Software Design Document

For

Multimedia-platform video downloader

Submitted by:

Muhammad Fahim

Roll No:

S22BINFT1M01124

Submitted to :

Dr. Musarat Kareem

Department Of Information Technology

Faculty of computing

The islamia University of Bahawalpur

Summary

This document outlines the software design for the Social Media Video Downloader, a web-based application enabling users to download videos from platforms like YouTube, Facebook, TikTok, and Twitter. The project supports guest and registered users, allowing them to paste URLs, select video resolution, and download content. Administrative users have additional controls for user and system management. Complex features like in-app payments or video editing are excluded to keep the system focused and streamlined.

The design follows a Django + React architecture with PostgreSQL as the backend database and a RESTful API layer. User authentication is managed via JWT tokens, and third-party libraries like yt-dlp or youtube-dl are used to process and fetch video content from supported platforms.

Key design considerations:

- Modular backend (Django REST Framework) for API-driven communication.
- Relational database (PostgreSQL) for managing user data, logs, and download history.
- Video platform detection and download handled via open-source libraries.
- React.js frontend with responsive Bootstrap 5 UI for compatibility across devices.

1. Application Architecture

The system follows a modular, MVC-inspired architecture using Django (backend), React.js (frontend), and PostgreSQL (database). The application is designed to be scalable, API-driven, and secure, with clearly separated responsibilities across components.

🔹 Architecture Layers

1.1 Frontend (React.js)

- Responsible for rendering the UI and interacting with the user.
- Makes API calls to the Django backend using `axios` or `fetch`.
- Provides pages like:
    - Home (URL input and download)
    - Login / Register
    - Dashboard (for registered users)
    - Admin Panel

1.2 Backend (Django + Django REST Framework)

- Handles business logic, authentication, user management, and download history.
- Implements RESTful APIs for frontend consumption.
- Interacts with video processing modules like `yt-dlp`.

1.3 Database (PostgreSQL)

- Stores:
    - User information
    - Download history
    - Admin logs
    - Platform details

1.4 Video Processing Layer

- Uses `yt-dlp` or `youtube-dl` (wrapped in Python) to:
  - Recognize platform from URL
  - Fetch available resolutions
  - Handle video downloading

