**TRƯỜNG ĐẠI HỌC VINH**

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**

****

**ĐỒ ÁN HỌC PHẦN  
CÔNG NGHỆ PHẦN MỀM**

**XÂY DỰNG WEBSITE**

**VẬN HÀNH VÀ QUẢN LÝ QUY TRÌNH**

**THI TRẮC NGHIỆM KHÁCH QUAN**

**NHÓM: 05**

|  |  |
| --- | --- |
| GVHD: | TS. Cao Thanh Sơn |
| SVTH: | Đặng Ngọc Anh, 215748020110333 (NT) |
|  | Thái Văn Tuấn, 215748020110086 |
|  |
|  |

**Nghệ An, 11/202****4**

MỞ ĐẦU

Với sự phát triển không ngừng của ngành công nghệ thông tin, trí tuệ nhân tạo hay cụ thể hơn là học máy thì các hệ thống quản lý thông minh đã và đang góp phần tối ưu hóa quy trình làm việc, nâng cao hiệu quả và đảm bảo tính minh bạch trong mọi hoạt động. Trong đó việc áp dụng các giải pháp công nghệ phần mềm vào quản lý và vận hành quy trình của giáo dục ngày càng trở nên cần thiết. Hiểu rõ tầm quan trọng của việc ứng dụng kiến thức vào thực tiễn, chúng tôi đã lựa chọn triển khai thực hiện đồ án với đề tài "*Xây dựng Website vận hành và quản lý quy trình thi trắc nghiệm khách quan*" nhằm đưa ra một giải pháp hỗ trợ tốt hơn cho quá trình tổ chức, vận hành, quản lý và giám sát các kỳ thi trắc nghiệm trong môi trường giáo dục bậc đại học.

Qua đồ án này, chúng tôi tập trung xây dựng một hệ thống Website hướng đến việc tự động hoá toàn bộ quy trình tổ chức thi với các chức năng nổi bật như quản lý thông tin sinh viên, tổ chức thi trắc nghiệm, giám sát, và lưu trữ dữ liệu kết quả một cách tự động và hiệu quả. Đặc biệt, hệ thống sử dụng công nghệ thị giác máy tính để xác minh danh tính thí sinh, giám sát quá trình thi, nâng cao tính khách quan, tính chính xác và tính bảo mật thông tin.

Đồ án sử dụng ngôn ngữ lập trình Java với framework Spring Boot cho phần xử lý backend, framework FastAPI của Python để xử lý hình ảnh, hệ quản trị cơ sở dữ liệu MongoDB và cuối cùng là giao diện người dùng bằng HTML, CSS, và JavaScript. Việc tiếp cận với các framework và công nghệ đang được ứng dụng rộng rãi trong thực tế là bài học rất lớn không chỉ giúp ôn lại kiến thức đã học mà còn là cái nhìn tổng quát hơn cho định hướng lựa chọn công việc sau này.

Tuy nhiên chúng tôi nhận thức rõ rằng để xây dựng một ứng dụng Website hoàn chỉnh cần có nhiều kinh nghiệm, sự nỗ lực, tính kỷ luật cao, và khả năng giải quyết các thách thức phát sinh. Do đó sản phẩm hoàn thiện không thể tránh khỏi nhiều thiếu sót, vì vậy nhóm mong nhận được sự đóng góp ý kiến của các thầy cô để có thể nhận ra, khắc phục và cải thiện hơn cho sản phẩm.

LỜI CẢM ƠN

Trước tiên, chúng tôi xin gửi lời cảm ơn sâu sắc và chân thành nhất đến TS. Cao Thanh Sơn, người đã tận tình hướng dẫn, đồng hành và hỗ trợ chúng tôi xuyên suốt quá trình thực hiện đồ án học phần Công nghệ phần mềm. Trong suốt thời gian qua, thầy không chỉ truyền đạt những kiến thức nền tảng vững chắc mà còn chia sẻ nhiều chỉ dẫn thực tế quý báu, giúp chúng tôi có thể áp dụng lý thuyết vào thực tiễn một cách hiệu quả. Sự tận tâm và những góp ý chân thành của thầy đã giúp chúng tôi hoàn thành đồ án này, đồng thời mang lại cho chúng tôi nhiều bài học bổ ích và kinh nghiệm quý giá cho chặng đường nghề nghiệp sau này.

Chúng tôi cũng xin bày tỏ lòng biết ơn sâu sắc đến toàn thể các thầy cô trong Viện Kỹ thuật Công nghệ, những người đã luôn tận tình giảng dạy, trang bị cho chúng tôi một nền tảng kiến thức vững chắc trong lĩnh vực công nghệ thông tin. Sự nhiệt huyết trong giảng dạy và tinh thần hỗ trợ của các thầy cô không chỉ giúp chúng tôi hiểu rõ các vấn đề học thuật mà còn khuyến khích chúng tôi sáng tạo và phát triển tư duy phản biện, tạo động lực mạnh mẽ để chúng tôi vững bước trên hành trình học tập và nghiên cứu.

Chúng tôi cũng không thể không nhắc đến sự đồng hành và nỗ lực của các thành viên trong nhóm, những người đã cùng nhau vượt qua những khó khăn và thử thách trong suốt quá trình làm việc. Đặc biệt, chúng tôi xin gửi lời cảm ơn sâu sắc đến bạn Thái Văn Tuấn, người đã luôn sát cánh, đóng góp ý kiến và hỗ trợ nhóm với tinh thần hợp tác, trách nhiệm và đoàn kết. Sự hỗ trợ lẫn nhau, sự kiên trì và nhiệt huyết của tất cả thành viên đã giúp chúng tôi hoàn thành đồ án này một cách tốt nhất.

Cuối cùng, chúng tôi chân thành mong muốn nhận được những góp ý, nhận xét từ quý thầy cô và các bạn để đồ án của chúng tôi ngày càng hoàn thiện và có giá trị thực tiễn hơn. Những ý kiến đóng góp ấy sẽ là nguồn động lực to lớn để chúng tôi tiếp tục phấn đấu và học hỏi trong tương lai.

Trân trọng cảm ơn,

*Đặng Ngọc Anh*,

*Thái Văn Tuấn.*

DANH MỤC CÁC HÌNH

[Hình 2.1. Biểu đồ ca sử dụng 14](#_Toc183516809)

[Hình 3.x: Giao diện sau khi hoàn thiện 31](#_Toc183516810)

[Hình 3.x: Test case đầu vào thẻ sinh viên 32](#_Toc183516811)

[Hình 3.x: Kết quả sau khi xử lý 33](#_Toc183516812)

[Hình 3.x: Tìm phòng thi thông qua mã phòng 34](#_Toc183516813)

[Hình 3.x: Danh sách sinh viên có trong phòng thi 34](#_Toc183516814)

[Hình 3.x: Nhập danh sách vào hệ thống đối chiếu khuôn mặt để giám thị kiểm tra 35](#_Toc183516815)

DANH MỤC CÁC BẢNG

[Bảng 2.x: Bảng chức năng xử lý đầu vào và trích xuất thông tin trên thẻ sinh viên 14](#_Toc182908732)

[Bảng 2.x. Bảng chức năng đối chiếu với danh sách sinh viên trong phòng thi 17](#_Toc182908733)

[Bảng 2.x: Collection lưu trữ thông tin sinh viên 21](#_Toc182908734)

[Bảng 2.x: Collection lưu trữ thông tin giám thị 21](#_Toc182908735)

[Bảng 2.x: Collection lưu trữ thông tin kết quả kiểm tra trước khi thi 22](#_Toc182908736)

[Bảng 2.x: Collection lưu trữ thông tin phòng thi 22](#_Toc182908737)

MỤC LỤC

[LỜI MỞ ĐẦU ii](#_Toc184670211)

[LỜI CẢM ƠN iii](#_Toc184670212)

[DANH MỤC CÁC HÌNH iv](#_Toc184670213)

[DANH MỤC CÁC BẢNG v](#_Toc184670214)

[MỤC LỤC 1](#_Toc184670215)

