TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO
THỰC TẬP HỆ THỐNG THÔNG TIN QUẢN LÝ

ĐỀ TÀI: XÂY DỰNG WEB NHẬN DIỆN VĂN BẢN TIẾNG VIỆT SỬ DỤNG TESSERACT

Sinh viên thực hiện	**: NGUYỄN VĂN NAM**
Giảng viên hướng dẫn	**: HOÀNG THANH TÙNG**
Mã sinh viên	**: 18810310428**
Chuyên ngành	**: CÔNG NGHỆ PHẦN MỀM**
Lớp	**: D13CNPM5**
Khóa	**: 2018– 2023**

Hà Nội, tháng 7 năm 2022

TRƯỜNG ĐẠI HỌC ĐIỆN LỰC
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Hà Nội, ngày 27 tháng 06 năm 2022

ĐỀ CƯƠNG THỰC TẬP MÔN
THỰC TẬP HỆ THỐNG THÔNG TIN QUẢN LÝ

1. Tên đề tài:

"Xây dựng web nhận diện văn bản tiếng việt sử dụng Tesseract".

2. Sinh viên thực hiện:

Họ và tên: Nguyễn Văn Nam MSSV: 18810310428

Số điện thoại: 0398727881 Email: nguyenvannamtgdd35@gmail.com

Vị trí thực tập: Lập trình viên .Net

Thời gian thực tập: Từ ngày 04/04/2022 đến ngày 27/06/2022

3. Giảng viên hướng dẫn :

Họ và tên : Hoàng Thanh Tùng

Số điện thoại : 0978421326 Email : tunght@epu.edu.vn

4. Cán bộ hướng dẫn tại đơn vị thực tập

Họ và tên: Lê Xuân Trường Chức vụ: Quản lý dự án

Số điện thoại : 0964038801 Email : truong.lexuan@vietis.com.vn

Phòng/Bộ phận : Phòng sản xuất số 3

Tên đơn vị thực tập : Công ty Cổ phần đầu tư và Giải pháp VietIS

Địa chỉ: 3A Building, 82 Duy Tân, Hà Nội.

4. Mô tả tóm tắt đề tài

Xây dựng web nhận diện văn bản tiếng việt sử dụng công nghệ Tesseract, xây dựng giao diện cho web nhận diện.

5. Nội dung hướng dẫn:

- Chương 1: Giới thiệu công ty – đơn vị thực tập.
- Chương 2: Khảo sát hiện trạng, mô tả hiện trạng bài toán nhận diện văn bản tiếng việt sử dụng Tesseract.
- Chương 3: Phân tích hệ thống
- Chương 4: Thiết kế giao diện chương trình

6. Kết quả cần đạt được:

- Phân chia công việc theo đầu mục.
- Xây dựng giao diện chương trình.
- Xây dựng được modul cho chương trình.

7. Các yêu cầu đối với sinh viên:

- Có khả năng học và tìm hiểu các tài liệu bằng Tiếng Anh.
- Có khả năng lập trình với Nodejs, …
- Sử dụng công cụ lập trình Visual studio, Studio 3T, MongoDB, …

Giảng viên hướng dẫn	**Sinh viên thực hiện**
Hoàng Thanh Tùng	**Nguyễn Văn Nam**

PHIẾU ĐIỂM

Sinh viên thực hiện:

Họ và tên	Nhiệm vụ	Chữ ký
Nguyễn Văn Nam 18810310428		

Giảng viên chấm:

Họ và tên	Nhận xét	Điểm
Giảng viên chấm 1:		
Giảng viên chấm 2:		

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường **Đại học Điện Lực**, đặc biệt là các thầy cô **Công Nghệ Thông Tin** của trường đã tạo điều kiện cho em thực hiện báo cáo. Và em cũng xin chân thành cảm ơn thầy **Hoàng Thanh Tùng** đã nhiệt tình hướng dẫn hướng dẫn em hoàn thành tốt báo cáo.

Trong quá trình thực tập, cũng như là trong quá trình làm bài báo cáo thực tập, khó tránh khỏi sai sót, rất mong các thầy, cô bỏ qua. Đồng thời do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp thầy, cô để em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn bài báo cáo.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Nguyễn Văn Nam

LỜI NÓI ĐẦU

Trong vài thập kỉ trở lại đây, với sự bùng nổ của ngành công nghệ thông tin đã mang lại cho chúng ta những thành tựu công nghệ mới, việc này tạo điều kiện cho sự phát triển và ra đời của công nghệ AI trong ứng dụng văn phòng cũng như tòa nhà, việc quản lý nhân sự…

Dựa trên nền tảng đó, em đã kết hợp sử dụng công nghệ nhận diện văn bản để "**Xây dựng website nhận dạng văn bản tiếng việt sử dụng Tesseract**".

Trong quá trình này, với sự giúp đỡ và tư vấn nhiệt tình từ thầy **Hoàng Thanh Tùng**, cùng với các thầy cô và toàn thể các anh chị đồng nghiệp tại **Công ty Cổ phần đầu tư và Giải pháp VietIS** đã giúp em xây dựng sản phẩm này.

Em xin chân thành cảm ơn!

CHƯƠNG 1: GIỚI THIỆU CÔNG TY - ĐƠN VỊ THỰC TẬP

1.1. Giới thiệu về VIETIS

Hình 1.1: Công ty Cổ phần Đầu tư và Giải pháp VietIS

Công ty Cổ Phần Đầu tư và Giải pháp VietIS (gọi tắt là VietIS Corporation) được thành lập từ năm 2009 và bắt đầu cung cấp dịch vụ phát triển phần mềm do đối tác Nhật Bản từ năm 2013.

VietIS Corporation hiện có 1 trung tâm phát triển tại Hà Nội và 1 văn phòng tại Tokyo Nhật Bản với đội ngũ kỹ sư lên đến hơn 250 người. VietIS luôn theo đuổi mục tiêu tạo ra những đột phá và trở thành một trong những công ty công nghệ hàng đầu Việt Nam, là đối tác đáng tin cậy của khách hàng Nhật Bản.

Sở hữu đội ngũ kỹ sư, chuyên gia và quản lý dự án nhiều năm kinh nghiệm tham gia phát triển ứng dụng điện thoại thông minh, dịch vụ ứng dụng, phát triển hệ thống cốt lõi và phát triển công nghệ tiên tiến như IoT, AI và điện toán đám mây.

VietIS luôn là đối tác làm việc của nhiều khách hàng Nhật Bản lớn như Honda, NEC, Fujitus, JBS, Hoya… Với định hướng tăng trưởng mạnh về số lượng nhân sự ở cả Việt Nam và Nhật Bản, VietIS hiện có nhiều cơ hội việc làm, thực tập hấp dẫn chờ đợi học viên FUNiX.

Hình 1.2: Các đối tác tiêu biểu của VietIS Corporation

Trong bối cảnh Covid – 19, công ty vẫn liên tục tuyển mới các vị trí từ nhân viên đến quản lý với mức lương tới 35 triệu đồng một tháng. Để đối phó với dịch bệnh, công ty tổ chức phỏng vấn online qua skype từ ngày 09/03, làm việc từ xa từ ngày 01/04. Những vị trí mà công ty liên tục tuyển dụng gồm: Junior Java, Mobile Techlead, BrSE Mobile, Junior Tester, Sale IT, …

Cùng với mức lương hấp dẫn, công ty có nhiều chương trình đào tạo giúp ứng viên phát triển như: Đào tạo Fresher cho sinh viên mới ra trường: đào tạo quy trình sản xuất phần mềm – Software Process Development cho nhân viên mới ra nhập; các khóa đào tạo Project Management, Khóa đào tạo kỹ năng mềm: Horensho, Leadership, … Khóa đào tạo ngôn ngữ: Tiếng Nhật (N5), tiếng anh, …

Ngoài ra, VietIS thường xuyên tổ chức seminar chia sẻ công nghệ giúp cán bộ, nhân viên nâng cao tri thức, trình độ. Cùng với đó công ty có chính sách hỗ trợ nhân viên học và thi các chứng chỉ cần thiết trong công việc cùng với mức trợ cấp hấp dẫn.

Hình 1.3: Seminar nghề BrSE

1.2. Lịch sử hình thành

Công ty Cổ phần Đầu tư và Giải pháp VietIS có 2 chi nhánh hiện nay, đó là một chi nhánh tại Hà Nội và một chi nhánh tại Nhật Bản.

Chi Nhánh tại Nhật Bản có địa chỉ là Tokyo, Shinagawa, Nishigotanda, 2-25-1 Intex 2F. Gồm các kỹ sư hàng đầu tại VietIS.

Với Chi nhánh tại Hà Nội, hiện này VietIS có 2 địa điểm chính là tầng 3 và tầng 5, tòa nhà 3A, ngõ 82 Duy Tân và tầng 7 Cầu Giấy, Hà Nội.

1.3. Tầm nhìn, sứ mệnh, giá trị cốt lõi

VietIS Corporation luôn mong muốn trở thành một trong những công ty công nghệ hàng đầu Việt Nam, là đối tác đáng tin cậy đối với khách hàng Nhật Bản.

VietIS là một công ty Outsourcing chuyên cung cấp các giải pháp phần mềm như Cloud Service, Business Application Development, Web/Mobile Development, … cho thị trường Nhật Bản. Chính vì vậy, VietIS luôn mong muốn trở thành một đối tác đáng tin cậy với tất cả các đối tác.

1.4. Văn hóa doanh nghiệp

Môi trường và các hoạt động: VietIS hiểu rằng môi trường làm nên con người, cho nên VietIS luôn chú trọng để phát triển môi trường học hỏi, vui chơi tốt nhất cho nhân viên.

