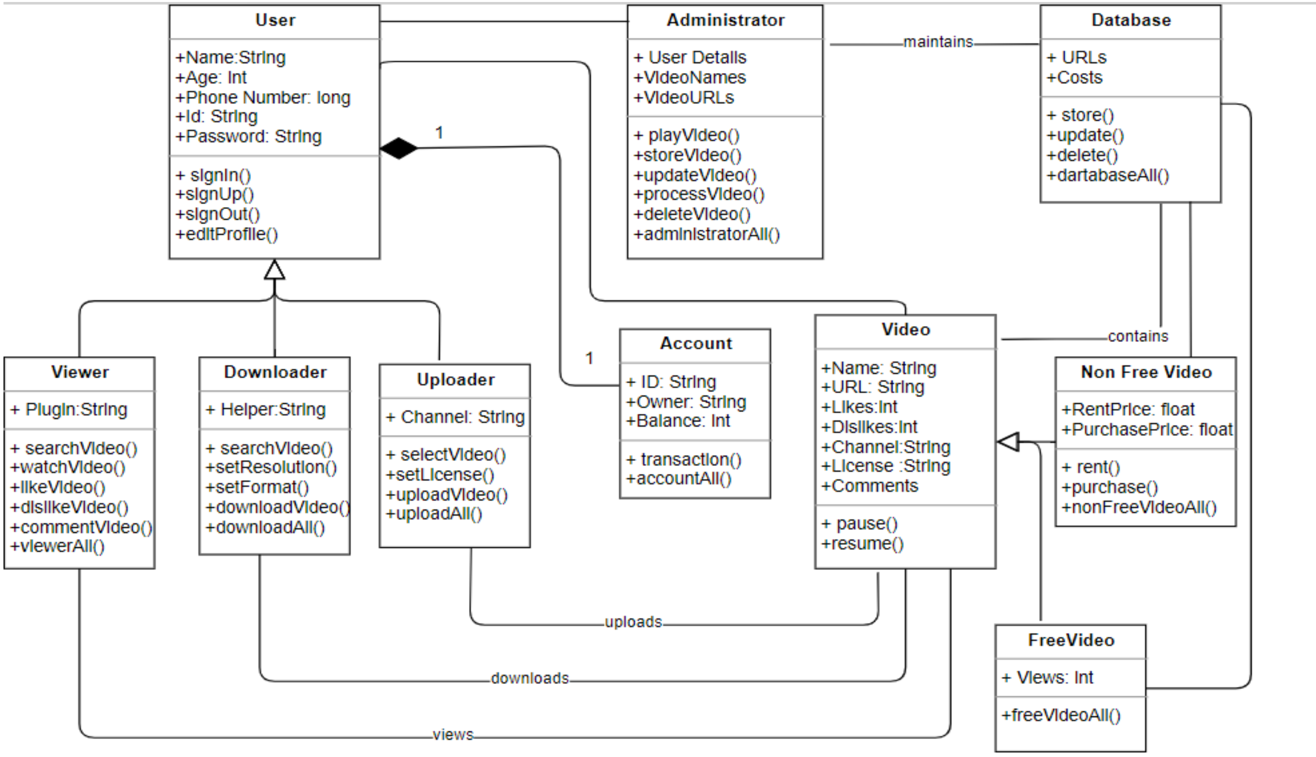
- Analysis stage with the domain model with respect to relationship between data.



4.2.1.Figure

4.3 Complete Class Diagram

- Design stage with domain model.



4.3.1.Figure

- **User** class represent the user who use the system.
- Name, Age, Phone number, Id & Password are the attributes of that class.
- User is a Parent class & Viewer, Downloader, Uploader inherits with it as child classes.
- **Viewer, Downloader, Uploader** classes owns some specific functions own for each other.
- **Account** class represent the user's account and its relationship between User & Account is **Composition**. That means when there isn't any customer there isn't any Account. Account is compulsory for an user, every user possess an account, new user must create an account.
- **Video** is a parent class & Free Video. Non-Free Video are its child classes. Relationship between them is **Inheritance**.
- **Viewer, Downloader, Uploader** classes are **Associated** with **Video** class. **Viewer** views video**, Downloader** downloads video, **Uploader** uploads video.
- **Administrator** maintains **Database,** so Administrator class & Database class are associated.
- **Database** contains **Video,** so Video class & Database class are associated.
- **Administrator** class & **Video** class also associated with **User** class.

