**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Lò Văn Kiên**

**THIẾT KẾ VÀ XÂY DỰNG GIẢI PHÁP HỖ TRỢ BIÊN TẬP VÀ HỌC TRỰC TUYẾN CÁC BÀI GIẢNG MULTIMEDIA DỰA TRÊN CÔNG NGHỆ WEBRTC**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành:** **Công nghệ thông tin**

## **HÀ NỘI - 2017**

## HÀ NỘI - 20**< hai số cuối của năm bảo vệ ĐATN>**

*(chữ hoa, 12pt, đậm, căn giữa)*

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

# **Lò Văn Kiên**

**THIẾT KẾ VÀ XÂY DỰNG GIẢI PHÁP HỖ TRỢ BIÊN TẬP VÀ HỌC TRỰC TUYẾN CÁC BÀI GIẢNG MULTIMEDIA DỰA TRÊN CÔNG NGHỆ WEBRTC**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành:** **Công nghệ thông tin**

**Cán bộ hướng dẫn: TS. Hoàng Xuân Tùng**

**HÀ NỘI - 2017**

## HÀ NỘI - 20**<hai số cuối của năm bảo vệ ĐATN>**

*(chữ hoa, 12pt, đậm, căn giữa)*

**LỜI CẢM ƠN**

Để hoàn thành khóa luận tốt nghiệp này, em đã nhận được rất nhiều sự ủng hộ động viên đến từ các thầy cô, bạn bè. Chính vì thế những lời đầu tiên này em xin được gửi lời cảm ơn đến những người đã giúp đỡ em trong thời gian qua. Đầu tiên, em xin được gửi lời cảm ơn sâu sắc đến TS.Hoàng Xuân Tùng, người đã trực tiếp hướng dẫn, chỉ bảo giúp đỡ em tận tình trong suốt quá trình phát triển sản phẩm cũng như hoàn thành báo cáo. Thầy là người đã định hướng, chia sẻ kinh nghiệm quý báu của mình và tạo điều kiện tốt nhất cho em hoàn thành khóa luận. Em cũng xin được chân thành cảm ơn các quý thầy cô trong khoa công nghệ thông tin, trường Đại học Công Nghệ - ĐHQG Hà Nội đã trang bị cho sinh viên lượng kiến thức nền tảng vững chắc trong quá trình học tập tại trường. Các thầy cô đã chia sẻ tới sinh viên những kinh nghiệm quý báu qua công việc cũng như qua trải nghiệm thực tế của mình giúp cho sinh viên có những trang bị tốt nhất để phục vụ cho cuộc sống cũng như công việc sau này. Do thời gian làm khóa luận còn hạn chế cũng như sự hạn chế về kiến thức và kinh nghiệm nên trong khóa luận này khó tránh khỏi những sai sót. Em rất mong nhận được sự góp ý của quý thầy, cô và các bạn để khóa luận được hoàn thiện hơn. Em xin chân thành cảm ơn!

Mục Lục

Danh mục các bảng

[Đăng nhập google driver 25](#_Toc480988614)

[Upload video 27](#_Toc480988615)

[Usecase remove video 37](#_Toc480988616)

[Usecase Share Video 38](#_Toc480988617)

# Mở Đầu

## Đặt vấn đề

Hiện nay nhu cầu học tập online đang là một xu thế khi thế giới interet bùng nổ. Các phương pháp giáo dục trực tuyến Elearning được áp dụng trên nhiều trường đại học, hay trên youtube. Học sinh sinh viên thường học tập qua video giảng dạy hơn là đọc các ebook. Học tập qua video sẽ giúp người học có cái nhìn trực quan hơn, dễ hiểu hơn và có thể học mọi lúc mọi nơi, bất kỳ khi nào người dùng muốn. Nhưng đối với người dạy hay những người muốn chia sẻ kiến thức thì phải dùng một công cụ chuyên biệt để có thể truyền đạt dữ liệu. Một trong những phương thức được sử dụng phổ biến để chia sẻ kiến thức là quay 1 đoạn video rồi chia sẻ trên các mạng xã hội hoặc diễn đàn (youtube,facebook …). Còn đối với các tổ chức hay các trang web chuyên cung cấp các bài giảng trực tuyến cho người dùng đăng ký học lại cung cấp một số giải pháp riêng để hỗ trợ việc soạn thảo và biên tập video bài giảng. Mà hầu hết chúng đều là ứng dụng bên thứ ba không gắn kết hay liên quan tới các trang web đó cả. Điển hình như sử dụng các ứng dụng desktop để ghi hình video rồi upload lên cơ sở dữ liệu của ứng dụng. Các thao đó rất mất thời gian và phụ thuộc vào sự hiểu biết của người sử dụng ứng dụng.

Vì vậy khi công nghệ webrtc ra đời, nó có thể hỗ trợ những thao tác mà người dùng thường phải làm khi tạo ra một bài giảng trực tuyến bằng video một cách dễ dàng, và đặc biệt hơn là nó chỉ cần trình duyệt để chạy, đồng nghĩa với nó có thể là một modul hay một chức năng của một trang web học trực tuyến dành cho quản trị viên hay giảng viên.

## Phạm vi và mục tiêu của khóa luận

Khóa luận này sẽ tập trung vào tìm hiểu các mô hình học trực tuyến và tìm giải pháp tích hợp việc soạn thảo biên tập bài giảng video ngay trên các trang web học trực tuyến, Khóa luận sẽ nghiên cứu công nghệ WebRTC và các API của trình duyệt để ứng dụng vào giải pháp trên. Ngoài ra trong khóa luận cũng sử dụng các API của google driver dùng để lưu trữ video trực tuyến. Vì hiện tại chưa có mô hình thực tiễn đang hoạt động để áp dụng trực tiếp giải pháp vào nên trong khóa luận này chỉ xây dựng một ứng dụng đơn giản mô phỏng các chức năng chính sẽ được sử dụng như các ứng dụng bên thứ ba mà các modul hiện tại có thể đang sử dụng. Và nó có thể được tích hợp dễ dàng vào các mô hình học trực tuyến kia.

# Tổng Quan Về Mô hình học trực tuyến.

* Nêu ra các mô hình tiêu biểu,các công nghệ su dụng, so sánh
* Ket luan hen tai ko can cac plugin mà co the dung ngay tnh dyet
* Lam phan 3 4 xong da

## Khảo sát vấn đề nghiên cứu

Hiện nay các mô hình học trực tuyến phổ biến trong và ngoài nước có thể kể đến: courser, topica,edx,kyna,edumail. Cùng tim hiểu qua các mô hình này để thấy nó có sự tác động đến đề tài này như thế nào.

Coursera:

Cousrsera được thành lập vào năm 2012 bởi các giáo sư khoa học máy tính của đại học Stanford :  [Andrew Ng](https://en.wikipedia.org/wiki/Andrew_Ng) , [Daphne Koller](https://en.wikipedia.org/wiki/Daphne_Koller) Ng, Koller.

Coursera hợp tác với nhiều trường đại học trên thế giới để cung cấp một số khoá học trên mạng của các trường này cho người đăng ký, các khoá học có thể thuộc ngành [khoa học kỹ thuật](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_k%E1%BB%B9_thu%E1%BA%ADt), [nhân văn học](https://vi.wikipedia.org/wiki/Nh%C3%A2n_v%C4%83n_h%E1%BB%8Dc), [y học](https://vi.wikipedia.org/wiki/Y_h%E1%BB%8Dc), [sinh học](https://vi.wikipedia.org/wiki/Sinh_h%E1%BB%8Dc), [khoa học xã hội](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_x%C3%A3_h%E1%BB%99i), [toán học](https://vi.wikipedia.org/wiki/To%C3%A1n_h%E1%BB%8Dc), [kinh tế học](https://vi.wikipedia.org/wiki/Kinh_t%E1%BA%BF_h%E1%BB%8Dc), [khoa học máy tính](https://vi.wikipedia.org/wiki/Khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh) và một số lĩnh vực khác. Coursera cung cấp rất nhiều khóa học trên mạng miễn phí , mỗi khóa bao gồm mộ đoạn phim về các bài giảng của giảng viên cung các bài tập về nhà. Các bài giảng có thể sẽ tính phí nếu học chuyên sâu hoặc tính phí cả toàn khóa. Với khóa học tính phí người học sẽ được cấp chứng chỉ khi hoàn thành và giao tiếp trực tiếp với người giảng bằng webcam. Hiện nay tổng số đại học liên kết với công ty là 83 và cung cấp các khóa học bằng nhiều thứ tiếng.

