**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)*

## **HÀ NỘI - 2017**

## 

## 

## 

## 

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

*(chữ hoa, 12pt, đậm, căn giữa)*

## **HÀ NỘI - 2017**

## 

## 

## 

## 

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

*(chữ hoa, 12pt, đậm, căn giữa)*

## **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)*

**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)*

**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)*

**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!

**LỜI CAM ĐOAN**

Tôi Lò Văn Kiên xin cam đoan nội dung trong khóa luận này là công trình nghiên cứu do chính tôi thực hiện dưới sự hướng dẫn của TS. Hoàng Xuân Tùng. Các số liệu sử dụng phân tích trong khóa luận có nguồn gốc rõ ràng, đã công bố theo đúng quy định. Các kết quả nghiên cứu trong khóa luận do tôi tự tìm hiểu, phân tích một cách trung thực, khách quan và phù hợp với thực tiễn. Nếu hình ảnh được lấy từ nguồn bên ngoài, tôi đều có trích dẫn nguồn rõ ràng và đầy đủ.

Hà Nội, ngày 28 tháng 04 năm 2017

Sinh viên

Lò Văn Kiên

[Chương 1: Mở Đầu 1](#_Toc481150594)

[1.1 Đặt vấn đề 1](#_Toc481150595)

[1.2 Phạm vi và mục tiêu của khóa luận 1](#_Toc481150596)

[Chương 2: Mô hình học trực tuyến và phương pháp giải quyết vấn đề. 2](#_Toc481150597)

[2.1. Khảo sát các mô hình trực tuyến và công nghệ sử dụng 2](#_Toc481150598)

[2.2. WebRTC và các thư viện sử dụng 8](#_Toc481150599)

[2.1.1 Webrtc là gì 8](#_Toc481150600)

[2.1.2 Lịch sử phát triển 8](#_Toc481150601)

[2.1.3 Lợi ích của WebRTC 9](#_Toc481150602)

[2.1.4 Các ứng dụng của WebRTC 10](#_Toc481150603)

[2.1.5 Các thư viện sử dụng trong khóa luận 10](#_Toc481150604)

[2.3. Web APIs (MDN) 12](#_Toc481150605)

[2.4. NodeJS Server 14](#_Toc481150606)

[2.5. Google driver API 14](#_Toc481150607)

[Chương 3: Xây Dựng Ứng Dụng 16](#_Toc481150608)

[3.1. Kiến trúc của ứng dụng 16](#_Toc481150609)

[3.2. Chức năng của ứng dụng 18](#_Toc481150610)

[3.3. Giao diện của ứng dụng 38](#_Toc481150611)

[Chương 4: Demo Và Thực Nghiệm 40](#_Toc481150612)

[Chương 5: Chương 5: Tổng Kết 51](#_Toc481150613)

[5.1 Các kết quả đã đạt được 51](#_Toc481150614)

[5.2 Những hướng phát triển 51](#_Toc481150615)

[DANH MỤC TÀI LIỆU THAM KHẢO 53](#_Toc481150616)

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Cụm từ tiếng anh |
| 1 | API | Application Programming Interface |
| 2 | MDN | Mozilla developer network |
| 3 | NPM | Node Package Manager |
| 4 | HTML5 | HyperText Markup Language 5 |
| 5 | HTTPS | Hypertext Transfer Protocol Secure |
| 6 | WebRTC | Web with Real-Time Communications |

Danh mục các bảng

[Bảng 2.5‑1 Các Api Google driver sử dụng 15](#_Toc481146882)

[Bảng 3‑1 Đăng nhập google driver 21](#_Toc481146883)

[Bảng 3‑2 Record video 21](#_Toc481146884)

[Bảng 3‑3 Stop record 22](#_Toc481146885)

[Bảng 3‑4 Play Video 22](#_Toc481146886)

[Bảng 3‑5 Download video 23](#_Toc481146887)

[Bảng 3‑6 Upload video 24](#_Toc481146888)

[Bảng 3‑7 Đăng nhập google driver 32](#_Toc481146889)

[Bảng 3‑8 Usecase Playvideo 33](#_Toc481146890)

[Bảng 3‑9 Usecase remove video 33](#_Toc481146891)

[Bảng 3‑10 Usecase Share Video 34](#_Toc481146892)

[Bảng 4‑1 Đầu vào thực nghiệm 43](#_Toc481146893)

[Bảng 4‑2 Kết quả của thực nghiệm 44](#_Toc481146894)

DANH MỤC CÁC HÌNH

[Hình 2.1‑1 Mô hình tổng quan tương tác giữa các đối tượng 4](#_Toc481146984)

[Hình 2.1‑2 Mô hình dữ liệu và hạ tầng của các hệ thống học trực tuyến 5](#_Toc481146985)

[Hình 2.1‑3 Top 10 capture tools 6](#_Toc481146986)

[Hình 2.1‑4 Top 10 editing tools 7](#_Toc481146987)

[Hình 2.1‑5 Top 10 output record tools 7](#_Toc481146988)

[Hình 3‑1 Biểu đồ data flow 19](#_Toc481146989)

[Hình 3‑2 Biểu đồ usecase cho trang record 20](#_Toc481146990)

[Hình 3‑3 Biểu đồ tuần tự đăng nhập 25](#_Toc481146991)

[Hình 3‑4 Biểu đồ tuần tự record 26](#_Toc481146992)

[Hình 3‑5: Biểu đồ tuần tự stop record 27](#_Toc481146993)

[Hình 3‑6: Biểu đồ tuần tự play video 28](#_Toc481146994)

[Hình 3‑7: Biểu đồ tuần tự download video 29](#_Toc481146995)

[Hình 3‑8 Biểu đồ tuần tự upload video 30](#_Toc481146996)

[Hình 3‑9 Biểu đồ use trang mydocument 31](#_Toc481146997)

[Hình 3‑10 Biểu đồ tuần tự getdocument 35](#_Toc481146998)

[Hình 3‑11 Biểu đồ tuần tự play video trên mydocument 36](#_Toc481146999)

[Hình 3‑12 Biểu đồ tuần tự remove item trên mydocument 37](#_Toc481147000)

[Hình 3‑13 Biểu đồ tuần tự share video trên mydocument 38](#_Toc481147001)

[Hình 4‑1 Cấu hình máy test 39](#_Toc481147002)

[Hình 4‑2 Tốc độ mạng đo bằng speedtest 39](#_Toc481147003)

[Hình 4‑3 Đăng nhập thành công 40](#_Toc481147004)

[Hình 4‑4 CPU và RAM trước khi thực nghiệm 1 41](#_Toc481147005)

[Hình 4‑5 CPU và RAM trong lúc thực nghiệm 1 42](#_Toc481147006)

[Hình 4‑6 Giao diện quá trình record thực nghiệm 2 43](#_Toc481147007)

[Hình 4‑7 Các file đã được upload thành công 46](#_Toc481147008)

[Hình 4‑8 Giao diện Chrome App 47](#_Toc481147009)

# 

# 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.

# Mô hình học trực tuyến và phương pháp giải quyết vấn đề.

# Khảo sát các mô hình trực tuyến và công nghệ sử dụng

Học trực tuyến đó là một xu hướng vì sự tiện dụng của nó, giảm chi phí cơ sở hạ tầng, người học có thể thụ động, lưu trữ kiến thức và xem lại một cách dễ dàng. Cho nên từ các trung tâm đào tạo đến các trường đại học danh tiếng thế giới đều tham gia mảng này.Nó vừa mang lợi ích cho cộng đồng vừa tạo ra chi phí cho các trường đại học, đồng thời cũng tạo ra người lao động chất lượng hơn.

Hiện nay các mô hình học trực tuyến phổ biến trong và ngoài nước có thể kể đến: Coursera, Edx, Kyna, Edumail, Topica… Cùng tìm hiểu qua các mô hình này để thấy nó có sự tác động đến đề tài này như thế nào.

**Coursera** [1]**:**

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 cùng 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.

**Edx** [2]**:**

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)

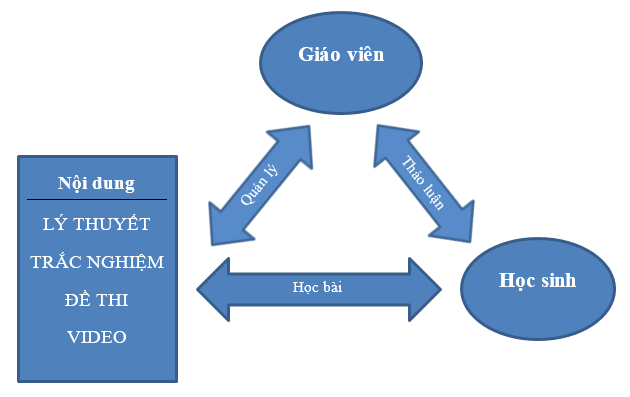
Trang chủ: <https://www.edx.org/>

**Udemy** [3]**:**

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

Trang chủ: <https://www.udemy.com/>

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:



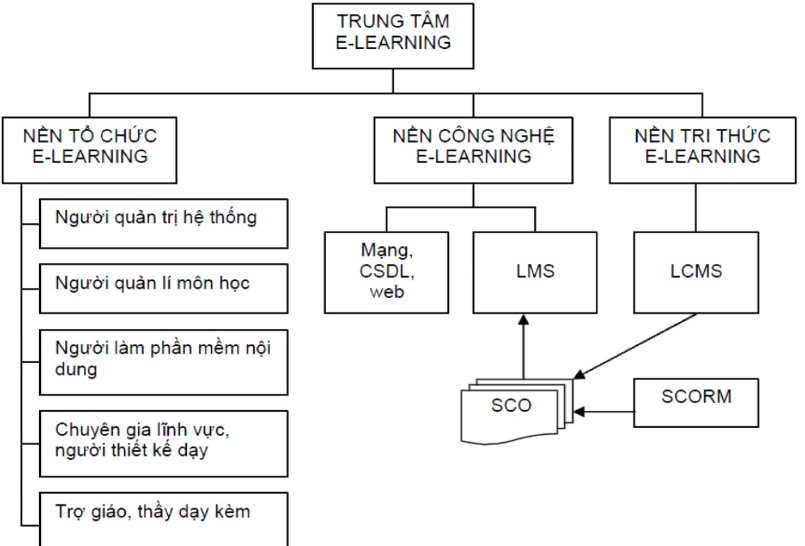
Hình 2.1‑1 Mô hình tổng quan tương tác giữa các đối tượng

**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.



Hình 2.1‑2 Các thành phần của các hệ thống học trực tuyến

Dù công nghệ được sử dụng cho các mô hình học trực tuyến kia có thể khác nhau (php, nodejs,python ….) nhưng tất cả đều có một điểm chung đó là đa số các bài giảng đều là video. Vậy video có tác động đến các mô hình học trực tuyến như thế nào?

