Software Requirement Specifications

Video Platform Analytics System

Version: 1.2

Project Code	https://github.com/k232003-Talal/Video_Platform_Analytics_System
Supervisor	-
Co Supervisor	-
Project Team	Syed Umer Taiyab Daniyal Ahmed Talal Ali
Submission Date	8/May/2025

Document History

[Revision history will be maintained to keep a track of changes done by anyone in the document.]

Version	Name of Person	Date	Description of change
1.0	Umer Taiyab		Document Created
1.1	Talal Ali		Added Functional Requirements
1.2	Daniyal Ahmed		Added Non-Functional Requirements

Distribution List

[Following table will contain list of people whom the document will be distributed after every sign-off]

Name	Role
Talal Ali	Team Lead
Daniyal Ahmed	Frontend Engineer
Umer	Backend Engineer

Document Sign-Off

[Following table will contain sign-off details of document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.

Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]

Version	Sign-off Authority	Sign-off Date

[Not Applicable]

Table of Contents

1. *Introduction*

1.1. *Purpose of Document*

The purpose of this document is to define the comprehensive functional and non-functional requirements for the Video Platform Analytics System. It serves as a definitive reference for the development team during implementation and provides stakeholders with a clear understanding of the system's capabilities and limitations. This document will also be used as a basis for system testing and validation.

1.2. *Intended Audience*

This document is intended for:

- Development team members responsible for implementing the Video Platform Analytics System
- Project supervisor and co-supervisor overseeing the project progress
- Any stakeholders interested in understanding the system's capabilities

1.3 Abbreviations

Abbreviation	Description
SRS	Software Requirements Specification
API	Application Programming Interface
UI	User Interface
MVC	Model View Controller
SQL	Structured Query Language
VPAS	Video Platform Analytics System
DB	Database

1.4 *Document Convention*

This document uses the following conventions:

- Main body text is in Arial, font size 12 points
- Main headings (e.g., 1 Introduction) are in Arial, font size 16 points, bold
- Subheadings (e.g., 1.1 Purpose of Document) are in Arial, font size 14 points, bold

2. *Overall System Description*

2.1. *Project Background*

The Video Platform Analytics System addresses a significant need among content creators to track and analyze their performance metrics. In today's digital landscape, video content creators need comprehensive analytical tools to understand audience engagement and optimize their content strategy. Currently, many creators struggle with accessing and interpreting their performance data in an intuitive way, especially when historical trend analysis is needed. This project aims to fill this gap by providing a standalone application that offers detailed analytics in an accessible format, particularly focusing on visualization and trend analysis.

2.2. *Project Scope*

The Video Platform Analytics System will provide the following key functionalities:

- User registration and authentication system for content creators
- Dashboard displaying key channel metrics (views, subscribers, revenue)
- Content management page with sorting and filtering capabilities
- Detailed analytics for individual videos
- Historical data visualization through graphs and charts
- User profile management

The system will be implemented as a standalone Windows application using Flutter framework with SQLite for local data storage.

2.3. *Not In Scope*

The following features are explicitly excluded from the scope of this project:

- Online database connectivity or cloud storage - all data will be stored locally
- Integration with actual video hosting platforms' APIs (such as YouTube, Vimeo, etc.)
- Video upload or content management capabilities
- Live streaming analytics
- Mobile platform compatibility
- Online user authentication
- Real-time data updates from external sources

2.4. *Project Objectives*

The primary objectives of the Video Platform Analytics System are to:

- Provide content creators with an intuitive interface to access their channel and video analytics
- Enable visualization of performance metrics through interactive graphs and charts
- Allow historical trend analysis to identify patterns in viewer engagement
- Facilitate content strategy optimization by highlighting top-performing videos
- Demonstrate effective implementation of MVC architecture with Flutter and SQLite
- Deliver a responsive and user-friendly application that meets content creators' analytical needs

2.5. *Stakeholders*

The key stakeholders for this project include:

- **Content Creators (Primary Users)**: Individuals who will use the system to track and analyze their video performance
- **Project Team**: Syed Umer Taiyab, Daniyal Ahmed, and Talal Ali who are responsible for system design and implementation
- **Course Instructor**: Evaluates the project deliverables and adherence to academic requirements

2.6. *Operating Environment*

The Video Platform Analytics System will operate in the following environment:

- **Operating System**: Windows 10 or higher
- **Platform**: Desktop application
- **Development Framework**: Flutter
- **Database**: SQLite (local storage)
- **Minimum Hardware Requirements:**
  - **Processor:** Intel Core i3 or equivalent
  - **RAM:** 4GB or higher
  - **Storage**: 200MB free space for application and additional space for data storage
  - **Display:** 1366x768 resolution or higher
- **Connectivity**: No internet connection required for core functionality

2.7. *System Constraints*

The Video Platform Analytics System is subject to the following constraints:

- Software constraints
  - Limited to Windows operating system only
  - Restricted to local data storage using SQLite
  - No real-time data updates from external sources
- Hardware constraints
  - **Processor:** Intel Core i3 or equivalent
  - **RAM:** 4GB or higher
  - **Storage**: 200MB free space for application and additional space for data storage
  - **Display:** 1366x768 resolution or higher
- Cultural constraints (includes language etc.)

  [Not Applicable]
- Legal constraints

  [Not Applicable]
- Environmental constraints (e.g., the environment where the software will be installed, It could be a noisy environment, which may require that there is no sound event in the project).

  [Not Applicable]
- User constraints (e.g., the project is developed for children, so it may be required that the project has more graphic controls rather than textual controls).

  [Not Applicable]
- Off the shelf components that might be used in the project may have their constraints that are consequently transferred to the project.]

  [Not Applicable]

2.8. *Assumptions & Dependencies*

Assumptions:

- Users have basic familiarity with analytics concepts and metrics
- The system will be used by individual content creators rather than large organizations
- Users have access to a Windows computer meeting the minimum requirements

Dependencies:

- Flutter framework and its compatibility with Windows
- SQLite database engine performance and reliability
- Availability of suitable Flutter packages for charting and visualization
- Flutter's ability to handle potentially large datasets efficiently

3. *External Interface Requirements*

3.1. *Hardware Interfaces*

The Video Platform Analytics System will interface with the following hardware components:

- **Display**: The system will support standard display resolutions (minimum 1366x768) and will adapt to different screen sizes within reasonable limits
- **Input Devices**: Standard keyboard and mouse functionality for navigation and data entry
- **Storage Devices**: Local storage drives for the application installation and database storage

3.2. *Software Interfaces*

The Video Platform Analytics System will interface with the following software components:

- **Operating System**: Windows 10 or higher, utilizing standard system APIs for file operations and UI rendering
- **SQLite Database Engine**: Version 3.35.0 or higher for local data storage and retrieval
- **Flutter Framework**: For UI rendering and application logic
- **Charting Libraries**: Integration with Flutter-compatible charting packages (e.g., fl_chart, Syncfusion Flutter Charts) for data visualization
- **File System**: Direct access to the local file system for importing/exporting data files

3.3. *Communications Interfaces*

[Not Applicable]

4. *Functional Requirements*

4.1. *Functional Hierarchy*

The Video Analytics System provides content creators with tools to authenticate, view channel statistics, and manage profiles. This functional hierarchy outlines the system's capabilities based on the defined use cases.

1. User Authentication (UC001)

- 1.1. Login
    - 1.1.1. Credential Verification
    - 1.1.2. Session Management
- 1.2. Registration
    - 1.2.1. Account Creation
    - 1.2.2. Initial Profile Setup
- 1.3. Invalid Credential Management