[CHƯƠNG 1. CHƯƠNG 1: MÔ TẢ BÀI TOÁN VẬN HÀNH VÀ QUẢN LÝ QUY TRÌNH THI TRẮC NGHIỆM KHÁCH QUAN 3](#_Toc184670216)

[1.1. Mô tả tổng quan bài toán 3](#_Toc184670217)

[1.2. Tìm hiểu bài toán vận hành và quản lý thi trắc nghiệm khách quan 4](#_Toc184670218)

[1.3. Một số ứng dụng liên quan đến đề tài 6](#_Toc184670219)

[1.4. Các yêu cầu cho hệ thống mới 7](#_Toc184670220)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG VẬN HÀNH VÀ QUẢN LÝ QUY TRÌNH THI TRẮC NGHIỆM KHÁCH QUAN 11](#_Toc184670221)

[2.1. Phân tích yêu cầu và xác định tác nhân 11](#_Toc184670222)

[2.2. Sprint 1: Xử lý ảnh đầu vào và trích xuất thông tin trên thẻ sinh viên 17](#_Toc184670223)

[2.3. Sprint 2: Chức năng so khớp thông tin sinh viên với danh sách phòng thi 23](#_Toc184670224)

[2.4. Sprint 3: Đối chiếu khuôn mặt trên thẻ với ảnh thực tế 27](#_Toc184670225)

[2.5. Sprint 4: Xuất phiếu thi và giám sát quy trình thi 31](#_Toc184670226)

[2.6. Sprint 5: Các chức năng quản lý của quản trị viên 33](#_Toc184670227)

[CHƯƠNG 3. XÂY DỰNG HỆ THỐNG VẬN HÀNH VÀ QUẢN LÝ QUY TRÌNH THI TRẮC NGHIỆM KHÁCH QUAN 34](#_Toc184670228)

[3.1. Môi trường phát triển 34](#_Toc184670229)

[3.2. Chức năng xử lý ảnh đầu vào và trích xuất thông tin trên thẻ sinh viên 36](#_Toc184670230)

[3.3. Chức năng so khớp thông tin sinh viên với danh sách phòng thi 42](#_Toc184670231)

[3.4. Chức năng đối chiếu khuôn mặt trên thẻ sinh viên với ảnh thực tế 48](#_Toc184670232)

[3.5. Chức năng xuất phiếu và giám sát quy trình thi 52](#_Toc184670233)

[3.6. Các chức năng quản lý của quản trị viên 52](#_Toc184670234)

[KẾT LUẬN 53](#_Toc184670235)

[Kết quả đạt được 53](#_Toc184670236)

[Những tồn tại và hạn chế 53](#_Toc184670237)

[TÀI LIỆU THAM KHẢO 54](#_Toc184670238)

1. MÔ TẢ BÀI TOÁN VẬN HÀNH VÀ QUẢN LÝ QUY TRÌNH THI TRẮC NGHIỆM KHÁCH QUAN

Việc sử dụng hình thức thi trắc nghiệm khách quan để kiểm tra và đánh giá kết quả học tập của sinh viên đang được các cơ sở giáo dục bậc đại học lựa chọn, dù vậy vẫn còn tồn tại nhiều khó khăn và thách thức. Trong chương đầu tiên này, chúng tôi sẽ trình bày bối cảnh thực tế dẫn đến việc chọn lựa đề tài, nhấn mạnh những yêu cầu cần thiết để xây dựng một hệ thống quản lý thi hiệu quả, và phân tích những thách thức đặt ra khi triển khai. Qua đó, chương cũng sẽ nêu ra và giới thiệu sơ bộ các công nghệ được sử dụng để cải thiện và khắc phục các nhược điểm hiện tại.

* 1. Mô tả tổng quan bài toán

Thực tế, tổ chức thi trắc nghiệm khách quan cho số lượng lớn sinh viên đã mang lại nhiều cải tiến so với hình thức thi truyền thống, như tăng tốc độ xử lý kết quả, chuẩn hóa cách đánh giá và giảm bớt áp lực chấm thi. Tuy nhiên, phương thức này vẫn còn nhiều điểm hạn chế cần khắc phục. Một trong những bất cập lớn nhất hiện nay là **quá trình xác minh danh tính sinh viên** trước mỗi ca thi vẫn chủ yếu được thực hiện thủ công. Phương pháp này không chỉ tiêu tốn thời gian mà còn dễ dẫn đến sai sót, nhất là với các kỳ thi có số lượng thí sinh tham gia lớn. Thêm vào đó, việc **phát phiếu thi chứa thông tin quan trọng** như tài khoản và mật khẩu truy cập thường diễn ra trước ca thi và trước cả bước xác minh danh tính. Đây là một lỗ hổng lớn trong khâu quản lý, tiềm ẩn nguy cơ gian lận, khi thông tin này có thể bị lạm dụng hoặc trao đổi trước giờ thi. Ở khâu **giám sát thi cử**, hầu hết các cơ sở giáo dục vẫn sử dụng phương pháp truyền thống, bố trí từ 1-2 giám thị cho mỗi phòng thi để quan sát trực tiếp sinh viên làm bài. Mặc dù phương pháp này đã được áp dụng từ lâu, nhưng nó đang bộc lộ nhiều hạn chế trước sự phát triển ngày càng tinh vi của các hành vi gian lận. Những bất cập trong quá trình vận hành này không chỉ làm giảm hiệu quả mà còn gây lãng phí thời gian, nguồn lực và thậm chí có thể ảnh hưởng đến uy tín của tổ chức giáo dục. Từ thực trạng trên,việc ứng dụng các **giải pháp công nghệ hiện đại** để tự động hóa quy trình thi là một đòi hỏi cấp thiết. Các công nghệ như nhận diện khuôn mặt để xác minh danh tính, hệ thống giám sát thông minh sử dụng trí tuệ nhân tạo, hoặc phần mềm bảo mật để quản lý dữ liệu thi đều có thể giúp khắc phục những hạn chế hiện tại. Những giải pháp này không chỉ giúp giảm thiểu tối đa sai sót và gian lận mà còn tối ưu hóa thời gian, nhân lực trong tổ chức thi. Quan trọng hơn, chúng tạo ra một môi trường thi cử minh bạch, công bằng và hiệu quả hơn, góp phần nâng cao chất lượng giáo dục cũng như sự tin cậy của xã hội đối với hệ thống giáo dục hiện đại.

* 1. Tìm hiểu bài toán vận hành và quản lý thi trắc nghiệm khách quan

Hiện tại, trong công cuộc tổ chức thi trắc nghiệm khách quan yêu cầu giám thị thực hiện tuần tự nhiều bước nhằm đảm bảo tính chính xác và minh bạch. Đầu tiên, giám thị nhận danh sách sinh viên thông qua văn bản giấy được cấp phát ngay trước giờ thi, sau đó gọi tên từng sinh viên để xác nhận sự có mặt. Tiếp theo, giám thị đối chiếu thông tin cá nhân trên thẻ sinh viên, bao gồm họ tên và mã số, với danh sách để đảm bảo khớp đúng. Đồng thời, giám thị kiểm tra khuôn mặt thực tế của sinh viên với ảnh trên thẻ nhằm xác minh danh tính. Chỉ khi hoàn thành đầy đủ các bước xác minh này, sinh viên mới nhận được phiếu thi để tham gia thực hiện bài thi, nhằm đảm bảo quá trình diễn ra chặt chẽ, hạn chế gian lận và sai sót.

Qua thực tiễn từ bài toán này, để hướng đến tự động hóa và tối ưu quy trình, nhóm chúng tôi đề xuất các bước thực hiện theo thứ tự ưu tiên dựa trên mức độ quan trọng, tạo cơ sở cho các bước tiếp theo:

* + 1. Giới thiệu chức năng xử lý đầu vào và trích xuất thông tin thẻ sinh viên

Trong thực tế, khi vận hành thủ công, giám thị phải trực tiếp kiểm tra thông tin trên thẻ sinh viên, đối chiếu mã số và họ tên với danh sách sinh viên. Quy trình này dễ gặp phải những nhược điểm như mất nhiều thời gian khi số lượng sinh viên lớn, dễ xảy ra sai sót khi giám thị đọc nhầm thông tin hoặc không phát hiện kịp thời các gian lận như sử dụng thẻ giả hoặc thẻ không hợp lệ.