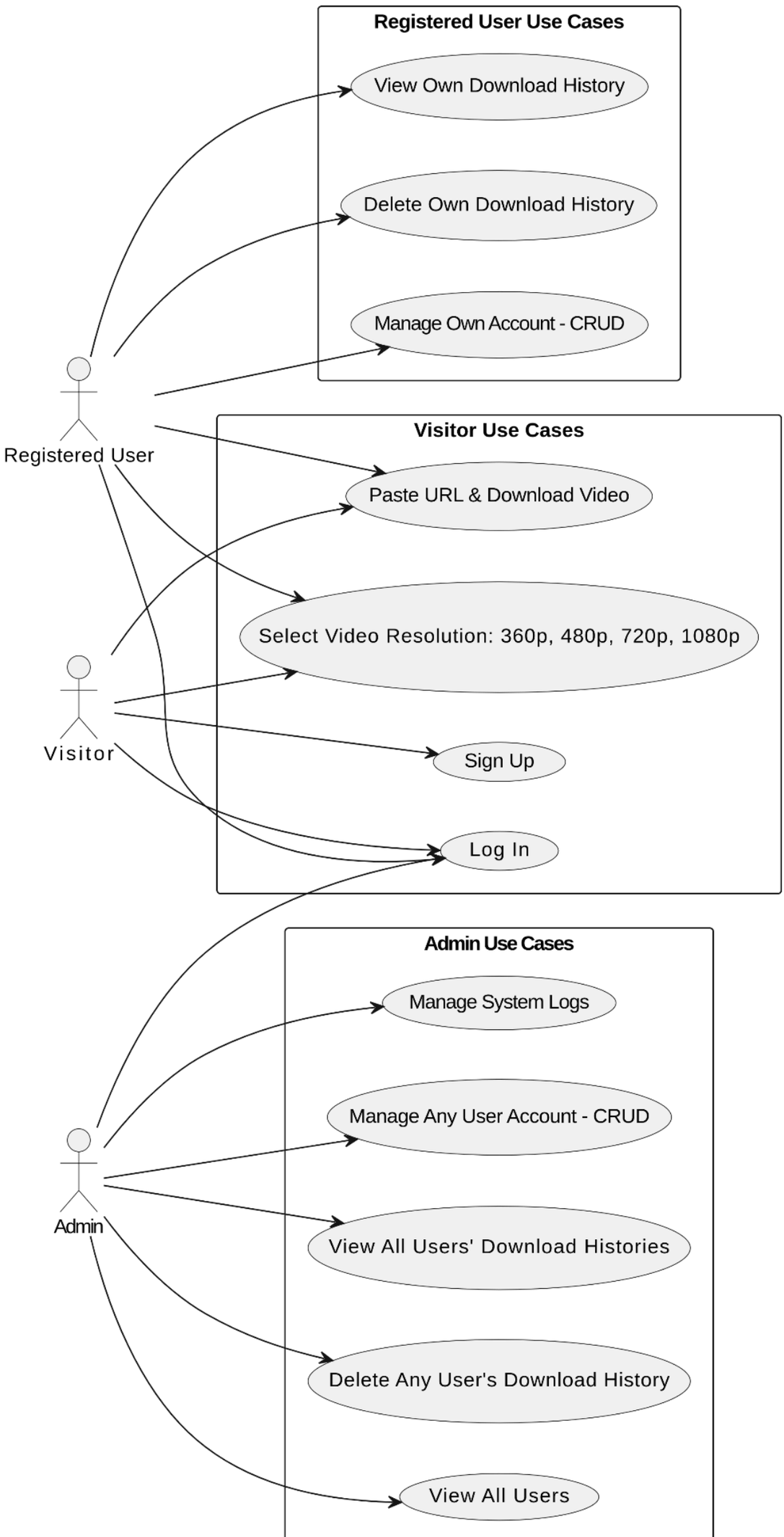
1.5 Authentication & Authorization

- Uses JWT (JSON Web Token) for secure API authentication.
- Role-based access:
  - Visitor: no login, can download
  - Registered User: full personal history and account features
  - Admin: full access to all users' data and logs

