Paletta – Backend Design Architecture

last updated: 29.01.2025

Table of Contents

This document is subject to ongoing changes and acts as a communication between stakeholders and engineers.

1. User role-based features

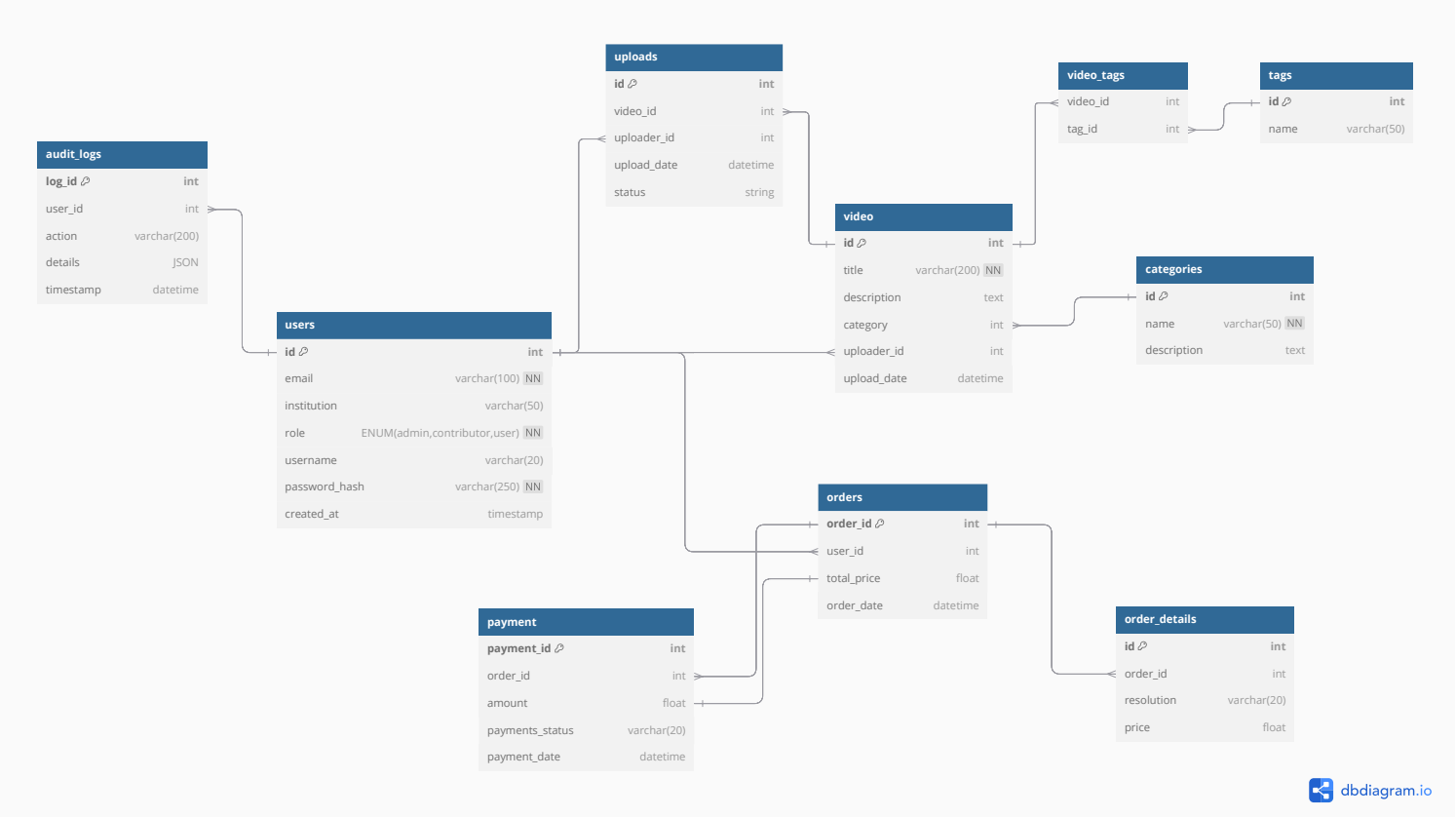
This section of the report defines the different user roles (Customer, Contributor, Admin) and their corresponding permissions within the system. It presents a role-based access control table, outlining which functionalities each user type can access.