Trang chủ : <https://www.coursera.org/>

Edx:

EdX là một dự án phi lợi nhuận của MIT và Harvard nhằm mang đến các khóa học trình độ đại học một cách miễn phí tới cho bất cứ ai trên trái đất. Được bắt nguồn từ MIT (Viện Công Nghệ Massachusetts), edX có tất cả những điều mà trước kia là ước mơ của rất nhiều người: các khóa học chất lượng cao được giảng dạy bởi các giáo sư hàng đầu ở các trường đại học danh tiếng, khả năng học ở bất cứ đâu, bất cứ hoàn cảnh nào và bất cứ thời gian nào, mà đặc biệt nó free (trừ khi bạn muốn nhận chứng chỉ)

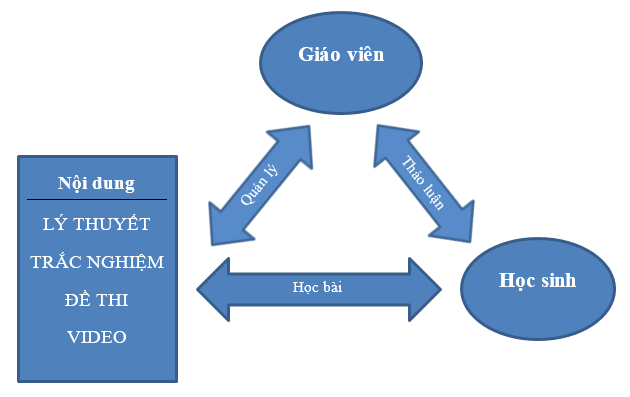
Trang chủ: <https://www.edx.org/>

Udemy:

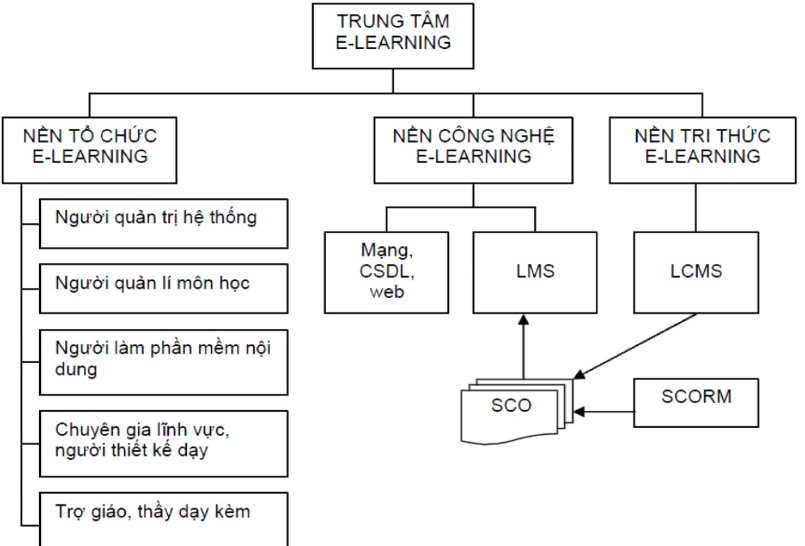
Thành lập năm 2010 và nhanh chóng trở thành MOOC có nhiều khóa học Online nhất. Udemy hoạt động vì lợi nhuận và thu một mức phí nhỏ phổ biến từ 20-40USD/khóa học. Ở thời điểm hiện tại Udemy đang có hơn**40.000** khóa học.

Udemy có ưu điểm cực lớn là tập trung chuyên sâu vào dạy thực hành. Và gần như bạn có thể học mọi thứ mà bạn thể nghĩ ra ở Udemy. Những khóa đông học viên nhất trên Udemy thì vẫn luôn là những khóa học về CNTT, lập trình với những khóa học có đến cả trăm nghìn học viên cùng tham gia học

Trên đây là 3 ví dụ điển hình cho việc mô tả các mô hình học trực tuyến. Tuy có thể có các business hoạt động khác nhau nhưng chúng đều tuân theo mô hình kiến trúc cơ bản sau:



Hình 1.1: Mô hình tổng quan tương tác giữa các đối tượng



Hình 1.2: Mô hình dữ liệu và hạ tầng

**Admin**: Là người quản trị hệ thống, có quyền cao nhất, chức năng chính thường là tạo khóa học/lớp học, quản lý giảng viên, quản lý tài liệu, quản lý sinh viên và các công việc liên quan đến thống kê và phân tích dữ liệu khác.

**Giảng viên**: là người trực tiếp giảng dạy trên hệ thống, sở hữu các tài liệu tự soạn thảo hoặc thu thập. Có quyền thêm sửa xóa tài liệu. Tùy hệ thống có thể tạo lớp học và quản lý sinh viên. Tương tác trực tiếp với sinh viên/người học.

**Sinh viên**: đăng ký tài khoản, tham gia các lớp học, xem và tải các tài liệu. Có thể tương tác trực tiếp với giảng viên. Các hành động của sinh viên có thể miễn phí hoặc mất phí.

**Tài liệu**: có thể là đề thi, ebook, âm thanh, hình ảnh … được giảng viên biên soạn hoặc thu thập. Tài liệu cũng có thể là do hệ thống có sẵn trong dữ liệu. Tài liệu thường được lưu trữ online (trên mây) hoặc offline (local). Cũng có tài liệu phải trả phí hoặc miễn phí.

Các mô hình học trực tuyến trên đã khá hoàn chỉnh cho việc dạy và học online, nhưng qua tìm hiểu thì ta vẫn có thể cải thiện hoặc thêm một số chức năng đã có. Một trong số đó phải kể đến biên soạn bài giảng bằng video.

Hiện tại thì khi muốn làm một video giảng dạy giảng viên phải cài các tool chuyên .Các công cụ hay được sử dụng là OBS (Open Broadcaster Software), Cam Studio…. Ưu điểm của các công cụ này là nhiều tính năng và ghi được video sắc nét, chỉnh sửa và cấu hình được việc quay hình. Nhưng nhược điểm là nặng nề, có thể phải cài thêm các plugin phụ trợ. Đặc biệt khó sử dụng với những người không chuyên và nhiều chức năng không cần thiết khi yêu cầu đơn giản chỉ là ghi hình. Đa số các công cụ này đều không gắn với trang web tức là người dung không thể sử dụng mọi lúc mọi nơi bất kể khi nào mạng, muốn dùng thì phải cài lại.

Nhận ra điểm yếu đó khóa luận hướng tới tạo ra một ứng dụng online có thể tích hợp vào các trang web học trực tuyến giúp ghi hình một cách dễ dàng quá trình giảng dạy hoặc hỗ trợ giảng viên soạn thảo các bài giảng video một cách dễ dàng.

* 1. Đối tượng và phạm vi

Hướng tới mục tiêu hỗ trợ người dùng có một công cụ đơn giản, nhẹ mà không phải cài đặt phức tạp, khóa luận này sẽ tìm hiểu và xây dựng một ứng dụng giúp giảng viên có thể dễ dàng ghi hình video (cả màn hình và webcam ) để phục vụ cho mục đích chia sẻ kiến thức. Khóa luận sẽ chỉ xây dựng công cụ để hỗ trợ ghi hình và lưu trữ video chứ chưa được tích hợp trực tiếp với mô hình học trực tuyến cụ thể nào.

## Phương pháp nghiên cứu và thực hiện đề tài

# Tổng Quan Về Kỹ Thuật Sử Dụng

1. WebRTC và các thư viện sử dụng

### Webrtc là gì

**WebRTC** (**Web Real-Time Communication**) là một web API được phát triển bởi World Wide Web Consortium (W3C), khả năng hỗ trợ trình duyệt (browser) giao tiếp với nhau thông qua VideoCall, VoiceCall hay transfer data "Peer-to-Peer" (P2P) mà không cần browser phải cài thêm plugins hay phần mềm hỗ trợ nào từ bên ngoài.

Nó tập hợp các api được sử dụng cho nhiều mục đích khác nhau, Có hàm chỉ để làm những việc đơn giản như đòi quyền truy cập vào webcam và microphone của máy tính, có hàm phức tạp hơn thì để thiết lập kết nối giữa hai người dùng với nhau, có hàm còn dùng để chia sẻ màn hình với người khác. Và rồi có hàm để hai người gọi video cho nhau, cũng là chức năng "nổi tiếng" nhất của WebRTC tính đến thời điểm hiện tại. Tuy nhiên, tất cả mọi hàm lập trình nằm trong bộ API có một điểm chung vô cùng quan trọng: chúng thực thi hầu hết các tác vụ theo thời gian thực. Đó là lý do vì sao chữ Real-Time xuất hiện trong cái tên của bộ hàm này.

WebRTC hỗ trợ các trình duyệt google chrome , firefox, opera, mục đích của nó là  "mang lại các ứng dụng phong phú, chất lượng cao và chạy theo thời gian thực có thể được phát triển bởi lập trình viên cho các trình duyệt, nền tảng di động, thiết bị Internet of Things, và cho phép tất cả bọn chúng liên lạc với nhau thông qua một bộ các giao thức chung".

