VIEWESTA- System Diagrams Documentation

This document contains detailed explanations for all system diagrams of the VIEWESTA project. Each section provides the purpose, involved components or actors, relationships, and placeholders for adding screenshots of the actual diagrams.

1. Use Case Diagram

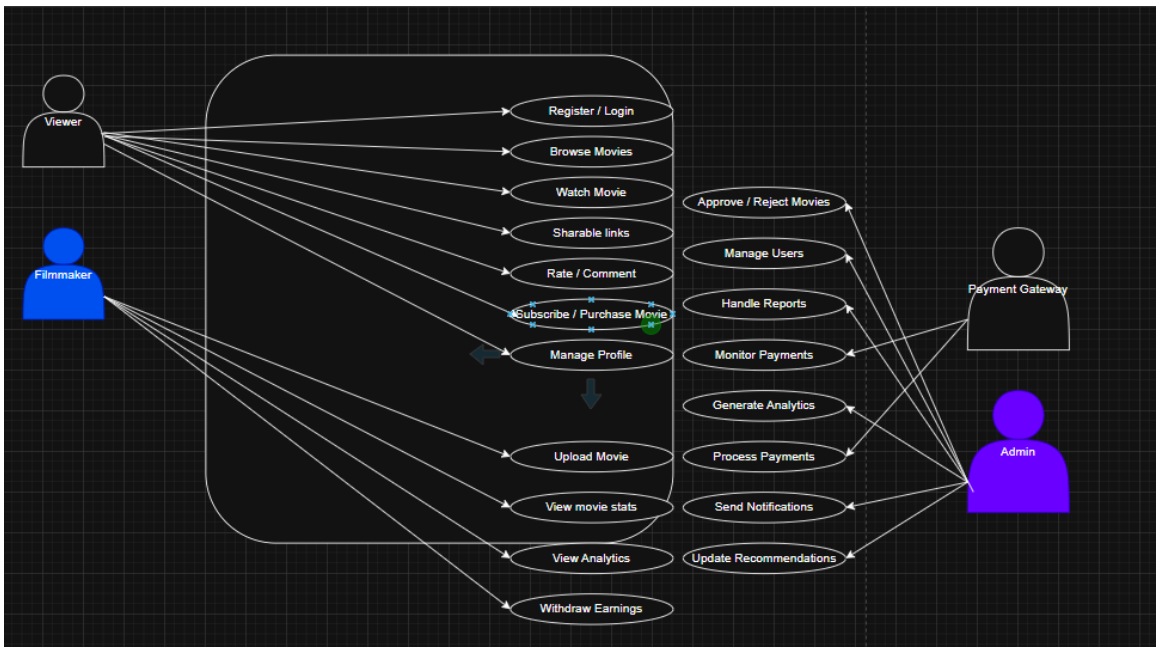
Purpose: Visualizes the main interactions between external actors (Viewers, Filmmakers, Admins, Payment Systems) and the VIEWESTA system.

Main Elements:

- Actors: Viewer, Filmmaker, Admin, Payment Gateway
- System: VIEWESTA Platform
- Key Use Cases: Register/Login, Watch Movies, Upload Movies, Approve Movies, Manage Payments, Rate & Review

Relationships & Interactions:

- Viewer interacts with VIEWESTA through browsing, watching, rating, and purchasing movies.
- Filmmaker interacts through uploading and managing content.
- Admin oversees and manages system moderation and approvals.
- Payment Gateway handles all monetary transactions between users and system.