User role-based feature access table			
Feature	**Customer**	**Contributor**	**Admin**
Register, log in, and manage their profile	✓	✓	✓
Search for clips using filters (keyword, category, tags)	✓	✓	✓
View video details (e.g., title, description, resolution options)	✓	✓	✓
View and re-download purchased videos via order history	✓	✓	✓
Add clips to a shopping cart	✓	✓	✓
Purchase clips	✓	✓	✓
Contributor inherits **Customer** access **++** the upload features:		✓	✓
Access to all Customer functionalities		✓	✓
Create/delete categories and tags at upload stage		✓	✓
Upload video clips with metadata (title, description, tags, attributes)		✓	✓
Preview thumbnail selection during uploads		✓	✓
Edit and delete their own uploaded clips		✓	✓
Full control over the system			✓
Manage user accounts (create, edit, delete)			✓
Access system analytics and logs			✓
View and manage all orders			✓
View all upload history			✓
Moderate uploaded videos			✓
Handle refunds and payment disputes			✓

2. Database design diagram

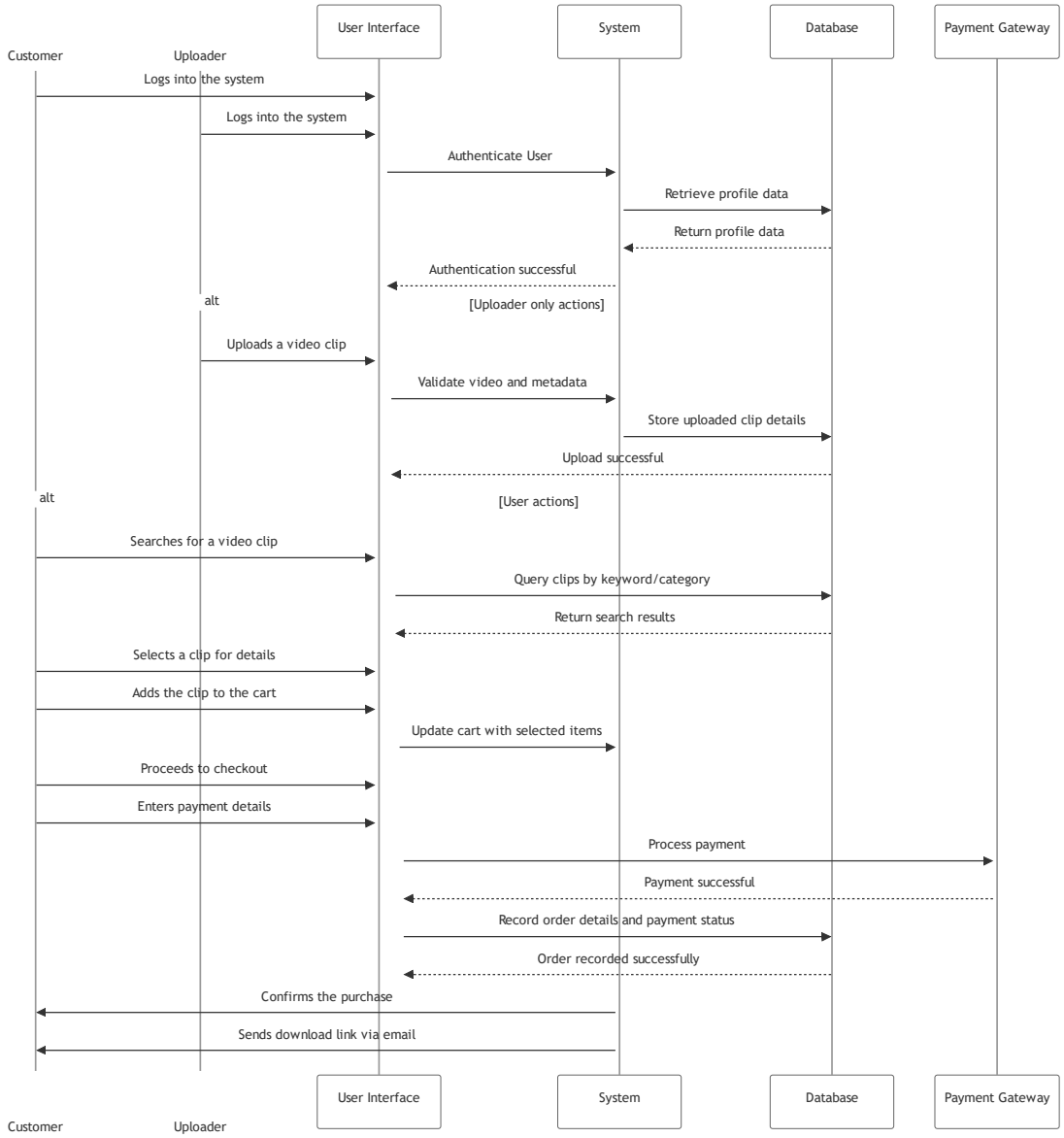
This section presents the backend relational database schema, outlining key entities and their relationships. The database is structured into multiple tables, illustrating how

data flows within the system. The focus is on designing efficient storage, retrieval, and management of user actions, video metadata, and transactions.