Để sử dụng các hàm lập trình WebRTC, các lập trình viên có thể dung rất nhiều ngôn ngữ như javascript, android,ios…

### Lịch sử phát triển

Ý tưởng về WebRTC được đưa ra bởi nhóm phát triển Google Hangouts từ năm 2009. Tại thời điểm đó, để thực hiện truyền thông đa phương tiện trên nền Web, người sử dụng phải cài đặt Flash và các plugin khác. Năm 2010, Google mua lại hai công ty là On2 và Global IP Solutions (GIPSVào đầu năm 2010, Google đã hoàn tất việc mua lại On2, một công ty chuyên về mã hoá hình ảnh đã phát triển bộ mã VP, mới nhất hiện này là phiên bản VP9. On2 đã xác định bộ mã của mình như một sản phẩm bản quyền miễn phí thay thế cho bộ mã H.26x, mà sẽ được tiêu chuẩn hoá, đăng ký bản quyền và sử dụng rộng rãi. Google sau đó đã mở các công nghệ của On2 cho thế giới và phát hành mã nguồn mở VP8 dưới tên gọi WebM. Ý tưởng của việc này là thay thế H.264 cho video trên web và thông qua đó, giảm chi phí bản quyền cho mọi người, đặc biệt là cho chính Google.

Google đã tiếp tục mua lại Global IP Solution (GIPS) trong năm 2010, một công ty nổi tiếng với sản phẩm media frameworks – công nghệ giúp phát triển các ứng dụng VoIP và Video Call dễ dàng hơn. Vào thời điểm đó, GIPS đang có một thị phần lớn trong thị trường VoIP, điều này khiến phần lớn các công ty cùng ngành phải chạy đi tìm kiếm các giải pháp thay thế. Cũng giống như với On2, Google sử các tài sản của GIPS và công khai mã nguồn mở. Lần này, với một thay đổi thú vị: họ đã vứt bỏ tất cả các bộ mã audio, video có bản quyền và thêm vào một lớp JavaScript API như là một lớp giao tiếp với các trình duyệt web. Ý tưởng là gì? Giúp mọi trình duyệt có khả năng xử lý media hai chiều và tích hợp sẵn công nghệ mã hoá media. Sau đó họ tiếp tục xúc tiến để các thứ này trở thành chuẩn tại 2 tổ chức IETF và W3C. Và [WebRTC](http://how.vndemy.com/glossary/webrtc/) đã ra đời từ đó.

### Lợi ích của WebRTC

Ở góc độ nhà phát triển, nếu không có WebRTC, việc tạo ra ứng dụng RTC đòi hỏi phải mất nhiều công sức từ việc lấy dữ liệu từ thiết bị camera, microphone đến việc thiết lập phiên, xử lý tín hiệu, truyền tín hiệu, … Nhưng với WebRTC, tất cả công việc để tạo ra một cuộc hội thoại chỉ nằm trong vài chục dòng lệnh. Việc phát triển ứng dụng với chức năng gọi điện, video chat và chia sẻ file,.. là rất đơn giản khi dùng WebRTC kết hợp giữa JavaScript và HTML5.

Ở góc độ người sử dụng, sử dụng WebRTC chỉ cần thông qua trình duyệt Web. Tính sẵn sàng cao cho phép thực hiện cuộc gọi mà không cần đăng ký tài khoản hay cài đặt thêm thành phần nào ngoài một trình duyệt có hỗ trợ WebRTC. Ví dụ, hai người dùng chỉ cần truy cập vào cùng một đường dẫn web để gọi video với nhau sử dụng trình duyệt Google Chrome hay Mozilla Firefox.

### Các ứng dụng của WebRTC

Trang web :[www.wello.com](http://www.wello.com/) cung cấp kênh thông tin cho việc giảng dạy cũng như tham gia vào các khóa học rèn luyện sức khỏe tại nhà thông qua hướng dẫn trực tiếp bằng video. Giảng viên hướng dẫn trực tiếp cho các học viên, có thể quan sát họ thực hiện các động tác và chỉnh sửa khi họ thực hiện sai. WebRTC giúp việc luyện tập trở nên thoải mái và tiện lợi hơn.

* Video call của Facebook: Thiết lập cuộc gọi giữa 2 người 1 cách dễ dàng.
* Skyper chat Web: Chat realtime ngay trên trình duyệt, gửi file dễ dàng.
* WebRTC cũng được xài để tạo ra các game nhiều người chơi mà không cần cài gì thêm, người ta chỉ cần xài trình duyệt có hỗ trợ WebRTC là đủ. Hầu hết những trò đó đều chỉ là các tựa game giải trí nhẹ nhàng nhưng cũng có vài cái tên nổi bật như [The Hobbit: The Battle for Five Armies](https://tinhte.vn/go-to/aHR0cDovL21pZGRsZS1lYXJ0aC50aGVob2JiaXQuY29tLw==)
* Trang web Appear.in hỗ trợ người dùng tạo một phòng chat video cực kì nhanh chóng chỉ bằng cách dùng Chrome hoặc [Firefox](https://tinhte.vn/tags/firefox/) gốc, không cần phải cài thêm bất kì một plugin nào. Thậm chí người ta còn không cần phải đăng nhập hay tạo tài khoản như các app chat video hiện nay

### Các thư viện sử dụng trong khóa luận

1. ***Screen Capturing plugin***

Là một plugin miễn phí được viết bởi MuazKhan (một chuyên gia trong lĩnh vực webrtc) nhằm phục vụ cho việc lấy dữ liệu màn hình để chia sẻ hoặc phục vụ cho một số thư viện webrtc. Một số thư việc webrtc được viết bởi tác giả yêu cầu phải cài plugin này. Plugin này hỗ trợ cho việc lấy dữ liệu của các màn hình, chương trình, tab bằng đoạn code sau:

// for chrome

mandatory: {chromeMediaSource: 'screen'}

// or desktop-Capturing

mandatory: {chromeMediaSource: 'desktop'}

// for Firefox

video: {

mediaSource: 'window' || 'screen'

Hiện tại plugin chỉ hỗ trợ 2 trình duyệt là chrome và firefox. Trong khóa luận này ta chỉ sử dụng để hỗ trợ cho thư viện getScreenId mà không thể chỉnh sửa gì trong plugin.

1. ***GetScreenid***

Để đơn giản hóa việc lấy dữ liệu (stream) màn hình desktop hoặc các tab trên trình duyệt cũng như có thể tận dụng tốt các chức năng mà WebRTC có thể hỗ trợ thì thư viện getScreenId là không thể thiếu.Thực ra thư viện này là một file javascript bao gồm các hàm được tác giả viết sẵn đầy đủ cung cấp cho việc lấy id của màn hình cũng như lấy dữ liệu media của màn hình đó.

Thư viện trên github:

<https://github.com/muaz-khan/getScreenId>

Tác giả : MuazKhan

Là một chuyên gia trong lĩnh vực webrtc, ông xây dựng ra rất nhiều thư viện, hàm hỗ trợ cho việc dùng webrtc rất đơn giản. Các hướng dẫn và thư viện của ông tại: <https://www.webrtc-experiment.com/>

Về thư việc getScreenId được sử dụng như một file javascript nên nó sẽ được link như một file javasript bình thường,có thể tải về offline để sử dụng. Những lưu ý của tác giả: Phải cài đặt plugin Screen Capture, yêu cầu khi tải trang phải là https. Hàm getScreenId sẽ trả về một đối tượng MediaStream. Có thể sử dụng đối tượng này với các thư viện khác để truyền dữ liệu tới server hoặc peer-to-peer.

Trong khóa luận này đơn giản ta chỉ sử dụng chức năng chính của nó là lấy stream từ màn hình hoặc các chương trình đang chạy trên máy tính để record thành video.

1. Web APIs (MDN)

 Mozilla Developer Network (MDN) là một nền tảng học tập và phát triển các công nghệ web và các phần mềm làm tăng sự mạnh mẽ cho web,bao gồm:

* Các chuẩn Web như CSS,HTML,Javascript
* Phát triển các ứng dụng web mở
* Phát triển tiện ích cho firefox

Nó cung cấp rất nhiều interface hỗ trợ cho việc phát triển web một cách dễ dàng hơn. Trong khóa luận này ta sẽ dùng tới 2 API của nó.

<https://developer.mozilla.org/en-US/docs/Web/API>

1. **MediaDevice.getuserMedia**

MediaDevice là một interface cung cấp các quyền truy cập kết nối tới các thiết bị đầu vào như camera, microphone, cũng như việc chia sẻ màn hình. Về bản chất nó cho phép bạn truy cập vào phần cứng của dữ liệu media.

**MediaDevices.getUserMedia() là một hàm hiển thị** các quyền cho phép người dùng sử dụng một thiết bị đầu vào là video hoặc/và audio giống như camera hoặc/và microphone. Nếu người dùng cung cấp các quyền trên , một **Promise**  sẽ được trả về đối tượng MediaStream. Các đổi tượng Promise được sử dụng để tính toán bất đông bộ. Một Promise đại diện cho một giá trị mà có thể có sẵn ngay bây giờ, hoặc trong tương lai, hoặc không bao giờ.Nếu người dùng từ chối các quyền hoặc media không có sẵn thì promise sẽ trả về rejected PermissionDeniedError hoặc NotFoundError tương ứng.