Với môi trường làm việc trẻ trung, năng động và chuyên nghiệp và nhiều cơ hội thăng tiến. Sức trẻ và nhiệt huyết của VietIS được tạo nên từ chính mỗi cá nhân công ty, cho nên VietIS không ngừng thay đổi để hoàn thiện hơn nữa bản thân.

Với việc tổ chức sinh nhật tháng của nhân viên, cũng như các hoạt động khác do công đoàn tổ chức như Teambuilding ngoài trời định kì, các hoạt động vui chơi giải trí khác, tham gia vào các câu lạc bộ vui chơi như CLB bóng đá, … đã tạo ra một môi trường năng động phù hợp với rất nhiều bạn trẻ.

Hình 1.4: Sự kiện Teambuilding

Hình 1.5: VietIS tổ chức chương trình trải nghiệp tiếng Nhật cho nhân viên

VietIS dành chỗ cho sinh viên tới thực tập: Công ty VietIS còn thiết kế riêng một góc với 20 chỗ ngồi cho học viên FUNIX tới thực tập. VietIS là đơn vị ký kết hợp tác đào tạo – tuyển dụng với FUNIX trong tháng ba, thuộc chương trình "100 doanh nghiệp IT cùng FUNIX phát triển nguồn nhân lực"

Hợp tác nhằm đảm bảo bám sát và đáp ứng nhu cầu tuyển dụng thực tế của doanh nghiệp, phù hợp xu hướng phát triển năng lực lao động của ngành phần mềm toàn cầu. VietIS sẵn sàng cử chuyên gia có trình độ cao tham gia trực tiếp vào đào tạo trong vai trò cố vấn chuyên môn học tập (memtor) để đồng hành cùng sinh viên trong suốt quá trình học tập.

Hình 1.6: Hình ảnh cho biết doanh nghiệp có nhiều cơ hội làm việc thực tập hấp dẫn đang chờ đợi học viên FUNiX

CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG VÀ MÔ TẢ HIỆN TRẠNG BÀI TOÁN NHẬN DIỆN VĂN BẢN TIẾNG VIỆT

2.1. Mô tả bài toán

OCR là thuật ngữ được viết tắt bởi cụm từ Optical Character Recognition (dịch là: nhận dạng ký tự quang học). Đây là ứng dụng công nghệ chuyên dùng để đọc text ở file ảnh. Được biết đến là một công cụ scan kỹ thuật số chuyên nhận dạng các ký tự, chữ viết tay, hay chữ đánh máy, công nghệ này chuyên dùng để truyền tải, nhập liệu dữ liệu. Đặc biệt, ở OCR có khả năng kỹ thuật số nhiều dưới nhiều dạng tài liệu khác nhau: hóa đơn, hộ chiếu, danh thiếp, tài liệu…

Đến với OCR, những văn bản số hóa, tìm kiếm và chỉnh sửa sẽ được thực hiện điện tử. Đồng thời, chúng giúp tiết kiệm không gian lưu trữ tài liệu bằng việc hiển thị trên trực tiếp.

Trong hoạt động công việc thường nhật, việc cần scan những tài liệu dưới dạng ghi chú viết tay hay là những cuốn sách tài liệu thường khó tránh khỏi. Giờ đây, với công nghệ nhận dạng ký tự quang học OCR sẽ đem đến cho bạn những trải nghiệm thú vị.

OCR giúp phân tích các văn bản dưới dạng in hoặc viết tay thành dạng file số có thể chỉnh sửa TIF.

2.2. Giới thiệu bài toán

Nhận dạng kí tự quang học (OCR) là một lĩnh vực nghiên cứu chuyển ảnh số được chụp hoặc quét từ tài liệu viết tay, đánh máy hay in thành dạng văn bản mà máy tính có thể hiểu được.

Trên thế giới, công nghệ OCR đã có những tác động sâu sắc đến nhiều lĩnh vực trong đời sống và sản xuất. Việc chuyển các văn bản in trên giấy thành dạng điện tử nhỏ gọn, dễ tìm kiếm giúp hàng triệu trang báo đến được với các bạn đọc khắp nơi trên thế giới. Bằng cách kết hợp với phần mềm text – to – speech lượng tài liệu này có thể đọc thành tiếng cho những người khiếm thị. Nhiều bưu điện đã áp dụng hệ thống phân loại thư tự động dựa trên máy đọc bì thư có cài phần mềm OCR. Các ngân hàng đọc nội dung của séc để chống rửa

tiền, gian lận và cả phát hiện khủng bố. OCR còn đi vào đời sống hàng ngày qua những thiết bị thông tin cá nhân giúp người sử dụng nhập dữ liệu bằng cách viết lên màn hình cảm ứng thay vì đem theo bộ bàn phím cồng kềnh.

Ở Việt Nam, công nghệ OCR mới chỉ phát triển ở giai đoạn đầu với một vài bộ phần mềm nhận dạng kí tự in như VnDOCR, VietOCR, ABBYY trong đó lĩnh vực nhận djang chữ viết tay vẫn còn bỏ ngỏ.

Qúa trình OCR gồm nhiều bước như phân tích cấu trúc văn bản, tách dạng, tách kí tự, kiểm tra ngữ nghĩa để tăng độ chính xác, … Hiện này Tesseract là một OCR hàng đầu. Công cụ này hỗ trợ nhận diện kí tự trên các tập hình ảnh và xuất ra dưới dạng kí tự thuần, html, … Người sử dụng có thể sử dụng trực tiếp hoặc các chức năng thông qua API.

Hiện nay, Tesseract có thể hoạt động trên các hệ điều hành phổ biến như Window, Mac, Linux. Công cụ này hỗ trợ nhận diện kí tự của hơn 100 ngôn ngữ khác nhau trong đó bao gồm cả tiếng Việt. Không những thế chúng ta có thể huấn luyện chương trình dùng Tesseract để có thể nhận diện một ngôn ngữ nào đó.

2.3. Tính cấp thiết của đề tài trong thực tiễn

Hiện nay việc rút trích từ ngữ từ hình ảnh đang ngày càng phát triển, bên cạnh sự gia tăng về nhu cầu là sự phát triển nhận dạng ký tự quang học hay còn được gọi tắt là OCR. Đây là một công nghệ giúp chuyển đổi hình ảnh của chữ viết tay hay chữ đánh máy thành các ký tự đã được mã hóa trong máy tính.

Giả sử chúng ta cần chỉnh sửa một số tài liệu giấy như: các bài viết trên tạp trí, tờ rơi hoặc một tập tin PDF hình ảnh. Rõ ràng chúng ta không thể sử dụng máy quét để chuyển các tài liệu này thành tập tin văn bản để có thể chỉnh sửa. Tất cả những gì máy quét có thể làm là tạo ra một hình ảnh hoặc một bản chụp của các tài liệu. Để giải nén và sử dụng lại dữ liệu từ tài liệu được quét, hình ảnh máy ảnh và hình ảnh của các tập tin PDF chúng ta cần một phần mềm OCR. Nó sẽ xuất ra kí tự trên hình ảnh, ghép chúng thành các từ và sau đó ghép các từ thành các câu. Nhờ vậy, chúng ta có thể truy cập và chỉnh sửa nội dung của tài liệu gốc.

Tương tự, những tài liệu cũ đang bị hư hại theo thời gian và việc viết tay hay đánh máy lại những tài liệu đó sẽ tốn rất nhiều thời gian cũng như chi phí cho việc này, ngoài ra độ chính xác còn không được đảm bảo cũng như sự an toàn cho tài liệu nền. Việc này rất cần một công nghệ lấy từ ngữ từ hình ảnh chụp.

Trong bài báo cáo nay, tôi sẽ tìm hiểu, chỉnh sửa công cụ Tesseract để thực thiện được việc rút trích từ các văn bản từ tập tin hình ảnh cũng như từ các tập tin PDF.

2.4. Các chức năng tổng quát của module

- Thu thập dữ liệu
- Chuyển đổi các dạng như file pdf sang text, từ ảnh sang text.
- Quản lý đăng nhập, đăng xuất.
- Quản lý phân quyền admin và user.

2.5. Các yêu cầu phi chức năng

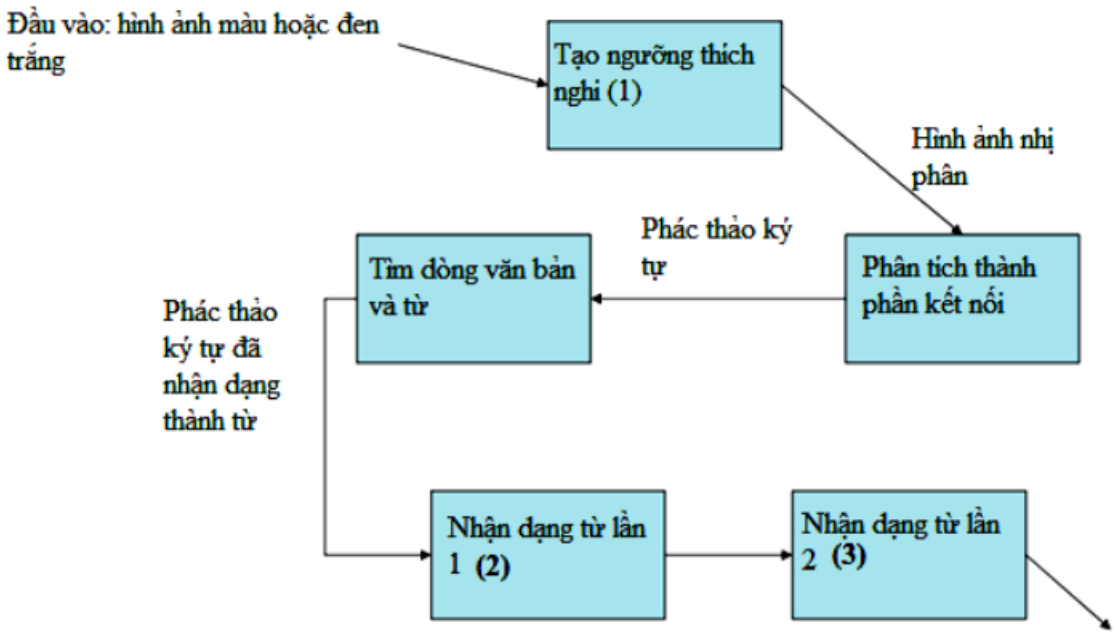
- Ngôn ngữ lập trình: Nodejs
- Công cụ lập trình: Visual studio
- Database: MongoDB