Việc tự động hóa bước đối chiếu thông tin từ thẻ sinh viên với danh sách không chỉ giúp giảm tải áp lực cho giám thị mà còn tăng cường độ chính xác và tốc độ xử lý. Thay vì giám thị kiểm tra từng thẻ một, hệ thống sẽ tự động trích xuất các thông tin văn bản từ thẻ sinh viên, đối chiếu trực tiếp với cơ sở dữ liệu để xác minh nhanh chóng. Điều này đảm bảo quy trình vận hành kỳ thi trở nên hiệu quả, minh bạch và giảm thiểu tối đa sai sót thủ công.

Trong đó để xác minh danh tính của một sinh viên ta chỉ cần: ***mã số sinh viên*** và ***họ tên sinh viên***. Lý do lựa chọn hai trường này là bởi chúng là các yếu tố cơ bản, quan trọng và dễ dàng xác minh trong quá trình đối chiếu. *Mã số sinh viên* là một thông tin duy nhất, không trùng lặp, giúp nhận diện chính xác từng cá nhân, trong khi *họ tên sinh viên* giúp xác nhận thông tin một cách bổ sung. Các trường khác dù thẻ sinh viên có trên thẻ sinh viên như *ngành học*, *khóa học*, *đơn vị* nhưng để tối ưu tốc độ xử lý, hệ thống tập trung vào hai trường thông tin cốt lõi này.

Tiếp theo, hệ thống sẽ thực hiện bước so khớp sinh viên với danh sách phòng thi, đảm bảo rằng mỗi sinh viên được phân vào đúng phòng thi và không có sai sót nào trong việc phân bổ phòng.

* + 1. Giới thiệu chức năng so khớp sinh viên với danh sách phòng thi

Chức năng so khớp thông tin sinh viên với danh sách phòng thi đóng vai trò quan trọng trong việc xác minh sinh viên có đến đúng phòng thi đã được phân công hay không. Trong thực tế, giám thị sẽ nhận danh sách sinh viên bằng bản giấy ngay trước ca thi, sau đó thực hiện việc gọi tên từng sinh viên theo danh sách và đối chiếu với thông tin trên thẻ sinh viên. Tuy nhiên, quy trình này khá rườm rà và không tối ưu. Khi giám thị gọi tên mà sinh viên chưa có mặt, họ sẽ phải đánh dấu và tạm bỏ qua để xác minh sinh viên khác, tạo ra khoảng thời gian trống không cần thiết, làm chậm tiến độ xác minh cho cả giám thị và sinh viên đã có mặt.

Với việc tự động hóa quy trình này, dựa trên thông tin đã được trích xuất từ thẻ sinh viên, hệ thống có thể tự động so sánh thông tin với danh sách sinh viên trong phòng thi, từ đó xác nhận đúng sinh viên và phòng thi một cách nhanh chóng. Điều này giúp tối ưu hóa thời gian và tiện lợi hơn rất nhiều so với phương pháp thủ công. Được biết mỗi phòng thi sẽ bao gồm các trường thông tin như: *phòng thi, mã đề thi, mã số giám thị, số lượng tối đa sinh viên, và danh sách mã số sinh viên*. Qua các trường thông tin này ta chỉ cần lấy *mã số sinh viên* trong *danh sách mã số sinh viên* để đối chiếu với phần đã đọc được trên thẻ từ trước.

* + 1. Giới thiệu chức năng đối chiếu khuôn mặt trên thẻ với ảnh thực tế

Việc đối chiếu khuôn mặt trên thẻ sinh viên với khuôn mặt thực tế của sinh viên là một công đoạn quan trọng để đảm bảo tính xác thực của danh tính. Hiện tại, giám thị sẽ phải dựa vào sự quan sát và kinh nghiệm của mình để so sánh hai khuôn mặt, một trên thẻ và một ngoài đời. Tuy nhiên, phương pháp này có nhược điểm lớn như việc dễ bị ảnh hưởng bởi yếu tố chủ quan, sự mệt mỏi của giám thị hoặc điều kiện ánh sáng không phù hợp, dẫn đến khả năng nhầm lẫn hoặc sai sót trong quá trình đối chiếu. Nếu sinh viên có thay đổi nhỏ về ngoại hình hoặc khuôn mặt bị mờ do góc chụp, quá trình xác minh cũng có thể gặp khó khăn.

Để tối ưu hóa và tự động hóa quá trình này, hệ thống sẽ sử dụng công nghệ nhận diện khuôn mặt để so sánh khuôn mặt trên thẻ sinh viên và khuôn mặt thực tế của sinh viên. Sau khi đã trích xuất được thông tin từ thẻ sinh viên, bao gồm mã số sinh viên, họ tên và đặc biệt là ảnh thẻ, hệ thống sẽ tiến hành tách và nhận diện khuôn mặt trong ảnh thực tế được chụp từ camera. Các thuật toán nhận diện khuôn mặt sẽ phân tích các đặc điểm như mắt, mũi, miệng, và cấu trúc khuôn mặt để so sánh với ảnh trên thẻ.

Quá trình này không chỉ giúp tăng cường tính chính xác trong việc xác minh danh tính mà còn giảm thiểu tối đa sai sót do yếu tố con người. Việc sử dụng công nghệ nhận diện khuôn mặt cũng giúp tăng tốc độ kiểm tra và đảm bảo an toàn hơn so với phương pháp thủ công truyền thống, mang lại một giải pháp hiệu quả và dễ dàng hơn trong việc quản lý và giám sát sinh viên.

* + 1. Giới thiệu chức năng xuất phiếu và giám sát quy trình thi

Sau khi xác minh danh tính sinh viên, giám thị sẽ phát phiếu thi chứa các thông tin như tài khoản, mật khẩu bài thi cho sinh viên tương ứng. Hiện tại, việc phát phiếu thi thủ công gặp nhiều vấn đề: phiếu thi được sắp xếp theo thứ tự nhưng sinh viên đến xác thực có thể không theo thứ tự hoặc có sinh viên vắng mặt, dẫn đến phiếu thi bị bỏ qua hoặc gửi nhầm cho sinh viên khác. Việc in phiếu sẵn cho tất cả sinh viên tham gia thi cũng gây lãng phí khi có sinh viên vắng thi trong ca thi đó.

Để giải quyết các bất cập này, chúng tôi đề xuất chức năng xuất phiếu thi ngay sau khi xác thực thành công. Cụ thể, phiếu thi sẽ được in ngay lập tức sau khi sinh viên xác thực thành công, với các thông tin như: *mã số sinh viên, tên môn thi, tên đăng nhập, mật khẩu, thời gian thi, và vị trí ngồi*. Việc này giúp sinh viên dễ dàng tìm được vị trí của mình và tránh các tình huống phát sinh như trong quy trình truyền thống.