Cú pháp được sử dụng là:

navigator.mediaDevices.getUserMedia(constraints).then(function(stream) {

/\* use the stream \*/

}).catch(function(err) {

/\* handle the error \*/

});

Với constraints là các thông số cũng như thiết bị đầu vào muốn truy cập:

Vd: lấy dữ liệu camera và micro với kích thước 1280x720 :

{

audio: true,

video: { width: 1280, height: 720 }

}

Trong khóa luận này để record video thì trước tiên ta phải có dữ liệu của video. Mà dữ liệu của video chính là giá trị trả về của hàm trên “stream”.

1. **MediaRecorder**

MediaRecorder là một interface của MediaStreamRecorder API cung cấp cho việc record media.

var mediaRecorder = new MediaRecorder(stream[, options]);

các tham số ở đây gồm có stream và options.

**Stream**: chính là dữ liệu mà ta lấy được sau khi gọi hàm getUserMedia. Do vậy việc khởi tạo đối tượng record này sẽ thực hiện chỉ khi việc lấy được stream ở phần trên.

Options: bao gồm:

* + - mimeType : kiểu dữ liệu mà bạn sẽ sử dụng để ghi dữ liệu. ví dụ như “video/webm” hoặc “video/mp4”
    - audioBitsPerSecond : tốc độ bit cho phần âm thanh
    - videoBitsPerSecond : tốc độ bit cho phần video
    - bitsPerSecond : tốc độ bit cho phần video và âm thanh. Có thể thay thế 2 tham số ở trên.

Nếu các options này không được thiết lập thì mặc định cho video alf 2.5Mbps, trong khi audio phụ thuộc số kênh âm thanh đầu vào.

1. NodeJS Server

**Nodejs** là một [hệ thống phần mềm](https://vi.wikipedia.org/w/index.php?title=H%E1%BB%87_th%E1%BB%91ng_ph%E1%BA%A7n_m%E1%BB%81m&action=edit&redlink=1) được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là [máy chủ web](https://vi.wikipedia.org/w/index.php?title=M%C3%A1y_ch%E1%BB%A7_web&action=edit&redlink=1). Chương trình được viết bằng [JavaScript](https://vi.wikipedia.org/wiki/JavaScript), sử dụng kỹ thật điều khển theo sự kiện, [nhập/xuất không đồng bộ](https://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%ADp/xu%E1%BA%A5t_kh%C3%B4ng_%C4%91%E1%BB%93ng_b%E1%BB%99&action=edit&redlink=1) để tối tiểu [tổng chi phí](https://vi.wikipedia.org/w/index.php?title=T%E1%BB%95ng_chi_ph%C3%AD&action=edit&redlink=1) và tối đại [khả năng mở rộng](https://vi.wikipedia.org/w/index.php?title=Kh%E1%BA%A3_n%C4%83ng_m%E1%BB%9F_r%E1%BB%99ng&action=edit&redlink=1). Node.js bao gồm có [V8 JavaScript engine](https://vi.wikipedia.org/w/index.php?title=V8_(JavaScript_engine)&action=edit&redlink=1) của Google, [libUV](http://github.com/joyent/libuv), và một vài thư viện khác.

Tại sao lại sử dụng Nodejs. Vì khóa luận này client này sử dụng javascript chính là ngôn ngữ của Nodejs và việc xây dựng một server nodejs khá đơn giản. Ta chỉ cần server trả về nội dung của trang web chứ không cần tương tác với cơ sở dữ liệu.

**HTTPS là** viết tắt của “Hypertext Transfer Protocol Secure”, Nó **là** một sự kết hợp giữa giao thức **HTTP**và giao thức bảo mật SSL hay TLS cho phép trao đổi thông tin một cách bảo mật trên Internet.

1. Google driver API
2. Các api sử dụng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stt | Tên api | Link | Phương thức | Tham số | Kết quả | Chú ý |
| 1 | Đăng nhập |  | Post | Email  Password | Success: 200  Erors: 400 |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

# Xây Dựng Ứng Dụng

1. Kiến trúc của ứng dụng

Ứng dụng gồm 4 phần chính:

Googler driver

Server

https

api

Trình duyệt

Người dùng

**Server:** Để truy cập và sử dụng ứng dụng ta sẽ gửi một request tới một tên miền đã đăng ký (hiện tại đang sử dụng là localhost cổng 3000). Sau khi nhận được request thì nó sẽ trả về mã html và các file javascript,css để ta có thể sử dụng chức năng chính của chương trình. Với chức năng record video thì sau khi load trang ta không cần thiết dùng đến mạng nữa. bởi vì chức năng này hoàn toàn có thể chạy trên web tĩnh. Vậy tại sao ta lại dùng server trong khi web tĩnh đã có thể chạy được ? Hãy chú ý ở đây, giao thức của chúng ta là https. Bởi vì là do cơ chế bảo mật của chrome nên một vài chức năng cũng nhưng extension mà ta sử dụng bắt buộc phải yêu cầu URL bắt đầu bằng https. Cụ thể trong ứng dụng này có sử dụng một extension của webrtc đó là Screen Capturing yêu cầu https. Hoặc cũng có một vài giải pháp thay thế nếu không muốn dùng https, ta sẽ nhắc tới nó ở các phần sau.

**Trình duyệt:** Trình duyệt hiện đại ngày nay đã có những chức năng tiên tiến vượt bậc so với các thời kỳ như IE cũ. Hiện nay nó có thể làm nhiều thứ tương tác với hệ thống của máy. Một số chức năng điển hình phải kể đến đó là: render đồ họa 3d, xem video nhạc mà không cần flash, truy cập webcam, xác định vị trí của bạn,kéo thả tập tin để upload…. Đặc biệt trong khóa luận này ta sẽ tận dụng 2 chức năng sau, nó là tiền đề để phát triển webrtc:

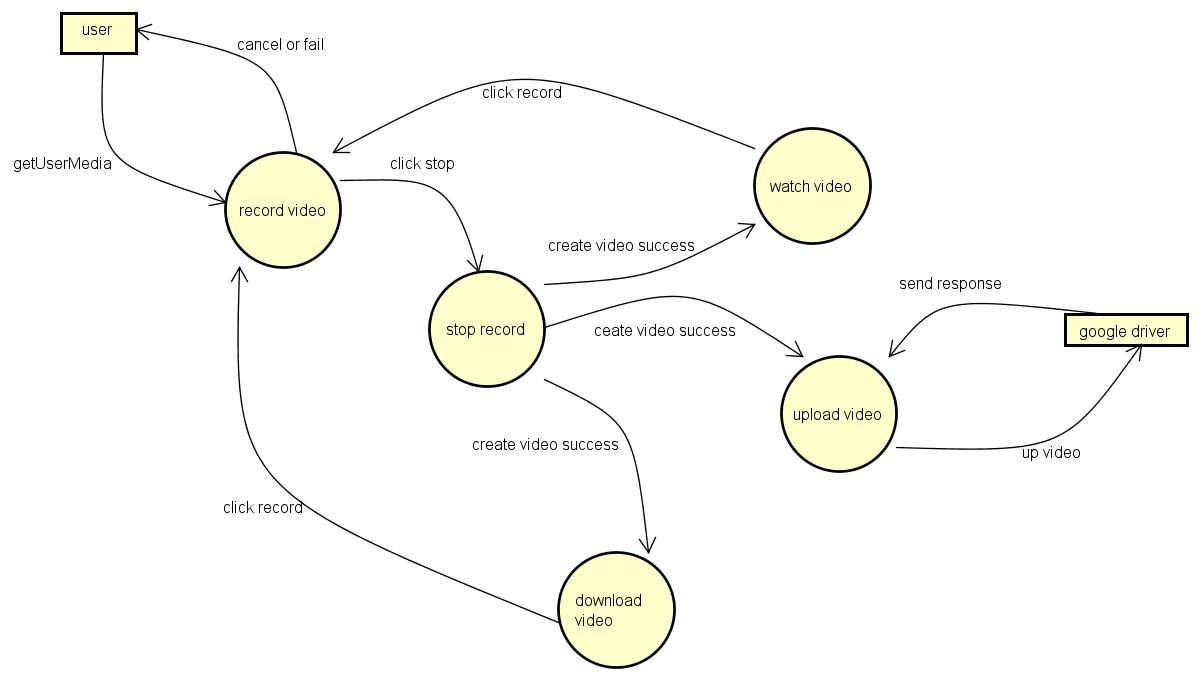
* Xem video nhạc mà không cần flash: Trước đây khi bạn vào một trang web bất kỳ hiển thị nội dung video là bắt buộc bạn phải cài flashplayer hay silverlight mới có thể xem được, nhưng với công nghệ tiên tiến hiện nay đặc biệt kể từ sự ra đời của html5 người dùng có thể xem video mà không cần quan tâm tới trình duyệt của mình đã cài plugin hỗ trợ như flashplayer chưa. Người dùng có thể xem video với định dạng: H.264 (MP4),WebM,Theora và công nghệ này đang dần dần thay thế flash và các plugin phụ trợ khác.
* Truy cập webcam: Từ bản beta của chrome 21, trình duyệt có khả năng truy cập vào các thiết bị đa phương tiện trên máy tính mà không cần plugin nào. Tính năng này được xây dựng nhờ vào hàm getUserMedia của Html5.  Điều này có nghĩa là chúng ta không cần phụ thuộc vào Adobe Flash hay Silverlight để có thể dùng được webcam, micro trong trình duyệt. Google cho biết đây là bước đi lớn đầu tiên của việc đem [WebRTC](https://tinhte.vn/tags/webrtc/) vào Chrome, một tiêu chuẩn về việc giao tiếp thời gian thực nhằm mang lại những trải nghiệm kết nối tốt thông qua hình ảnh và âm thanh chất lượng cao.

