**Nhóm 23**

**Đề tài: Phát triển service xử lý và lưu trữ video**

**Họ và tên: Nguyễn Minh Sơn**

**MSSV: 20110713**

**Milestone 3**

**Task 6: Mô tả Overall Description**

1. **Product Perspective**

Sản phẩm được thiết kế để cung cấp một nền tảng tập trung để quản lý nội dung video, cho phép người dùng tải lên, lưu trữ và truy xuất video, cũng như chỉnh sửa và xem thông tin về dữ liệu của video.

Được dự định tích hợp với các hệ thống và ứng dụng khác, chẳng hạn như hệ thống quản lý nội dung và các nền tảng phát trực tuyến. Việc tích hợp này sẽ cho phép người dùng dễ dàng truy cập và quản lý nội dung video từ trong quy trình và nền tảng hiện có của họ, tăng năng suất và hiệu quả

Phần này cũng mô tả kiến trúc tổng thể của ứng dụng, bao gồm các thành phần chính và cách chúng tương tác với nhau. Điều này bao gồm các hệ thống lưu trữ và quản lý video

Sử dụng Cơ sở dữ liệu MongoDB lưu trữ những thông tin chi tiết về project video: tên video file, ảnh thumbnail, đường dẫn, thông tin cơ bản về video (định dạng, độ dài, độ phân giải…), trạng thái xử lý video, phiên bản, vị trí lưu file,…

1. **Product Features**

* Tải lên video: Khả năng tải lên các tệp video lên máy chủ, thông qua giao diện dựa trên web hoặc một API.
* Quản lý video: Khả năng quản lý các tệp video trên máy chủ, bao gồm lưu trữ, tổ chức và phân loại chúng.
* Quản lý siêu dữ liệu: Khả năng thêm và chỉnh sửa siêu dữ liệu cho các tệp video, bao gồm tiêu đề, mô tả và thẻ.
* Chỉnh sửa video: Khả năng chỉnh sửa video như cắt khung ảnh, xoay, cắt nội dung video, chỉnh sửa độ phân giải.
* Phát video: Khả năng phát video trực tiếp trong ứng dụng hoặc thông qua trình phát bên ngoài.
* Quản lý người dùng: Khả năng quản lý người dùng và quyền hạn của họ trong ứng dụng.
* Phân phối nội dung: Khả năng phân phối nội dung video đến nhiều kênh và nền tảng, chẳng hạn như mạng xã hội và dịch vụ phát trực tuyến video.
* Phát trực tiếp: Khả năng phát nội dung video trực tiếp đến người dùng theo thời gian thực.
* Phân tích và báo cáo: Khả năng theo dõi việc sử dụng video và sự tương tác thông qua các tính năng phân tích và báo cáo.
* Tự động hóa quy trình làm việc: Khả năng tự động hóa quy trình sản xuất video và các nhiệm vụ, chẳng hạn như chuyển mã và mã hóa.

1. **User Classes and Characteristics**

* **User Classes:** Hệ thống chỉ hỗ trợ một loại người dùng nói chung là người dùng cuối (End user). Dịch vụ được thiết kế cho người dùng có nhu cầu upload, chỉnh sửa và stream video. Những người dùng này có thể bao gồm người tiêu dùng, khách hàng hoặc nhân viên nội bộ cần truy cập vào nội dung video cho mục đích đào tạo, tiếp thị hoặc mục đích khác. Người dùng cuối có thể có các cấp độ kỹ thuật khác nhau và có thể truy cập ứng dụng từ nhiều thiết bị và nền tảng khác nhau.
* **User** **Characteristics:**. Đây mới chỉ là API, đòi hỏi người dùng phải có kiến thức cơ bản để có thể sử dụng được. Người dùng đều có một số yêu cầu chung
  + Yêu cầu ứng dụng dễ sử dụng và thân thiện với người dùng
  + Yêu cầu các tính năng
    - Tạo, xem, xoá project video
    - Chỉnh sửa video: scale, trim, rotate, crop
    - Xem video
    - Nhân đôi project video
    - Tạo ảnh thumbnail

1. **Operating Environment**

* Mục đích của phần này là cung cấp cho đội phát triển và các bên liên quan một cái nhìn tổng quan về các điều kiện vật lý và hệ thống mà ứng dụng sẽ hoạt động trong đó.
* Ứng dụng được thiết kế để hoạt động trên một môi trường máy chủ tương đối phức tạp, với các yêu cầu về phần cứng, hệ điều hành và các thành phần hệ thống khác. Điều này bao gồm việc yêu cầu sử dụng các thành phần phần cứng như bộ vi xử lý mạnh mẽ, bộ nhớ RAM lớn, ổ đĩa cứng với dung lượng lớn và kết nối mạng nhanh chóng,ổn định.
* Ứng dụng được thiết kế để triển khai trên một môi trường đám mây, chẳng hạn như Amazon Web Services hoặc Microsoft Azure, để đảm bảo tính khả dụng cao và khả năng mở rộng
* Yêu cầu hệ điều hành cho ứng dụng bao gồm các hệ điều hành dựa trên Linux, chẳng hạn như Ubuntu, CentOS hoặc Debian. Ứng dụng cũng yêu cầu một máy chủ web, chẳng hạn như Nginx hoặc Apache, để xử lý các yêu cầu HTTP
* Ứng dụng máy chủ video phụ thuộc vào một số thành phần phần mềm của bên thứ ba, bao gồm thư viện chỉnh sửa video FFmpeg, ngôn ngữ Python 3, cơ sở dữ liệu NoSQL MongoDB