2. Channel Analytics (UC002)

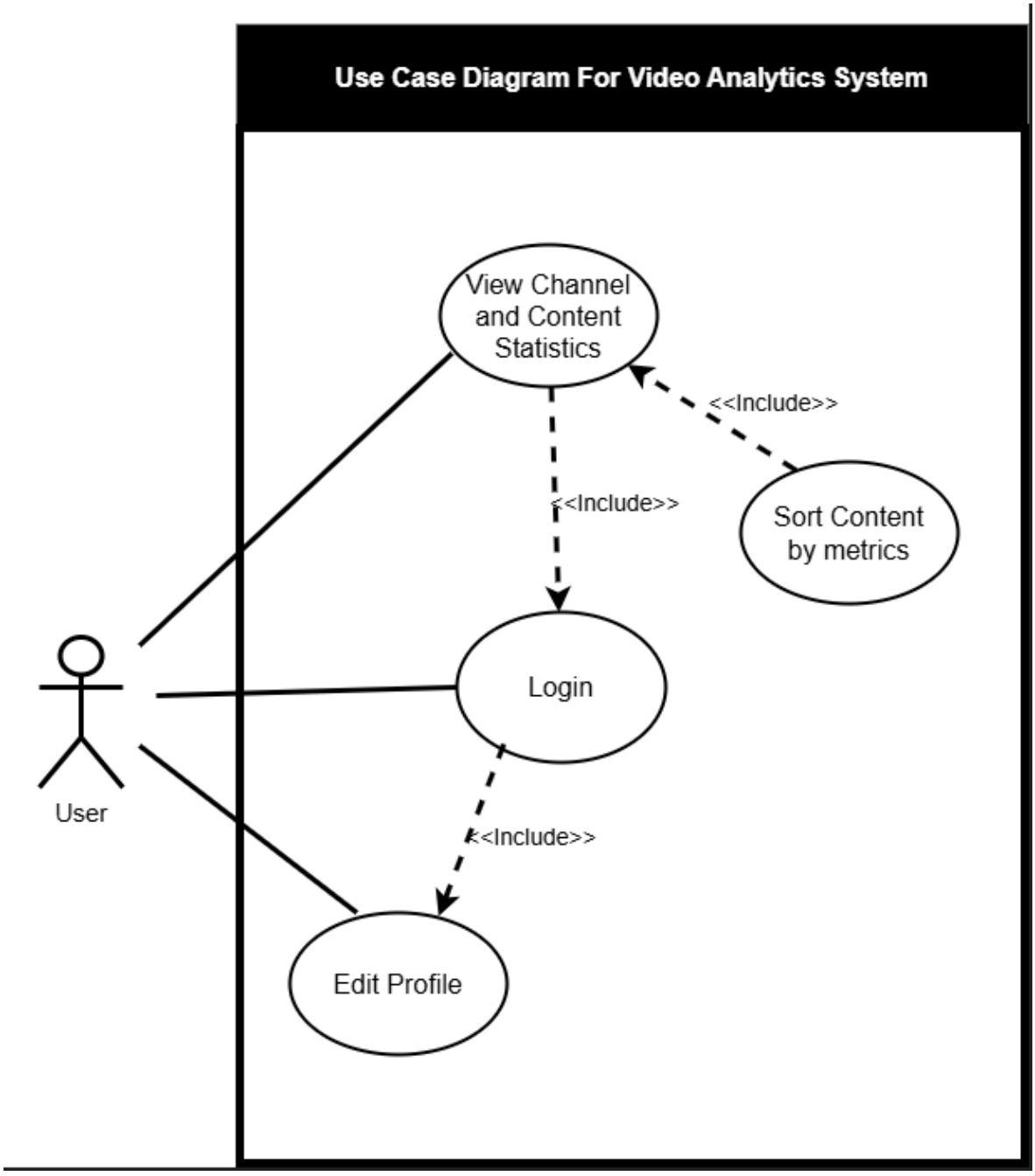
- 2.1. Dashboard Overview
    - 2.1.1. 30-Day Performance Summary
    - 2.1.2. Top Grossing Video Display
- 2.2. Content-Level Analytics
    - 2.2.1. Individual Video Statistics
    - 2.2.2. Detailed Video Performance Metrics
    - 2.2.3. Video Performance Visualization
- 2.3. Audience Feedback Analysis
    - 2.3.1. Viewer Comment Statistics
    - 2.3.2. Engagement Rate Tracking
- 2.4. Content Organization
    - 2.4.1. Metric-based Sorting
    - 2.4.2. Content Discovery

3. Profile Management (UC003)

- 3.1. Profile Editing
  - 3.1.1. Real-time Profile Updates
  - 3.1.2. Profile Data Persistence
- 3.2. Channel Information Presentation
  - 3.2.1. Public Profile View
  - 3.2.2. Creator Presentation Management

4.2. *Use Cases*

4.2.1. Video Analytics System Use Case Diagram



<Use case name: Login>		
Use case Id:	UC001	
Actors: User		
Feature: User Authentication		
Pre-condition:	User has valid account credentials	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User Starts the application or logs out	User taken to login page. prompted for credentials
2.	User submits login form	Verify credentials, take User to Account Dashboard (UC002)
3.	User Clicks Signup button	User taken to signup form. prompted to make new account
Alternate Scenarios:		
2b.	User enters invalid credentials	Prompt user to enter credentials again
3a.	User Clicks Signup button	User taken to signup form. prompted to make new account
3b.	User Submits Sign-in form	Display Success. Take user to Login page again.
Post Conditions		
Step#	**Description**	
2.	User has access to view channel statistics, edit profile, and other authorized features	
3.a.	User is added to the system	
Use Case Cross referenced	UC002 (View Channel and Content Statistics)	