**Google driver:** Hiện tại để nghĩ xem có nơi nào có thể giúp lưu trữ miễn phí tài liệu cũng như các tập tin đa phương tiện đầu tiên ta nghĩ đến là google driver hay dropbox. Nhưng hầu hết ai ai cũng có một tài khoản google nên việc đăng nhập và quản lý tài liệu một cách dễ dàng hơn. Đặc biệt là với chức năng google docs của mình việc ứng dụng lưu trữ của goole đang thể hiện hỗ trợ hết mức cho người dùng liên quan đến học tập và giảng dạy. Google cũng hỗ trợ rất đầy đủ các api cho bên thứ 3 tích hợp vào ứng dụng của mình để truy cập và lưu trữ tới kho tài liệu của mình. Khóa luận này sẽ tích hợp api của google để lưu trữ cũng như quản lý các video mà người dùng ghi hình.

**Người dùng:** Là người trực tiếp thao tác và yêu cầu các lệnh đến trình duyệt. Các chức năng của người dùng sẽ được thể hiện ở phần sau.

1. Chức năng của ứng dụng
2. Màn hình Record

Record là chức năng chính của chương trình này. Để hiểu rõ hơn thì ta nhìn vào biểu đồ data flow sau:



Vẽ lại State transition và thuyết minh

Biểu đồ usecase cho trang record và các luồng sự kiện chính:



Figure 1Biểu đồ usecase cho trang record

User ở trang record sẽ có có các use case sau:

* Start record
* Stop record
* Play video
* Download video
* Đăng nhập tài khoản google driver
* Upload video

Chi tiết sẽ được mô tả ở bảng bên dưới:

Table 1: Đăng nhập google driver

|  |  |
| --- | --- |
| Tên Usecase | Đăng nhập |
| Tác nhân | Người dùng |
| Tiền điều kiện |  |
| Sự kiện kích hoạt | Click vào nút “ đăng nhập google driver” |
| Luồng sự kiện chính:  1. Người dùng bấm vào nút “ đăng nhập google driver”  2. Chấp nhận các quyền mà ứng dụng yêu cầu  3. Đăng nhập thành công thì về trang hiện tại. | |

Table 2: Record video

|  |  |
| --- | --- |
| Tên Usecase | Start Record |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | Đã cài plugin screen capturing của webrtc |
| Sự kiện kích hoạt | Click vào nút “record” |
| Luồng sự kiện chính:  Khi click vào nút “record” thì sẽ gọi đến hàm getUserMedia để lấy screenID của màn hình cần record  Chọn màn hình cần record và bắt đầu cấu hình các thông tin cần thiết để lấy stream.  Gọi hàm record nếu getStream thành công. | |

Table 3: Stop record

|  |  |
| --- | --- |
| Tên Usecase | Stop Record |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | Đã đang có sự kiện start record |
| Sự kiện kích hoạt | Click vào nút “stop record” |
| Luồng sự kiện chính:  Khi click vào nút “stop record” thì dừng việc get stream từ các thiết bị đa phương tiện lại. Gọi hàm stopRecord.  Gộp các blod đã record được trong 1 mảng thành 1 object video. | |

Table 4: Play Video

|  |  |
| --- | --- |
| Tên Usecase | PlayVideo |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | Sau khi sự kiện stop record kích hoạt |
| Sự kiện kích hoạt | Click vào nút “play video” |
| Luồng sự kiện chính:  Khi click vào nút “play video” sẽ lấy object video mà sau khi stoprecord tạo ra.  Tạo mới 1 object url từ object video và truyền vào video tag trong html  Gọi sự kiện play với video trên | |

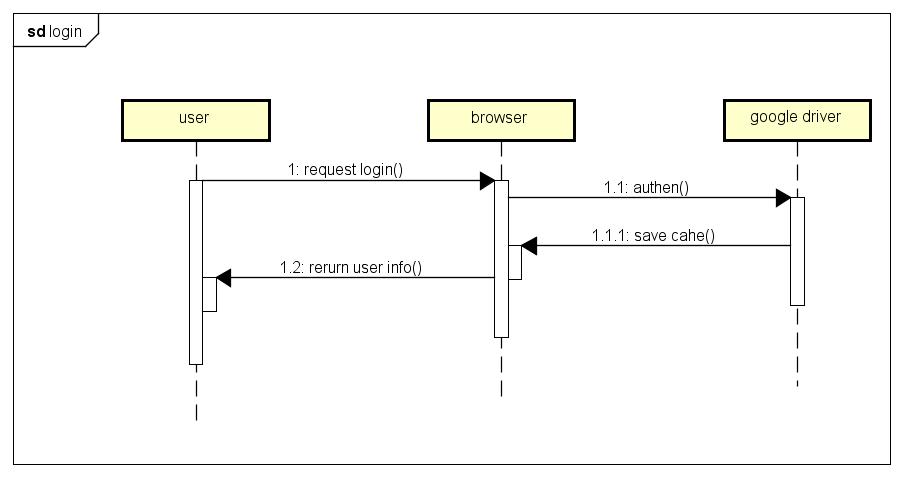
Table 5: Download video

|  |  |
| --- | --- |
| Tên Usecase | Downlaod video |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | Sau khi sự kiện stop record kích hoạt |
| Sự kiện kích hoạt | Click vào nút “downlaod video” |
| Luồng sự kiện chính:  Khi click vào nút “download video” sẽ lấy object video mà sau khi stoprecord tạo ra.  Tạo mới 1 object url từ object video và tạo 1 thẻ a với source là url video đó  Gọi sự kiện click tới thẻ a đó để download. | |

Table 2: Upload video

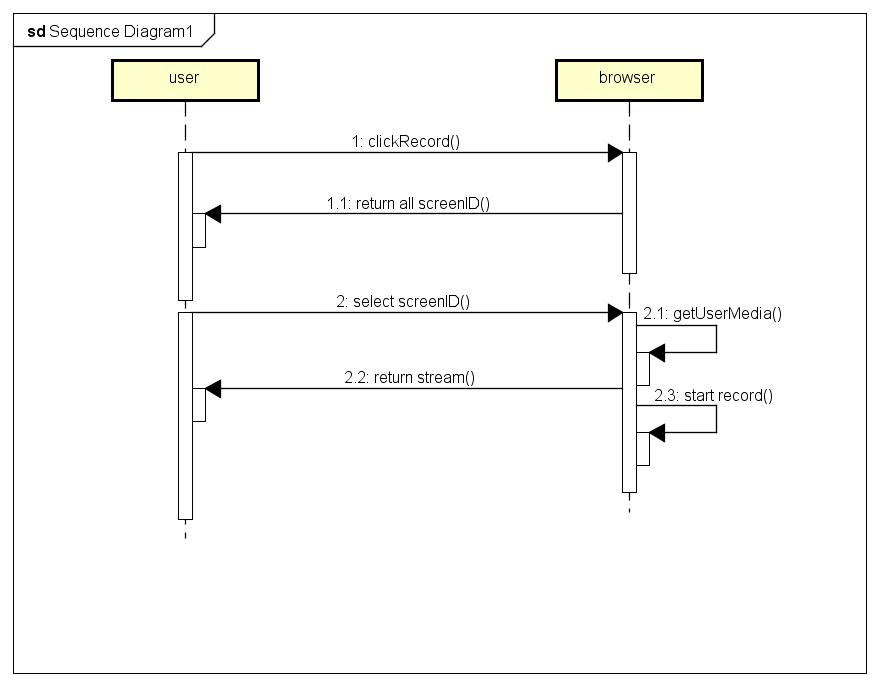
|  |  |
| --- | --- |
| Tên Usecase | Upload video |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | - Sau khi sự kiện stop record kích hoạt  - đã đăng nhâ[j google driver |
| Sự kiện kích hoạt | Click vào nút “upload video” |
| Luồng sự kiện chính:  Khi click vào nút “download video” sẽ lấy object video mà sau khi stoprecord tạo ra.  Dùng api google gửi object video đó lên driver lưu trữ  Sau khi thành công thì trả về các thông tin lưu trữ như link…. | |

Các biểu đồ tuần tự:



Hình 1: Biểu đồ tuần tự đăng nhập

Đầu tiên người dùng sẽ click vào nút login google driver sau đó trình duyệt sẽ gọi api để đăng nhập với tài khoản google. Người dùng nhập email/password (nếu chưa đăng nhập lần nào) và gửi lên api. Đăng nhập thành công khi người dùng chấp nhận các quyền mà ứng dụng muốn truy cập với tài khoản đó. Ở đây là các quyền: thông tin cá nhân, địa chỉ email, xem và quản lý các file trên google driver. Nếu lần đầu đăng nhập thành công trình duyệt sẽ tự động lưu tài khoản lại.