2.6. Giới thiệu các công cụ cần dùng

2.6.1. Thuật toán Tesseract

Tesseract là một công cụ nhận diện ký tự đang được phát triển bởi Google với bản quyền mã nguồn mở được nghiên cứu và phát triển bởi HP trong giai đoạn 1984 – 1994. Ó được biết đến như là một phần mềm thêm vào cho dòng sản phẩm máy quét của HP. Trong giai đoạn này, nó vẫn còn rất sơ khai và chỉ được sử dụng để cải thiện chất lượng của các bài báo in. Nó được phát triển đến năm 1994 thì ngừng lại. Sau đó, sau khi được cải thiện độ chính xác, nó được HP đưa vào cuộc kiểm tra thường niên về độ chính xác của các công cụ OCR và nó đã thể hiện được sự vượt trội của mình. Kể từ năm 2006 nó đã được cải thiện rộng rãi bởi Google. Nó được hoạt động trên Linux, Windows, và Mac OSX.

Cấu trúc của Tesseract: đầu vào sẽ là hình ảnh màu hoặc đen trắng (bây giờ còn có cả file pdf).



Hình 2.1: Cấu trúc Tesseract

Vậy với cơ chế như nào mà Tesseract có thể mang đến sự hiệu quả cũng như được sử dụng khá nhiều trong việc nhận dạng ký tự như hiện nay. Về cơ bản, quá trình nhận diện sẽ diễn ra từng bước trải qua bốn bước chính như phân tích layout, tìm kiếm dòng, tìm kiếm ký tự, nhận diện ký tự và chỉnh sửa kết quả.

Trước tiên, hình ảnh sẽ được phân tích để tìm ra các vùng kết nối. Bước này cho phép OCR dễ dàng nhận biết những vùng ký tự ngược để có thể nhận diện những ký tự bên trong. Thì trong Tesseract những vùng chứa ký tự này được gọi là Blob.

Tiếp đến, những Blob này sẽ tiếp tục được phân tích ra thành từng dòng, rồi đến các ký tự. Việc tìm dòng sẽ được xử lý bởi thuật toán dựa vào vùng ký tự, cỡ chữ cùng tọa độ (trục x). Trong quá trình này, các blob cũng có thể ghép với nhau nếu OCR nhận thấy chúng chứa các ký tự trong cùng một dòng. Những blob được ghép phải trùng ít nhất 50% theo chiều ngang. Sau đó các đường cơ sở (baseline) cũng được tìm kiếm nhờ vào việc quét các dòng đã được xác định.

Sau khi đã xác định được các dòng ký tự cùng các đối số tương ứng, dòng ký tự sẽ được chia nhỏ thành các từ dựa vào các ký tự phân cách. Lúc này văn bản cố định sẽ được chia nhỏ và tiến hành nhận diện. Trong khi đó văn bản không cố định hoặc chưa chắc chắn

thì sẽ được chia nhỏ thành các từ dù chưa chắc chắn. Nhưng nhờ vào bước nhận diện, chúng ta sẽ thu được kết quả cuối cùng chính xác hơn.

Bước vào quá trình nhận diện, input của chúng ta sẽ được đánh giá, phân tích 2 lần. ở lân đầu tiên, OCR sẽ nhận diện ký tự với kết quả phân tích ở bước trước đó. Các kết quả nhận diện thỏa mãn yêu cầu được đưa vào tập huấn luyện để hỗ trợ cho quá trình nhận diện lần thứ hai với các kết quả chưa đạt yêu cầu. Đương nhiên, việc xác nhận kết quả có thỏa mãn hay không cần phải dựa trên nhiều tiêu chí vì nhận diện nội dung trải qua một quá trình lặp đi lặp lại gồm các bước nhận diện ký tự, ghép ký tự và so khớp với từ điển. Các tiêu chí đó bao gồm khoảng cách của các ký tự, độ phù hợp với từ điển và khoảng cách đến các dấu câu.

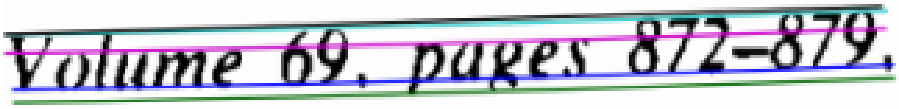
Cụ thể hơn về việc xác định dòng và từ:

Xác định dòng: mục đích của bước này là nhận dạng các dòng của hình ảnh bị nghiêng, giúp nó giảm bớt sự mất thông tin khi nhận dạng ảnh nghiêng. Các bộ phận quan trọng của quá trình này là lọc lấy dãy màu (blobs) và xây dựng dòng. Bước này cũng giúp loại bỏ các văn bản có drop – cap.

Thiết lập dòng:

Khi dòng văn bản được tìm thấy, các dòng cơ sở được thiết lập chính xác hơn bằng cách sử dụng một đường có tên là spline toàn phương (là dòng mà được kết hợp từ nhiều đoạn). Nó giúp cho Tesseract xử lý các trang có đường cơ sở là đường cong.

Các dòng cơ sở được thiết lập bằng cách phân vùng các blobs thành các nhóm cơ thể thay thế thích hợp liên tục trong đường cơ sở thẳng ban đầu. Một spline toàn phương được thiết lập cho phân vùng dày đặc nhất, Spline có lợi thế là tính toán ổn định, nhược điểm là sự gián đoạn có thể xảy ra khi nhiều phân đoạn spline được yêu cầu.



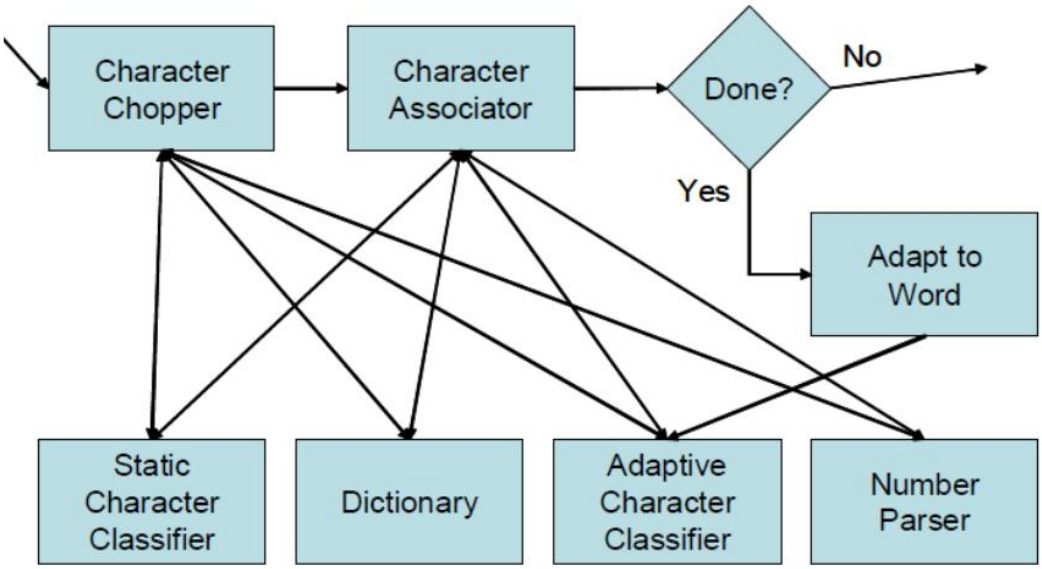
Hình 2.2: Ví dụ về một đường cơ sở dạng cong

Cắt từ nhỏ: Tesseract sẽ xác định xem có các dạng ký tự dính với nhau trong một từ hay không. Nếu có nó sẽ cắt nhỏ các ký tự ra thành các ký tự riêng lẻ.

Hình 2.3: Ví dụ về cắt các ký tự bị dính

Nhận dạng khoảng cách giữa các chữ hoặc số: Xác định khoảng cách giữa các số hoặc giữa các chữ là một vấn đề khá phức tạp. Tesseract giải quyết những vấn đề này bằng cách đo khoảng cách trong một phạm vị hạn chế theo chiều dọc giữa các dòng cơ sở và các dòng trung bình.

Nhận dạng từ: Qúa trình nhận dạng một từ là quá trình phân tích một từ được chia ra thành các ký tự như thế nào.



Hình 2.4: Quy trinh nhận diện từ của Tesseract

Khi kết quả xuất ra một từ nó không thỏa mãn nhu cầu thì Tesseract cố gắng cải thiện kết quả này bằng cách cắt nhỏ các từ có nghĩa không tốt nhất. Nếu việc cắt nhỏ không làm tăng chất lượng từ thì nó sẽ phục hồi lại từ trước đó.

2.6.2. Ngôn ngữ NodeJS

Hình 2.5: Hình ảnh ngôn ngữ nodejs

2.6.2.1. Nodejs là gì?

Nodejs là một nền tảng được xây dựng, vận hành tại V8 JavaScript runtime của Chrome. Với Nodejs bạn có thể chạy JavaScript trên server và có thể xây dựng và phát triển các ứng dụng mạng nhanh chóng và dễ dàng.

Nền tảng Nodejs bắt đầu được xây dựng, phát triển tại California từ năm 2009 với phần Core phía dưới được lập trình bằng C++ gần như 100%. Điều này tạo ưu thế về tốc độ xử lý cũng như hiệu năng của nền tảng này. Đến nay Nodejs vẫn đang gây bão trong cộng đồng công nghệ bởi khả năng phát triển ứng dụng vượt trội.