Theo thống kê của trang web elearningindustry [4] nói rằng:

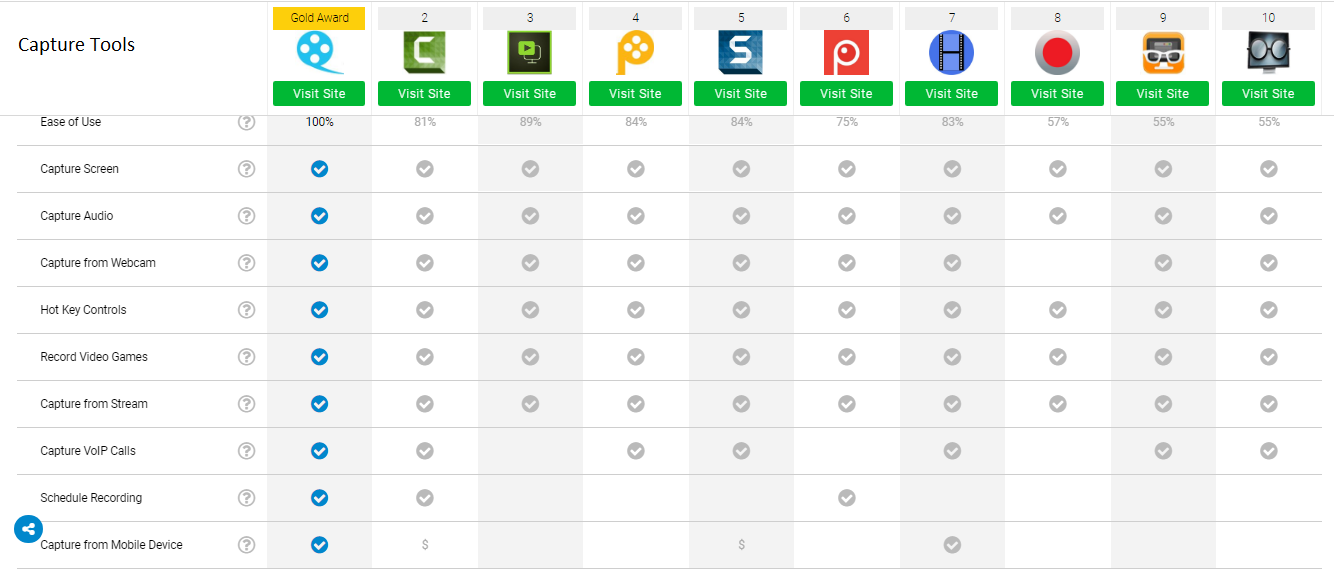
* Trang học trực tuyến Edudemic báo cáo rằng 67% giảng viên tin tưởng vào hiệu quả trong việc sử dụng video trong giáo dục. 46% giảng viên nói rằng họ ít nhất đã từng tạo một video bài giảng.
* Một nghiên cứu được tiến hành bởi trang skilljar.com năm 2013 chỉ ra rằng 67% số người học trực tuyến tham gia các lớp học bằng video. Hiện tại là năm 2017 chưa có con số cụ thể nhưng chắc chắn đã tăng lên rất nhiều.
* Các trang web học trực tuyến có tốc độ phát triển nhanh nhất đều chủ yếu dựa trên video. Phải kể đến như : Lynda.com, Skillshare, CreativeLive, Udacity, Udemy, Craftsy….

Việc dạy qua video quan trọng như vậy thì các mô hình học trực tuyến đã tạo ra video như thế nào. Cùng tìm hiểu một số cách thức có thể được sử dụng trong các mô hình trên.

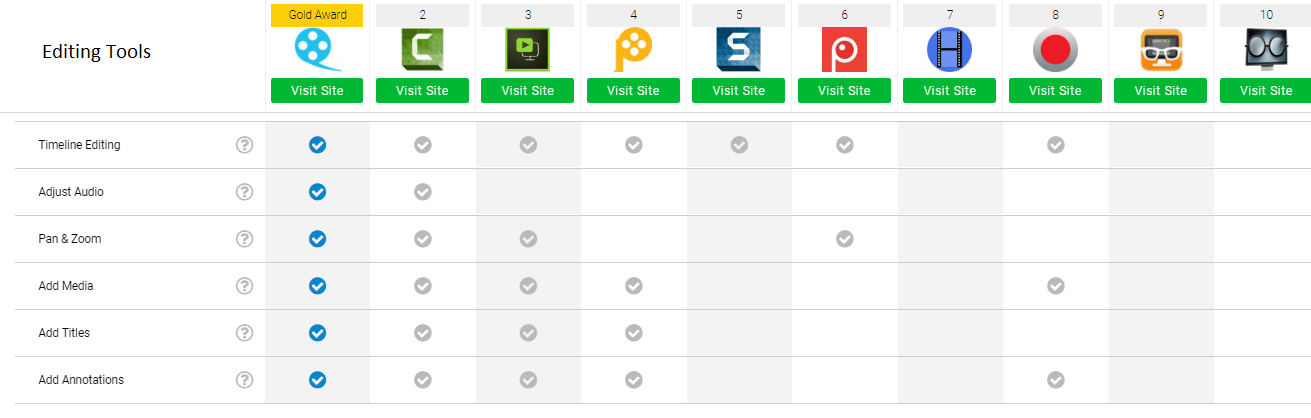
Đơn giản nhất có thể kể đến là các công cụ thuyết trình bài giảng, nó có hỗ trợ trực tiếp ta tạo video. Điển hình là Powerpoint và Keynote nó sẽ giúp ghi lại quá trình thuyết trình bao gồm cả bài giảng và âm thanh bằng cách nhấn vào Record Slide Show Button.

Nhưng công cụ được sử dụng nhiều nhất từ những người không chuyên đến chuyên nghiệp là các phần mềm chuyên sử dụng cho việc record. Ưu điểm của nó so với cách bên trên là có thể kết hợp giữa ghi hình màn hình và camera, ngoài ra người dùng còn có thể biên tập lại video. Nhưng nhược điểm lớn nhất của nó là nặng (từ vài chục đến vài trăm Mb) và cài đặt phức tạp, khó sử dụng.

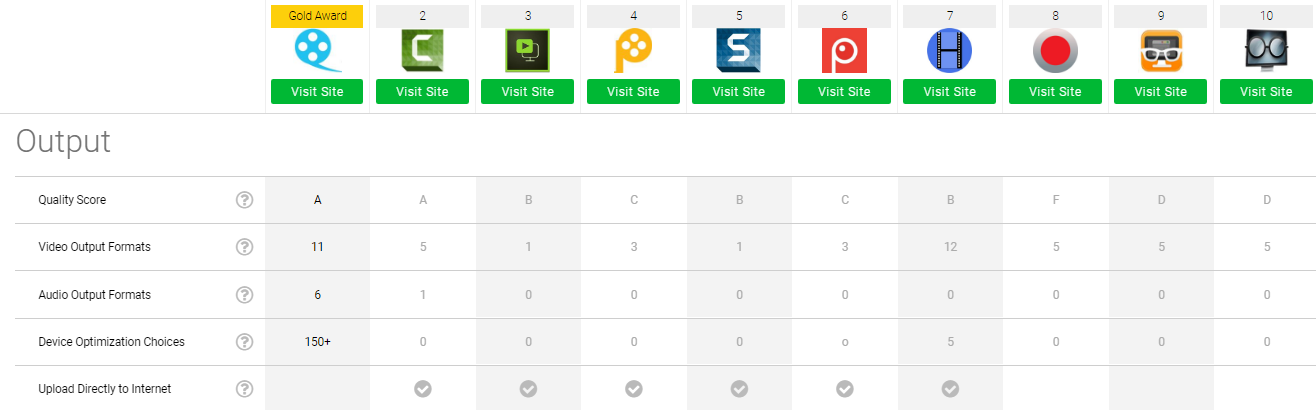
Dưới đây là một số hình ảnh so sánh các chức năng của danh sách 10 ứng dụng ghi hình tốt nhất được trang toptenreviews [5] thống kê vào tháng 05/8/2016.



Hình 2.1‑3 Top 10 capture tools



Hình 2.1‑4 Top 10 editing tools



Hình 2.1‑5 Top 10 output record tools

Ta có thể thấy các ứng dụng trên đều phải mất phí để kích hoạt (sẽ có bản dùng thử miễn phí 30 ngày). Mặc dù hiện nay đã có rất nhiều ứng dụng miễn phí hỗ trợ tương đương các ứng dụng trên như OBS (Open Broadcaster Software)… nhưng các nhược điểm của nó vẫn không được cải thiện. Ngoài ra khi người dùng upload video lên các trang web học trực tuyến với định dạng video này, người dùng bắt buộc phải cài plugin flash player hoặc quicksilver mới có thể xem được.

Nhận ra điểm yếu trên 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. Đồng thời khi upload lên các trang web người dùng không phải cài đặt thêm plugin phụ trợ nào vì nó sẽ áp dụng các chuẩn video mới.

# WebRTC và các thư viện sử dụng

### Webrtc là gì

**WebRTC** [6](**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, vân vân 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 : wello [7] 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*** [8]

Là một plugin miễn phí được viết bởi MuazKhan [9] (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*** [10]

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.

# Web APIs (MDN) [11]

 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.

1. **MediaDevice.getuserMedia** [12]

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 đối tượng Promise được sử dụng để tính toán bất đồng 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** [13]

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 là 2.5Mbps, trong khi audio phụ thuộc số kênh âm thanh đầu vào.

# NodeJS Server [14]

**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 thuật điều khiển 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 thiểu [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 gì thêm 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.

Do trong ứng dụng ta sử dụng mộtob số plugin của webrtc và api của trình duyệt để truy cập vào thiết bị phần cứng là camera và microphone nên trình duyệt yêu cầu trang web phải kết nối bằng giao thức bảo mật https.

# Google driver API [15]

Với mục tiêu là xây dựng ứng dụng đơn giản, không phụ thuộc quá nhiều vào server, đặc biệt là database (mỗi hệ thống đều có một database khác nhau như mysql hay mogodb). Do vậy việc có một nơi lưu trữ, quản lý và chia sẻ tài liệu mà miễn phí, triển khai một cách dễ dàng ta có thể nghĩ ngay tới dịch vụ Google Driver. Hiện tại google có cup cấp một số gói unlimited cho các tài khoản sinh viên. Nên trong phạm vi khóa luận ta có thể sử dụng chúng để đơn giản hóa việc lưu trữ. Sau đây là một số API mà ta sẽ sử dụng:

Link = Domain + path.

Domain = https://www.googleapis.com/

Bảng 2.5‑1 Các Api Google driver sử dụng

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stt | Tên api | Đường dẫn (path) | Phương thức | Tham số | Kết quả | Chú ý |
| 1 | Đăng nhập | auth/drive | Post | Email  Password | Success: 200  Trả về các quyền và profile  Erors: 400 |  |
| 2 | Upload file | Upload/driver/v2/files | Post | uploadType : upload một lần hay upload từng phần | Success: 200  Trả về link file  Erors: 400 |  |
| 3 | Get file | drive/v2/files | Get | folderId: id của folder muốn lấy danh sách file | Success: 200  Trả về danh sách các file  Erors: 400 |  |
| 4 | Create file | drive/v2/files | Post | Title: tiêu đề của file/folder  mimeType: kiểu file hay folder  parents: thư mục cha | Success: 200  Trả về link file  Erors: 400 |  |
| 5 | Delete file | drive/v2/files | Delete | Fileid: id của file muốn xóa | Success: 200  Erors: 400 |  |
| 6 |  | drive/v2/files/{fileid}/permissions | Post | Fileid: id của file muốn chia sẻ (path param)  Role: quyền của người dược chia sẻ  Type: chia sẻ với ai | Success: 200  Trả về link chia sẻ  Erors: 400 |  |

Để gọi các API trên google driver cung cấp cho ta một thư viện tên là Google API Client Library. Nó hỗ trợ các hàm truy vấn một cách dễ dàng.