4.4 Sequence Diagram & Collaboration Diagram

- Sequence diagram & Collaboration diagram use to show the interaction.
- Sequence diagram shows how objects in the system interaction sequence to complete a specific task & Collaboration diagram emphasize the structure of object interaction.
- Within sequence diagram the order of interaction is established by vertical positioning whereas in Collaboration diagram the sequence is given by numbering the interaction.
- Number of Sequence Diagrams/ Collaboration Diagrams for a system is equal to the number of Use Cases in the system (one sequence diagram/collaboration diagram for one use case).

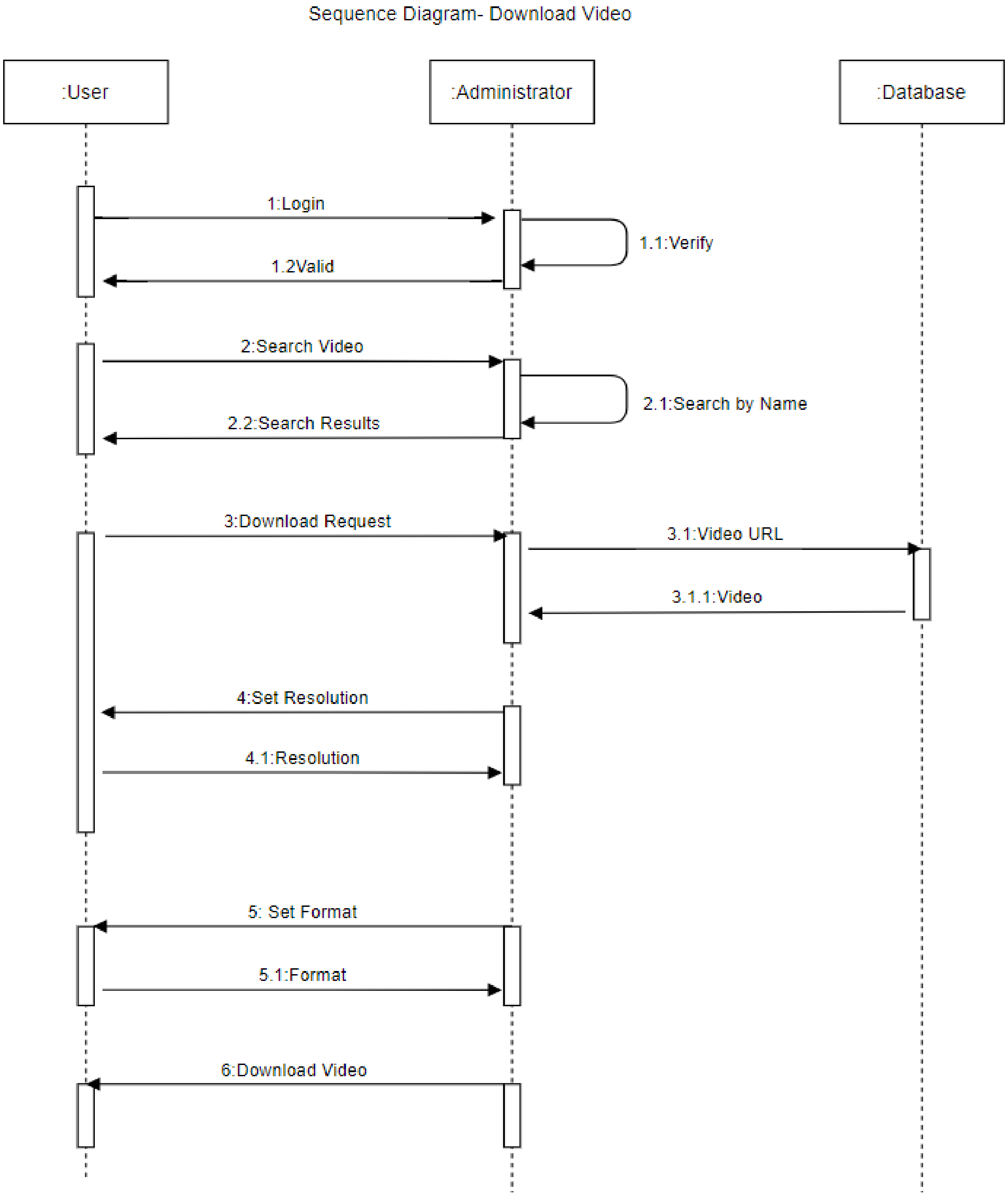
4.4.1. Scenarios

- Use Cases playing a key role in this Online Video Database Management System are Download Video, Watch Video and Upload Video.