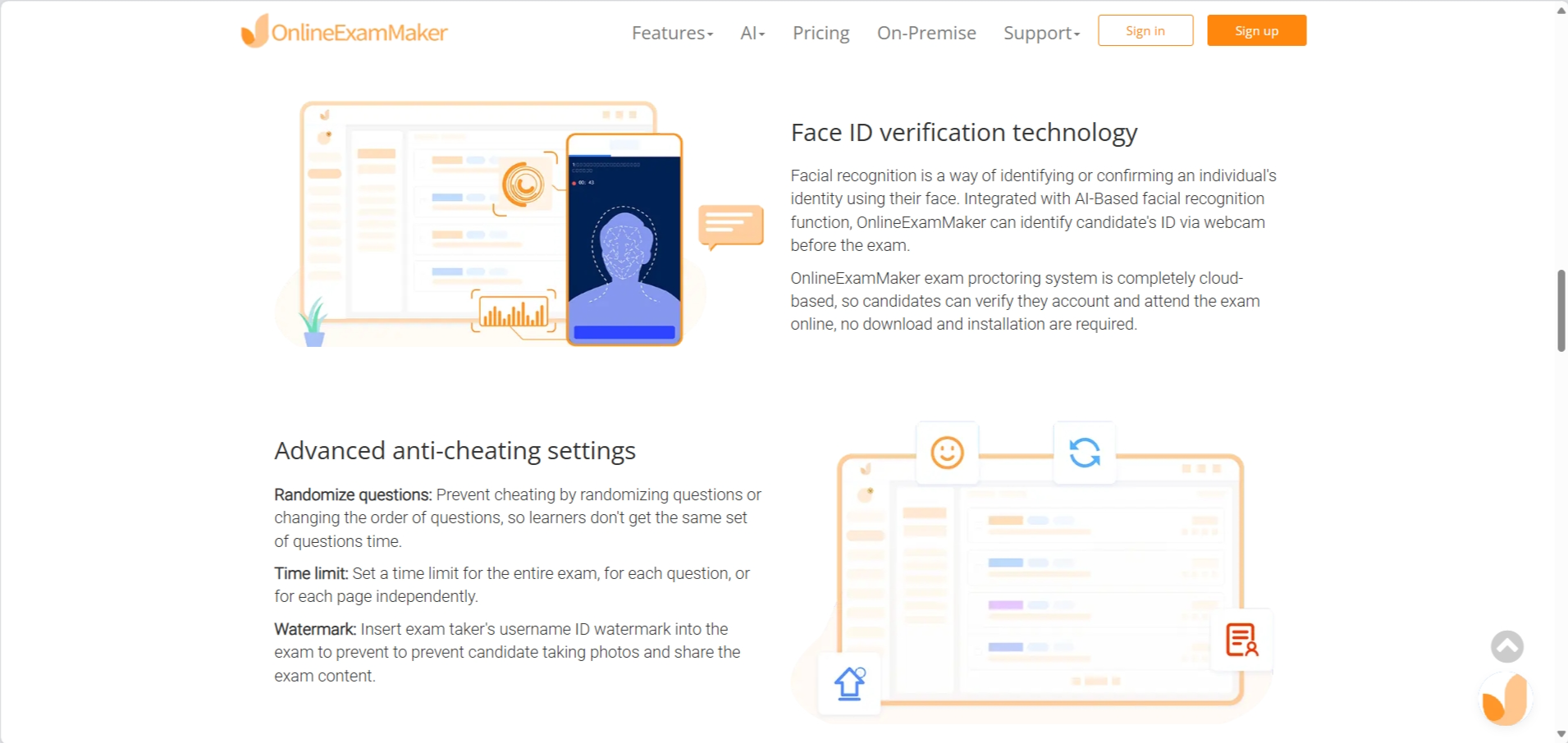
Về phần giám sát quy trình thi, mặc dù phòng thi đã được trang bị camera, nhưng các camera này chưa được ứng dụng vào việc phát hiện gian lận thời gian thực, mà vẫn cần đến sự giám sát trực tiếp của giám thị. Với số lượng sinh viên trong phòng thi và các hành vi gian lận ngày càng tinh vi (như trao đổi bài, thay đổi vị trí ngồi, sử dụng tài liệu hoặc thiết bị công nghệ cao), chúng tôi đề xuất ứng dụng công nghệ truy vết đối tượng và giám sát hành vi để nhanh chóng và chính xác phát hiện các hành vi gian lận trong quá trình thi.

* + 1. Giới thiệu thực tế các chức năng quản lý của quản trị viên
    2. Giới thiệu thực tế quy trình triển khai, kiểm thử và bảo trì hệ thống
  1. Một số ứng dụng liên quan đến đề tài

Đề tài phát triển hệ thống nhận diện và xác minh danh tính có thể ứng dụng trong nhiều lĩnh vực, đặc biệt là trong các hệ thống quản lý, kiểm tra, giám sát và bảo mật. Dưới đây là một số ứng dụng liên quan đến đề tài:

1) Hệ thống kiểm tra thi trực tuyến

Trang web <https://onlineexammaker.com/> là nền tảng hỗ trợ tổ chức thi trực tuyến có áp dụng nhận diện khuôn mặt để xác minh danh tính của thí sinh ngay trước ca thi.



Hình 1.x: Giao diện hệ thống kiểm tra thi trực tuyến

2) Hệ thống xác minh danh tính bằng công nghệ nhận diện khuôn mặt tại Vinpearl

Ứng dụng nhận diện khuôn mặt được triển khai đầu tiên tại Tổ hợp Vinpearl Nha Trang để xác thực danh tính du khách khi ra vào khách sạn, khu nghỉ dưỡng, khu vui chơi giải trí VinWonders và các nhà hàng, thay thế các thủ tục kiểm soát truyền thống bằng thiết bị quan sát nhận diện khuôn mặt.



Hình 1.x: Ứng dụng xác minh khuôn mặt tại Vinpearl



Hình 1.x: Ứng dụng xác minh khuôn mặt tại Vinpearl

3) Hệ thống thi lí thuyết trong sát hạch bằng lái xe ô tô

Trước khi học viên tiến hành ca thi lý thuyết trong phần thi sát hạch bằng lái xe ô tô, học viên cần cung cấp căng cước công dân và tiến hành đứng trước camera để xác minh danh tính, sau khi hoàn thành mới được tiến hành thi.

A person holding up a card

Description automatically generated

Hình 1.x: Ứng dụng thi lí thuyết trong sát hạch bằng lái xe ô tô

* 1. Các yêu cầu cho hệ thống mới

Hệ thống nhận diện và xác minh danh tính sinh viên thông qua thẻ sinh viên và khuôn mặt cần phải đáp ứng một số yêu cầu chức năng và phi chức năng để đảm bảo hoạt động hiệu quả và ổn định. Dưới đây là các yêu cầu chi tiết cho hệ thống.

* + 1. Yêu cầu chức năng

|  |  |  |
| --- | --- | --- |
| **STT** | **Nội dung yêu cầu** | **Mô tả** |
| 1 | Hệ thống phải nhận diện các trường thông tin trên thẻ sinh viên. | Các trường cần nhận diện gồm mã số sinh viên, họ tên, ngày sinh, ngành học, khóa học, đơn vị, địa chỉ email, số điện thoại. Thông tin trích xuất được lưu trữ chính xác vào cơ sở dữ liệu để phục vụ các quy trình sau. |
| 2 | Hệ thống phải nhận diện khuôn mặt sinh viên qua camera. | So khớp khuôn mặt sinh viên với dữ liệu đã lưu trong cơ sở dữ liệu. Đảm bảo nhận diện chính xác trong các điều kiện khác nhau (ánh sáng yếu, góc nhìn khác). |
| 3 | Hệ thống phải đối chiếu thông tin từ thẻ và khuôn mặt để xác minh danh tính sinh viên. | Đối chiếu các thông tin với cơ sở dữ liệu hiện có. Quá trình xác minh phải nhanh, chính xác, không xảy ra sai sót. |
| 4 | Hệ thống phải lưu trữ thông tin sinh viên vào cơ sở dữ liệu. | Lưu trữ cả ảnh thẻ sinh viên và ảnh khuôn mặt. Thông tin phải được bảo mật và dễ dàng truy xuất. |
| 5 | Hệ thống phải quản lý thông tin phòng thi và danh sách sinh viên. | Quản lý danh sách sinh viên, giờ thi, môn thi. Xác nhận sinh viên được tham gia thi khi thông tin khớp với dữ liệu trong danh sách phòng thi. |
| 6 | Hệ thống cần kiểm tra và thông báo kết quả xác minh ngay lập tức. | Cung cấp thông báo “thành công” hoặc “thất bại” dựa trên độ khớp của thông tin. |
| 7 | Hệ thống cần có giao diện người dùng dễ sử dụng. | Giao diện được thiết kế cho sinh viên, giám thị và quản trị viên.Đơn giản, dễ hiểu, không yêu cầu kiến thức kỹ thuật đặc biệt. |