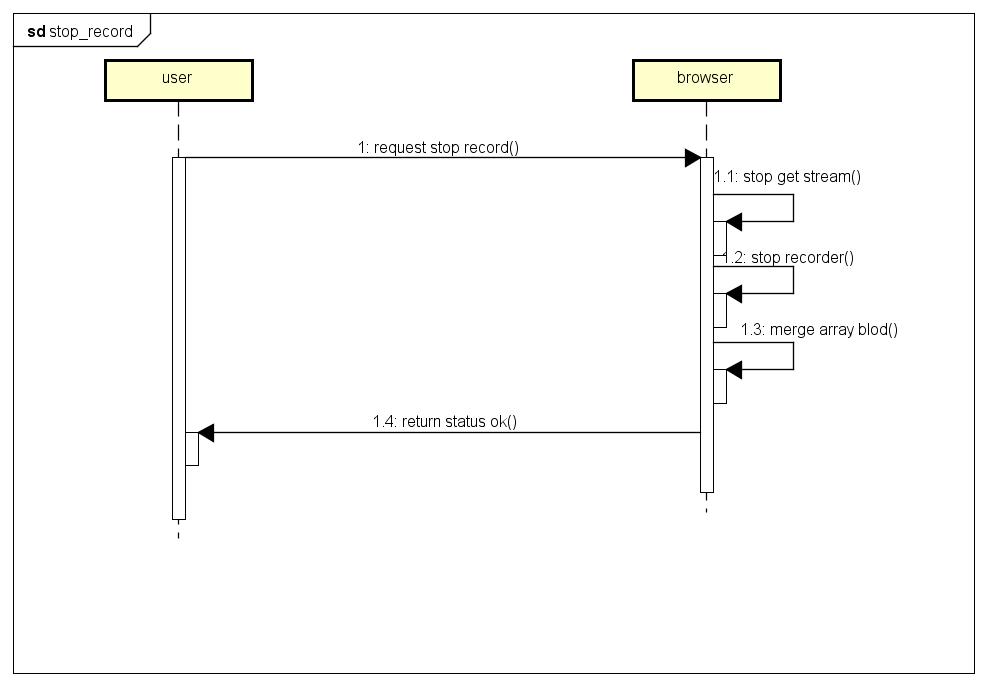
Hình 1: Biểu đồ tuần tự record

Khi người dùng click và nút “start record” thì sẽ gọi tới hàm getScreenId để lấy id các màn hình hiện tại có. Sau đó người dùng click vào một màn hình tương ứng với id muốn record thì trong javascript sẽ gọi tới hàm getUserMedia của html5 để lấy dữ liệu (là các stream) về màn hình đó.

Sau khi có stream rồi thì sẽ dùng đối tượng MediaRecorder (là một đối tượng trong html5 chuyên để record các stream) để record video. Mediarecorder này sẽ record stream thành các blod, mỗi blod sẽ ghi trong 5s. Vậy ta sẽ có một mảng các blod là các frame của video.

Với mỗi lần record thì có 2 kiểu record:

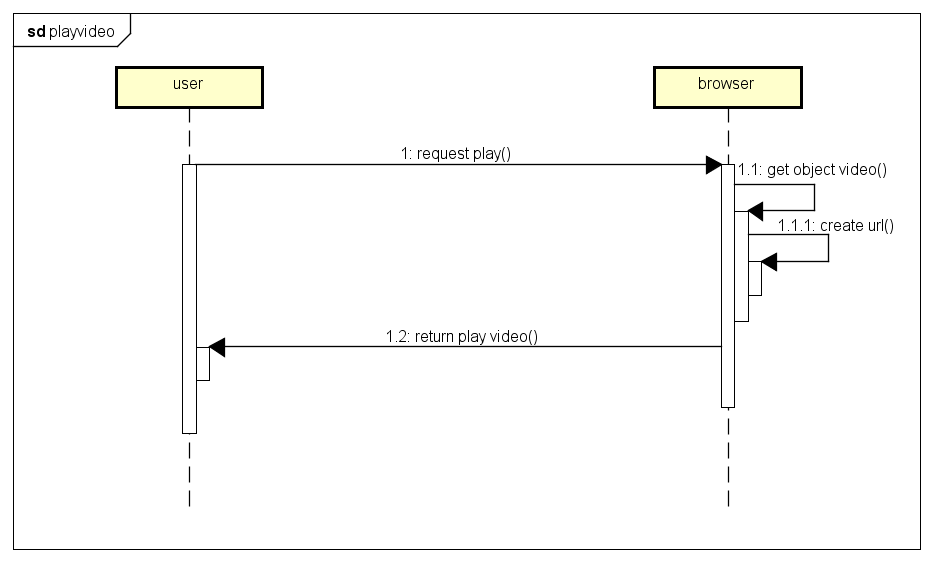
1. Record screen: là kiểu record duy nhất màn hình hoặc các chương trình đang chạy trên desktop. Có các option kèm theo là có audio hoặc không có audio (audio do mic máy tính thu được).
2. Record screen + camera: là kiểu ngoài record như màn hình ở trên còn kèm theo một video nhỏ được hiển thị ở góc phải màn hình do thu được từ camera. Và kiểu ghi này luôn có audio. Do vậy đầu ra của việc record sẽ có 2 video cung độ dài (Một là của màn hình và một là của camera).



Hình ‑: Biểu đồ tuần tự stop record

Nếu trạng thái đang là record thì nút “ stop record” sẽ được hiện lên và có thể click vào. Khi click vào thì trình duyệt sẽ dừng việc get Stream từ các thiết bị như camera hay màn hình, đồng thời gọi hàm stopRecord của đối tượng MediaRecorder. Sau đó sẽ kiểm tra nếu mảng các blod sau khi record khác rỗng thì tiến hành gộp lại tạo thành 1 object video. Và báo cho người dùng có thể sẵn sang tương tác với video này.

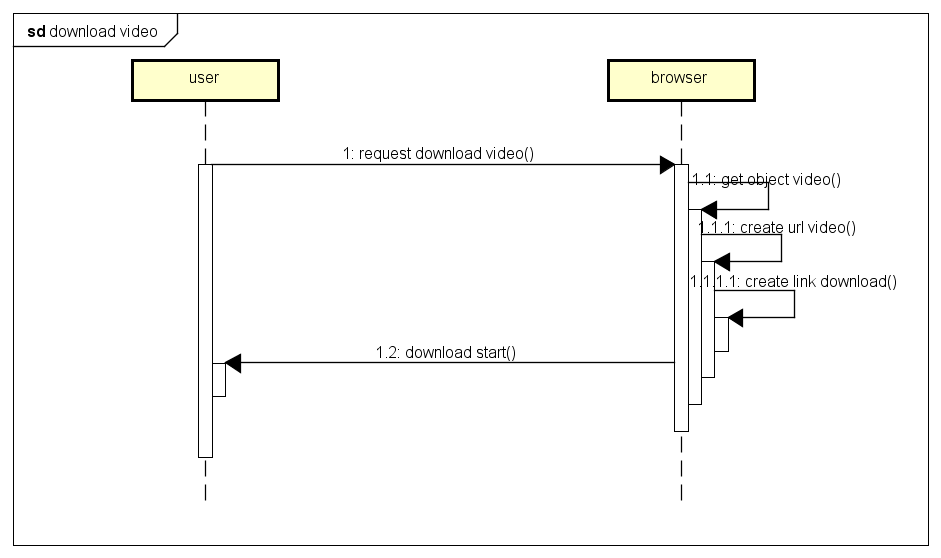
Đối với kiểu ghi có camera+screen thì sẽ có 2 object video được tạo tương ứng với từng mảng blod.



Hình ‑: Biểu đồ tuần tự play video

Nút “play video” chỉ có thể hoạt động được khi record video đã được stop. Và object video đã được tạo ra trước đó. Khi bấm vào nút này thì sẽ lấy object video đó ra và tạo thành một url video mới. Truyền url video này vào thuộc tính src của tag video. Sau đó gọi sự kiện video đó play. Người dùng sẽ xem được video vừa record chạy trên màn hình.