2.6.2.2. Đặc điểm của Nodejs

Không đồng bộ: đặc điểm đầu tiên của Nodejs là tính bất đồng bộ. Nodejs không cần đợi API trả dữ liệu về, vậy nên mọi APIs nằm trong thư viện Nodejs đều không được đồng bộ, hiểu đơn giản là chúng không hề blocking (khóa). Server có cơ chế riêng để gửi thông báo và nhận phản hồi về các hoạt động của Nodejs và API đã gọi.

Tốc độ nhanh: về phần core phía dưới lập trình gần như toàn bộ bằng ngôn ngữ C++ kết hợp với V8 Javascript Engine mà Google Chrome cung cấp, tốc độ vận hành, thực hiện code của thư viện Node.js diễn ra rất nhanh.

Đơn giản – hiệu quả: tiến trình vận hành của Node.js đơn giản song lại mang đến hiệu năng cao nhờ ứng dụng mô hình single thread và các sự kiện lặp. Một loạt cơ chế sự

kiện cho phép server trả về phản hồi bằng cách không block, đồng thời tăng hiệu quả sử dụng. Các luồng đơn cung cấp dịch vụ cho nhiều request hơn hẳn server truyền thống.

Không đệm: Nền tảng Nodejs không có vùng đệm, tức là không cung cấp khả năng lưu trữ dữ liệu buffer.

2.6.2.3. Lý do nên sử dụng Nodejs

Nodejs được nhiều lập trình viên, nhà phát triển sử dụng trong thiết kế web hay phát triển ứng dụng bởi một số lý do sau:

Ứng dụng Nodejs phần đông đều được viết bằng ngôn ngữ lập trình javascript – một ngôn ngữ thông dụng được sử dụng rộng rãi và chạy trên nhiều trình duyệt, nền tảng và hệ điều hành khác nhau.

Nodejs khá nhẹ nhưng lại hiệu quả nhờ vào cơ chế non – blocking I/O, chạy đa nền tảng trên Server và dùng Event – driven.

Tương thích với nhiều thiết bị. Bạn có thể chạy các ứng dụng phát triển bởi Nodejs trên bất cứ thiết bị nào, dù là Mac, Window, Linux, …

Cộng đồng Nodejs khá lớn và được cung cấp miễn phí cho người dùng.

Nodejs có tốc độ cực kỳ nhanh, xử lý được nhu cầu sử dụng của lượng khách truy cập "khổng lồ" trong thời gian cực ngắn.

ứng dụng được phát triển bởi Node.js có khả năng xử lý nhiều yêu cầu truy cập cùng một lúc, "cứu" website của bạn khỏi nguy cơ bị "sập" khi lượng truy cập quá nhiều.

2.6.3. MongoDB

MongoDB là hệ cơ sở dữ liệu mã nguồn mở, là cơ sở dữ liệu phi quan hệ hay được gọi là NoSQL (None – Relationship SQL hay còn được gọi là Not Only SQL).

NoSQL được phát triển trên JavaScript Framework với kiểu dữ liệu là JSON và dạng dữ liệu kiểu key và value.

NoSQL ra đời với sự bổ sung những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS (Relational Database Management System).

Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc.

NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng. NoSQL được sử dụng ở rất nhiều công ty hay các tập đoàn lớn. Ví dụ như Facebook sử dụng Cassandra do Facebook phát triển, Google phát triển và sử dụng BigTable.

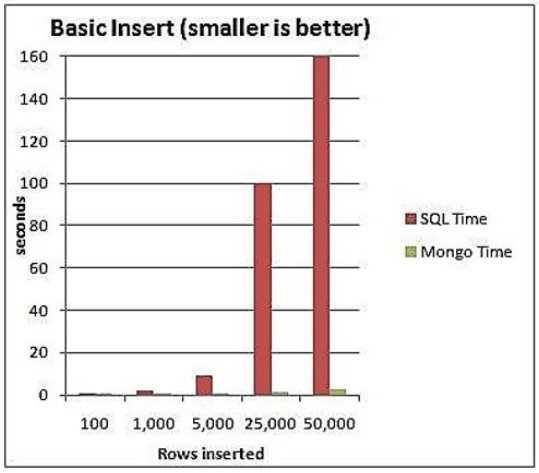
MongoDB là một database hướng tài liệu, các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như cơ sở dữ liệu quan hệ nên truy vấn rất nhanh.

Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở quan hệ như MySQL, SQL Server sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta dùng khái niệm collection thay vì bảng. So với RDBMS thì trong MongoDB collection ứng với table, còn document ứng với row.

Các collection trong MongoDB đươc cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định. Thông tin liên quan được lưu trữ cùng nhau để truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB. Chính vì vậy việc chèn thêm dữ liệu vào không bị hạn chế.

Dữ liệu trong MongoDB không ràng buộc. Do đó khi bổ sung hoặc xóa hoặc cập nhật sẽ không phải mất thời gian để kiểm tra xem có thỏa mãn các ràng buộc hay không.

Trường dữ liệu "_id" luôn được đánh chỉ mục tự động để tăng tốc độ truy vấn thông tin để đạt hiệu suất cao nhất.



Hình 2.6: So sánh thời gian chèn dữ liệu của MongoDB với SQL Server

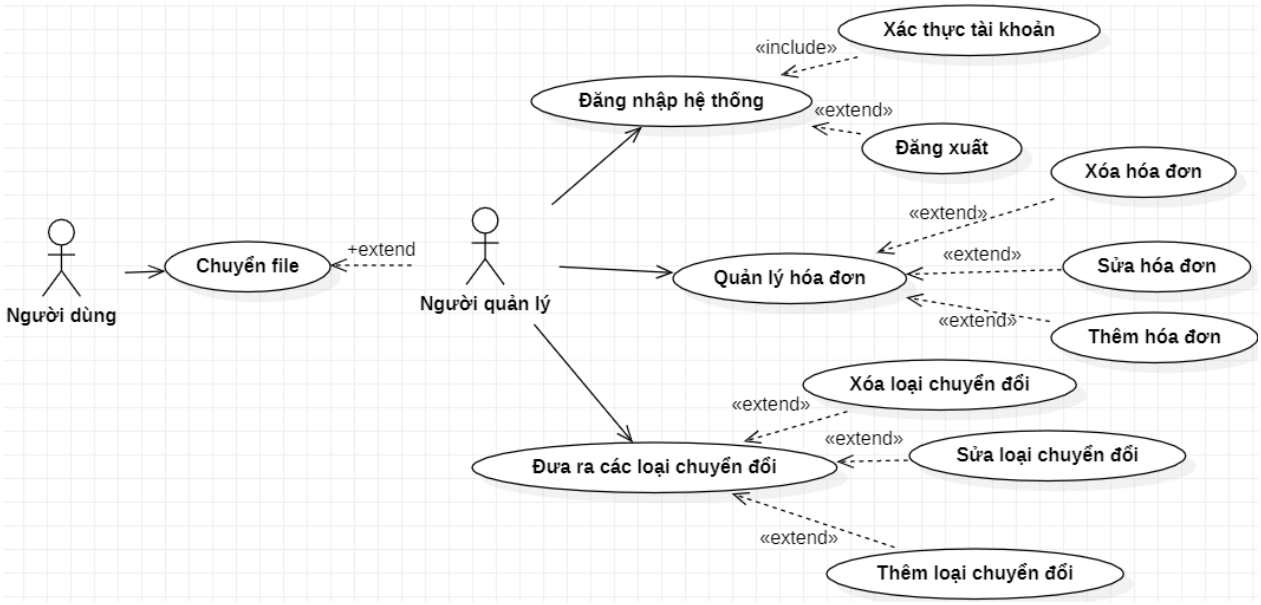
Ngoài những ưu điểm đã được kể ở trên thì MongoDB còn có một số nhược điểm khác như:

Do không có tính ràng buộc giống như RDBMS nên khi thao tác với MongoDB thì chuyên gia phải tự xử lý các mối quan hệ giữa các dữ liệu. Bộ nhớ lưu trữ bị tăng do dữ liệu lưu trữ dưới dạng key – value, các bộ dữ liệu chỉ khác về giá trị do đó các khóa sẽ bị lặp lại. MongoDB cũng không hỗ trợ liên kết nên dữ liệu bị dư thừa.

Một điều nữa là khi insert – update – remove bản ghi, Mongo sẽ chưa cập nhật ngay xuống ổ cứng mà sau đó khoảng thời gian 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống thiết bị lưu trữ điều này sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện, …

CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

3.1. Sơ đồ use case tổng quát của hệ thống

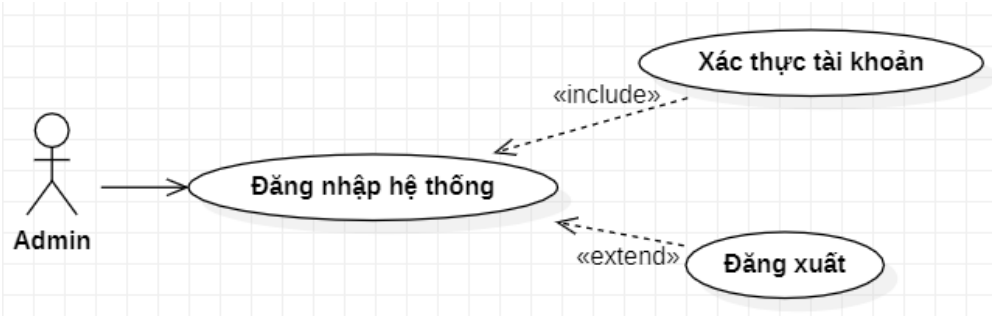


Hình 3.1: Biểu đồ use case tổng quát của hệ thống

3.2. Phân tích, thiết kế từng chức năng của hệ thống

3.2.1. Chức năng đăng nhập

3.2.1.1. Biểu đồ use case chức năng đăng nhập



Hình 3.2: Biểu đồ use case quản lý đăng nhập

Đặc tả use case đăng nhập

- **Tác nhân:** người dùng hoặc admin.
- **Mô tả:** tác nhân sử dụng use case để thực hiện chức năng đăng nhập hệ thống.