1. **Design and Implementation Constraints**

* Một ràng buộc thiết kế quan trọng là sử dụng các công nghệ và tiêu chuẩn để đảm bảo tính tương thích và dễ dàng bảo trì. Hệ thống cần phải hỗ trợ các chuẩn mã hóa và định dạng video phổ biến để đảm bảo tính linh hoạt trong việc nhập và xuất dữ liệu. Điều này sẽ đòi hỏi cân nhắc kỹ lưỡng về kiến trúc hệ thống, cũng như lựa chọn các công nghệ và framework phù hợp. Cụ thể
  + Thiết kế Web API: sử dụng framework Flask
  + Kiểm tra API: Swagger
  + Ngôn ngữ: Python 3
  + Thư viện sử dụng: FFmpeg, Celery, RabbitMQ…
  + Cài đặt cơ sở dữ liệu cho ứng dụng bằng MongoDB
  + Xử lý các công việc chỉnh sửa video đồng thời bằng thư viện: Celery, RabbitMQ
* Các hạn chế về hiệu suất và tài nguyên của hệ thống cũng phải được xem xét trong quá trình thiết kế và triển khai. Đặc biệt, hệ thống cần phải có khả năng xử lý đồng thời nhiều yêu cầu của người dùng và đảm bảo tính khả dụng và ổn định.
* Cần thiết lập ứng dụng an toàn và đáng tin cậy, với các biện pháp phù hợp để bảo vệ chống lại việc xâm nhập dữ liệu và đảm bảo tính toàn vẹn của dữ liệu. Điều này sẽ đòi hỏi việc triển khai các giao thức bảo mật mạnh mẽ, cũng như theo dõi và bảo trì liên tục
* Các ràng buộc tiềm năng khác có thể bao gồm tính tương thích với các hệ thống và công nghệ hiện có, chẳng hạn như codec và giao thức phát trực tuyến video, cũng như tuân thủ các yêu cầu pháp lý và quy định liên quan đến quyền riêng tư dữ liệu và phân phối nội dung

1. **User Documentation**

* Giúp người dùng cuối hiểu rõ về ứng dụng và sử dụng các chức năng một cách hiệu quả.
* Cung cấp cho người dùng cuối các thông tin cần thiết để bắt đầu sử dụng ứng dụng, bao gồm cách tải lên, lưu trữ, chỉnh sửa video, xem thông tin metadata của video và phát lại video trên trình duyệt.
* Giải thích các quy trình và quy trình làm việc cơ bản, giúp người dùng cuối hiểu rõ hơn về cách sử dụng ứng dụng và tối ưu hóa trải nghiệm người dùng.
* Được thiết kế để truy cập và thân thiện với người dùng, với mục tiêu giúp người dùng nhanh chóng và dễ dàng sử dụng ứng dụng
* Cần được cập nhật thường xuyên để phản ánh các tính năng và thay đổi mới nhất của ứng dụng.

1. **Assumptions and Dependencies**

Phần giả định và phụ thuộc giúp đảm bảo rằng nhóm phát triển hiểu rõ những yếu tố bên ngoài có thể ảnh hưởng đến việc phát triển hoặc sử dụng ứng dụng, để có thể chú ý khi triển khai ứng dụng.

* **Assumptions**
  + Ứng dụng sẽ được phát triển bằng các ngôn ngữ lập trình và framework hiện đại như Python, Flask.
  + Ứng dụng sẽ được đặt trên cơ sở hạ tầng máy chủ dựa trên đám mây như AWS hoặc Google Cloud để đảm bảo tính mở rộng và đáng tin cậy.
  + Người dùng của ứng dụng sẽ có kết nối internet tốc độ cao và trình duyệt web hiện đại để đảm bảo hiệu suất tối ưu.
  + Ứng dụng sẽ được phát triển bằng các phương pháp phát triển linh hoạt, với các bản phát hành và cập nhật thường xuyên để đảm bảo sự cải thiện liên tục.
* **Dependencies**

Ứng dụng phụ thuộc vào các thư viện và framework bên ngoài, chẳng hạn như

* + Flask 2.0 hoặc cao hơn để xây dựng API
  + FFmpeg, để xử lý và chuyển đổi video.
  + Database: MongoDB
  + Celery, RabbitMQ để xử lý các công việc đồng thời

Ứng dụng có thể cần tích hợp với các dịch vụ và nền tảng bên thứ ba, chẳng hạn như mạng lưới phân phối nội dung hoặc cổng thanh toán, để cung cấp các chức năng bổ sung