**Software Requirements Specifications**

A logo of a computer science

Description automatically generated

Session: 2022 – 2026

**Submitted by:**

Shahneela Iqbal 2022-CS-159

**Submitted to:**

Dr. Atif

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**YouTube Thumbnail Downloader**

**1. Introduction**

1.1 Purpose

This document outlines the requirements for the YouTube Thumbnail Downloader, a web-based tool that allows users to enter a YouTube video URL or keyword and retrieve the corresponding video thumbnail image and embed code. The tool is designed for ease of use and to provide a quick, simple solution for obtaining YouTube thumbnails. This document also includes user authentication features to allow users to save and manage their downloaded thumbnails securely.

1.2 Scope

The YouTube Thumbnail Downloader will be a web application that works across all major browsers. Users will be able to input a YouTube video link or the name of the video, retrieve the corresponding thumbnail in different resolutions, and utilize user authentication to save their preferences and history. This document describes the system’s requirements, including functionality, performance, usability, and design constraints.

1.3 Definitions, Acronyms, and Abbreviations

* **SRS**: Software Requirements Specification
* **URL**: Uniform Resource Locator
* **UI**: User Interface
* **API**: Application Programming Interface

1.4 References

* YouTube API Documentation
* HTML5, CSS3, JavaScript, and React Documentation

1.5 Overview

This document includes:

* Functional requirements
* Non-functional requirements
* Interface design
* System constraints

**2. Overall Description**

2.1 Product Perspective

The system will be developed as a standalone web-based application. It will rely on YouTube video URLs or names to retrieve thumbnails and display them to the user. Users will be required to create an account and log in to access features such as saving thumbnails and managing their download history.

2.2 Product Features

* User Authentication:
* Sign up, log in, and log out features.
* Password reset functionality.
* Input YouTube video URL or Name.
* Retrieve the thumbnail in different resolutions (e.g., default, medium, high).
* Download or save the retrieved thumbnail to the local device.
* Get Embed Code.

2.3 User Classes and Characteristics

**Regular Users:** Users who need to quickly download or save YouTube video thumbnails for personal or business use.

**Administrators:** Responsible for managing the application, updating code, and monitoring performance and usage statistics.

2.4 Operating Environment

The application will be web-based and supported on all major browsers such as Chrome, Firefox, Safari, and Edge. The system should run on any device that supports a modern web browser, including desktops, tablets, and mobile phones.

2.5 Design and Implementation Constraints

* The application must handle invalid YouTube URLs gracefully, providing the user with helpful error messages.
* The system must conform to YouTube’s API rate limits if used.
* The system should not store or manipulate video content directly due to legal constraints.
* User authentication must comply with security best practices, including password hashing and session management.

2.6 Assumptions and Dependencies

* The user has a stable internet connection.
* YouTube API availability and compliance with any legal requirements from YouTube for accessing its thumbnails.
* Users will follow security guidelines for managing their accounts.

**3. Functional Requirements**

3.1 User Authentication

3.1.1 Sign Up

**Description:** Users can create an account by providing a username, email, and password.  
**Input:** Username, email, password.  
**Process:** Validate the inputs, hash the password, and store the user information in the database.  
**Output:** Confirmation message or error message.

3.1.2Log In

**Description:** Users can log in to their account using their credentials.  
**Input:** Username or email and password.  
**Process:** Validate the credentials and create a session for the user.  
**Output:** Redirect to the dashboard or error message.

3.1.3 Password Reset

**Description:** Users can request a password reset link.  
**Input:** Email address.  
**Process:** Send a password reset email with a link to reset the password.  
**Output:** Confirmation message.