# 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.

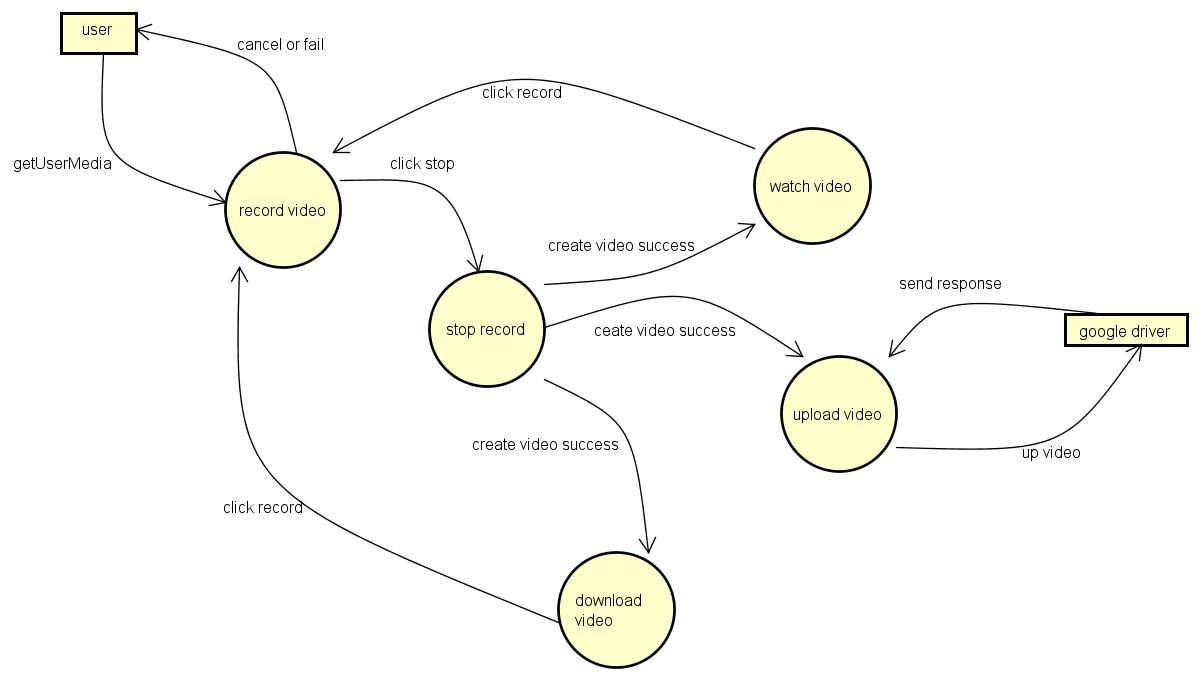
**Google driver:** Hiện tại để nghĩ xem có nơi nào có thể giúp lưu trữ miễn phí tài liệu cũng như các tập tin đa phương tiện đầu tiên ta nghĩ đến là google driver hay dropbox. Nhưng hầu hết ai ai cũng có một tài khoản google nên việc đăng nhập và quản lý tài liệu một cách dễ dàng hơn. Đặc biệt là với 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

1. 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:



Hình 3‑1 Biểu đồ data flow

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



Hình 3‑2 Biểu 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:

Bảng 3‑1 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. | |

Bảng 3‑2 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. | |

Bảng 3‑3 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. | |

Bảng 3‑4 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 | |

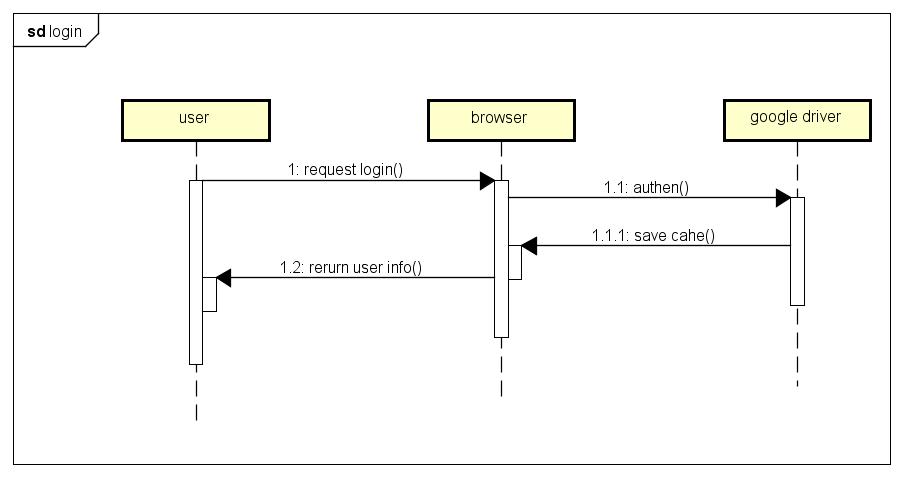
Bảng 3‑5 Download video

|  |  |
| --- | --- |
| 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. | |

Bảng 3‑6 Upload video

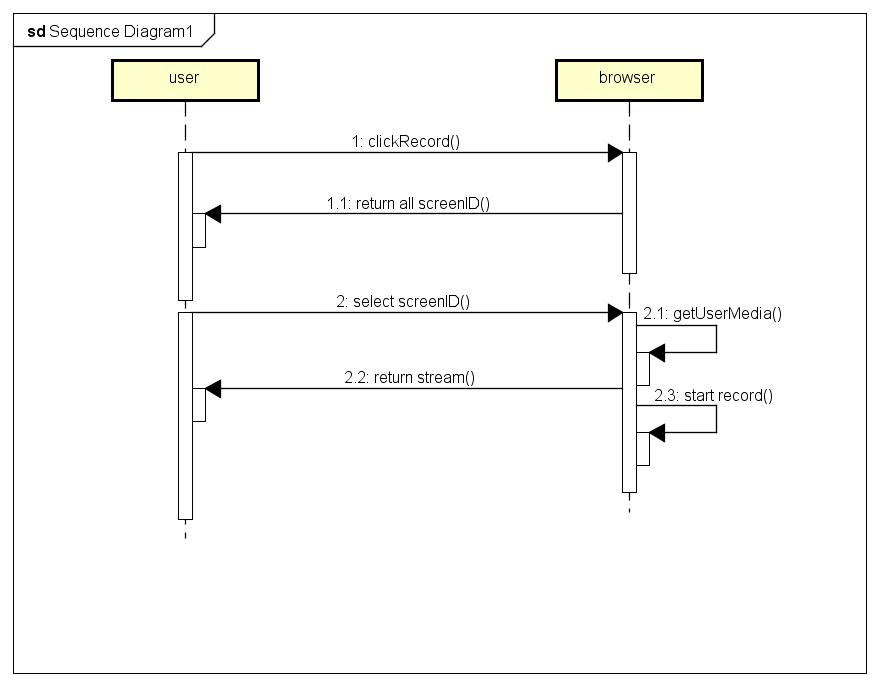
|  |  |
| --- | --- |
| Tên Usecase | Upload video |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | - Sau khi sự kiện stop record kích hoạt  - đã đăng nhâ[j google driver |
| Sự kiện kích hoạt | Click vào nút “upload video” |
| Luồng sự kiện chính:  Khi click vào nút “download video” sẽ lấy object 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 3‑3 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.



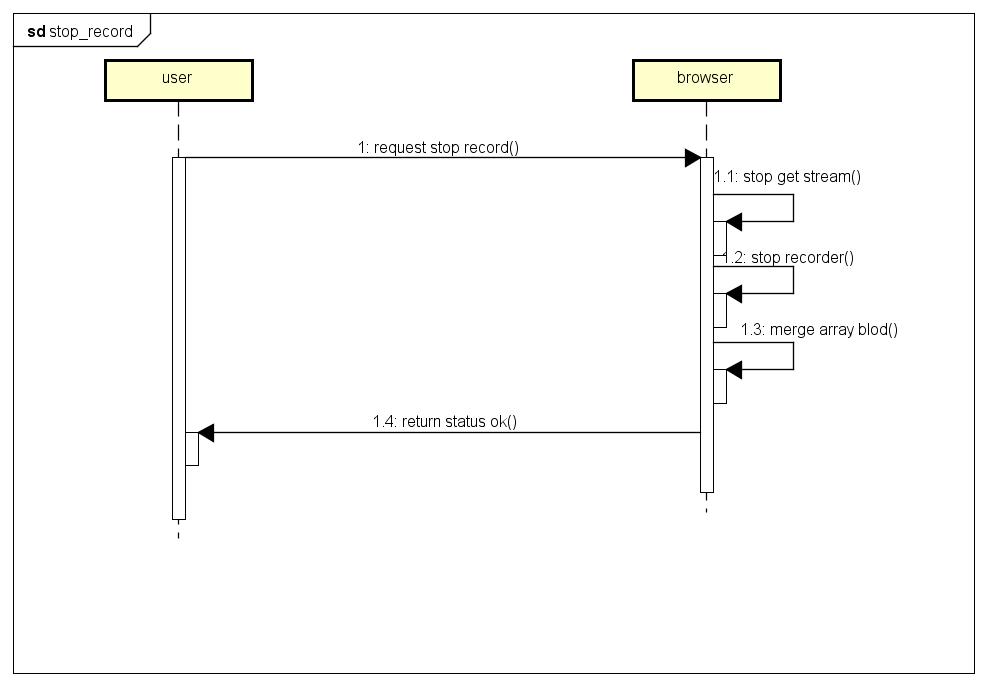
Hình 3‑4 Biểu đồ tuần tự record

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:

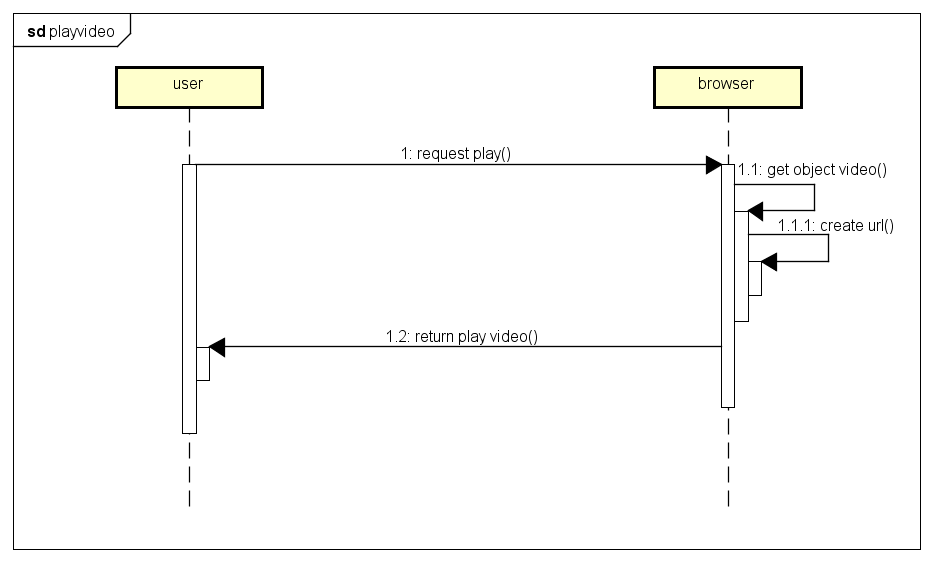
1. Record screen: là kiểu record duy nhất màn hình hoặc các chương trình đang chạy trên desktop. Có các option kèm theo là có audio hoặc không có audio (audio do mic máy tính thu được).
2. Record screen + camera: là kiểu ngoài record như màn hình ở trên còn kèm theo một video nhỏ được hiển thị ở góc 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 3‑5: 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 sàng tương tác với video này.