Bảng 1.x Bảng mô tả yêu cầu chức năng

* + 1. Yêu cầu phi chức năng

|  |  |  |
| --- | --- | --- |
| **STT** | **Nội dung yêu cầu** | **Mô tả** |
| 1 | Hệ thống phải xử lý nhanh chóng và hiệu quả. | Đảm bảo trích xuất và xác minh thông tin nhanh chóng, không quá 10 giây/sinh viên. Tránh gây trì hoãn trong các quy trình như tham gia thi hoặc sự kiện. |
| 2 | Hệ thống phải có khả năng mở rộng. | Đáp ứng nhu cầu gia tăng trong tương lai như số lượng sinh viên tăng hoặc tích hợp tính năng mới. Cơ sở dữ liệu phải xử lý tốt lượng lớn dữ liệu mà không làm giảm hiệu suất. |
| 3 | Hệ thống phải đảm bảo tính bảo mật cao. | Mã hóa các thông tin nhạy cảm như mã số sinh viên, ảnh thẻ, và thông tin cá nhân. Quy trình truy cập được bảo mật, chỉ người có quyền mới truy cập hoặc chỉnh sửa dữ liệu. |
| 4 | Hệ thống phải đảm bảo tính khả dụng và độ tin cậy. | Luôn hoạt động ổn định trong quá trình triển khai. Có khả năng tự động khôi phục sau sự cố và tránh mất dữ liệu quan trọng. |
| 5 | Hệ thống phải tương thích với các thiết bị đầu vào và nền tảng khác nhau. | Hỗ trợ máy ảnh, webcam và các thiết bị nhận diện khuôn mặt. Hỗ trợ trên nhiều nền tảng như máy tính để bàn và thiết bị di động. |
| 6 | Hệ thống phải dễ bảo trì và cập nhật. | Dễ dàng nâng cấp phần mềm, triển khai tính năng mới hoặc sửa lỗi mà không làm gián đoạn hoạt động. |
| 7 | Hệ thống phải tương tác tốt với các hệ thống khác. | Tích hợp với hệ thống quản lý sinh viên, hệ thống thi cử, và các phần mềm hành chính khác. Các API phải được thiết kế để hỗ trợ tương tác dễ dàng với các hệ thống bên ngoài. |

Bảng 1.x Bảng mô tả yêu cầu phi chức năng

1. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG VẬN HÀNH VÀ QUẢN LÝ QUY TRÌNH THI TRẮC NGHIỆM KHÁCH QUAN

Chương 2 sẽ tập trung vào việc phân tích và thiết kế hệ thống vận hành và quản lý quy trình thi trắc nghiệm khách quan. Nội dung của chương sẽ trình bày các yêu cầu chức năng của hệ thống, bao gồm việc quản lý kỳ thi, thí sinh, cũng như việc chấm điểm tự động và bảo mật dữ liệu. Đồng thời, chương cũng sẽ đề cập đến các công nghệ và kiến trúc phần mềm cần thiết để hệ thống hoạt động hiệu quả, đảm bảo tính chính xác và công bằng trong quá trình thi. Thông qua việc phân tích chi tiết các quy trình nghiệp vụ và thiết kế hệ thống, chương này sẽ cung cấp cái nhìn tổng quan về cách thức vận hành của hệ thống, từ khâu chuẩn bị kỳ thi cho đến việc xử lý kết quả và đánh giá thí sinh.

* 1. Phân tích yêu cầu và xác định tác nhân
     1. Phân tích yêu cầu hệ thống

Mục tiêu của phân tích yêu cầu hệ thống là xác định các tính năng và yêu cầu mà hệ thống phải đáp ứng trong quá trình phát triển. Để đảm bảo tính hoàn chỉnh, hiệu quả và khả năng mở rộng của hệ thống, các yêu cầu được phân chia thành các Sprint cụ thể, mỗi Sprint có mục đích và yêu cầu rõ ràng. Dưới đây là mô tả tổng thể các yêu cầu hệ thống được chia theo từng Sprint:

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Nội dung** | **Mục đích hướng đến** |
| 1 | Xử lý đầu vào và trích xuất thông tin trên thẻ sinh viên | Phát triển khả năng xử lý và trích xuất thông tin từ ảnh thẻ sinh viên. Sprint này tập trung vào việc phát triển tính năng tải ảnh, xử lý ảnh (chỉnh sửa độ sáng, độ tương phản, xoay ảnh) và trích xuất các thông tin như mã số sinh viên, tên, lớp học, khoa/ngành học từ thẻ. |
| 2 | So khớp thông tin sinh viên với danh sách phòng thi | Phát triển tính năng kiểm tra tính hợp lệ của thông tin trích xuất từ ảnh thẻ sinh viên và đối chiếu với cơ sở dữ liệu để xác nhận danh tính. |
| 3 | Đối chiếu khuôn mặt trên thẻ với ảnh thực tế | Tích hợp khả năng nhận diện khuôn mặt từ ảnh thẻ sinh viên và khuôn mặt thực tế trong môi trường thi để xác nhận đúng danh tính sinh viên. |
| 4 | Xuất phiêu và giám sát quy trình thi | Xây dựng khả năng quản lý, lưu trữ và truy xuất thông tin sinh viên, bao gồm việc lưu trữ dữ liệu ảnh thẻ, thông tin sinh viên và kết quả xác nhận. |
| 5 | Các chức năng quản lý của quản trị viên | Đảm bảo hệ thống hoạt động ổn định, hiệu quả và đạt được hiệu suất cao khi xử lý lượng lớn dữ liệu. |
| 6 | Triển khai, kiểm thử và bảo trì hệ thống | Triển khai hệ thống vào môi trường thực tế và thực hiện bảo trì, cập nhật khi cần thiết. |
| 7 | … | … |

Bảng 2.x Bảng mô tả yêu cầu phi chức năng

Thông qua các Sprint này, hệ thống sẽ dần được xây dựng từ các tính năng cơ bản đến các tính năng phức tạp hơn, đồng thời đảm bảo chất lượng và hiệu quả hoạt động của hệ thống trong môi trường thực tế.

* + 1. Xác định các tác nhân tác động lên hệ thống
       1. Quản trị viên

Quản trị viên là tác nhân quan trọng nhất trong hệ thống vì họ có quyền quản lý và điều phối tất cả các hoạt động trong hệ thống. Tác nhân này sẽ chịu trách nhiệm cho việc cấu hình hệ thống, quản lý người dùng, giám sát và xử lý các thông tin liên quan đến sinh viên, giám thị và kỳ thi. Các chức năng chính của tác nhân quản trị viên bao gồm:

* Quản lý người dùng: Quản trị viên có quyền tạo, chỉnh sửa và xóa tài khoản của giám thị và sinh viên trong hệ thống.
* Cấu hình hệ thống: Quản trị viên có thể thay đổi các cấu hình của hệ thống, bao gồm các tham số liên quan đến quy trình thi, thông tin kỳ thi, và các cài đặt bảo mật.
* Giám sát hoạt động hệ thống: Quản trị viên có thể theo dõi các hoạt động của giám thị và sinh viên trong hệ thống, bao gồm cả việc theo dõi thời gian tham gia thi, trạng thái nhận diện khuôn mặt và các hành vi khác.
* Quản lý dữ liệu: Quản trị viên sẽ chịu trách nhiệm duy trì và quản lý cơ sở dữ liệu, bao gồm việc sao lưu dữ liệu, phục hồi dữ liệu và đảm bảo tính bảo mật của hệ thống.
* Xử lý sự cố và hỗ trợ: Quản trị viên sẽ tiếp nhận các khiếu nại, báo lỗi từ giám thị và sinh viên, xử lý sự cố liên quan đến hệ thống và đưa ra giải pháp hỗ trợ khi cần thiết.

Quản trị viên là người điều hành chính trong việc đảm bảo hệ thống hoạt động hiệu quả và đáp ứng các yêu cầu của người dùng.