3.2 YouTube URL Input

**Description:** The user inputs a valid YouTube video URL.

**Input:** A YouTube video link.

**Process:** Validate the URL.

**Output:** Display either a success message (valid URL) or an error message (invalid URL).

3.3 YouTube Video Name Input

**Description:** The user inputs a valid YouTube video name.

**Input:** A YouTube video name.

**Process:** Validate the video’s name.

**Output:** Display either a success message (valid name) or an error message (invalid name).

3.4 Fetch Thumbnail

**Description:** The system retrieves the thumbnail images based on the input URL or name.

**Input:** Valid YouTube video URL or name.

**Process:** Connect to YouTube's servers, fetch the thumbnail images in available resolutions.

**Output:** Display the thumbnails to the user in different resolutions.

3.5 Download Option

**Description:** The user selects a resolution and downloads the corresponding thumbnail.

**Input:** User clicks on a "Download" button for the desired thumbnail resolution.

**Process:** Serve the selected thumbnail for download.

**Output:** Image file is downloaded to the user’s device.

3.6 Error Handling

**Description:** The system provides feedback for invalid URLs or names or network issues.

**Input:** Incorrect or incomplete URL or name.

**Process:** Validate the URL or name and check for API connection issues.

**Output:** Display error messages such as "Invalid URL or name" or "Unable to fetch thumbnail".

3.7 Copy Code Option

**Description:** The user gets a embed html code to use for blogs.

**Input:** User clicks on a "Generate Embed Code" button for the desired code.

**Process:** Fetch the code.

**Output:** Code displayed on the screen.

**4. Non-functional Requirements**

4.1 Performance

* The system should respond to user input and fetch thumbnails within 2-3 seconds on a normal internet connection.

4.2 Usability

* The user interface should be intuitive and responsive across different devices and screen sizes.

4.3 Reliability

* The system should handle high traffic and operate without crashes or major downtime.

4.4 Security

* No user information will be stored.
* The system will not expose any private YouTube data.

4.5 Scalability

* The system should scale to handle multiple users concurrently.

**5. External Interface Requirements**

5.1 User Interface

* A clean, minimal UI with a search bar for the user to input the YouTube URL.
* Buttons for fetching thumbnails.
* Thumbnails are displayed in a grid.

5.2 Hardware Interfaces

* None. The application is web-based and requires only a modern browser.

5.3 Software Interfaces

* YouTube API: Used to fetch the thumbnails associated with a video.