2. Entity-Relationship Diagram (ERD)

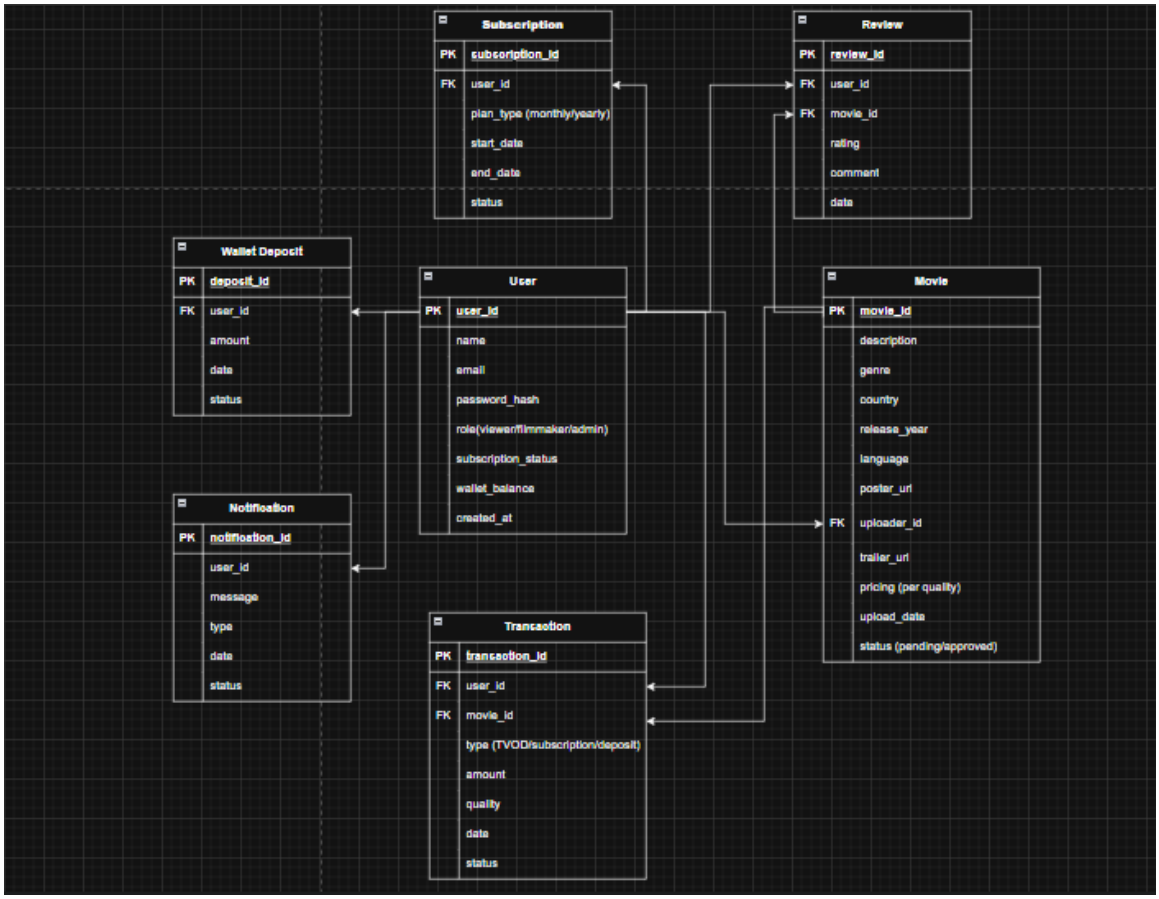
Purpose: Shows the logical data model of the VIEWESTA platform and how entities are connected.

Main Elements:

- Main Entities: User, Movie, Payment, Subscription, Review, Category, Transaction, Wallet

Relationships & Interactions:

- User —< Review >— Movie (One user can review many movies)
- User —< Payment >— Transaction (Each user can make multiple payments)
- Movie —< Category (Each movie belongs to one category)
- User —< Subscription (One user can have multiple subscriptions)