Download Video

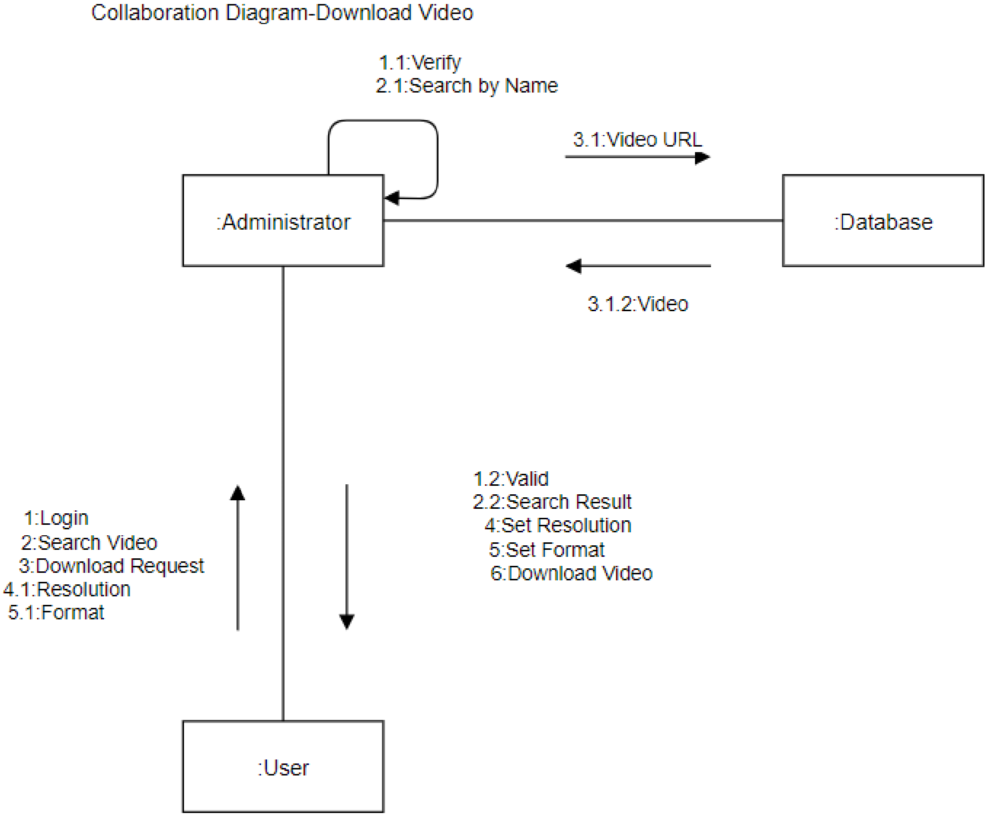
- *Scenario Name:* Download Video
- *Scenario Description:*
  - ✓ Actor: User
  - ✓ Main Success Scenario:
    1) The user initiates the scenario by login to the system.
    2) S(he) search the video they need.
    3) S(he) request permission for download.
    4) S(he) selects resolution.
    5) S(he) selects format.
    6) If everything goes smoothly, the system downloads video to the user’s device.
  - ✓ Precondition(s): User must have an account.
  - ✓ Postcondition(s): User downloads the selected video under selected resolution & format.