3. User actions sequence diagram

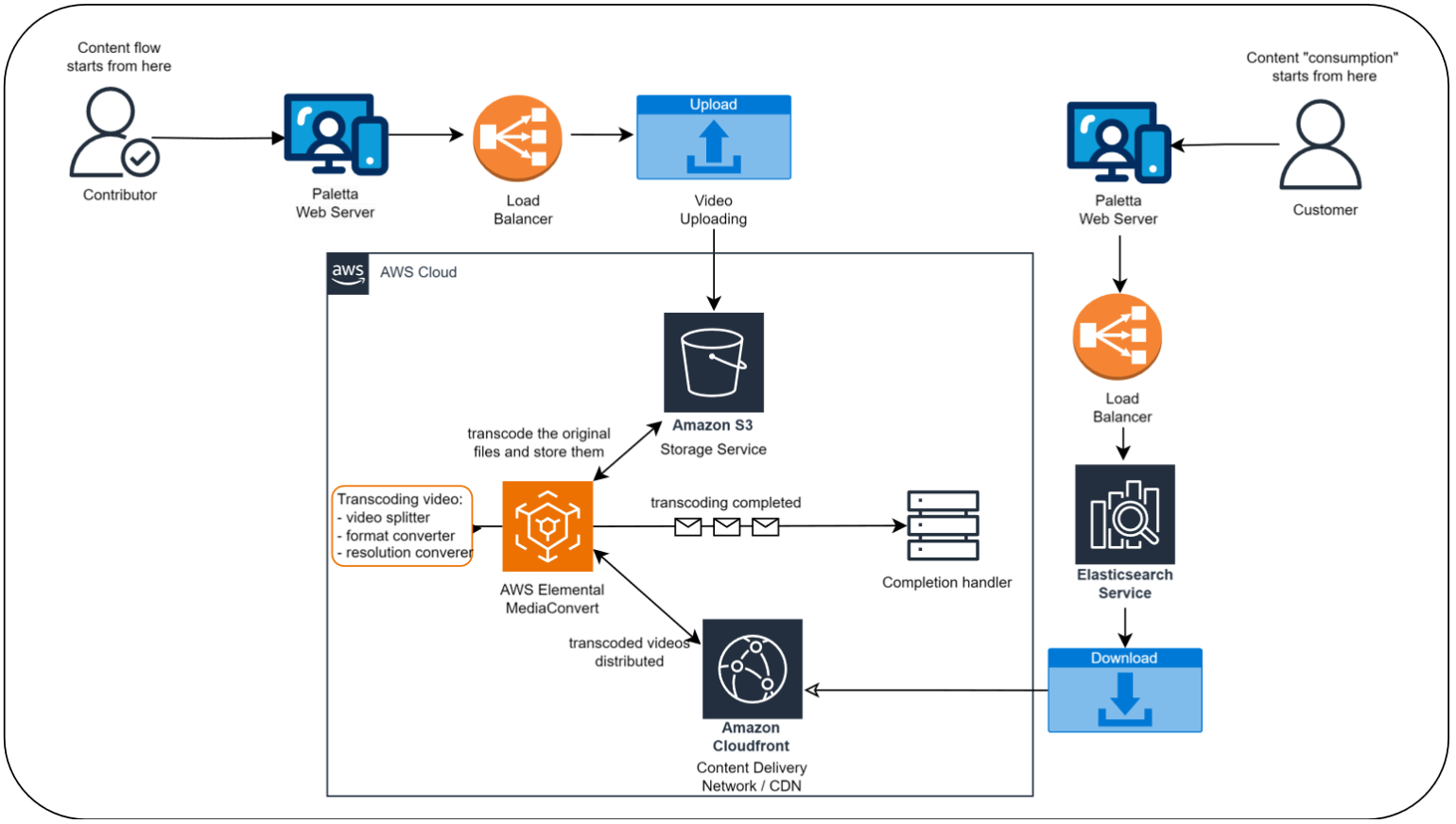
This section visualises the flow of interactions between users and the backend system, showing how different actions trigger system processes. The sequence diagram ensures efficient workflow processing, reducing system bottlenecks and maintaining a structured flow of operations.



4. High-level cloud architecture diagram

This section presents the high-level cloud-based architecture of the system, with focus on the key components for scalability, security, and performance. This architecture is designed, prioritising large-scale user interactions, both from **Contributor** and **Customer** side of view, focusing on availability, reliability, and performance. I will be setting up the backend framework using Python (FastAPI), also using AWS S3 for video

storage, PostgreSQL for the database, and will set up automated emails with purchased content download link generation. During my research, I have discovered the AWS Elemental MediaConvert tool which could automate the video transcoding process (need to research this).



***** I have not included the **Admin** in the diagram as from the discussion on Friday 27h January 2025, the focus is to get the website up and running as soon as possible – admin functionality is a feature that can be added later in the development process.

All diagrams can be found in the project's repository folder: Paletta\backend-documentation. Please, feel free to ask questions and provide me with critical feedback.