* + - 1. Giám thị

Giám thị là tác nhân chịu trách nhiệm giám sát kỳ thi, theo dõi hành vi của sinh viên trong suốt thời gian thi và đảm bảo quá trình thi diễn ra công bằng. Các chức năng của tác nhân giám thị bao gồm:

* Giám sát sinh viên: Giám thị theo dõi sinh viên trong suốt quá trình thi, bao gồm việc quan sát và ghi nhận hành vi của sinh viên để phát hiện các hành vi gian lận.
* Xác nhận danh tính sinh viên: Giám thị có thể sử dụng hệ thống để xác nhận danh tính của sinh viên qua nhận diện khuôn mặt, đối chiếu với ảnh thẻ sinh viên hoặc khuôn mặt thực tế.
* Quản lý phòng thi: Giám thị chịu trách nhiệm theo dõi và quản lý các phòng thi, bao gồm việc đảm bảo tất cả sinh viên đều tham gia thi đúng phòng, đúng thời gian.
* Ghi nhận vi phạm: Giám thị có quyền ghi nhận các vi phạm của sinh viên, bao gồm hành vi gian lận, sử dụng tài liệu trái phép, hoặc hành động không tuân thủ quy định kỳ thi.
* Báo cáo kết quả thi: Sau khi kỳ thi kết thúc, giám thị sẽ gửi báo cáo kết quả thi, thông tin về các vi phạm (nếu có) cho quản trị viên để xử lý.

Giám thị là người đảm bảo tính minh bạch và công bằng trong kỳ thi, chịu trách nhiệm bảo vệ quy trình thi và xử lý các tình huống phát sinh.

* + - 1. Sinh viên

Sinh viên là tác nhân tham gia vào quá trình thi, thực hiện các nhiệm vụ liên quan đến việc tham gia thi, bao gồm việc chuẩn bị thông tin cá nhân và tham gia các kỳ thi. Các chức năng của tác nhân sinh viên bao gồm:

* Cung cấp thông tin cá nhân: Sinh viên phải cung cấp các thông tin cá nhân cần thiết để hệ thống xác nhận danh tính, bao gồm thông tin thẻ sinh viên và khuôn mặt thực tế.
* Tham gia kỳ thi: Sinh viên tham gia kỳ thi theo lịch đã định, và phải tuân thủ các quy định của kỳ thi, bao gồm việc xác nhận danh tính trước khi vào phòng thi.
* Chấp hành quy chế thi: Sinh viên phải tuân thủ các quy định trong kỳ thi, không sử dụng tài liệu trái phép hoặc gian lận trong suốt quá trình thi.
* Theo dõi kết quả thi: Sinh viên có thể kiểm tra kết quả thi sau khi kỳ thi kết thúc, bao gồm điểm thi và các thông báo liên quan đến việc tham gia kỳ thi.
* Phản hồi về hệ thống: Sinh viên có thể cung cấp phản hồi về quá trình thi hoặc hệ thống nếu gặp vấn đề hoặc có yêu cầu cải tiến.

Sinh viên là tác nhân chủ yếu của hệ thống, chịu trách nhiệm tham gia kỳ thi đúng quy trình và tuân thủ các yêu cầu của hệ thống để đảm bảo tính minh bạch và công bằng.

* 1. Sprint 1: Xử lý ảnh đầu vào và trích xuất thông tin trên thẻ sinh viên
     1. Giới thiệu Sprint 1

Trong quy trình xác minh danh tính sinh viên, bước đầu tiên là xử lý ảnh thẻ sinh viên mà người dùng tải lên, trong bước này các giám thị sẽ yêu cầu sinh viên đưa thẻ để giám thị chụp và tiến hàng xử lý. Ảnh đầu vào sẽ được hệ thống tiếp nhận và trải qua quá trình tiền xử lý, bao gồm các thao tác như cắt thẻ, điều chỉnh độ sáng, độ tương phản và làm thẳng ảnh nếu cần thiết. Sau khi ảnh đã được chuẩn hóa, hệ thống sử dụng công nghệ nhận dạng ký tự quang học (OCR) để trích xuất các thông tin quan trọng từ thẻ, bao gồm mã số sinh viên, tên, ngày tháng năm sinh, và khoa hoặc ngành học. Kết quả sau bước này là các thông tin trên thẻ sinh viên sẽ được hệ thống ghi nhận và sẵn sàng cho các bước xác thực tiếp theo trong quá trình xác minh danh tính.

Thời gian thực hiện trong vòng 11 ngày từ 30/9/2024 – 13/10/2024, với các công việc cần làm:

* B1: Thu nhập và chuẩn bị dữ liệu mẫu (Thái Văn Tuấn phụ trách – 2 ngày)
* B2: Xây dựng chức năng trích xuất thông tin từ thẻ sinh viên (Đặng Ngọc Anh phụ trách – 5 ngày)
* B3: Huấn luyện với Model Yolo v11 với tập dữ liệu THE SINH VIEN (Đặng Ngọc Anh phụ trách – không giới hạn vì đây là quá trình huấn luyện)
* B4: Tạo giao diện và kết hợp chức năng đối chiếu vào giao diện thông qua backend, truy vấn với DB SINH VIEN (Đặng Ngọc Anh phụ trách – 2 ngày)
* B5: Kiểm thử Model, API trích xuất thông tin (Thái Văn Tuấn phụ trách – 2 ngày)

A diagram of a person with a circle with text

Description automatically generated with medium confidence

Hình 2.x. Biểu đồ ca sử dụng của giám thị với sprint 1

A diagram of a person with text

Description automatically generated

Hình 2.x. Biểu đồ ca sử dụng của sinh viên trong sprint 1

* + 1. Thiết kế dữ liệu sinh viên

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Diễn giải | Tên trường | Kiểu | Mặc định | Ràng buộc | Ghi chú |
| 1 | MongoDB tự tạo | \_id | ObjectId |  |  |  |
| 2 | Mã sinh viên | std\_id | String |  | PK |  |
| 3 | Họ tên sinh viên | std\_name | String |  |  |  |
| 4 | Ngày sinh | std\_dob | Date |  |  |  |
| 5 | Lớp | std\_class | String |  |  |  |
| 6 | Ngành | std\_faculty | String |  |  |  |
| 7 | Đơn vị | std\_course | String |  |  |  |
| 8 | Email | std\_email | String |  |  |  |
| 9 | Số điện thoại | std\_phone | String |  |  |  |
| 10 | Kết quả thi | exam\_result | Object |  | FK |  |
| 11 | Trạng thái | status | Boolean |  |  |  |

Bảng 2.x: Dữ liệu lưu trữ thông tin sinh viên

* + 1. Quy trình xử lý ảnh đầu vào và trích xuất thông tin

Cụ thể các bước thực hiện:

Bước 1:Tiếp nhận ảnh thẻ sinh viên thông qua giao diện hoặc nguồn dữ liệu.

Trong bước đầu tiên, hệ thống tiếp nhận ảnh thẻ sinh viên được tải lên hoặc chụp trực tiếp từ camera. Ảnh đầu vào sẽ được xử lý sơ bộ để chuẩn bị cho các giai đoạn trích xuất thông tin và xác minh danh tính. Các công việc chính trong bước này bao gồm:

* Tiếp nhận và lưu trữ ảnh đầu vào: Hệ thống nhận ảnh từ giao diện người dùng và lưu vào bộ nhớ tạm hoặc cơ sở dữ liệu nếu cần thiết.
* Kiểm tra định dạng ảnh: Xác minh định dạng tệp ảnh (JPEG, PNG, v.v.) để đảm bảo phù hợp với các bước xử lý tiếp theo.
* Xử lý ảnh cơ bản: Ảnh có thể được điều chỉnh về kích thước tiêu chuẩn, loại bỏ nhiễu cơ bản hoặc chuyển đổi không gian màu nếu cần để cải thiện độ chính xác khi xử lý sau này.

Sau khi hoàn thành, ảnh sẽ được chuyển sang bước tiếp theo là phát hiện và cắt thẻ sinh viên từ ảnh toàn cảnh.