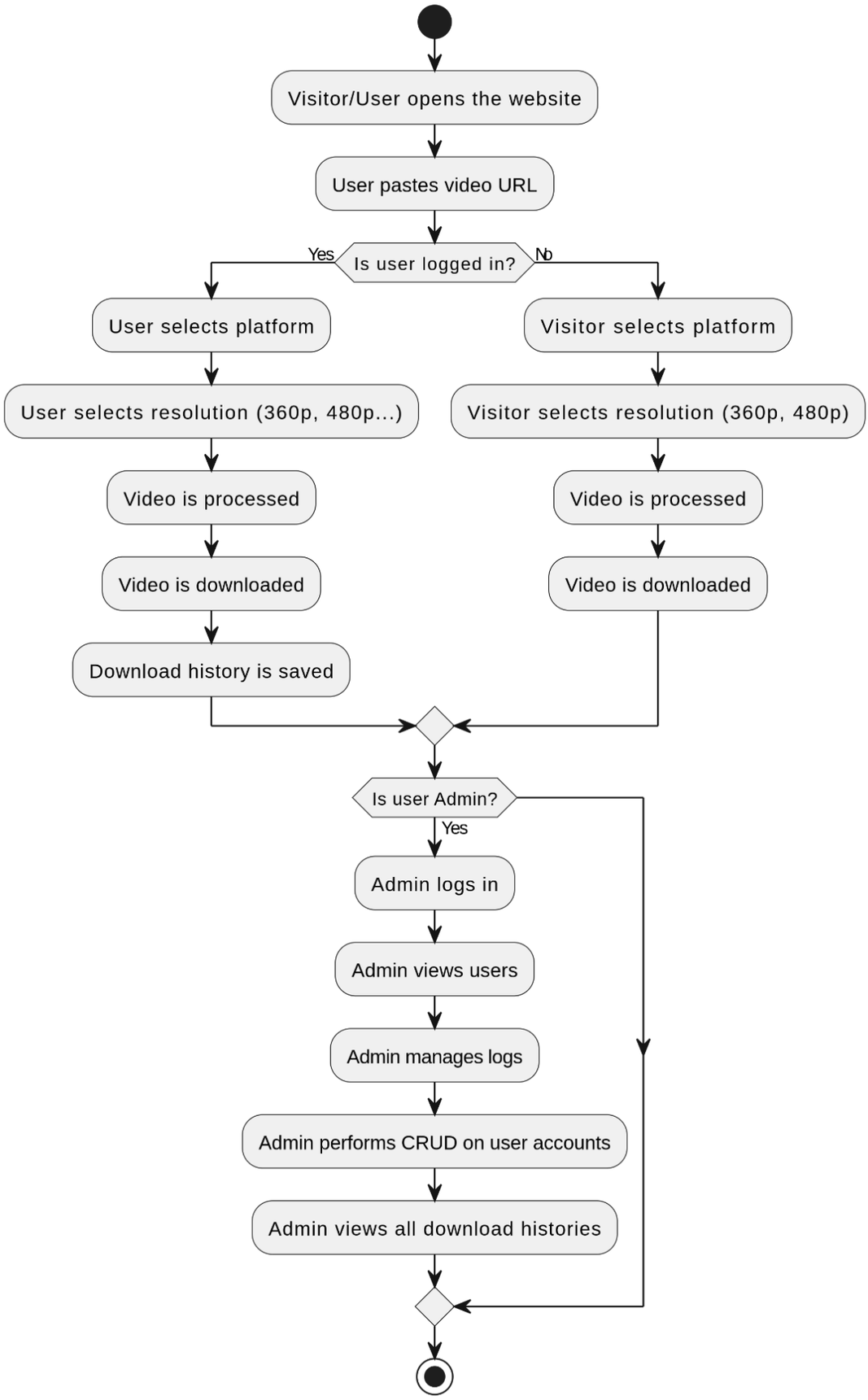
1.6 Key Controllers (Django Views / API Views)

Control Name	Functions
AuthController	`register`, `login`, `logout`
DownloadController	`process_url`, `fetch_resolutions`, `download_video`
UserController	`get_user_profile`, `update_user`, `delete_user`
AdminController	`get_all_users`, `view_logs`, `manage_user_account`, `view_all_histories`
HistoryController	`save_download`, `get_user_history`

3.1 Use Case Diagram



3.2 Flow of Process



2. Data Model Schema

This section outlines the database schema, including the main data entities, their fields, and relationships. The project uses a relational database (MySQL) to manage structured data for users, videos, and system logs. Below are the key models/entities:

2.1. User

Field	Type	Description
id	Integer (PK)	Primary key, auto-incremented
first-name	Varchar	First name of the user
last-name	varchar	Last name of the user
email	Varchar	Unique email address used for login
password	Varchar	Hashed password
profile_photo	Varchar	Path to uploaded display image
is_approved	Boolean	Approval status from superuser
created_at	DateTime	Timestamp of registration