<Use case name: View Channel Statistics>		
Use case Id:	UC002	
Actors: User		
Feature: Analytics Viewing		
Pre-condition:	User is logged in (UC001)	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User Views Dashboard	User is shown the overall statistics of the channel over the last 30 days. They are also shown their top grossing videos.
2	User navigates to Content page through Navigation bar	User taken to Content page. User is shown their videos along with their individual stats
3.	User selects desired metric for sorting	Content is displayed in the requested sort order
4.	User navigates to feedback page	The audience feedback statistics is shown to the user.
Alternate Scenarios:		
2.a.	User taps on particular video	The statistics for that video are shown in graphical format
2.b.	User taps on video already clicked before	The statistics for the video are hidden
4.a.	User logs out	Any changes made by user is saved (UC003) and user is taken back to login page (UC001)
Post Conditions		
Step#	**Description**	
1.	User is added to the system.	
2.	User has access to view channel statistics, edit profile (UC003), and other authorized features	
Use Case Cross referenced	UC001 (Login), UC003 (Edit Profile)	

<Use case name: Edit Profile>		
Use case Id:	UC003	
Actors: User		
Feature: Profile Management		
Pre-condition:	User is logged in (UC001)	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	User selects "Edit Profile" option	The changes made to the profile are saved and displayed on screen in real time
2.	User taps on view information	The channel information is displayed in the way it would be displayed to a viewer of the channel.
Alternate Scenarios:		
1.a.	User aborts process early OR does not click save	None of the change made (if any) are saved.
Post Conditions		
Step#	**Description**	
1.	The updated profile data is stored in the database.	
Use Case Cross referenced	UC001 (Login)	

5. *Non-functional Requirements*

Non-functional Requirements

1.1. Performance Requirements

Response Time

- *The system shall load the login page within reasonable time frame.*
- *Dashboard statistics shall be displayed immediately upon successful login.*
- *Profile updates shall be saved and reflected in real-time*
- *Content sorting operations shall complete without taking too much time.*
-

Capacity

- *The system shall be able to support at least 100 users without performance degradation.*
- *The system shall handle channels with up to 100 videos without significant performance impact.*
- *Statistical data shall be stored with consistency and accuracy.*

Availability

- *The system shall be available for use even without internet.*

Scalability

- *The architecture shall support horizontal scaling to accommodate growing user base and increasing data volume.*
- *The database shall be designed to efficiently scale as per the requirements in the future.*

1.2. Safety Requirements

Data Integrity

- *The system shall prevent corruption of statistical data during processing or storage.*
- *The system shall validate all input data to prevent injection of malicious content.*

User Protection

- *The system shall implement measures to prevent exposure of sensitive user information.*

1.3. Security Requirements

Authentication

- *The system shall enforce strong password policies (minimum 8 characters, mixture of uppercase, lowercase, numbers, and special characters).*

Authorization

- *Channel owners shall only have access to their own channel's data.*

1.4. User Documentation

System Documentation

- *Database schema documentation shall be maintained for system administrators.*
- *Architecture documentation shall be available for technical stakeholders.*

Documentation Updates

- *All documentation shall be updated with each major release.*
- *Version history shall be maintained for all documentation.*
- *Users shall be notified of significant documentation updates.*
- *Documentation shall be available in both online and offline formats.*

6. *References*

1. **Flutter Documentation:**
   - Flutter SDK Documentation (2025). https://docs.flutter.dev/
   - Flutter State Management Guide (2024). https://docs.flutter.dev/data-and-backend/state-mgmt/intro
2. **SQLite Documentation:**
   - SQLite Documentation (2025). https://www.sqlite.org/docs.html
3. **Flutter-SQLite Integration:**
   - sqflite Package Documentation (2025). https://pub.dev/packages/sqflite
4. **UI/UX Design Resources:**
   - Flutter Material Components (2025). https://docs.flutter.dev/ui/widgets/material
5. **Similar Applications and Systems:**
   - YouTube Studio Analytics Interface (2024)
   - TikTok Creator Analytics Dashboard (2024)

7. *Appendices*

[Not Applicable]