- **Dòng sự kiện chính:**

  1, Tác nhân yêu cầu giao diện đăng nhập tới hệ thống.

  2, Hệ thống sẽ hiển thị giao diện đăng nhập cho tác nhân.

  3, Tác nhân sẽ cập nhật:

  - Cập nhật tên đăng nhập hoặc Email.
  - Cập nhật mật khẩu đăng nhập.

  4, Hệ thống sẽ kiểm tra dữ liệu và xác nhận thông tin được nhập vào

  5, Khi thành công hệ thống sẽ hiển thị giao diện chính của phần mềm tùy vào từng chức năng của tác nhân.

  6, Kết thúc use case.

- **Dòng sự kiện phụ:**
  - **Dòng sự kiện phụ thứ nhất:**

    1, Tác nhân nhập sai thông tin đăng nhập.

    2, Hệ thống sẽ hiển thị thông báo lỗi.

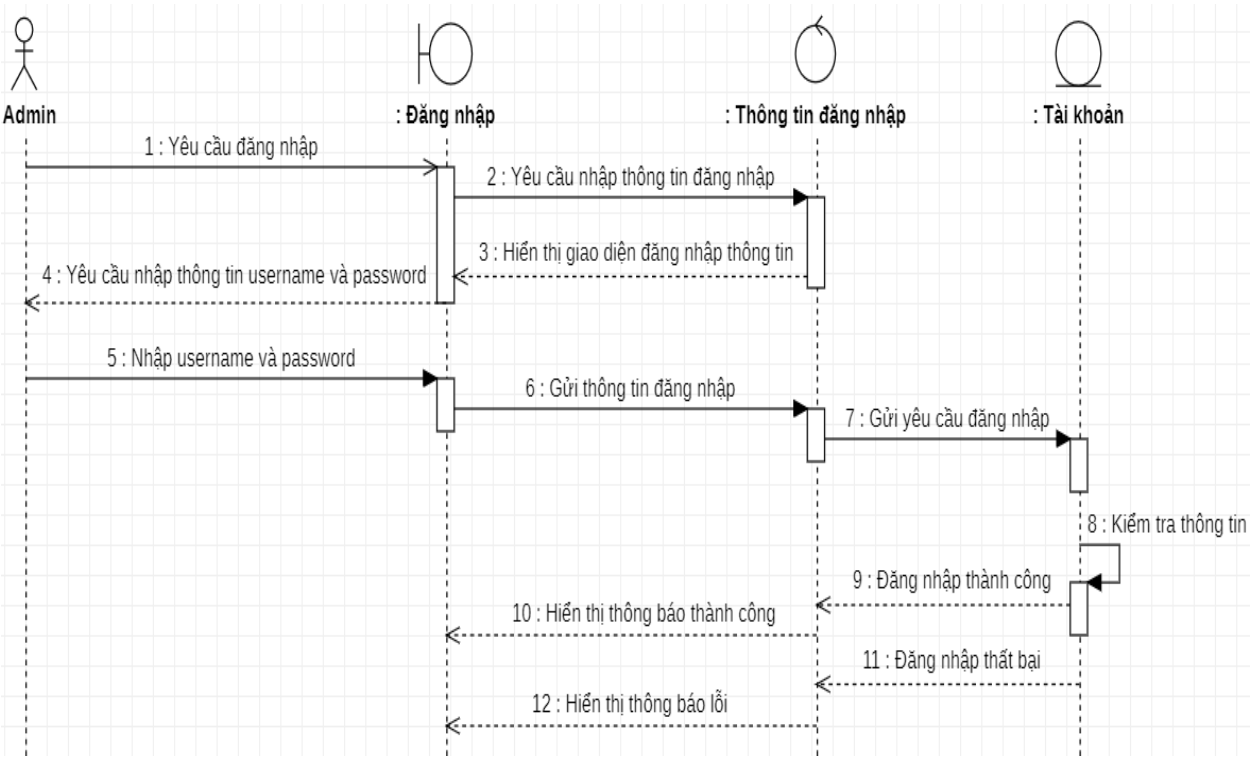
    3, Kết thúc use case.
  - **Dòng sự kiện phụ thứ hai:**

    1, Tác nhân không nhập đủ thông tin cần đăng nhập.

    2, Hệ thống sẽ hiển thị dòng chữ báo lỗi cho tác nhân nhìn thấy.

    3, Kết thúc use case.
- **Các yêu cầu đặc biệt:** Không có
- **Trạng thái hệ thống trước khi use case sử dụng:** không đòi hỏi gì trước đó.
- **Trạng thái hệ thống sau khi sử dụng use case:**
  - **Nếu thành công:** hệ thống hiển thị giao diện tùy vào quyền hạn của tác nhân.
  - **Nếu thất bại:** Hệ thống sẽ đưa ra thông báo lỗi.

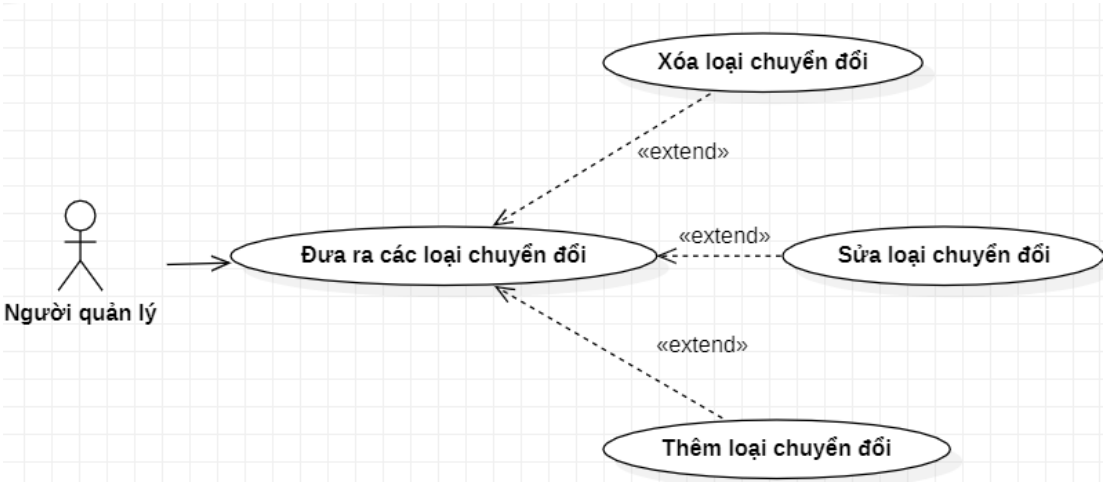
3.2.1.2. Biểu đồ tuần tự chức năng đăng nhập



Hình 3.3: Biểu đồ tuần tự chức năng đăng nhập

3.2.2. Chức năng câu hỏi

3.2.2.1. Biểu đồ use case chức năng đưa ra các trình tự chuyển đổi



Hình 3.4: Biểu đồ use case chức năng đưa ra các trình tự chuyển đổi

Đặc tả use case chức năng đưa ra các trình tự chuyển đổi

- **Tác nhân**: người dùng hoặc admin.

- **Mô tả:** tác nhân sử dụng use case để đưa ra các trình tự chuyển đổi.
- **Dòng sự kiện chính:**

  1, Tác nhân yêu cầu cập nhật thông tin các loại chuyển đổi tới hệ thống.

  2, Hệ thống sẽ hiển thị giao diện cập nhật cho tác nhân.

  3, Tác nhân sẽ cập nhật thông tin bằng cách chọn các chức năng như thêm cách chuyển đổi, xóa chuyển đổi đang có, sửa loại chuyển đổi.

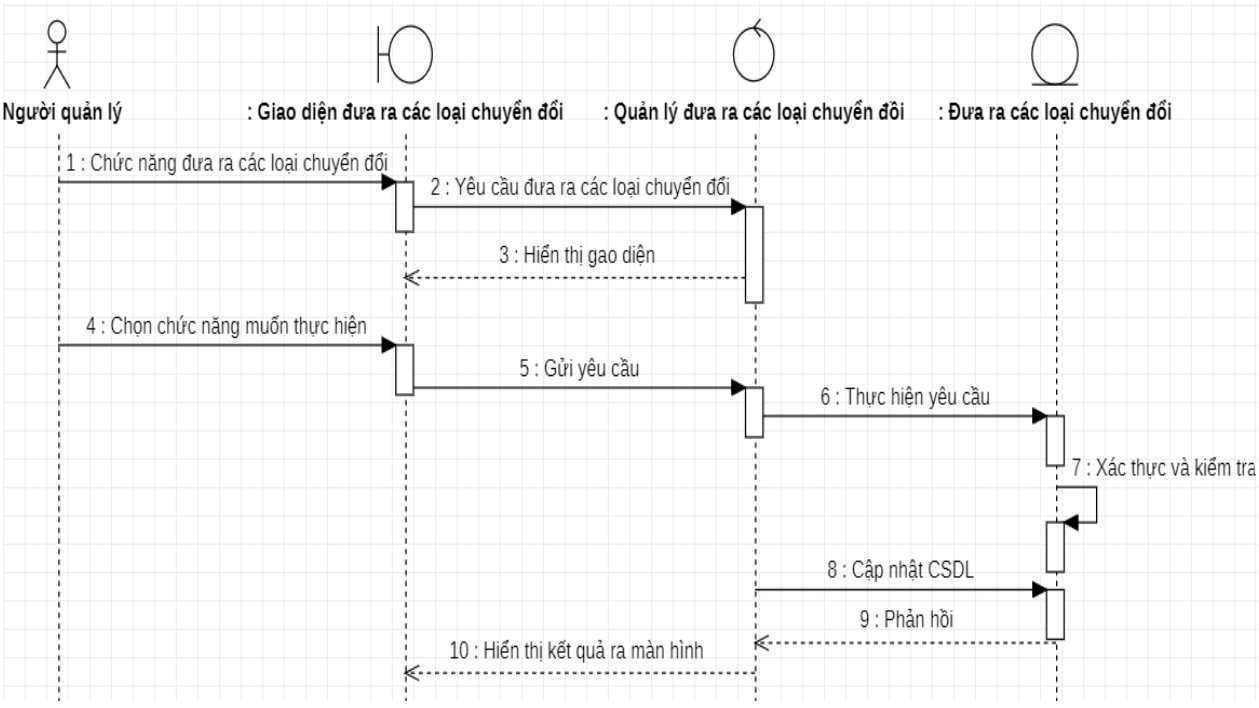
  4, Hệ thống xác nhận và kiểm tra việc cập nhật của tác nhân.