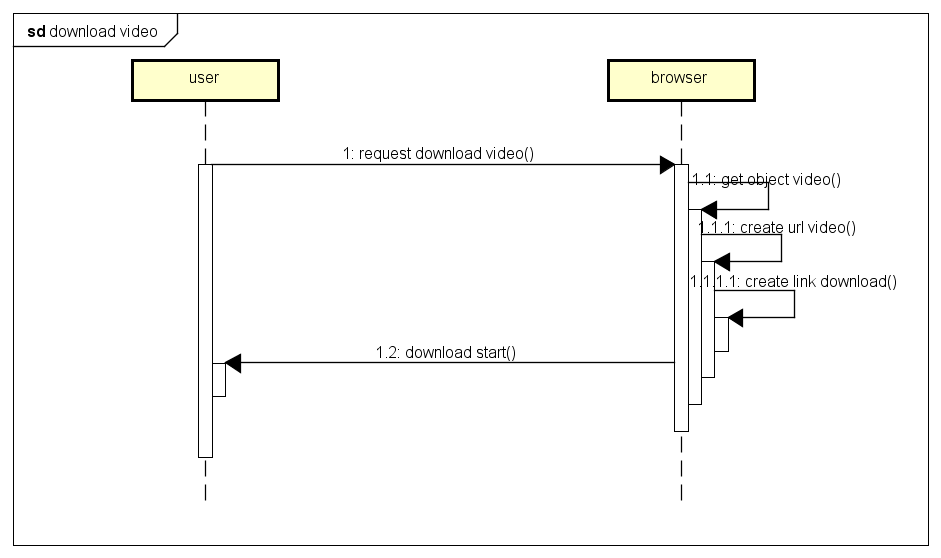
Đối với kiểu ghi có camera+screen thì sẽ có 2 object video được tạo tương ứng với từng mảng blod.



Hình 3‑6: 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.

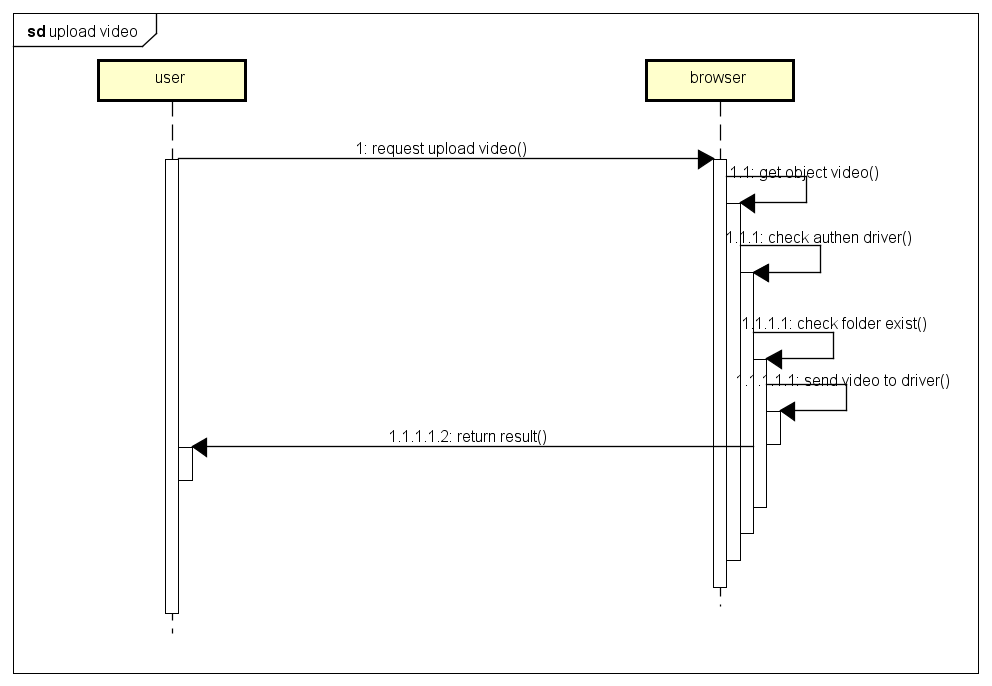
Đối với từng kiểu ghi thì sẽ có từng kiểu play khác nhau. Sẽ có 2 khung hình video được tạo nếu là kiểu ghi screen + camera. Truyền 2 object url video tương ứng vào. Và khi play sẽ play cùng lúc, pause hay seek cũng cùng lúc.



Hình 3‑7: 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.



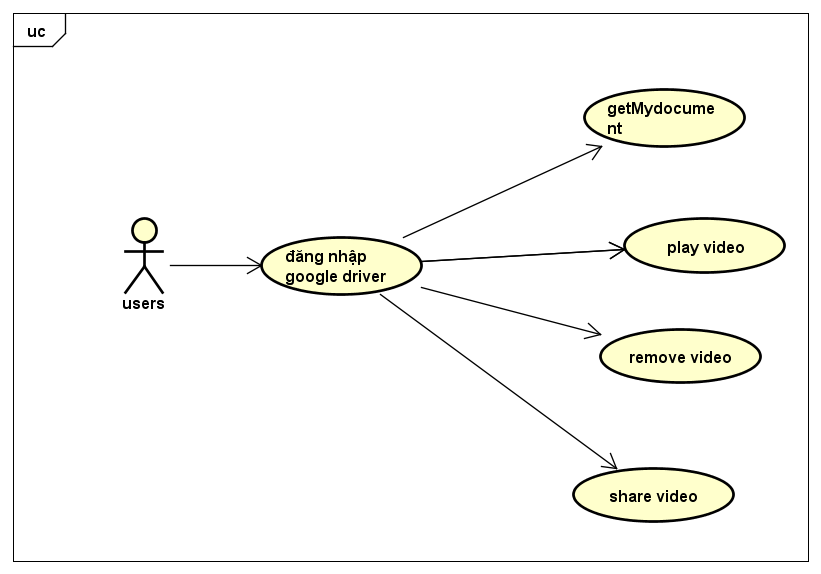
Hình 3‑8 Biểu đồ tuần tự upload video

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 dùng để lưu trữ dạng 2 video. Hai video này sẽ cùng 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 cửa 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 3‑9 Biểu đồ use trang 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 3‑7 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 3‑8 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 | |

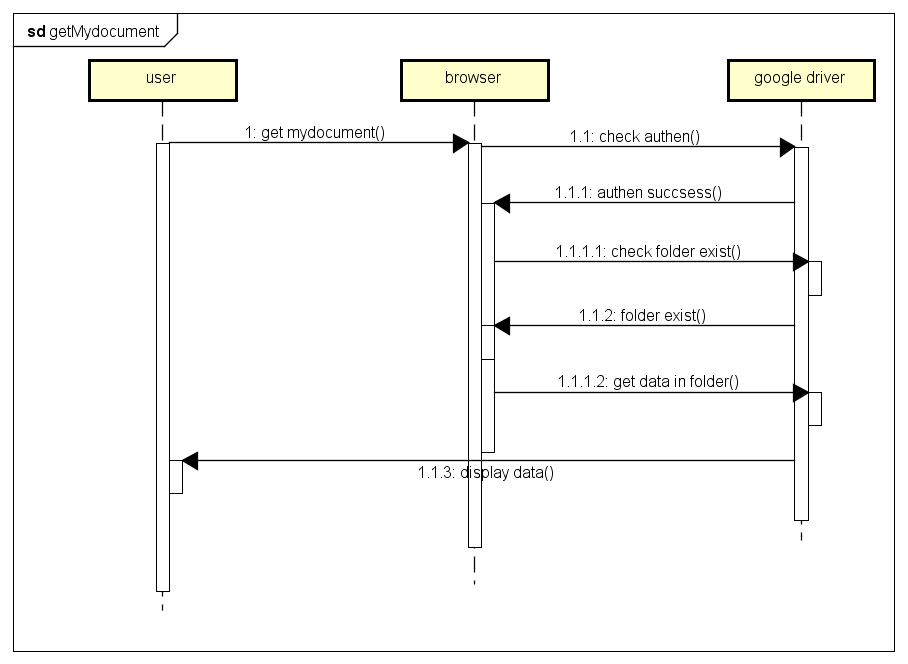
Bảng 3‑9 Usecase remove video

|  |  |
| --- | --- |
| 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 3‑10 Usecase Share Video

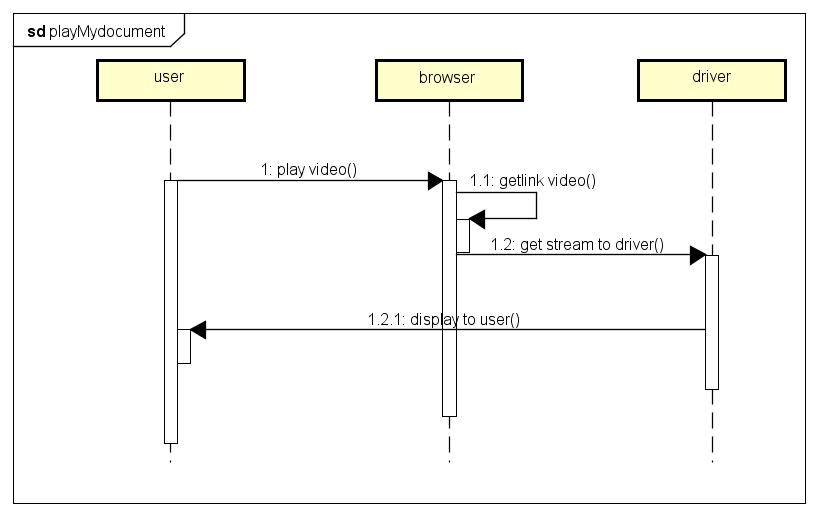
|  |  |
| --- | --- |
| Tên Usecase | Share Video |
| Tác nhân | Người dùng cuối |
| Tiền điều kiện | - Sau khi laod được dữ liệu từ folder chứa video và hiển thị |
| Sự kiện kích hoạt | Click vào biểu tượng nút “share” |
| Luồng sự kiện chính:  Khi click vào nút “share” thì sẽ lấy id của video và sau 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:



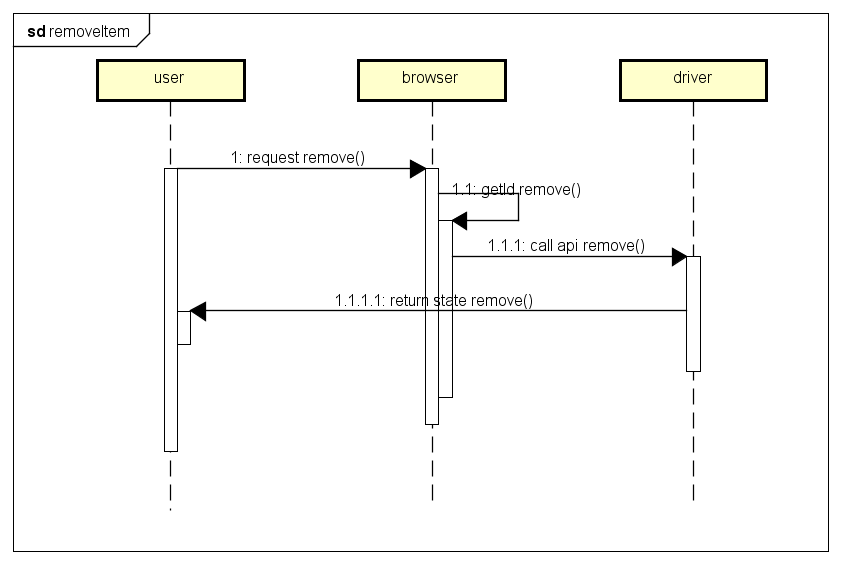
Hình 3‑10 Biểu đồ tuần tự getdocument

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 hiển thị cho người dùng. Nếu không có dữ liệu hoặc folder không tồn tại thì thông báo cho người dùng không tồn tại.