**6. System Features**

6.1 Feature 1: User Authentication

**Description:** The system will manage user registration, login, logout, and password reset functionalities.

6.2 Feature 2: Input Validation

**Description:** The system will validate YouTube URLs entered by the user.

6.3 Feature 3: Thumbnail Retrieval

**Description:** Upon validation, the system will fetch the thumbnails.

6.4 Feature 4: Thumbnail Download

**Description:** Users will be able to download thumbnails in available resolutions.

**7. Other Non-functional Requirements**

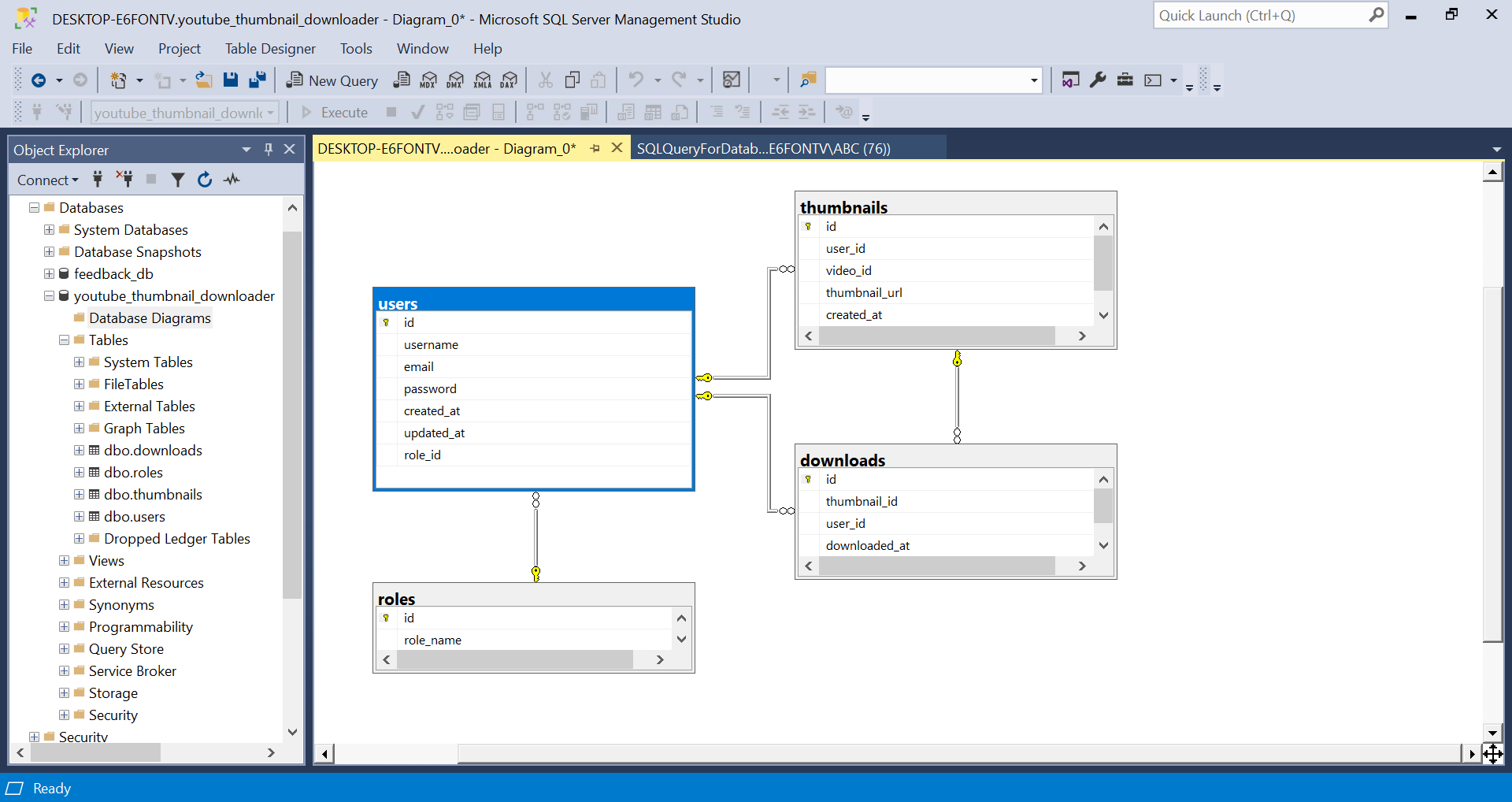
7.1 Availability

The application will be available 99.9% of the time, except for planned maintenance or unforeseen outages.

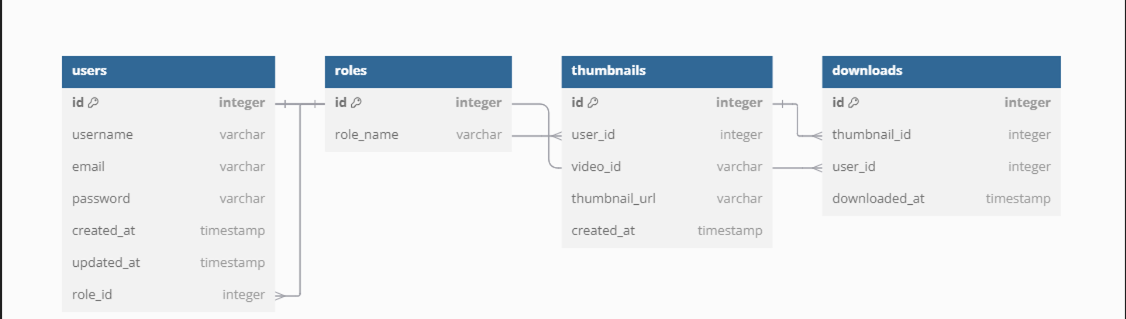
7.2 Maintainability

Code should be modular, following good development practices, and easily maintainable for future updates.