  5, Cập nhật vào cơ sở dữ liệu và sau đó hiển thị thông tin phản hồi cho tác nhân.

  6. Hiển thị kết quả sau khi cập nhật ra màn hình.

  7, Kết thúc use case.
- **Trạng thái hệ thống trước khi use case sử dụng:** không đòi hỏi gì trước đó.
- **Trạng thái hệ thống sau khi sử dụng use case:**
  - **Nếu thành công:** Hệ thống sẽ hiển thị kết quả sau khi cập nhật ra màn hình.
  - **Nếu thất bại:** Hệ thống sẽ đưa ra thông báo lỗi.

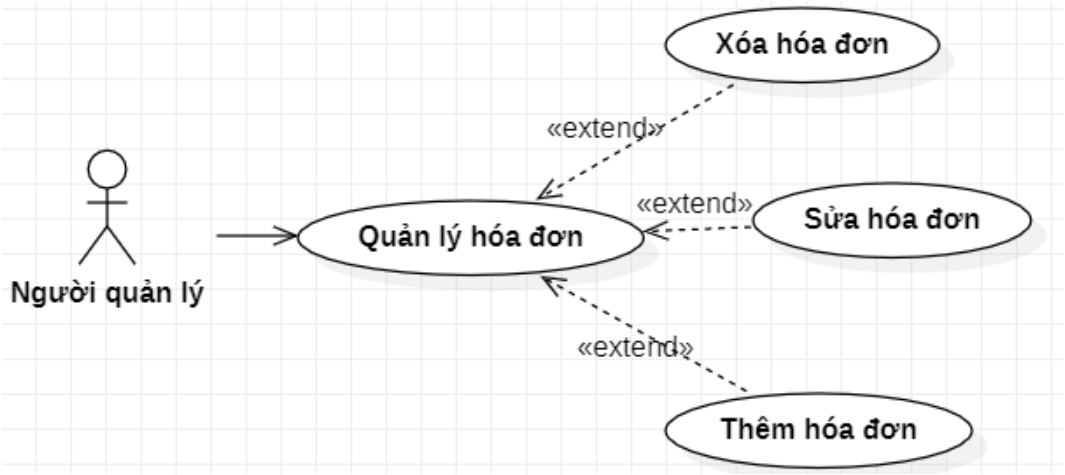
3.2.2.2. Biểu đồ trình tự chức năng đưa ra các trình tự chuyển đổi



Hình 3.5: Biểu đồ trình tự chức năng đưa ra các trình tự chuyển đổi

3.2.3. Chức năng quản lý hóa đơn

3.2.3.1. Biểu đồ use case chức năng quản lý hóa đơn



Hình 3.6: Biểu đồ use case quản lý hóa đơn

Đặc tả use case quản lý hóa đơn (đặc tả này phù hợp khi tác nhân đăng nhập thành công)

- **Tác nhân:** người quản lý.
- **Mô tả:** tác nhân sử dụng use case để thực hiện chức năng quản lý hóa đơn.
- **Dòng sự kiện chính:**

  1, Tác nhân yêu cầu giao diện quản lý hóa đơn

  2, Hệ thống sẽ hiển thị giao diện quản lý hóa đơn cho tác nhân.

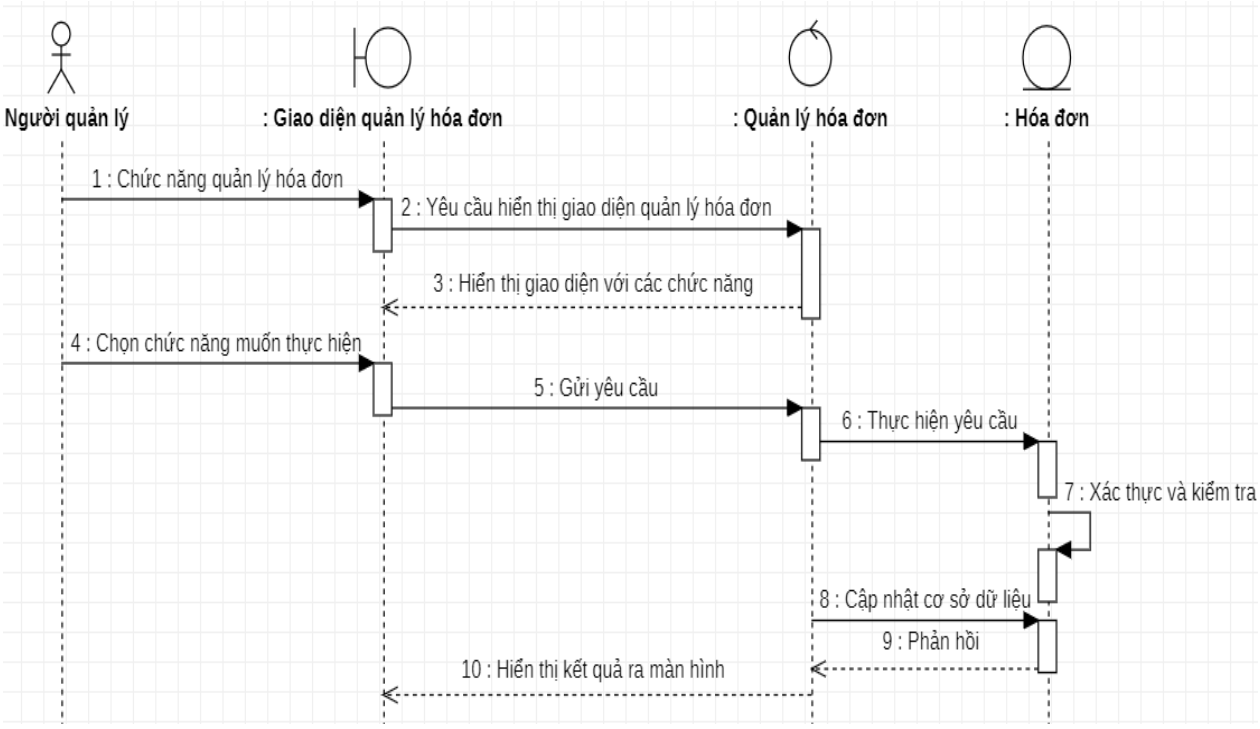
  3, Tác nhân sẽ chọn chức năng như thêm, sửa, xóa hóa đơn mà tác nhân muốn.

  4, Sau khi thực hiện các chức năng mong muốn của tác nhân nếu thành công hệ thống sẽ thống báo kết quả cho tác nhân.

  5, Kết thúc use case.

- **Trạng thái hệ thống trước khi use case sử dụng:** không đòi hỏi gì trước đó.
- **Trạng thái hệ thống sau khi sử dụng use case:**
  - **Nếu thành công:** hệ thống hiển thị giao diện tùy vào quyền hạn của tác nhân.
  - **Nếu thất bại:** Hệ thống sẽ đưa ra thông báo lỗi.

3.2.3.2. Biểu đồ trình tự chức năng quản lý hóa đơn



Hình 3.7: Biểu đồ trình tự quản lý hóa đơn

3.3. Cấu trúc bảng và kiểu dữ liệu thuộc tính

Collection accounts	
Field Name	Data Type
_id	ObjectId
username	String
avatar	String
email	String
password	String
status	String
birth	String
createAt	Date
UpdateAt	Date

Bảng 3.1: Collection account

Collection bills	
Field Name	Data Type
_id	ObjectId
userId	ObjectId
priceId	Object
totalPrice	Int32
day	String
status	String
birth	String
createAt	Date
UpdateAt	Date

Bảng 3.2: Collection bills

Collection converts	
Field Name	Data Type
_id	ObjectId
name	String
avatar	String
description	String
rate	Int32
web	String
status	String
totalFile	String
createAt	Date
UpdateAt	Date

Bảng 3.3: collection converts

Collection users	
Field Name	Data Type
_id	ObjectId
username	String
email	String
password	String
dayBuy	String
dayEnd	String
convertCount	Int32
totalFileConvert	Int32
createAt	Date
UpdateAt	Date

Bảng 3.4: collection users

Collection prices	
Field Name	Data Type
_id	ObjectId
name	String
ConvertCount	Int32
description	String
price	Int32
status	String
createAt	Date
UpdateAt	Date

Bảng 3.5: collection prices

Collection images	
Field Name	Data Type
_id	ObjectId
name	String
img	Array
size	String
numfile	String
rate	Null
comment	Null
user_id	ObjectId
isUserSet	Bool
day	String
type	String
createAt	Date
UpdateAt	Date

Hình 3.6: collection images

Collection reviews	
Field Name	Data Type
_id	ObjectId
convertId	ObjectId
rate	Int32
comment	String
day	String
createAt	Date
UpdateAt	Date