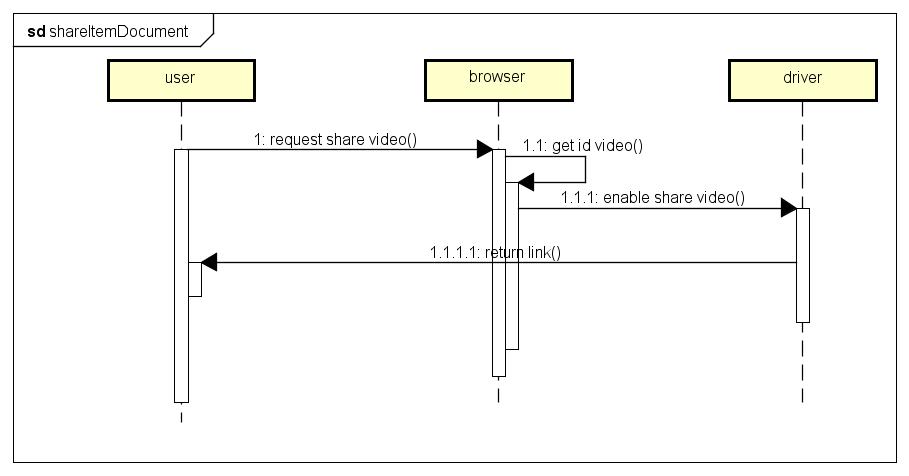
Hình 3‑11 Biểu đồ tuần tự play video trên mydocument

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 3‑12 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.

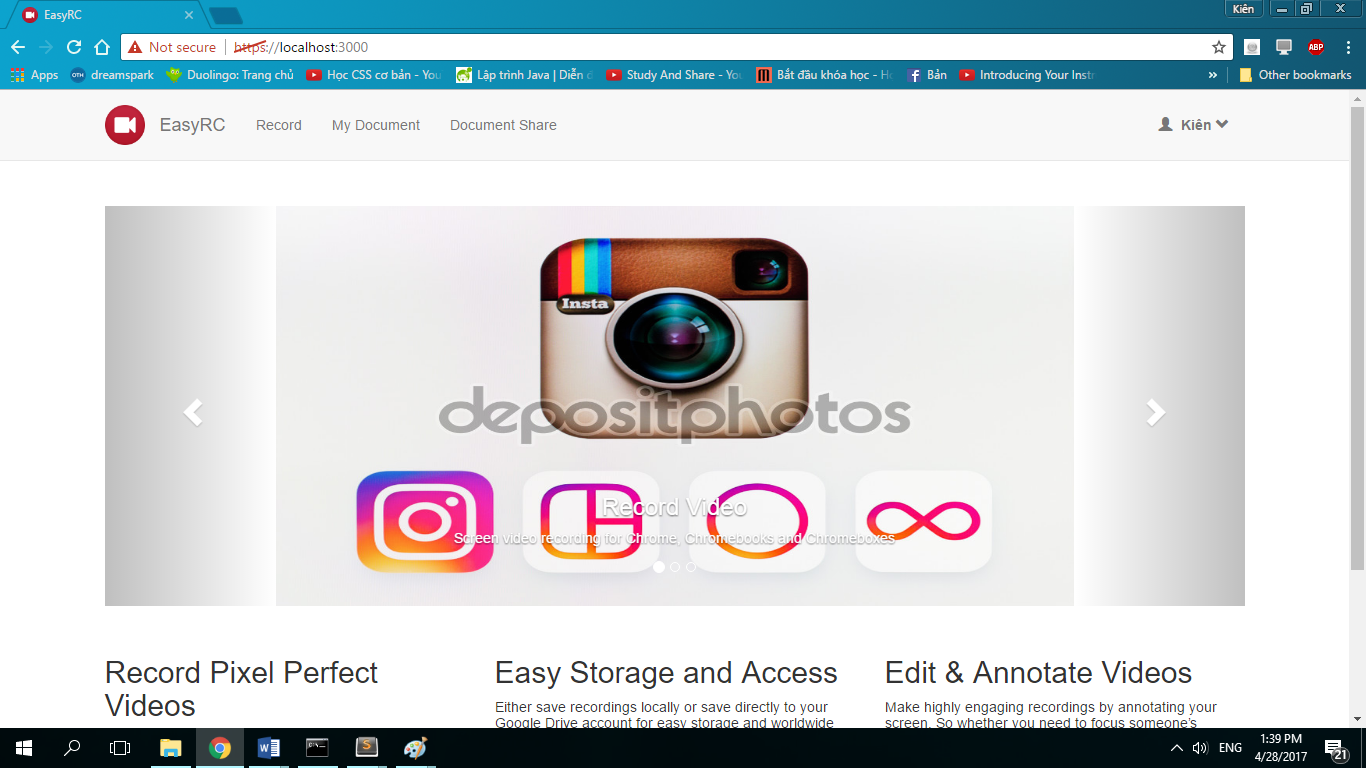


Hình 3‑13 Biểu đồ tuần tự share video trên mydocument

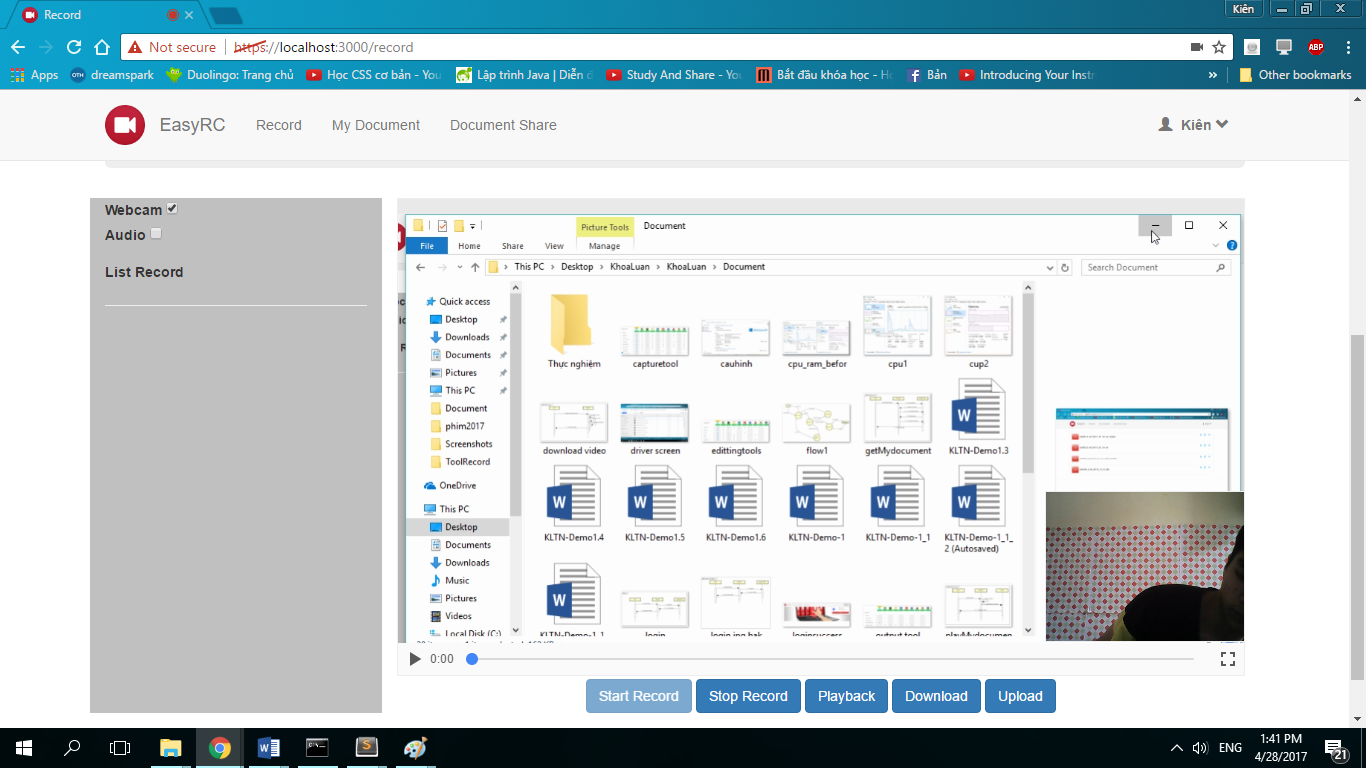
Khi click vào biểu tượng nút “share” thì sẽ lấy id của video hiện tại và gọi api chia sẻ file do google driver cung cấp. api này sẽ bật chế độ chia sẻ của google driver với một file xác định, sau đó sẽ trả về một link để người dùng có thể sao chép và dán bất kỳ nơi nào muốn. Đối với video có nhiều file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