- *Sequence Diagram:*



4.4.1.1.Figure

- *Collaboration Diagram:*

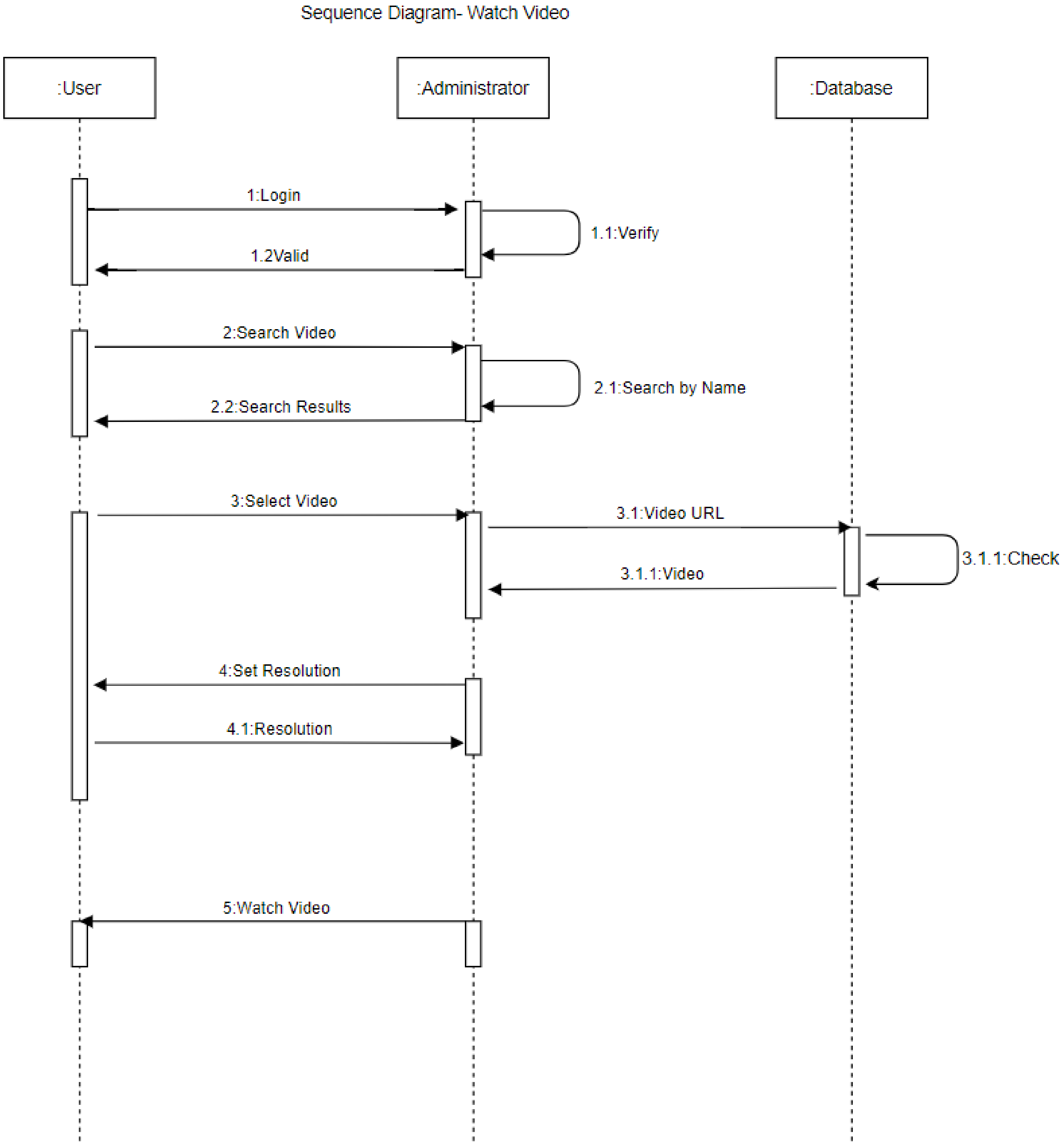


4.4.1.2.Figure

Watch Video

- *Scenario Name:* Watch Video
- *Scenario Description:*
  - ✓ Actor: User
  - ✓ Main Success Scenario:
    1) The user initiates the scenario by login to the system.
    2) S(he) search the video they need.
    3) S(he) selects the video.
    4) S(he) selects resolution.
    5) S(he) selects format.
    6) If everything goes smoothly, the system views the searched video to the user.
  - ✓ Precondition(s): User must have an account.
  - ✓ Postcondition(s): User can watch the searched video under selected resolution & format.