Bảng 3.7: collection reviews

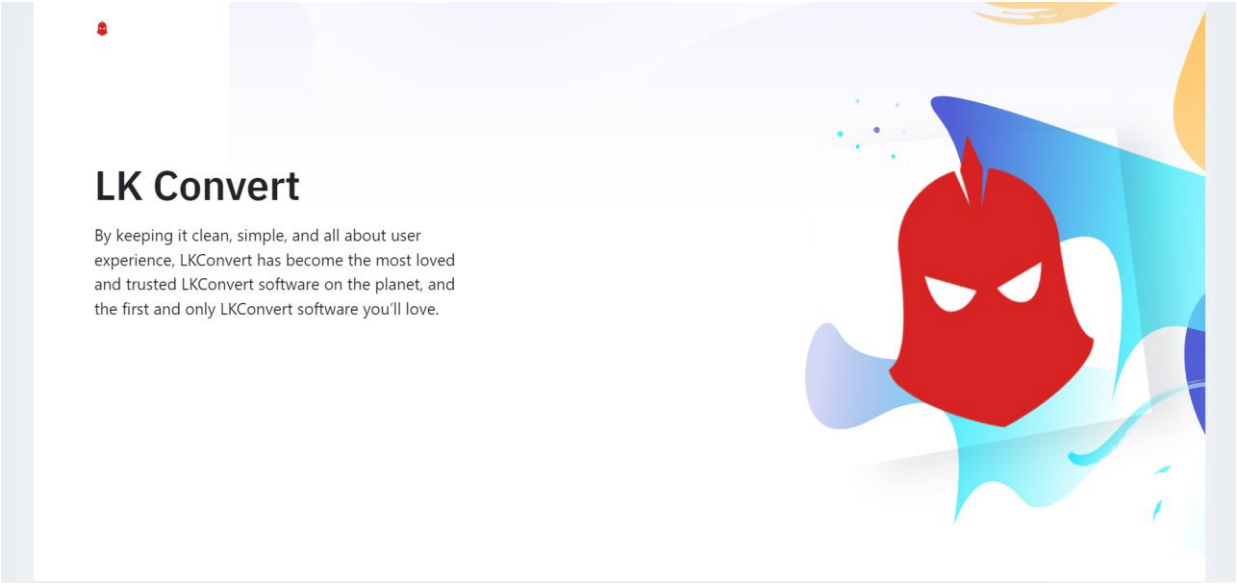
Collection reports	
Field Name	Data Type
_id	ObjectId
userId	ObjectId
comment	String
day	String
createAt	Date
UpdateAt	Date

Bảng 3.8: collection reports

CHƯƠNG 4: THIẾT KẾ GIAO DIỆN CHƯƠNG TRÌNH

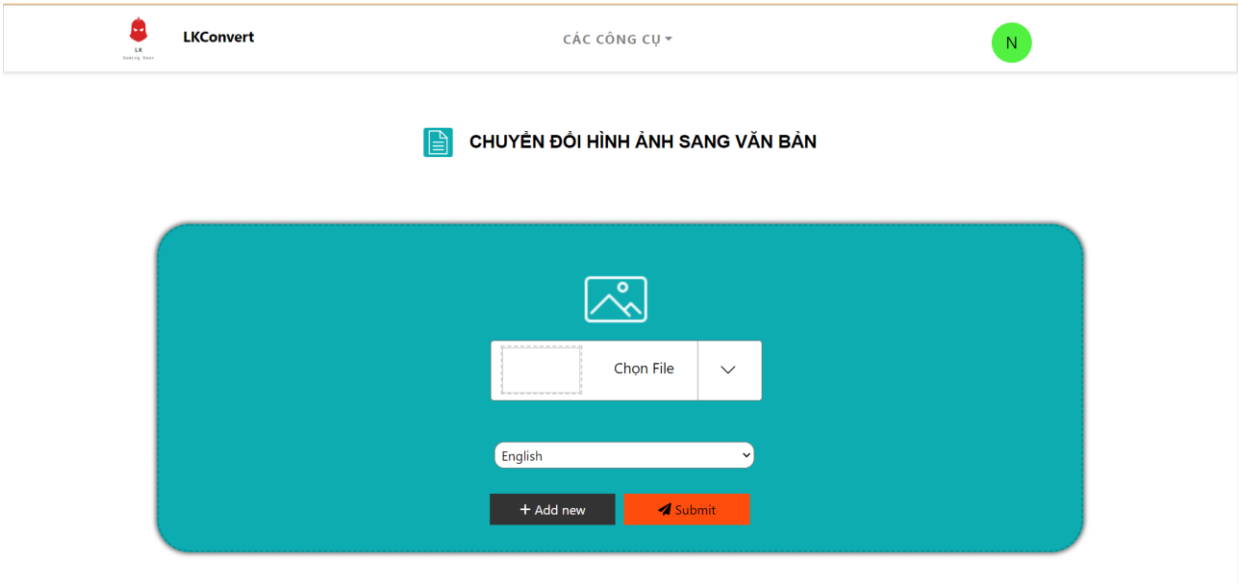
4.1. Giao diện người dùng

4.1.1. Trang chủ



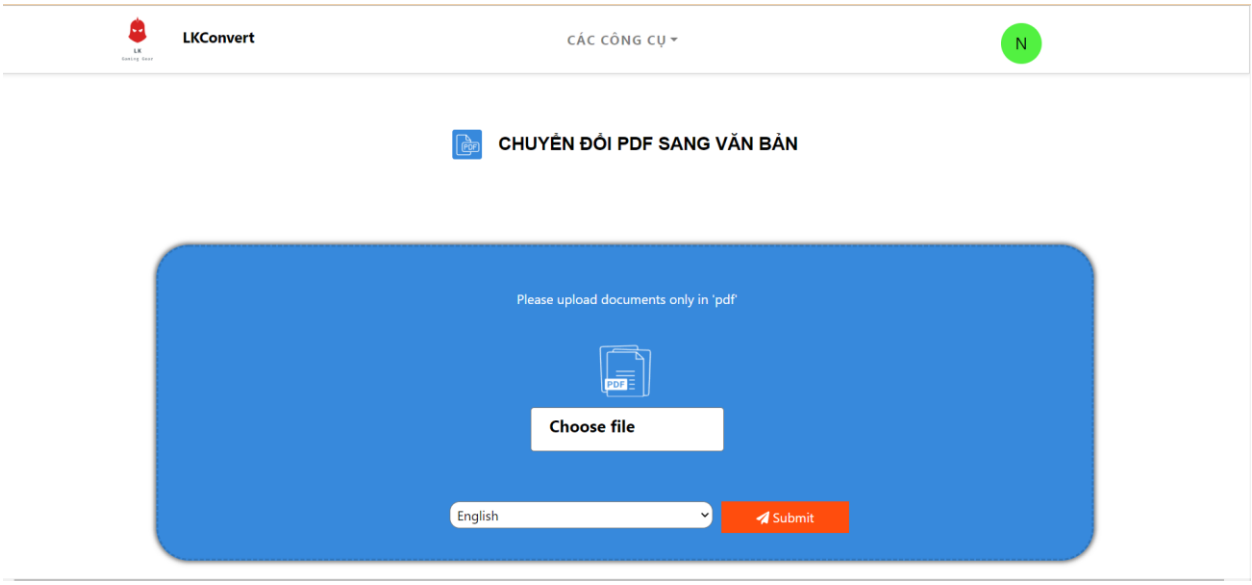
Hình 4.1: Giao diện trang chủ

4.1.2. Giao diện chuyển đổi ảnh sang văn bản



Hình 4.2: Giao diện chuyển đổi ảnh sang văn bản

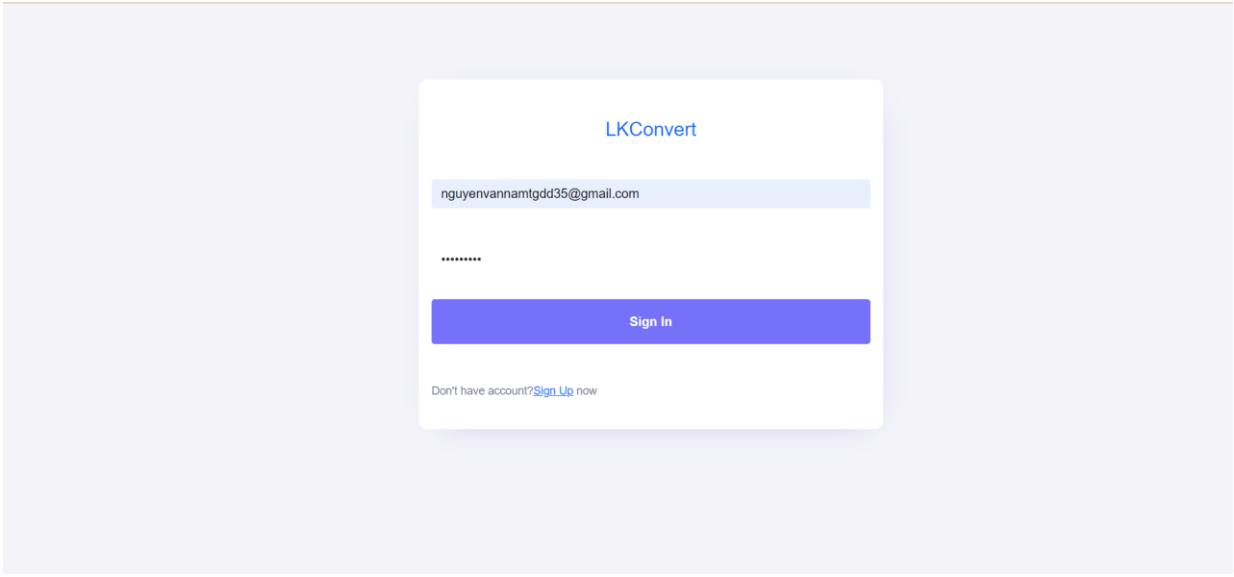
4.1.3. Giao diện chuyển đổi từ PDF sang văn bản