* 1. Giao diện của ứng dụng

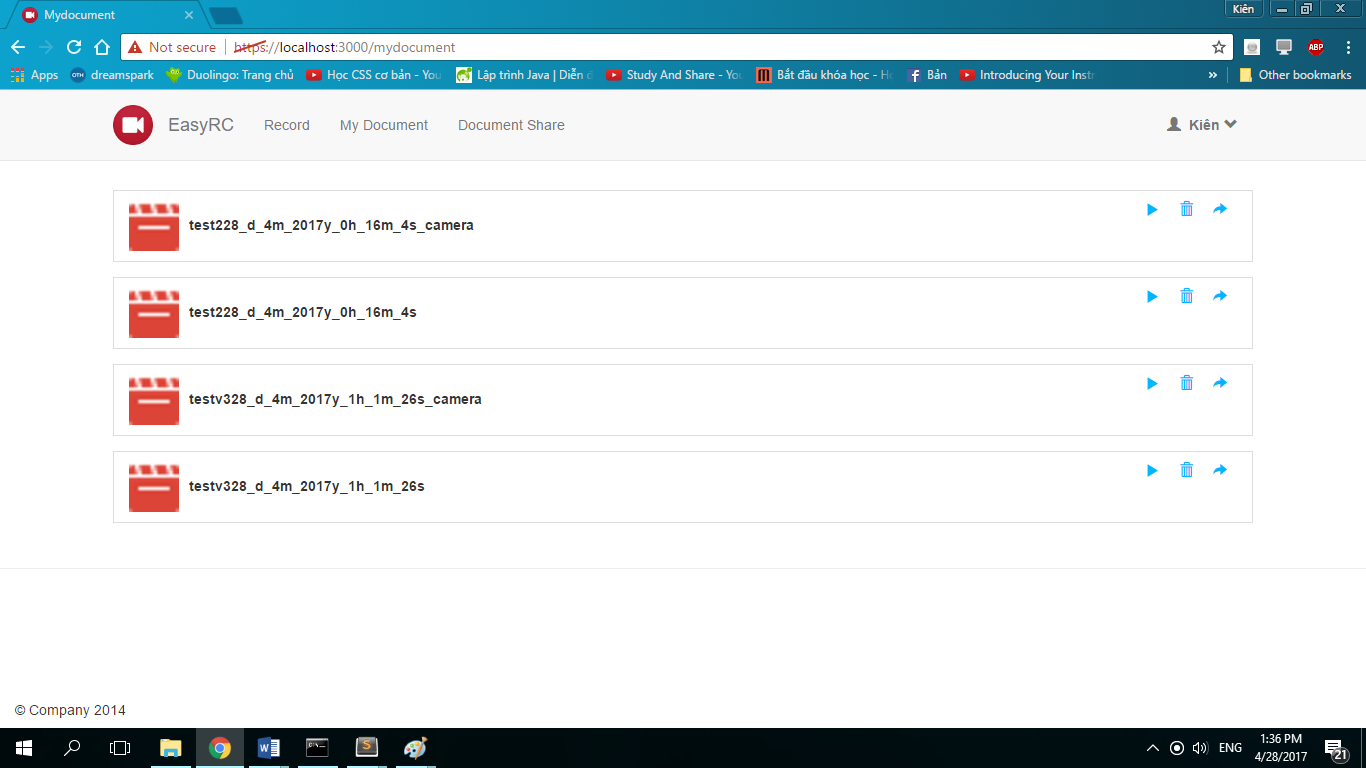
Trang chủ:



Trang Record:



Trang Mydocument:

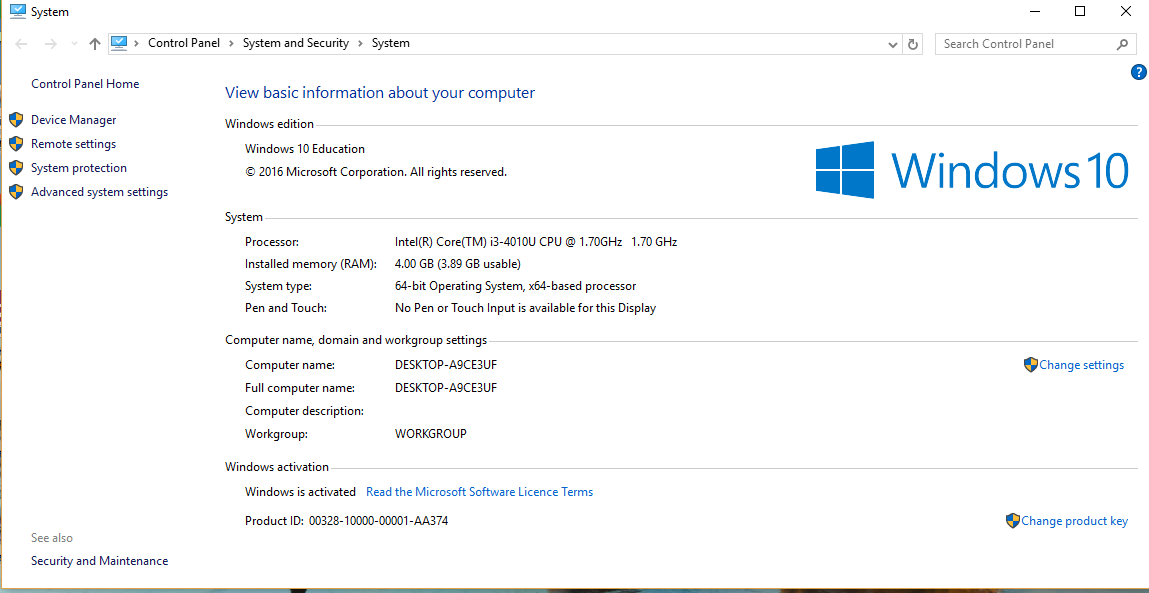


# Demo Và Thực Nghiệm

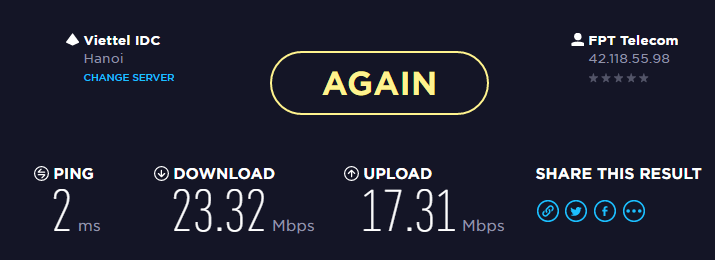
Đầu tiên ta cần môi trường triển khai mã nguồn để tạo ra trang web chạy. Trong demo này ta sẽ sử dụng laptop với cấu hình như sau:

Hệ điều hành win 10, Ram 4GB, chip Intel core i3-4010U, CPU 1.7GHz

Tốc độ mạng được đo bằn trang speedtest.com như hình bên dưới.



Hình 4‑1 Cấu hình máy test



Hình 4‑2 Tốc độ mạng đo bằng speedtest [16]

Cài đặt Nodejs và npm để có thể triển khai localhost trên máy laptop trên. Bạn có thể tải chúng trên trang chủ của nodejs. Sau khi cài đặt thì ta trỏ đường dẫn tới project và chạy câu lệnh sau:

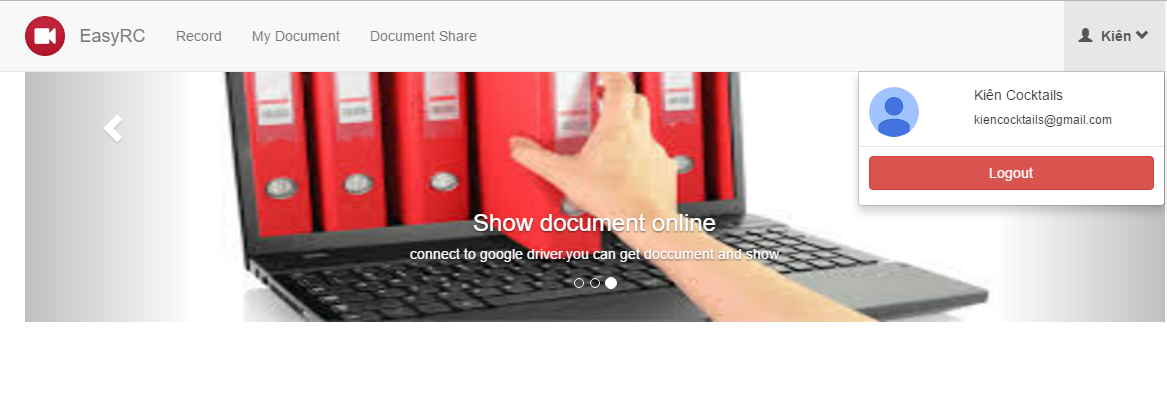
* Npm install
* npm start

Sau khi chạy xong thì web server của nodejs đã hoạt động, ta có thể truy cập vào đường dẫn sau để vào trang chủ của ứng dụng: <https://localhost:3000>

Bắt đầu thử nghiệm những chức năng đầu tiên:

**Đăng nhập:**

Đối với nhứng tài khoản đã từng đăng nhập trước đây ứng dụng không yêu cầu lại cấc quyền. Còn với tài khoản mới ứng dụng sẽ yêu cầu quyền quản lý file trong driver. Sau khi đăng nhập thành công sẽ có giao diện như sau:



Hình 4‑3 Đăng nhập thành công

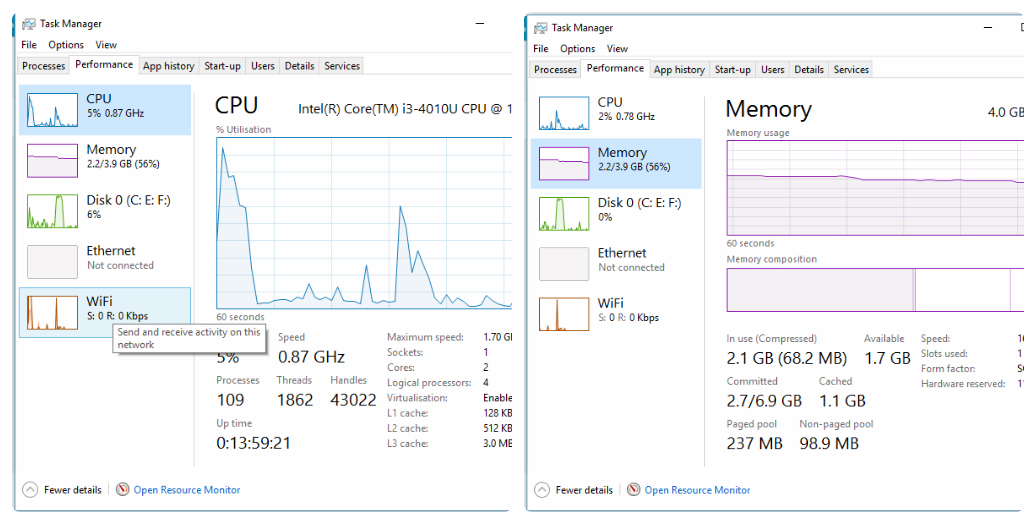
**Chức năng record**:

Đây là chức năng chính và quan trọng nhất của ứng dụng nên ta sẽ làm vài thử nghiệm sau:

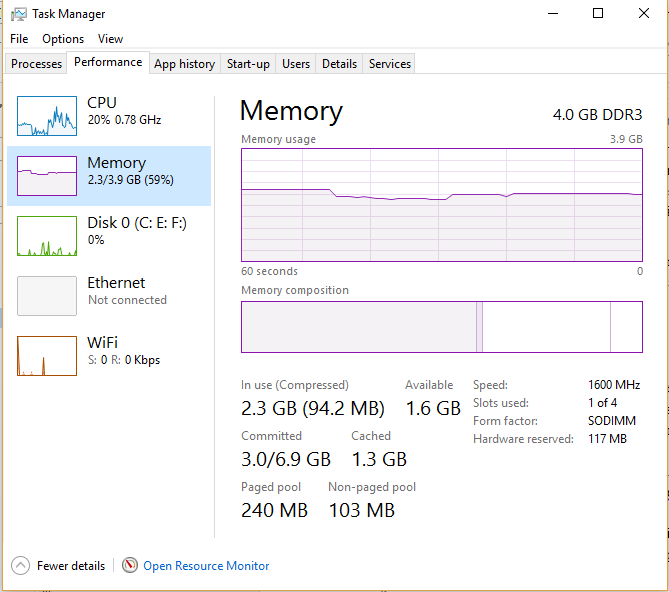
1. Record 1 video screen là chương trình word + camera
2. Record 1 video screen là chương trình youtube trên trình duyệt đang trình chiếu một video với độ phân giải HD + camera.
3. Record 3 video screen là các chương trình word + camera
4. Record 2 video screen là chương trình youtube trên trình duyệt đang trình chiếu các video với độ phân giải khác nhau (480p,720p,fullhd) + camera.
5. Record 1 video ghi lại thao tác toàn màn hình desktop (có microphone nhưng không có camera)
6. Record 1 video ghi lại một chương trình trên máy tính đang chạy (trong trường hợp này là sublimetext và có microphone nhưng không có camera )
7. Record 2 video ghi lại 2 trình duyệt chrome và Internet Explorer hoạt động (có microphone nhưng không có camera).