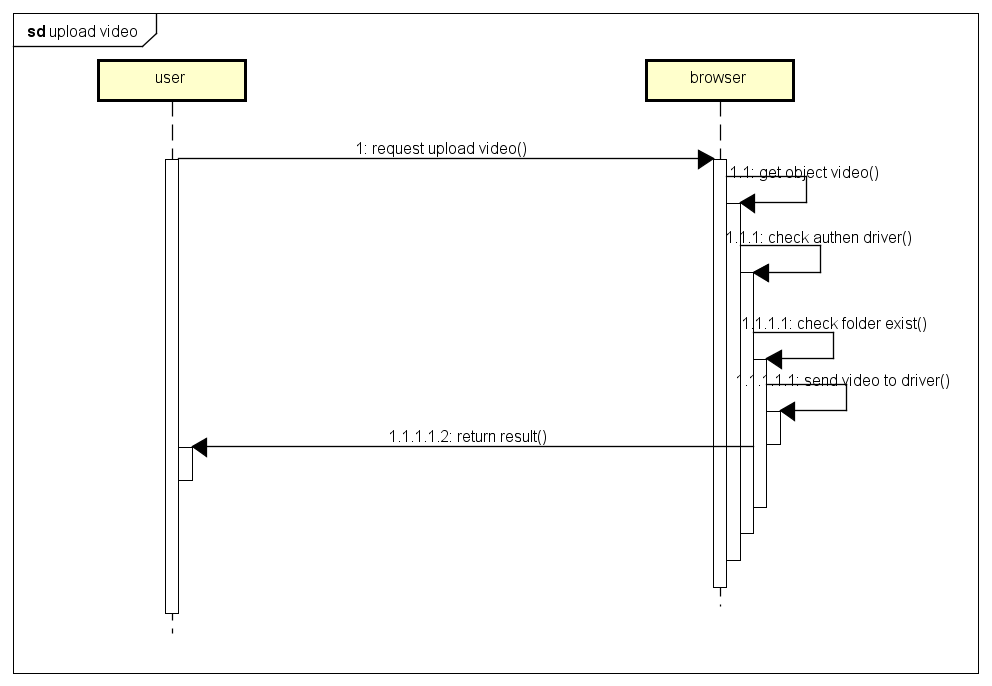
Đối với từng kiểu ghi thì sẽ có từng kiểu play khác nhau. Sẽ có 2 khung hình video được tạo nếu là kiểu ghi screen + camera. Truyền 2 object url video tương ứng vào. Và khi play sẽ play cùng lúc, pause hay seek cũng cùng lúc.



Hình ‑: Biểu đồ tuần tự download video

Giống như nút playvideo nút “download video” chỉ có thể kích hoạt khi sự kiện record video được stop. Sau đó nó cũng lấy object video đã được tạo ra trước đó để tạo ra một url video mới. Để download được thì url này phải gán vào một thẻ link trong htm nên ta sẽ tạo ra 1 thẻ a ẩn và gọi sự kiện click thẻ a đó trình duyệt sẽ tự động download video.

Việc download video không phụ thuộc vào kiểu ghi nào, nó luôn luôn lấy kiểu ghi chính là record screen.



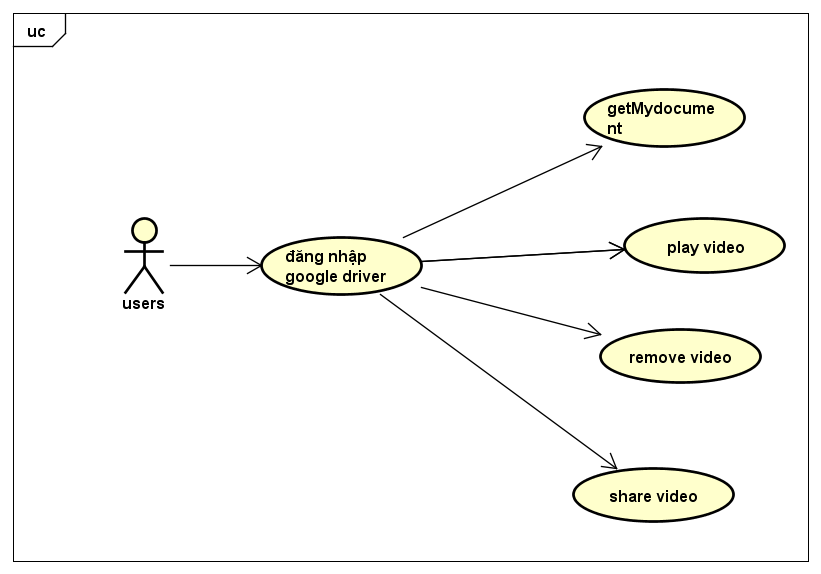
Hình 1: Biểu đồ tuần tự upload video

Giống download video, trước tiên muốn thực hiện một hành động nào đó thì ta phải có video đã. Sau khi việc record được dừng thì object video được tạo ra, ta chỉ việc từ object này tạo url video xác định. Nhưng muốn upload, cụ thể ở đây là đẩy lên google driver để lưu trữ trước tiên ta check người dùng đã đăng nhập chưa, Nếu người dùng đã đăng nhập thì kiểm tra xem folder chứa video đã có sẵn chưa. Nếu chưa có thì tạo folder mới chuyên dùng để lưu trữ các video mà ứng dụng đẩy lên, Sau đó thì đẩy video lên bằng api. Tên file của video sẽ được đặt theo thời gian tạo video. Trả về kết quả của việc upload video cho người dùng. Người dùng có thể kiểm tra các video của mình đã upload lên trong phần MyVideo.

Nếu là kiểu ghi record screen + camera thì ở trong folder gốc lưu trữ sẽ có một folder con tên là camera dùn để lưu trữ dạng 2 video. Hai video này sẽ cung tên nhưng khác folder lưu trữ.

1. Màn hình MyDocument

Sau khi record video và upload lên driver thành công thì video sẽ được nằm trên folder mặc định của ứng dụng tạo ra. Nhưng khi người dùng muốn xem tất cả các video mình đã đẩy lên thì trang MyDocument sẽ có chức năng đó. Trước tiên nếu bạn chưa đăng nhập thì trang này sẽ hiện ra của sổ yêu cầu bạn đăng nhập với tài khoản google driver với các quyền như ở phần trước đã nêu. Nhưng đặc biệt trang này sẽ dùng quyền “đọc và quản lý file trên google driver” của người dùng đăng nhập. Nếu đăng nhập thành công thì giao diện sẽ trả về danh sách các video. Mỗi video kèm theo ba chức năng chính sau : xem video, xóa video, chia sẻ video.



Hình 1: Biểu đồ usecase cho màn hình mydocument

User trước tiên phải đăng nhập tài khoản google driver (nếu trước đó chưa đăng nhập). Sau đó sẽ lấy dữ liệu về các video đã lưu trên google driver về (nằm trong một folder mặc định). Với mỗi video ta có thể có các thao tác tương ứng với các use case : play video, remove video, share video.

Sau đây ta sẽ đi chi tiết vào từng use case:

Bảng 1: Đăng nhập google driver

|  |  |
| --- | --- |
| Tên Usecase | getVideo |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | - Sau khi đăng nhập vào google driver |
| Sự kiện kích hoạt | Khi load trang MyDocument (Với trường hợp đã đăng nhập)hoặc sau khi đăng nhập. |
| Luồng sự kiện chính:  Sau khi đăng nhập hoặc trang được tải. Sẽ gọi một api để lấy id của folder mặc định mà ta quy ước để lưu các video do người dùng đẩy lên. Nếu folder này tồn tại thì gọi tiếp api để lấy các dữ liệu nằm trong foler với id này trả về cho trình duyệt. Nếu folder không tồn tại thì thông báo bạn không có video nào. | |

Bảng 2 : Usecase Playvideo

|  |  |
| --- | --- |
| Tên Usecase | Play Video |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | - Sau khi laod được dữ liệu từ folder chứa video và hiển thị |
| Sự kiện kích hoạt | Click vào biểu tượng nút “play” |
| Luồng sự kiện chính:  Khi click vào nút “play” thì sẽ lấy link của video và truyền vào thuộc tính src của video và trình chiếu cho người dùng | |