3. System Architecture Diagram

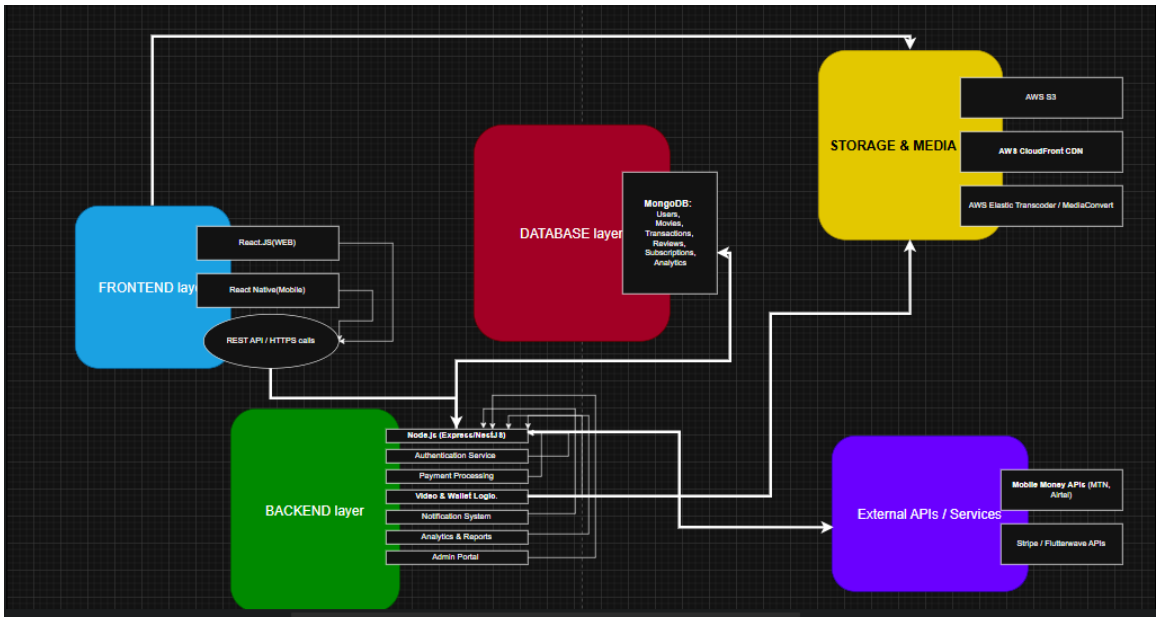
Purpose: Shows the overall structure of VIEWESTA technical ecosystem — frontend, backend, and cloud infrastructure.

Main Elements:

- Frontend: Web (React.js), Mobile App (React Native)
- Backend: Node.js (Express/VIEWESTA)
- Database: MongoDB or PostgreSQL
- Cloud Services: AWS (S3, EC2/Lambda, CloudFront, Cognito)
- External Services: Firebase, Stripe, VIEWESTA, Analytics

Relationships & Interactions:

- Frontend communicates with backend through REST APIs.
- Backend handles business logic and database communication.
- AWS handles cloud hosting, storage, CDN, and authentication.
- Payments are processed via Stripe and VIEWESTAAPIs.
- Notifications and analytics are managed using Firebase and AWS QuickSight.



4. Sequence Diagrams

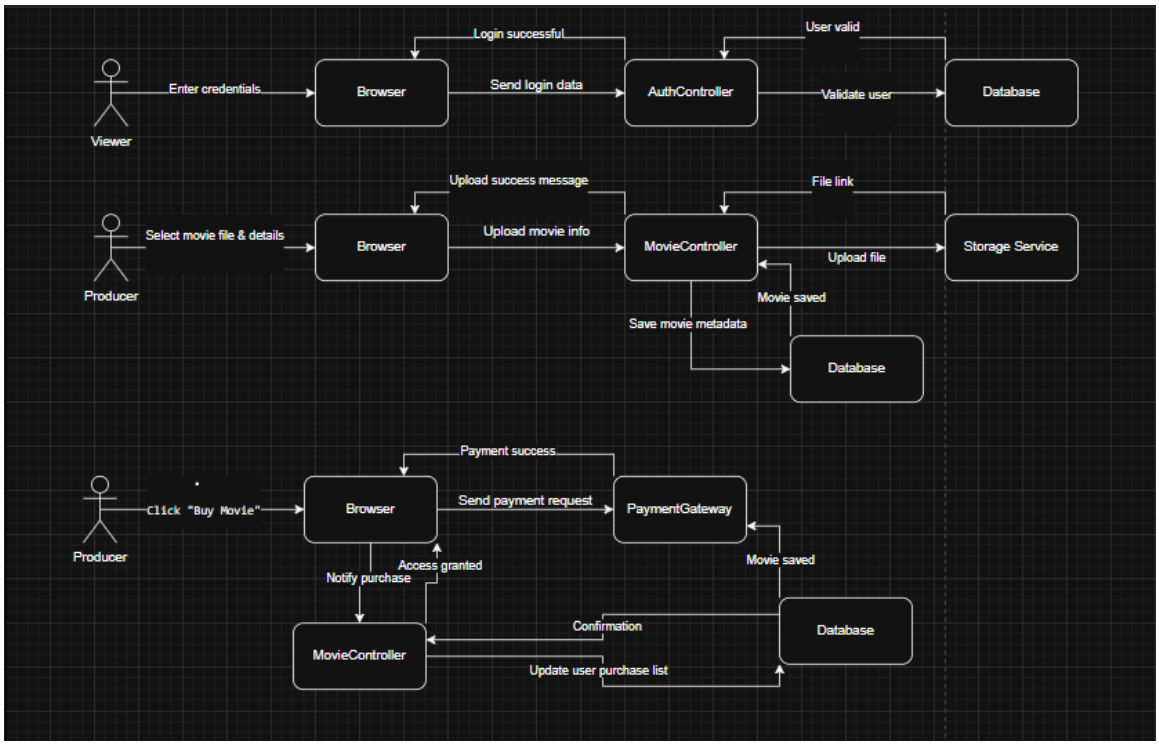
Purpose: Illustrate how system components interact over time to complete different processes.