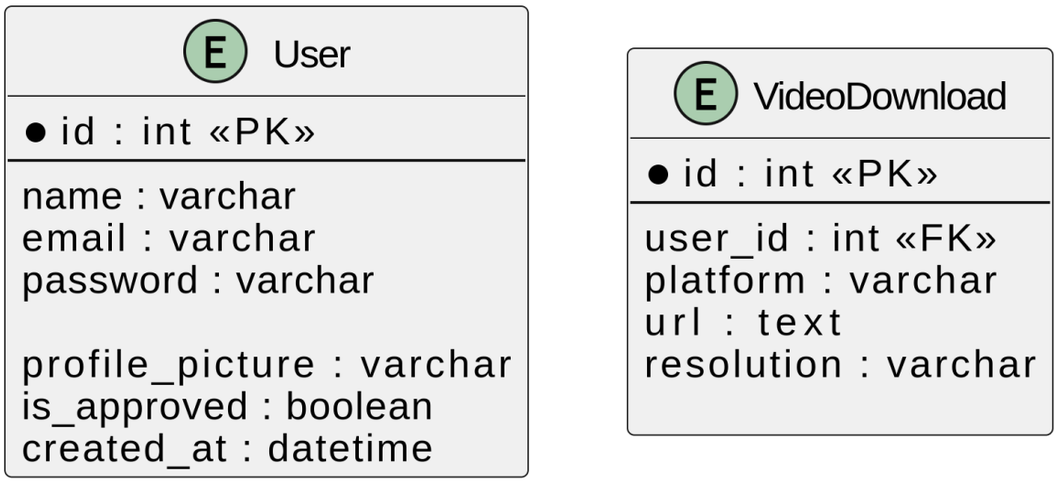
2.2 VideoDownload

Field	Type	Description
id	Integer (PK)	Primary key, auto-incremented
user_id	ForeignKey	Links to User model (`null` if visitor)
url	text	URL of the video to be downloaded
resolution	Varchar	Selected resolution (e.g., 360p, 480p)

Entity Relationship Summary

- A `User` can have many `VideoDownload` records.
- `VideoDownload.user_id` is nullable to allow anonymous (visitor) downloads.

2.3 ER Diagram



Explanation:

- **User** table is connected with:
    - `VideoDownload` (each user can have many download records).
    - `Log` (admins' actions are recorded).
- **VideoDownload.user_id** is optional (to allow downloads by visitors).

3. User Interface

The user interface (UI) of the application is designed with simplicity, responsiveness, and accessibility in mind. The main objectives are to ensure ease of use for all user types—Visitors, Registered Users, and Admins—and to provide a seamless experience across devices (desktop, tablet, and mobile).

3.1 Design Philosophy

- Color Scheme: Light mode interface with primary colors like blue for action buttons, white for background, and grey for borders and inactive elements.
- Typography: Simple and clean fonts like *Roboto* or *Open Sans* are used for readability.
- Layout: Grid-based layout with consistent padding, margins, and alignment.
- Responsiveness: Built using Bootstrap 5, ensuring that the UI adapts well to different screen sizes.

3.2 Page/Screen List and Descriptions

Screen/Page	Description
Home Page	Landing page for both visitors and users; includes sign-up, log-in options, and URL input form.
Login / Register Page	Forms for user authentication with validation constraints.
Dashboard	Personalized page for users to view download history and manage account.
Admin Panel	Admin-only interface to manage users, logs, and view download history of all users.
Download Page	Interface to paste URL, resolution, and trigger download.

3.3 UI Components and Controls

Component	Functionality
Textbox	For entering video URL.
Resolution Dropdown	For selecting video resolution (360p, 480p, etc.).
Buttons	Actions like Log In, Register, Download, Submit.
Table/Grid View	To display download history and user management data.

3.4 Input Constraints

- URL Field: Must be a valid video URL (regex validated).
- Dropdowns: Default values and validations to prevent empty submissions.
- Forms: All forms include client-side and server-side validation (e.g., password min. 8 characters).

3.5 wireframe sketches

3.5.1. Home Page / Landing Page

Elements:

- Logo / Branding
- Navigation Bar (Home, Login, Sign Up)
- URL Input Box
- Resolution Dropdown (360p, 480p)
- Download Button

3.5.2. Login Page

Elements:

- Username / Email Input
- Password Input
- Login Button
- “Forgot Password?” link
- Link to Sign Up page

3.5.3. Sign Up Page

Elements:

- Full Name Input
- Email Input
- Password Input
- Confirm Password Input
- Sign Up Button
- Link to Login page

3.5.4. Dashboard (for Registered Users)

Elements:

- Welcome Message
- Sidebar (Dashboard, History, Account, Logout)
- Central Panel: Paste URL, Select Platform, Select Resolution
- Download Button
- Table of recent downloads

3.4.5. Admin Dashboard

Elements:

- Navigation Tabs: View Users, View Logs, Download Histories
- User Table (with Edit/Delete)
- Full Download History Table (Searchable)

3.4.6. Account Settings Page

Elements:

- Update Profile Fields (Name, Email, Password)
- Save Changes Button
- Delete Account Button

3.4.7 Landing page