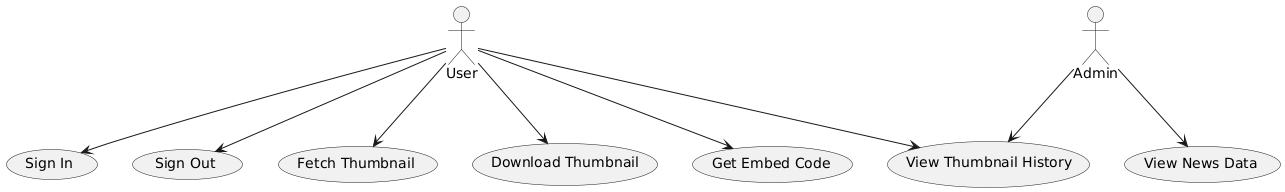
Data Base



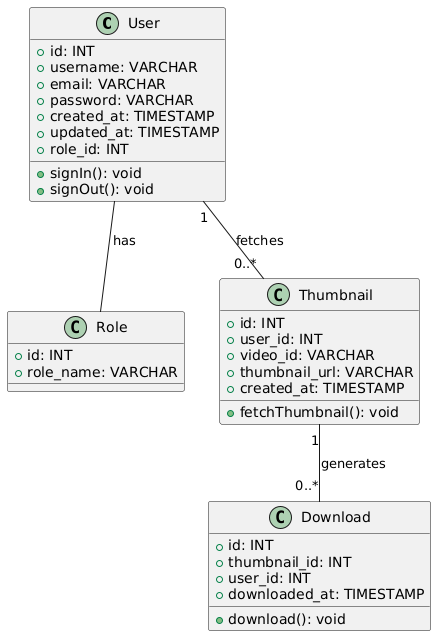
ERD



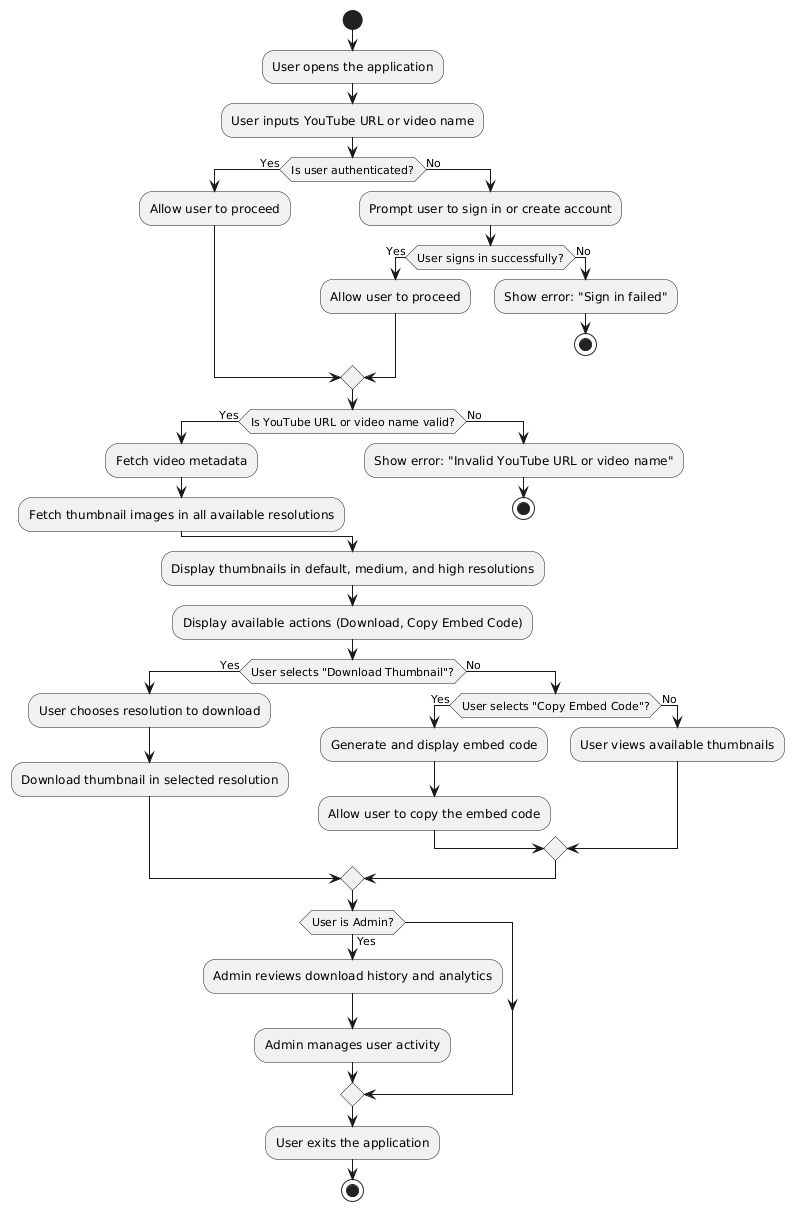
User Case Diagram



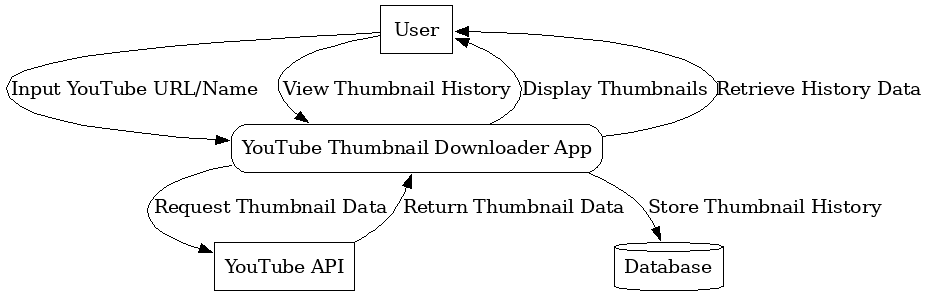