Main Elements:

- Main Sequences: User Registration/Login, Movie Upload, Movie Purchase, Admin Approval, Movie Review & Rating

Relationships & Interactions:

- Each sequence represents the flow of data between users, controllers, databases, and external services.



5. DFD Diagram

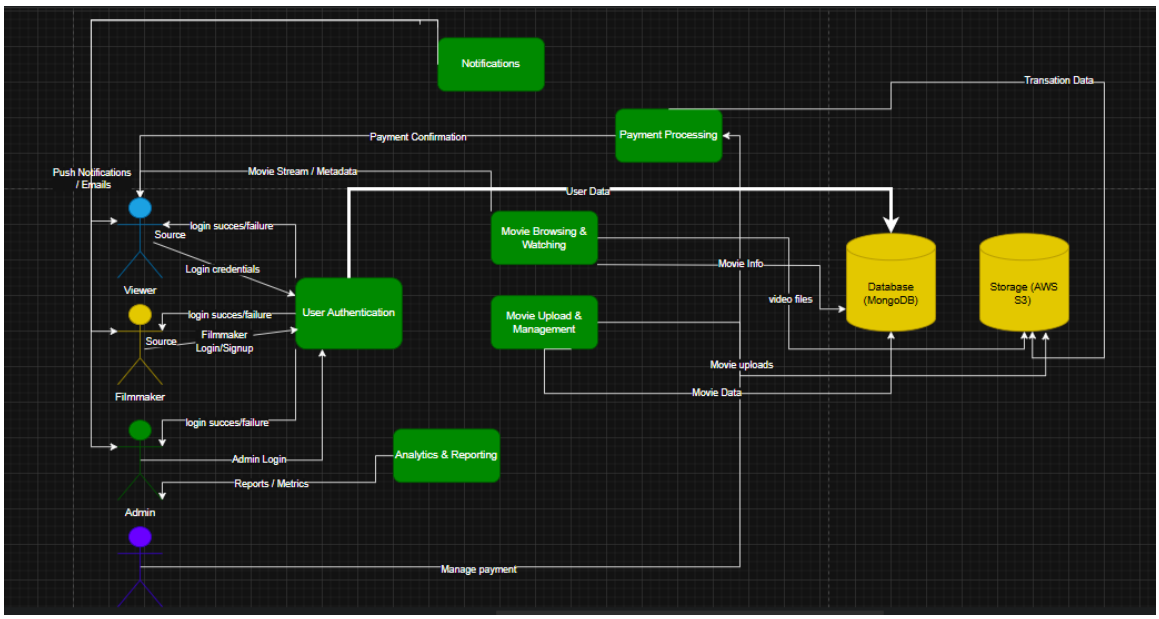
Purpose: Shows the flow of control and activities in a process — e.g., watching or uploading a movie.

Main Elements:

- Start → Login → Browse → Select Movie → Payment → Watch → Rate → End

Relationships & Interactions:

- Each step flows into the next depending on user decisions.
- Conditional branches exist for actions like 'Subscribe' or 'Top-up Wallet'.


Notifications
Transation Data
Payment Confirmation
Payment Processing
Push Notifications / Emails
Movie Stream / Metadata
User Data
Source
login succes/failure
Login credentials
Viewer
Movie Browsing & Watching
Movie Info
Database (MongoDB)
Storage (AWS S3)
video files
User Authentication
Source
login succes/failure
Filmmaker
Login/Signup
Movie Upload & Management
Movie uploads
Movie Data
Filmmaker
login succes/failure
Admin Login
Analytics & Reporting
Reports / Metrics
Admin
Manage payment