Sau khi có được danh sách các thực nghiệm, với mỗi thực nghiệm ta đo hiệu suất của cpu, lượng ram sử dụng trước khi record và sau khi record. Kiểm tra các file đầu ra, chất lượng hình ảnh,âm thanh, dung lượng của file và đưa ra nhận xét tổng quát nhất.

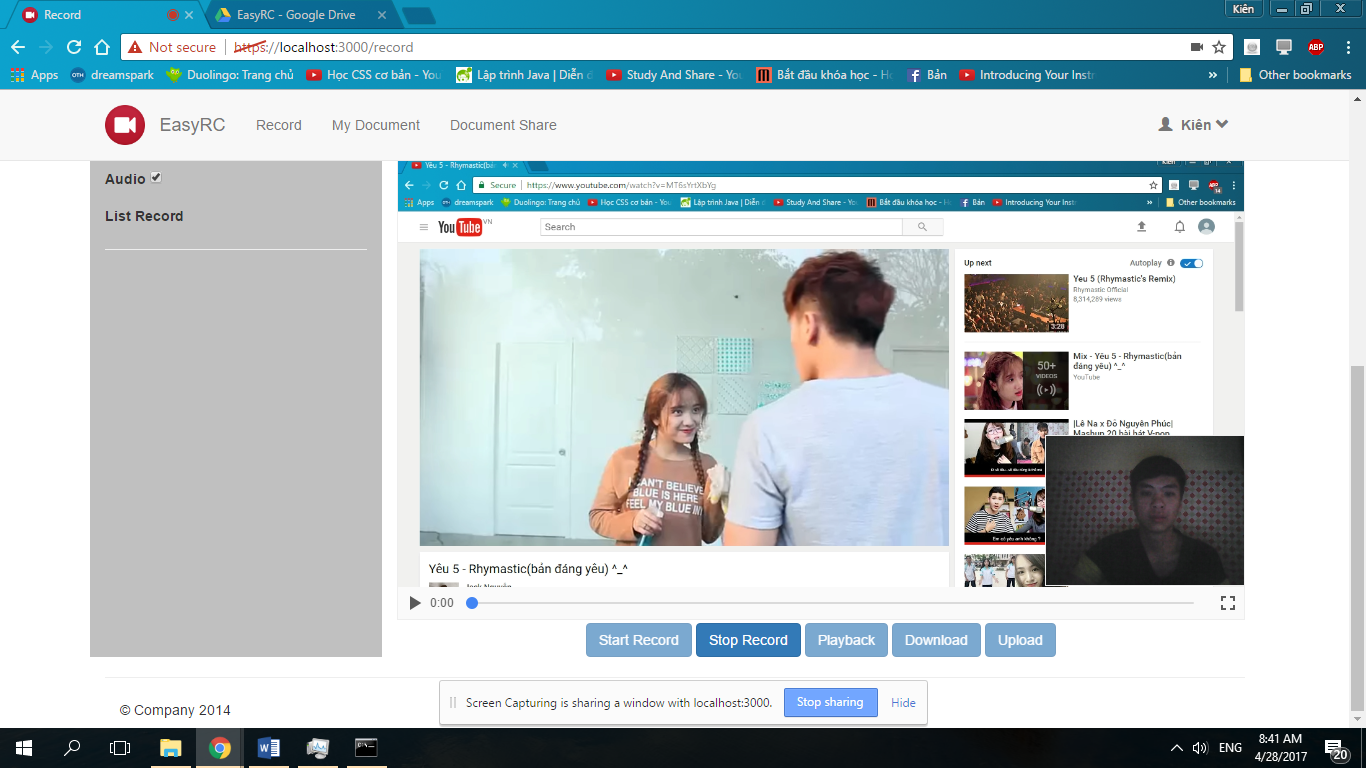
Sau đây là một số hình ảnh trong các lần thực nghiệm:



Hình 4‑4 CPU và RAM trước khi thực nghiệm 1



Hình 4‑5 CPU và RAM trong lúc thực nghiệm 1



Hình 4‑6 Giao diện quá trình record thực nghiệm 2

Dựa vào số lần thực nghiệm trên ta thống kê dược bảng sau:

**Đầu vào:**

Bảng 4‑1 Đầu vào thực nghiệm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Thực nghiệm | Screen | camera | Audio | Số tab | Loại record |
| 1 | 1280x720 | Vga | Microphone | 1 | File word |
| 2 | 1280x720 | Vga | Microphone | 1 | Video youtube độ phân giải 1280x720 |
| 3 | 1280x720 | vga | microphone | 3 | File word |
| 4 | 1280x720 | vga | microphone | 3 | Video youtube độ phân giải 1280x720 |
| 5 | 1280x720 |  | microphone | 1 | Toàn màn hình desktop |
| 6 | 1280x720 |  | microphone | 1 | Chương trình sublimetext |
| 7 | 1280x720 |  | microphone | 2 | 2 trình duyệt web firefox và chrome |

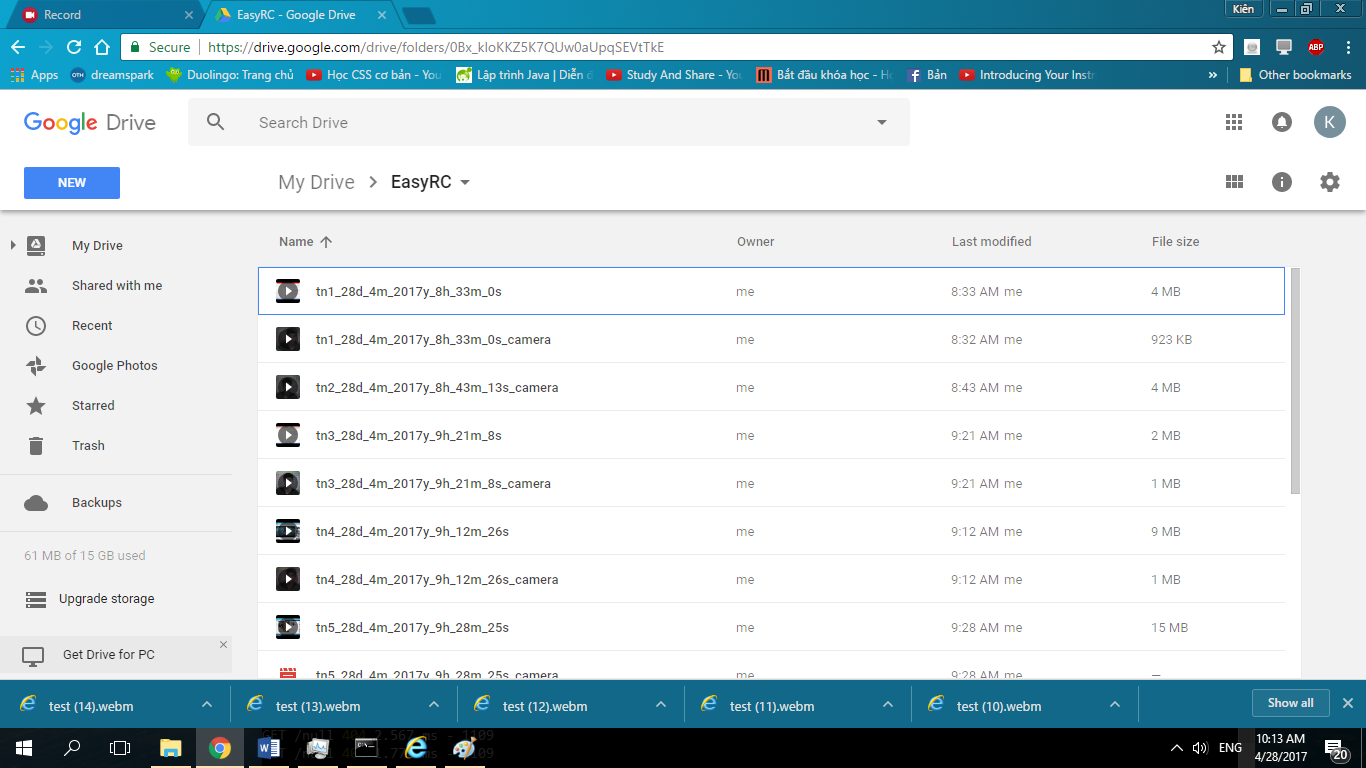
**Đầu ra:**

Bảng 4‑2 Kết quả của thực nghiệm

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Thực nghiệm | Dung lượng file screeen | Dunng lượng file camera | Thời gian | Cpu trước khi chạy | Cpu sau khi chạy | Ram trước khi chạy | Ram sau khi chạy |
| 1 | 4mb | 923kb | 46s | 5% | 20% | 54% | 59% |
| 2 | 226mb | 4mb | 3’42 | 8% | 75% | 64% | 69% |
| 3 | * File1: 3,6mb * File2: 2,6mb * File3: 2,3mb | 2mb | * 1’44s * 1’16 * 1’5 | 5% | 40% | 54% | 62% |
| 4 | * File1: 8,9mb(hd) * File2: 2,7mb(480p) * File3: 8,1mb (hd) | 1mb | * 1’29s * 1’ * 1’5s | 55% | 100% | 64% | 69% |
| 5 | 14,6mb |  | 1’22s | 3% | 10% | 46% | 49% |
| 6 | 2mb |  | 33s | 13% | 19% | 51% | 54% |
| 7 | * File1: 13,7mb * File2: 7,5mb |  | * 1’29s * 1’09s | 30% | 64% | 80% | 72% |

Sau khi quá trình record thành công ta có thể download file hoặc upload file, trong các thực nghiệm trên ta sẽ upload chúng lên driver. Quá trình upload lâu hay chậm tùy thuộc vào dung lượng của file và tốc độ mạng.