Bước 2: Phát hiện và xác định vùng chứa thẻ sinh viên

Trong bước này, hệ thống sử dụng các thuật toán xử lý ảnh và nhận diện đối tượng để tìm vùng chứa thẻ sinh viên trong ảnh đầu vào. Quá trình này bao gồm:

* Áp dụng các thuật toán phát hiện đối tượng: Sử dụng các mô hình như YOLO, Haar Cascade, hoặc các mô hình mạng nơ-ron tiên tiến để phát hiện vị trí của thẻ sinh viên trong ảnh.
* Xác định và đánh dấu vùng thẻ: Khi phát hiện, hệ thống sẽ khoanh vùng khu vực có khả năng chứa thẻ sinh viên bằng cách vẽ khung bao quanh (bounding box) chính xác.
* Tiền xử lý bổ sung (nếu cần): Nếu ảnh bị nghiêng hoặc méo, thực hiện các bước căn chỉnh và hiệu chỉnh hình học để đưa vùng thẻ về đúng tỷ lệ chuẩn.

Sau khi xác định chính xác vùng chứa thẻ sinh viên, hệ thống sẽ tiếp tục trích xuất thông tin chữ và khuôn mặt từ vùng này để phục vụ cho bước xác minh danh tính.

Bước 3: Tiền xử lý trước OCR

Trong bước này, hệ thống tiếp tục xử lý vùng thẻ sinh viên đã được phát hiện để chuẩn hóa hình ảnh, đảm bảo khả năng nhận diện thông tin chính xác nhất:

* Cắt ảnh thẻ: Dựa trên khung bao quanh đã được xác định ở bước trước, hệ thống sẽ cắt chính xác vùng chứa thẻ sinh viên từ ảnh gốc.
* Căn chỉnh góc độ và làm thẳng thẻ: Nếu thẻ bị nghiêng hoặc không thẳng, áp dụng các phép biến đổi hình học như xoay hoặc làm phẳng để đưa thẻ về trạng thái chuẩn, đảm bảo thẻ hiển thị rõ ràng và nằm đúng góc độ.
* Chuyển ảnh về dạng ma trận màu thống nhất: Chuẩn hóa ảnh thẻ về cùng một không gian màu (chẳng hạn như RGB) và điều chỉnh độ sáng, độ tương phản để thẻ được rõ nét, dễ dàng trích xuất thông tin sau này.

Bước này rất quan trọng để đảm bảo chất lượng hình ảnh thẻ sinh viên đạt chuẩn, giúp các thuật toán nhận diện và trích xuất thông tin hoạt động chính xác và hiệu quả.

Bước 4: Trích xuất thông tin bằng OCR

Ở bước này, hệ thống sẽ áp dụng công nghệ nhận dạng ký tự quang học (OCR) để trích xuất chính xác các thông tin quan trọng từ ảnh thẻ sinh viên đã được chuẩn hóa:

* Sử dụng PaddleOCR: Áp dụng thuật toán PaddleOCR để nhận diện và chuyển đổi các ký tự trên ảnh thẻ thành dữ liệu văn bản.
* Trích xuất các trường thông tin: Hệ thống tập trung lấy ra các thông tin quan trọng như mã số sinh viên, tên sinh viên, ngày sinh, và lớp học.
* Xử lý sau OCR: Thực hiện các bước tiền xử lý như lọc nhiễu, loại bỏ ký tự không hợp lệ, và chuẩn hóa định dạng dữ liệu trích xuất để đảm bảo thông tin chính xác và nhất quán.

Bước này là nền tảng để có được dữ liệu đầu vào cần thiết phục vụ cho các bước tiếp theo trong quy trình xác thực danh tính sinh viên.

Bước 5: Xử lý và chuẩn hóa dữ liệu trích xuất để đưa vào hệ thống quản lý

Sau khi trích xuất thông tin từ thẻ sinh viên, hệ thống sẽ thực hiện các bước xử lý và chuẩn hóa dữ liệu để đảm bảo tính nhất quán và sẵn sàng sử dụng:

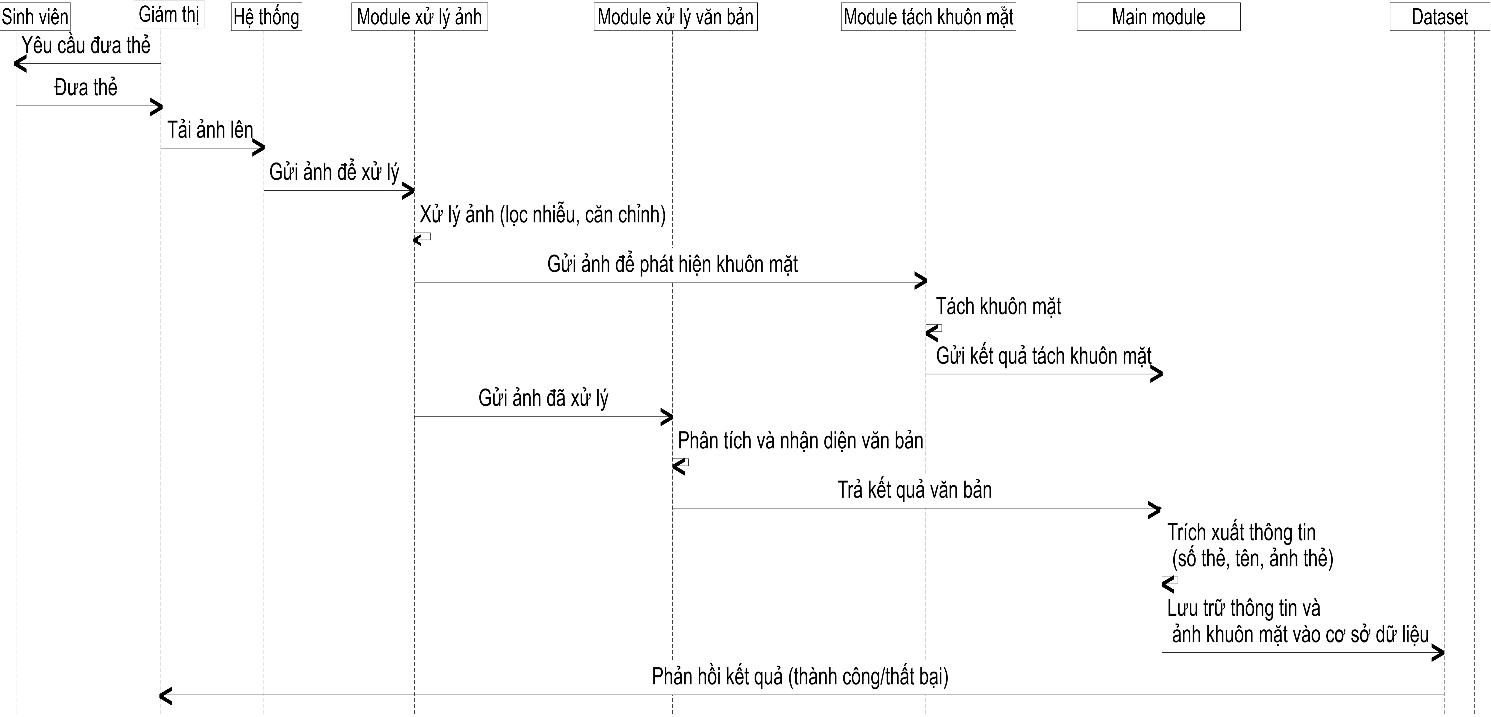
* Xử lý dữ liệu thô: Loại bỏ các ký tự không hợp lệ, sửa lỗi ký tự do nhận dạng OCR sai, và điều chỉnh định dạng cho phù hợp (ví dụ, định dạng ngày tháng hoặc viết hoa chữ cái đầu trong tên sinh viên).
* Chuẩn hóa thông tin: Đảm bảo rằng các trường dữ liệu như mã số sinh viên, tên, ngày sinh, và lớp học được định dạng theo đúng chuẩn của hệ thống quản lý.
* Lưu trữ dữ liệu: Chuẩn bị dữ liệu để tích hợp vào hệ thống quản lý, bao gồm chuyển đổi thông tin sang cấu trúc phù hợp với cơ sở dữ liệu MongoDB và sẵn sàng để sử dụng trong các chức năng khác, như kiểm tra danh tính và theo dõi hồ sơ sinh viên.