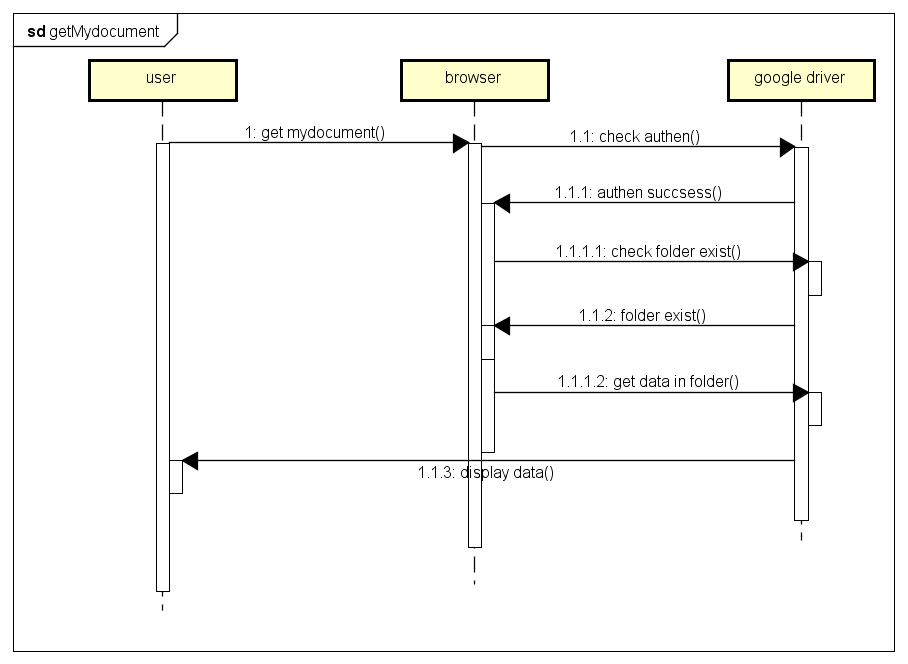
Bảng 3: Usecase remove video

|  |  |
| --- | --- |
| Tên Usecase | Remove video |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | - Sau khi laod được dữ liệu từ folder chứa video và hiển thị |
| Sự kiện kích hoạt | Click vào biểu tượng nút “remove” |
| Luồng sự kiện chính:  Khi click vào nút “remove” thì sẽ lấy id của video và gọi tới api remove do google driver cung cấp, truyền tham số là id của video để xóa. Nếu video được tạo thành 2 file (gồm 1 file màn hình, 1 file camera) thì sẽ xóa cả hai. Trả lại trạng thái sau khi xóa và reload lại trang. | |

Bảng 4: Usecase Share Video

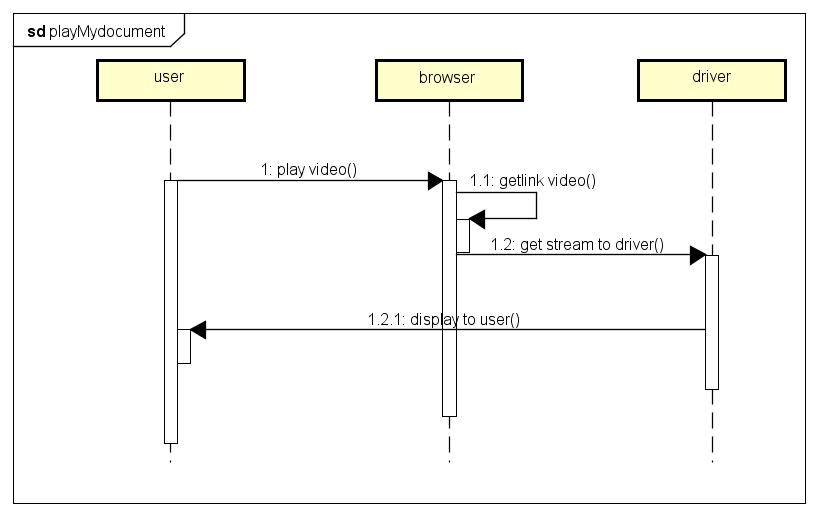
|  |  |
| --- | --- |
| Tên Usecase | Share Video |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | - Sau khi laod được dữ liệu từ folder chứa video và hiển thị |
| Sự kiện kích hoạt | Click vào biểu tượng nút “share” |
| Luồng sự kiện chính:  Khi click vào nút “share” thì sẽ lấy id của video và sau đó gọi api của google driver cung cấp cho việc share video. Truyền tham số id vào video đó. Sau khi gọi api thành công thì trả về link của video chia sẻ. Người dùng có thể coppy để dán vào bất cứ đâu muốn. | |

Biểu đồ tuần tự cho các usecase:



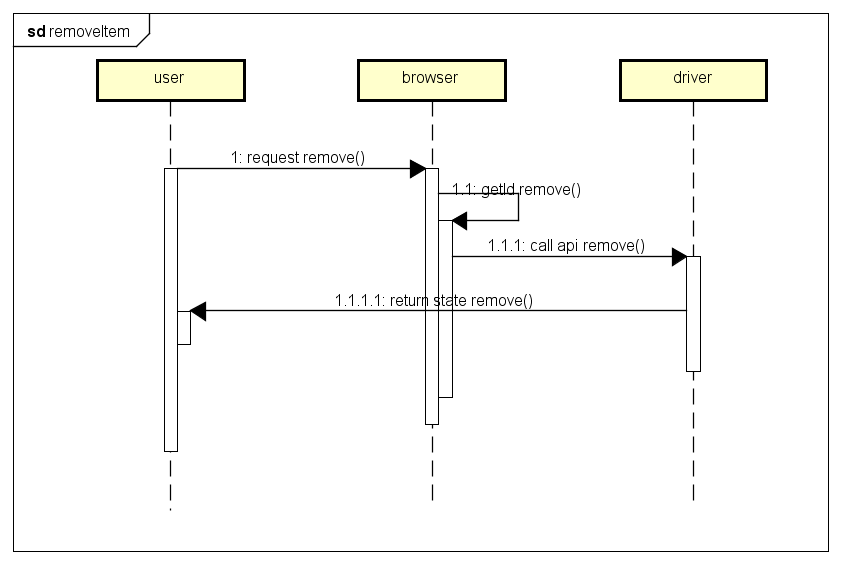
Hình 1: Biểu đồ tuần tự getdocument

Khi người dùng chuyển sang trang myDocument thì việc đầu tiên là kiểm tra xem người dùng này đã đăng nhập chưa. Nếu chưa đăng nhập thì yêu cầu đăng nhập tài khoản google driver, nếu đăng nhập thành công hoặc đã đăng nhập rồi thì sẽ kiểm tra xem trên googler driver có tồn tại thư mục mặc định mà ứng dụng tạo ra để lưu trữ các video mà người dùng đẩy lên hay không. Nếu tồn tại thì lấy tất cả các video nằm trong đó trả về hiện thị cho người dùng. Nếu không có dữ liệu hoặc folder không tồn tại thì thông báo cho người dùng không tồn tại.



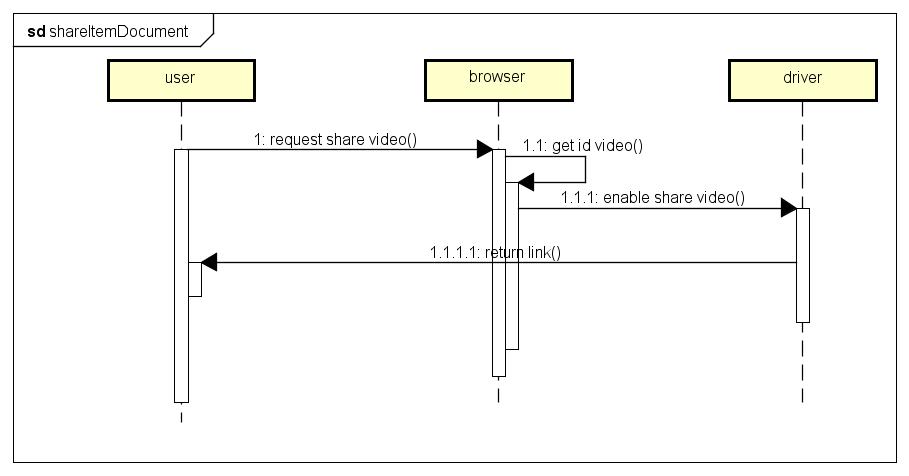
Hình 1: Biểu đồ tuần tự play video trên mydocument

Sau khi lấy được dữ liệu video về thì người dùng có thể xem trực tiếp video đó bằng click vào biểu tượng nút “play” trên mỗi item. Khi đó đoạn mã javascript sẽ lấy link của video tương ứng gán vào src và trình chiếu cho người dùng xem. Đối với video có 2 khung hình (1 khung hình quay màn hình, 1 khung hình quay camera ) nó sẽ hiển thị cả 2. Và khung hình nhỏ hơn sẽ nằm ở góc bên phải.



Hình 1: Biểu đồ tuần tự remove item trên mydocument

Khi người dùng click vào biểu tượng “remove” thì sẽ lấy id của video hiện tại sau đó gọi api dùng để xóa file với id xác định do google driver cung cấp. Nếu video này được tạo từ 2 video con thì sẽ getID của cả 2 video và cũng xóa cả 2 video. Sau khi xóa xong thì cập nhật lại giao diện của trang myDocument.



Hình 1: Biểu đồ tuần tự share video trên mydocument

Khi click vào biểu tượng nút “share” thì sẽ lấy id của video hiện tại và gọi api chia sẻ file do google driver cung cấp. api này sẽ bật chế độ chia sẻ của google driver với một file xác định, sau đó sẽ trả về một link để người dùng có thể sao chép và dán bất kỳ nơi nào muốn. Đối với video có nhiều file thì file chính sẽ được chia sẻ (thường là file ghi màn hình).

1. Giao diện của ứng dụng

# Demo Và Thực Nghiệm

* Các thiết đặt thí nghiệm

Hình ảnh , đo cpu…..mô tả các thí nghiệm đã làm, cấu hình máy…

# Chương 5: Tổng Kết