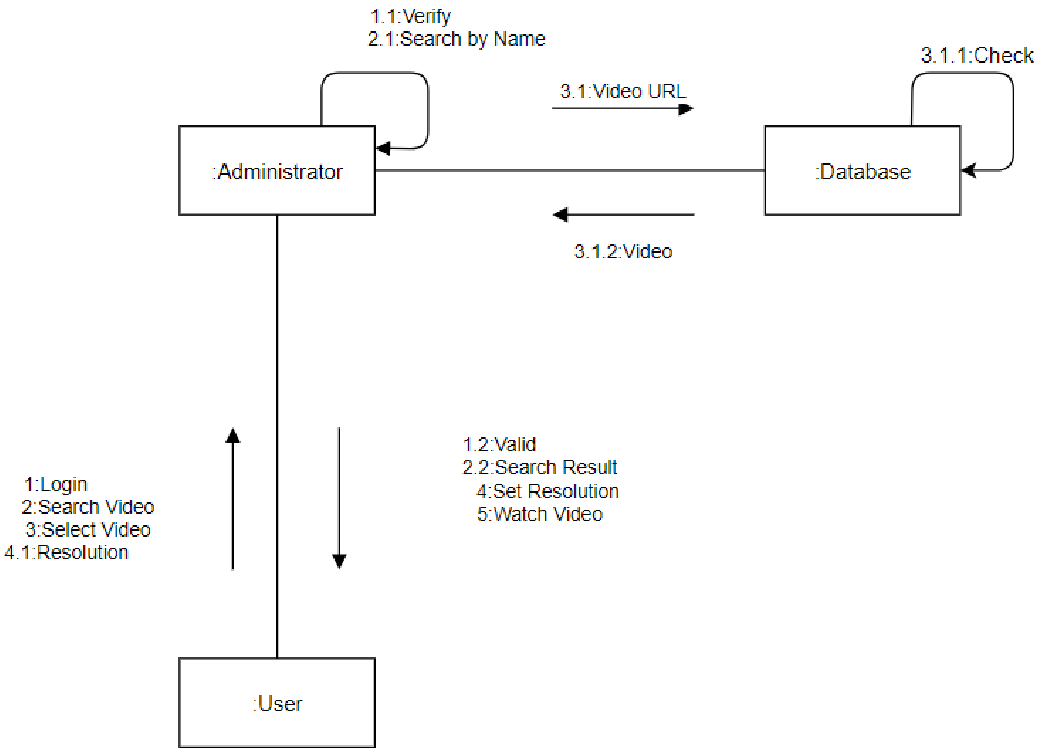
- *Sequence Diagram:*



4.4.1.3.Figure

- ***Collaboration Diagram:***

Collaboration Diagram-Watch Video

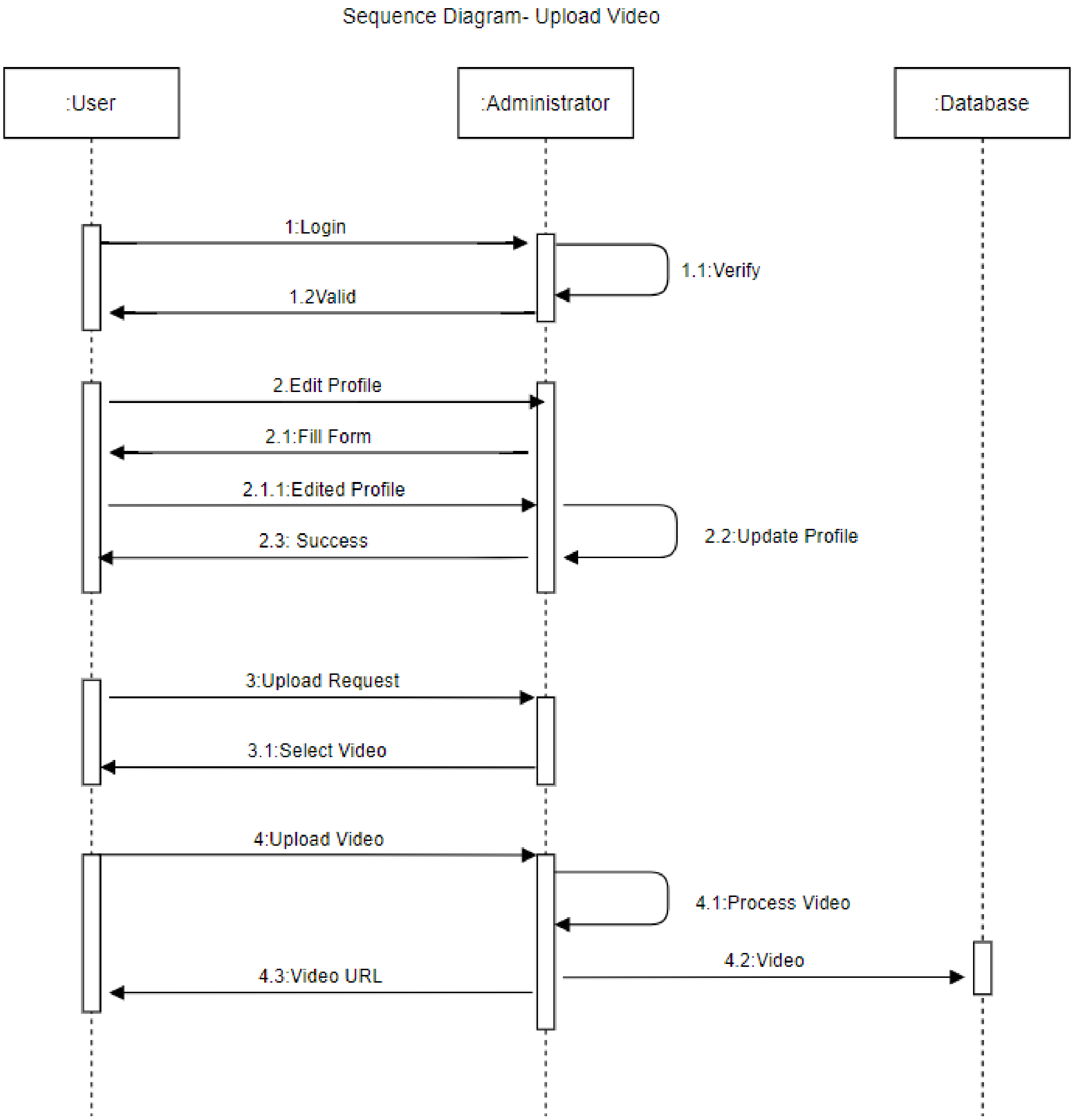


4.4.1.4.Figure

Upload Video

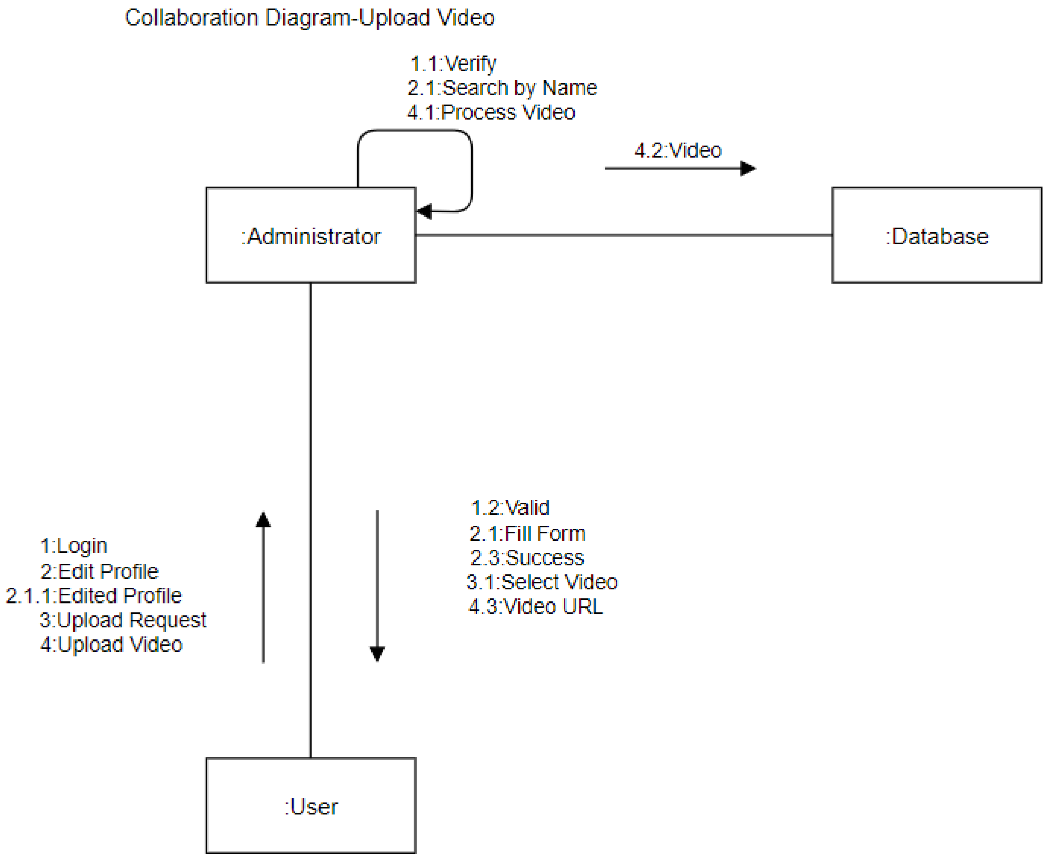
- *Scenario Name: Upload* Video
- *Scenario Description:*
    - ✓ Actor: User
    - ✓ Main Success Scenario:
        1) The user initiates the scenario by login to the system.
        2) S(he) edit the profile in order to get permission to upload a video.
        3) S(he) uploads the request.
        4) S(he) uploads the video.
        5) If everything goes smoothly, the system gives the video URLto the user.
    - ✓ Precondition(s): User must have an account.
    - ✓ Postcondition(s): User owns the URL of the video s(he) uploaded.