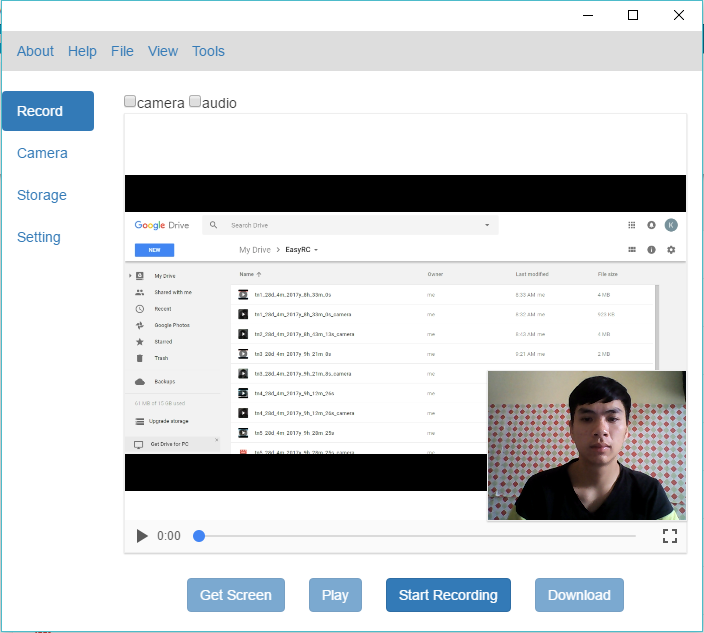
Hình ảnh sau khi upload thành công các file sẽ có trên folder EasyRC:



Hình 4‑7 Các file đã được upload thành công

Trong khóa luận này do yêu cầu các plugin của Webrtc yêu cầu sử dụng giao thức https nên để tìm giải pháp thay thế, ta có thể sử dụng các mã nguồn của chương trình để xây dựng thành một chrome app. Tuy chưa phát triển đầy đủ các chức năng như bản web (thiếu upload file lên google driver) nhưng các chức năng chính như ghi hình, play hay download đều hoạt động ổn định.

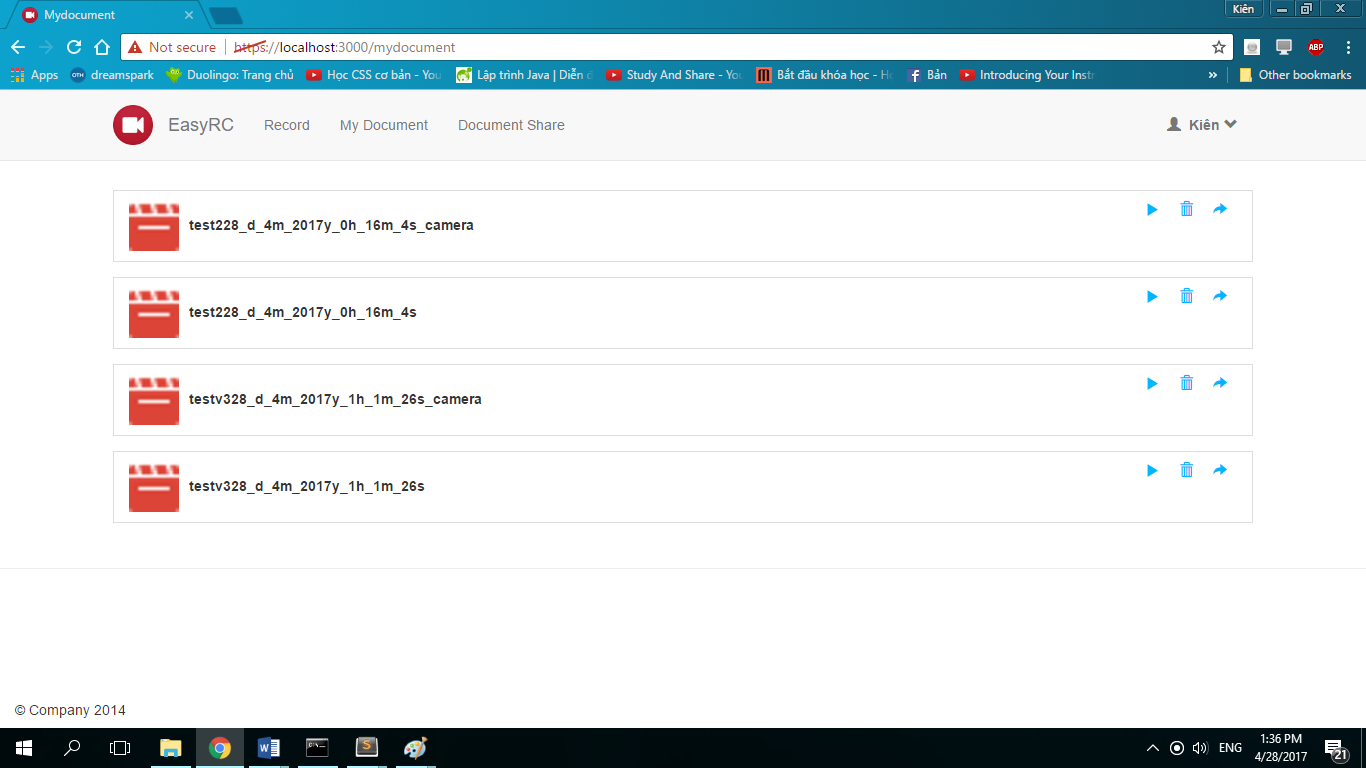
Hình ảnh của chrome app:



Hình 4‑8 Giao diện Chrome App

**Chức năng xem video đã lưu trên google driver:**

Hiển thị danh sách các video đã được lưu trong folder EasyRC, click vào nút play thì có thể play được video.



**Nhận xét - Đánh giá:**

Qua quá trình thực nghiệm với nhiều đầu vào khác nhau ta nhận ra kết quả là: Việc ghi hình được thực hiện nhanh mượt mà kể cả đối với 1 tab hay nhiều tab, chất lượng video ổn định nhưng đối với ghi hình nhiều tab là các video đang phát với chất lượng cao thì cpu sẽ lên cao (3 video hd record cùng một lúc trên 3 trình duyệt cpu sẽ lên 100% đối với cấu hình máy đang thực nghiệm chưa record thì cpu đã xử lý mất trên 50%), lượng ram dùng không đáng kể (tăng từ 20% lên 40-50%) còn đối với các ứng dụng bình thường như các desktop app hay trình duyệt lướt web nó có thể hoạt động mượt mà. Dung lượng file video screen thường lớn sau khi record lâu với chất lượng cao (video youtube), còn dối các video bình thường thì dung lượng nằm trong khoảng 69-70mb cho 3-4 phút ghi. Do chất lượng webcam thấp nên trong các thực nghiệm này chất lượng video camera không thay đổi, dung lượng trong khoảng 3-4mb cho 3-4 phút ghi hình. Tốc độ upload lên google driver khá nhanh, video có dung lượng 3-4mb có thể được upload trong khoảng thời gian 10 giây (đã bật log để đo) đối với mạng tốc độ mạng được đo như trong thực nghiệm.

**Kết luận :**

Đối với ghi hình một tab ứng dụng này có thể hoàn toàn đáp ứng được các yêu cầu về video chất lượng cao, tốc độ xử lý nhanh (xuất file download), dễ dàng thực hiện trong một vài thao tác. ứng dụng rất thích hợp để tích hợp thay thế cho các phần mềm cồng kềnh và khó sử dụng hiện nay với các chức năng chính là ghi hình, đặc biệt các file video có thể chạy trực tiếp trên trình duyệt mà không cần cài plugin phụ trợ như những video khác.Nó có tính thực tiễn cao nếu áp dụng làm một thành phần module giúp giảng viên dễ dàng soạn thảo video trong các mô hình trực tuyến.

# Tổng Kết

## 5.1 Các kết quả đã đạt được

Qua quá trình nghiên cứu và xây dựng ứng dụng thử nghiệm, khóa luận đã thu được các kết quả sau:

* Tìm hiểu được công nghệ Webrtc, các thư viện và plugin hữu ích có thể áp dụng được nhiều dự án trong cuộc sống.
* Tìm hiểu được các mô hình học trực tuyến, cách hoạt động cơ bản của chúng.
* Tìm được giải pháp giúp cải thiện hoặc hỗ trợ việc soạn bài giảng video cho các mô hình trực tuyến.
* Tìm hiểu và ứng dụng lưu trữ trực tuyến của google driver vào một app.
* Phân tích yêu cầu, thiết kế xây dựng một modul có thể triển khai trên các web khác.
* Tìm hiểu thêm về chrome app, cách xây dựng và hoạt động.
* Biêt và sử dụng thêm nhiều APIs của trình duyệt có sẵn hỗ trợ cho việc truy cập phần cứng cũng như lập trình.

## 5.2 Những hướng phát triển

Qua quá trình xây dựng và thử nghiệm ứng dụng, em nhận thấy ứng dụng còn thiếu nhiều chức năng nâng cao và có thể xử lý được nhanh mượt hơn. Sau đây là một số hướng em sẽ có thể nghiên cứu và phát triển thêm sau khóa luận:

* Thêm các chức năng cho phép chỉnh sửa video trực tiếp trên trình duyệt như : cắt video, nối video, thêm hiệu ứng cho video sinh động….
* Ghép 2 video camera và screen và xuất ra thành một file
* Tìm cách nén và giảm dung lượng file đầu ra mà không làm giảm quá nhiều chất lượng video.
* Xây dựng thành một thư viện hoàn chỉnh có thể tích hợp thẳng vào các trang web cần sử dụng tới record video.

DANH MỤC TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | Wiki, "Coursera," [Online]. Available: https://wikipedia.org/wiki/Coursera. |
| [2] | Wiki, "Edx," wikipedia, [Online]. Available: https://wikipedia.org/wiki/EdX. |
| [3] | Wiki, "Udemy," wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Udemy. |
| [4] | B. S. Lin, "3 Easy Methods to Create eLearning Videos," elearningindustry, 11 September 2014. [Online]. Available: https://elearningindustry.com/3-easy-methods-create-elearning-videos. |
| [5] | D. Chadwick, "The Best Video Capture Software of 2017," toptenreviews, 2016 AUGUST 05. [Online]. Available: http://www.toptenreviews.com/software/multimedia/best-video-capture-software/. |
| [6] | Google, "About WebRTC," Google, [Online]. Available: https://webrtc.org/. |
| [7] | bloggeek, "WebRTC & Wello: An Interview With Amol Kher," Wello, 2012 December 6. [Online]. Available: https://bloggeek.me/webrtc-wello-interview/. |
| [8] | M. Khan, "Chrome app plugin," www.webrtc-experiment.com, [Online]. Available: https://chrome.google.com/webstore/detail/screen-capturing/ajhifddimkapgcifgcodmmfdlknahffk. |
| [9] | M. Khan, "WebRTC captureStream API," [Online]. Available: http://www.muazkhan.com/. |
| [10] | M. Khan, "getScreenId | Capture Screen on Any Domain," [Online]. Available: https://www.webrtc-experiment.com/getScreenId/. |
| [11] | Mozilla Developer Network, "Web APIs," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API. |
| [12] | MDN, "MediaDevices.getUserMedia() API," Mozilla Developer Network, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia. |
| [13] | MDN, "MediaRecorder," Mozilla Developer Network, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/MediaRecorder. |
| [14] | R. Dahl, "About Nodejs," [Online]. Available: https://nodejs.org/en/. |
| [15] | Google, "Google Drive Api Reference," Google, [Online]. Available: https://developers.google.com/drive/v2/reference/. |
| [16] | Ookla, "Test Network," [Online]. Available: http://www.speedtest.net/. |
| [17] | M. Khan, "getScreenId | Capture Screen on Any Domain!," [Online]. Available: https://www.webrtc-experiment.com/getScreenId/. |
| [18] | M. Khan, "WebRTC captureStream API," [Online]. Available: http://www.muazkhan.com/. |