Hình 4.3: Giao diện chuyển đổi từ PDF sang văn bản

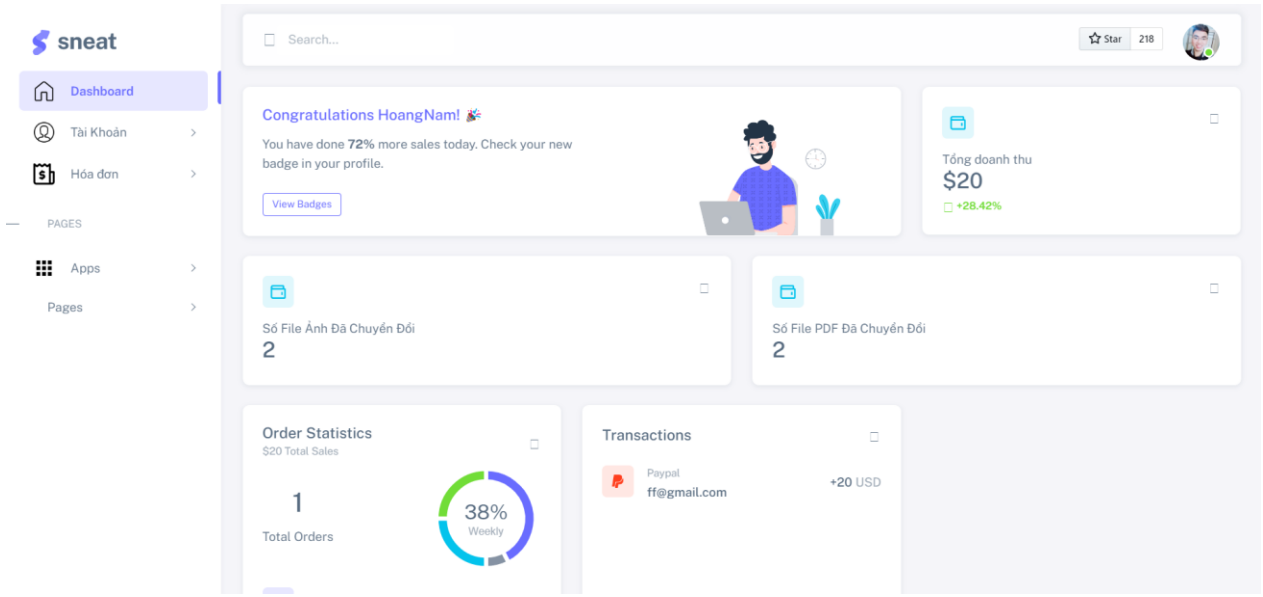
4.2. Giao diện Admin

4.2.1. Giao diện đăng nhập



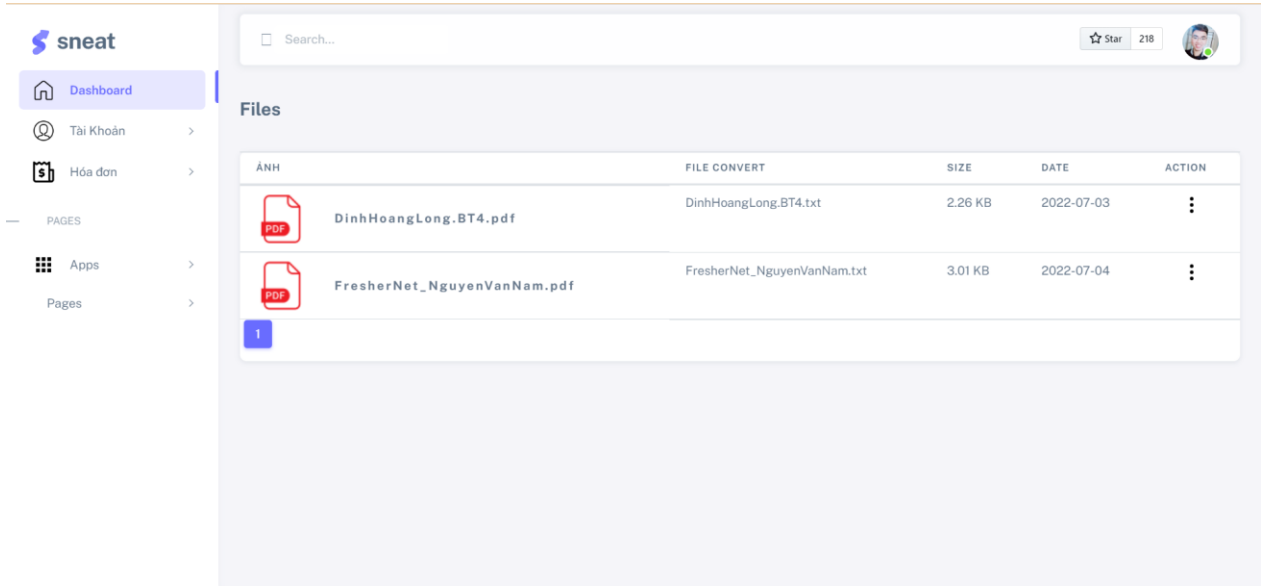
Hình 4.4: Giao diện đăng nhập

4.2.2. Giao diện Admin



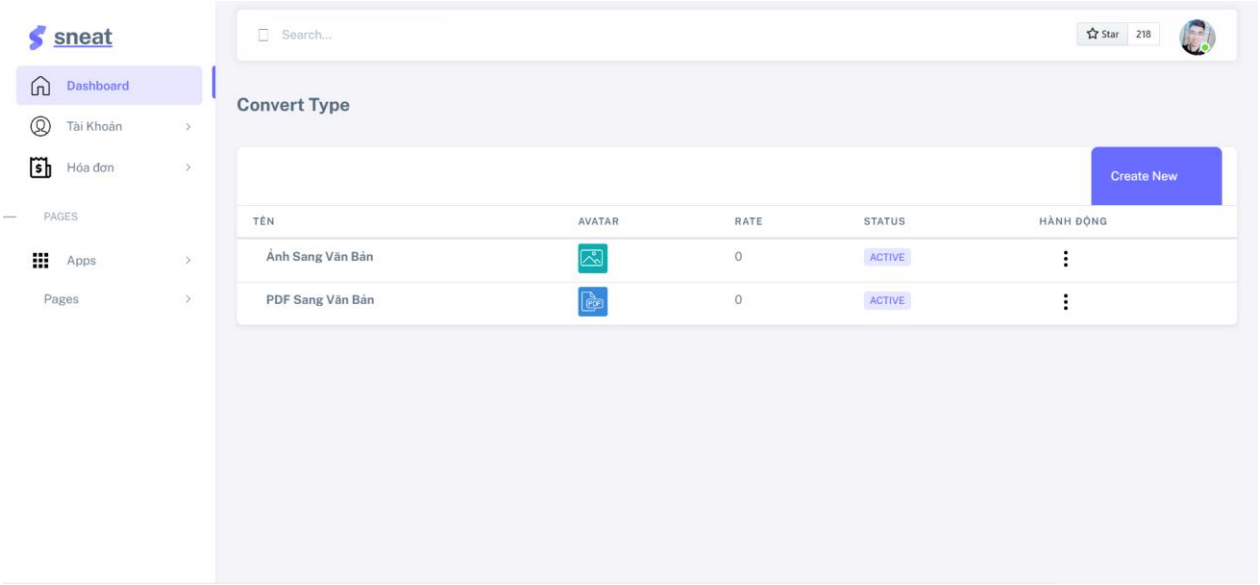
Hình 4.5: Giao diện Admin

4.2.3. Giao diện quản lý file người dùng



Hình 4.6: Giao diện quảy lý file người dùng

4.2.4. Giao diện các loại chuyển đổi



Hình 4.7: Giao diện quản lý các loại chuyển đổi

KẾT LUẬN

Qua việc nghiên cứu đề tài "**Xây dựng Web nhận dạng văn bản tiếng việt dùng Tesseract**", em cũng đã một phần nào biết thêm được các kiến thức cũng biết được cách tìm kiếm dữ liệu và triển khai đối với dữ liệu mà mình tìm được. Em cũng tổng kết được một số kết quả mà mình đạt được cũng như những công việc chưa đạt được khi làm đề tài báo cáo thực tập này.

Kết quả đạt được: Biết các tìm kiếm dữ liệu cũng như cách triển khai đối với dữ liệu đã tìm được. Thực hiện được việc nhận dạng văn bản thông qua việc chuyển đổi từ file pdf hay file ảnh sang dạng text. Đồng thời còn quản lý được số lượng file, …

Trong quá trình nghiên cứu với đề tài báo cáo thực tập này vẫn còn một số sai sót do kiến thức còn hạn hẹp nhưng em đã cố gắng hết sức có thể để tạo ra một chương trình hoàn chỉnh và đẹp nhất để đưa tới thầy cô. Mong rằng sẽ được sự góp ý, chia sẻ từ các thầy cô để em có thể hoàn thiện chương trình một cách tốt nhất, chỉnh chu nhất.

Một lần nữa, em cảm ơn thầy **Hoàng Thanh Tùng**, người đã tận tình hỗ trợ, giúp đỡ em trong suốt quá trình làm báo cáo thực tập.

TÀI LIỆU THAM KHẢO

[1] https://github.com/tesseract-ocr/tessdata

[2] https://tech.miichisoft.net/ocr-tesseract-js/

[3] https://en.wikipedia.org/wiki/Tesseract_(software)

[4] Video Node.js Express Project to Extract Text From Image Using Tesseract OCR Library in Browser Using JS