Class Diagram



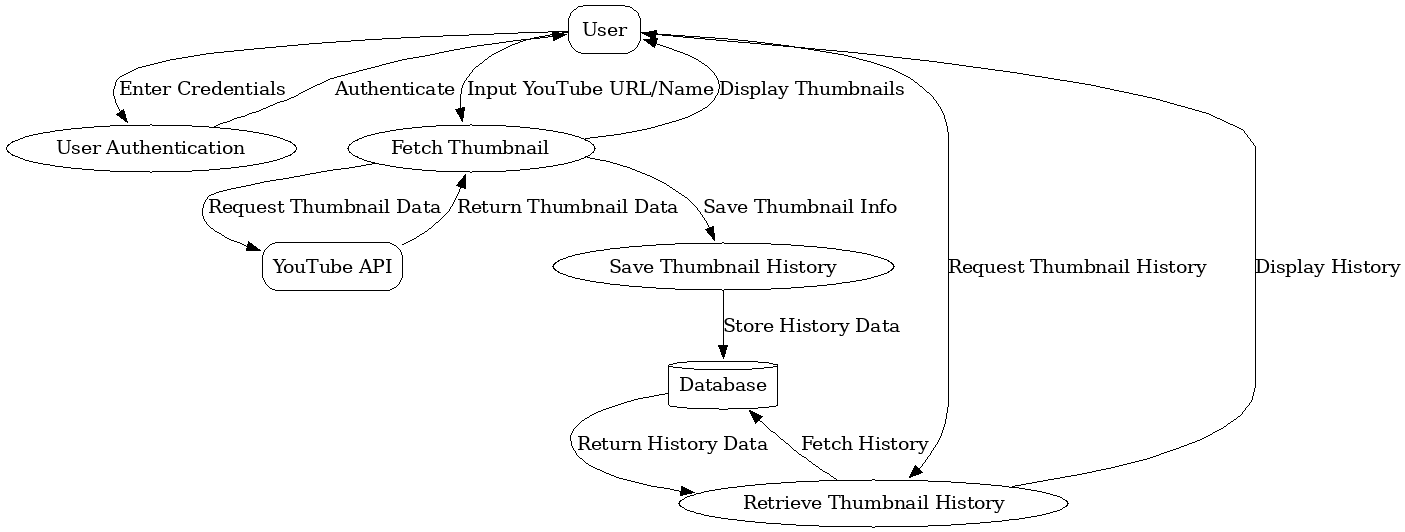
User Activity



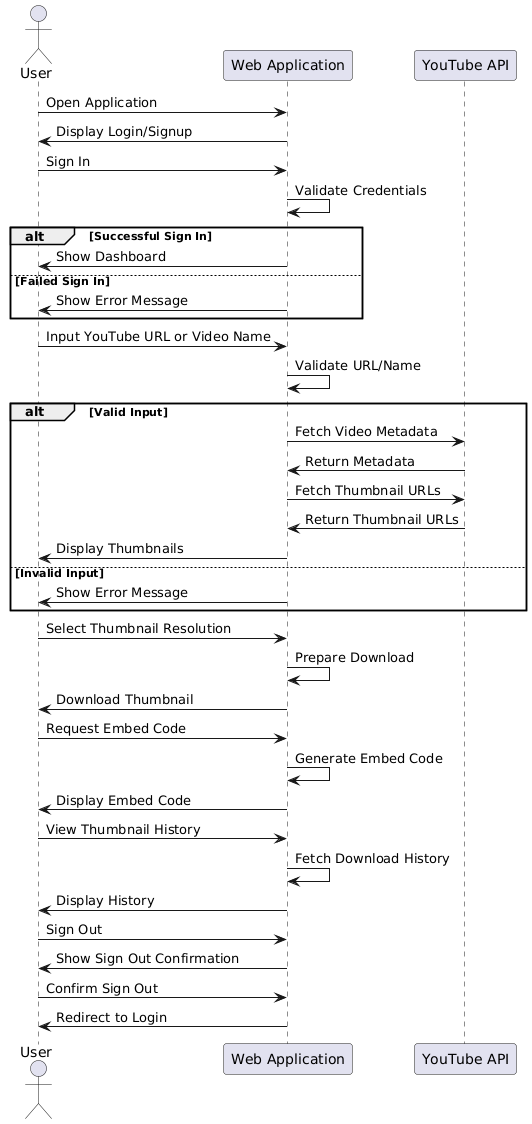
DFD Level 0



DFD Level 1



Sequence Diagram



Wireframes