- *Sequence Diagram:*



4.4.1.5.Figure

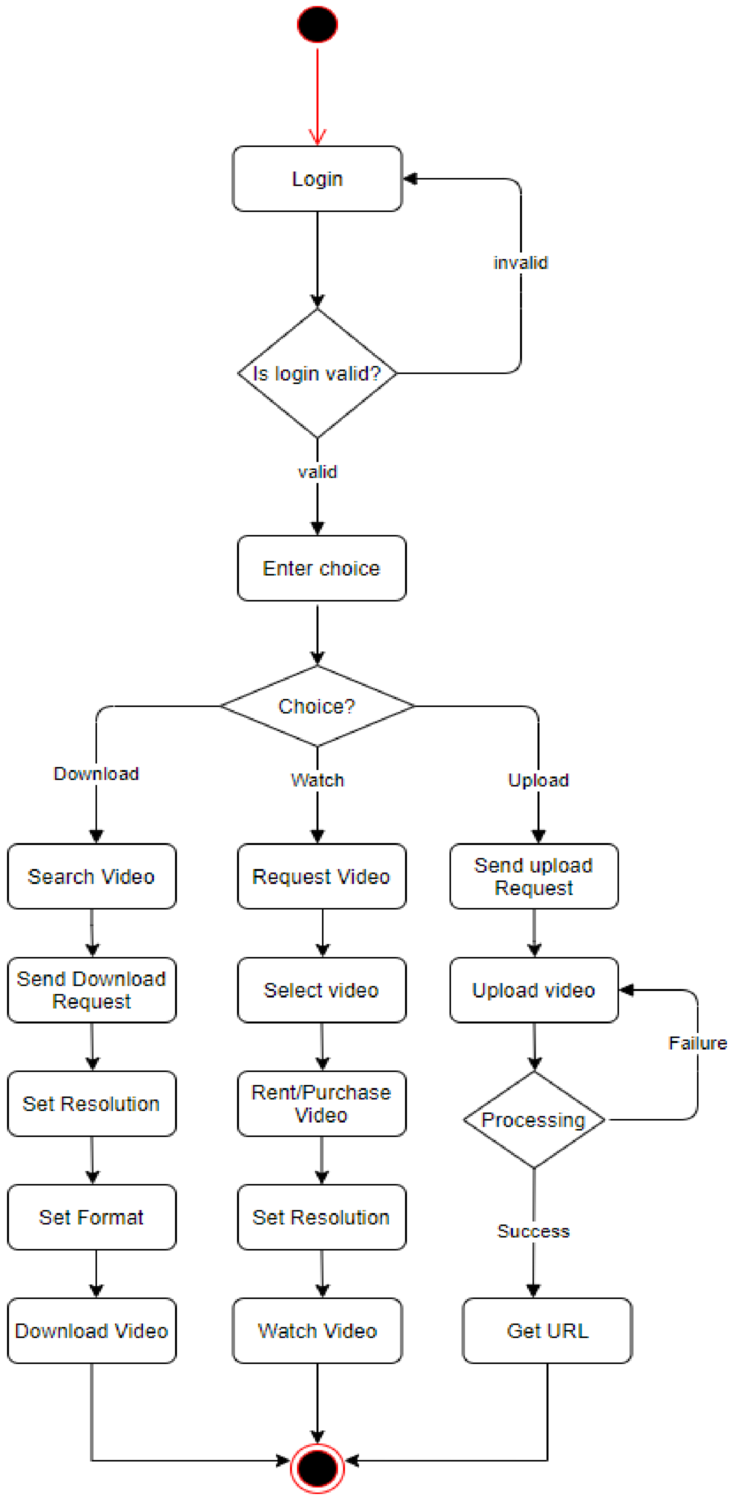
- *Collaboration Diagram:*



4.4.1.6.Figure

4.5 Activity Diagram

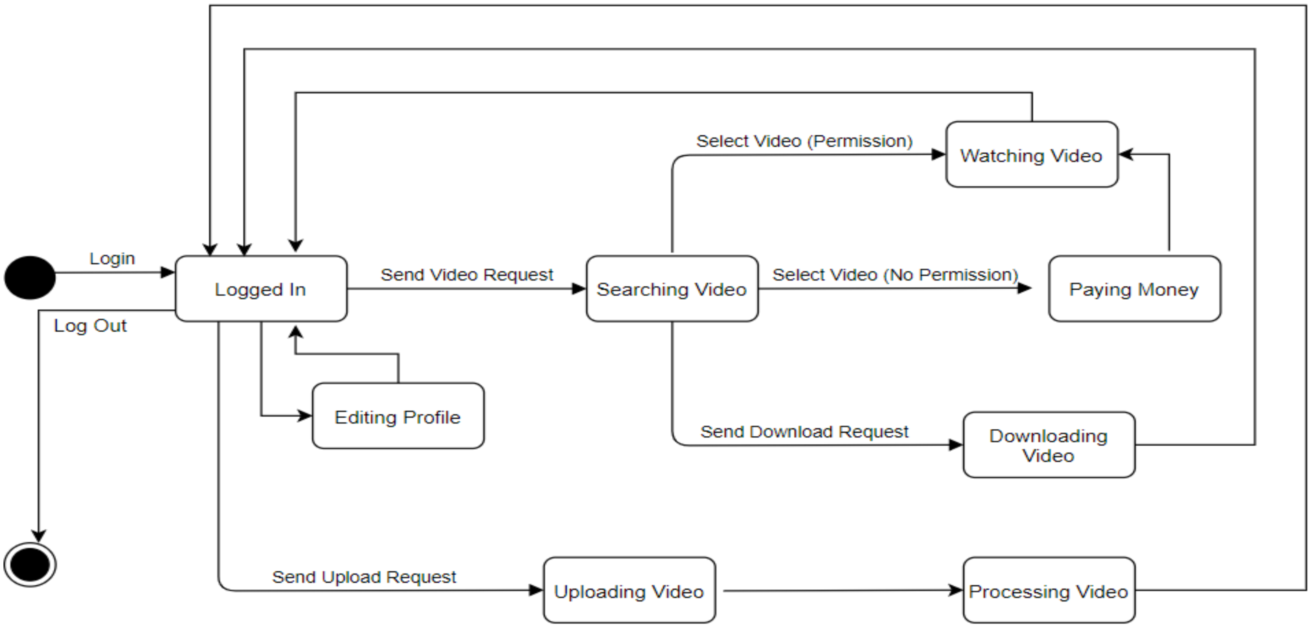
- Represents the system as a flow of activities.



4.5.1.Figure

4.6 State Diagram

- Represents the sequence of states an object goes through in working of a system. Therefore, number of state diagrams is equal to number of classes in the system.
- The object which plays a key role in the system is User.
- Hence, the state diagram is for User.



4.6.1.Figure

5 Conclusion

This system has appropriate documentation so that development team can easily resume work in improving this VoD system & successfully deliver a commercially viable VoD system.
This system has a lot of scope for improvements in future for better performance.

Future works:

- The user interface of the current system is built using Java Swing & hoping to explore to develop a better user interface.

- Hasn't dealt with any of the security issue so far, but in order to make it a complete product the security issues have to be addressed. At the present, the viewer can access the video files downloaded to his/her edge device & copy the video files outside the application. This will interfere with the copyright issues of the video. Therefore, the video files downloaded to his/her edge device should be encrypted so far that user cannot access it.

- Database management.