Bước này là yếu tố quan trọng để đảm bảo dữ liệu trích xuất từ thẻ được sử dụng hiệu quả và chính xác trong toàn bộ hệ thống.

* + 1. Thiết kế quy trình nghiệp vụ

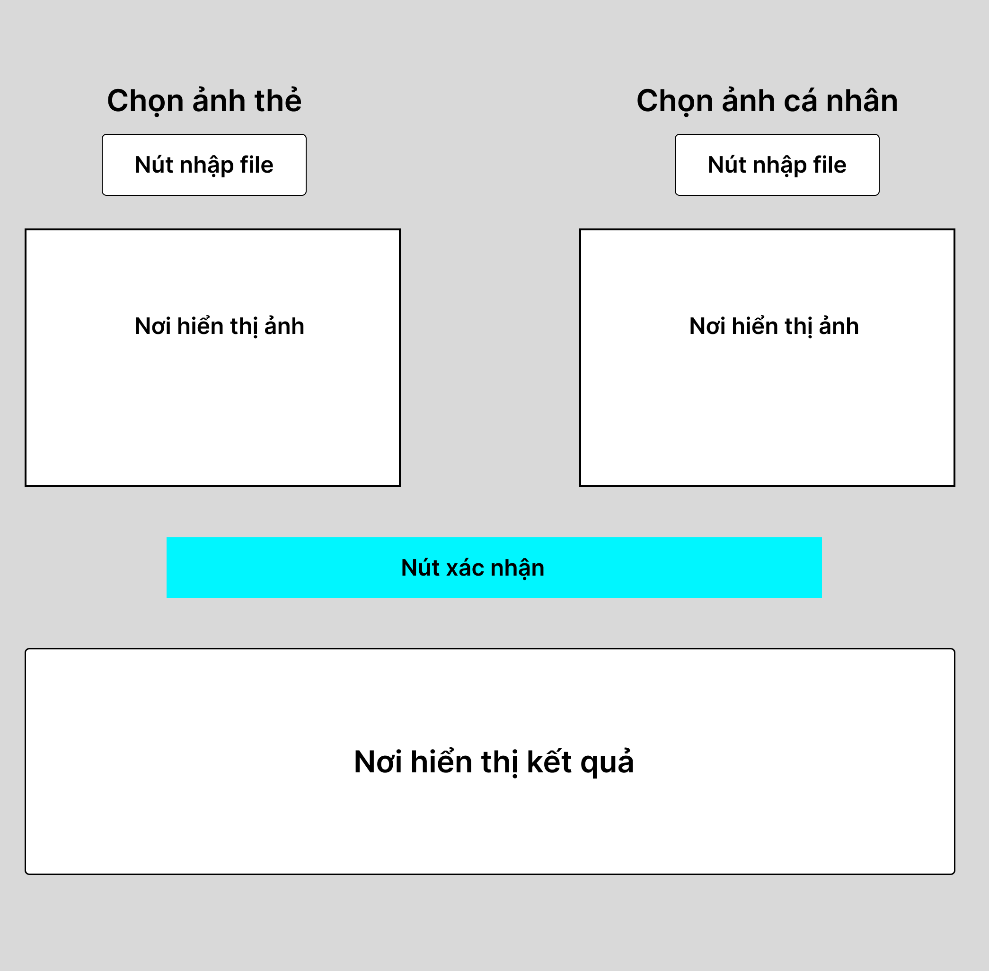
|  |  |
| --- | --- |
| **Mục đích** | Đảm bảo trích xuất chính xác thông tin sinh viên từ ảnh thẻ để phục vụ cho quá trình xác nhận danh tính. |
| **Các bước thực hiện** | **Bước 1:** Tiếp nhận ảnh thẻ sinh viên từ người dùng thông qua giao diện hoặc nguồn dữ liệu.  **Bước 2:** Áp dụng các thuật toán phát hiện, nhận diện để xác định vùng chứa thẻ sinh viên trong ảnh.  **Bước 3:** Cắt và làm thẳng ảnh thẻ, chuyển thẻ về cùng 1 dạng ma trận màu, đảm bảo thẻ hiển thị rõ ràng và đúng góc độ.  **Bước 4:** Sử dụng OCR (như PaddleOCR) để trích xuất thông tin chữ trên thẻ sinh viên, như mã số sinh viên, tên, ngày sinh, lớp.  **Bước 5:** Xử lý và chuẩn hóa dữ liệu trích xuất để đưa vào hệ thống quản lý. |
| **Các bước**  **bổ sung** | * Áp dụng các phương pháp tiền xử lý ảnh, như làm mịn, điều chỉnh độ tương phản, để cải thiện độ chính xác của OCR. * Sử dụng thuật toán sửa lỗi ký tự để tăng tính chính xác cho dữ liệu trích xuất từ thẻ. |
| **Các ngoại lệ** | * Ảnh đầu vào mờ hoặc bị nhòe, dẫn đến việc không thể trích xuất chính xác thông tin. * Thẻ sinh viên bị hư hỏng hoặc thông tin không rõ ràng. * OCR không nhận diện được các ký tự đặc biệt, cần có cơ chế sửa thủ công hoặc thông báo lỗi. |
| **Tài liệu liên quan** | PaddleOCR, OpenCV, các phương pháp tiền xử lý ảnh, tài liệu phân tích và xử lý ký tự. |
| **Tham chiếu** | Mục 1.6.1.1: Xử lý đầu vào và trích xuất thông tin trên thẻ sinh viên  Mục 3.3.1.1: Mô-đun xử lý đầu vào và trích xuất thông tin trên thẻ sinh viên. |

Bảng 2.x: Bảng chức năng xử lý đầu vào và trích xuất thông tin trên thẻ sinh viên



Hình 2.x: Biểu đồ tuần tự sprint

* + 1. Thiết kế giao diện



Hình 2.x: Thiết kế giao diện của trích xuất thông tin sinh viên

* 1. Sprint 2: Chức năng so khớp thông tin sinh viên với danh sách phòng thi
     1. Giới thiệu Sprint 2

Sau khi đã đảm bảo hoàn tất Mô-đun trích xuất thông tin trên thẻ sinh viên, bước tiếp theo là sẽ hiển thị danh sách sinh viên trong phòng thi kết hợp với Mô-đun xác nhận danh tinh sinh viên để các giám thị có thể dễ dàng kiểm tra. Các giám thị sẽ đăng nhập vào hệ thống và tìm phòng thi mình trực, hệ thống sẽ hiển thị danh sách sinh viên trong phòng thi đó và nếu các giám thị đã chắc chắn tất cả thông tin của sinh viên đều đúng thì giám thị có thể trích xuất danh sách thành file Excel. Sau khi đã có file, các giám thị sẽ nhập nó vào trong giao diện kiểm tra khuôn mặt và trích xuất thông tin thẻ sinh viên để giám thị có thể dễ dàng kiểm tra từng sinh viên một.

Thời gian thực hiện trong vòng 2 tuần từ 13/10/2024 – 27/10/2024, với các công việc cần làm:

* B1: Chuẩn bị danh sách thí sinh (Thái Văn Tuấn phụ trách – 2 ngày)
* B2: Xây dựng chức năng đối chiếu danh sách sinh viên trong phòng thi (Đặng Ngọc Anh phụ trách – 5 ngày)
* B3: Thêm sửa xóa với danh sách sinh viên (Đặng Ngọc Anh phụ trách – 2 ngày)
* B4: Xem danh sách sinh viên có trong phòng thi chỉ định, xuất kết quả ra file excel (Đặng Ngọc Anh phụ trách – 1 ngày)
* B5: Kiểm thử đối chiếu danh sách, CRUD sinh viên vào cơ sở dữ liệu (do Đặng Ngọc Anh và Thái Văn Tuấn phụ trách – 4 ngày)

A diagram of a file

Description automatically generated

Hình 2.x. Biểu đồ ca sử dụng của giám thị trong sprint 2

* + 1. Thiết kế dữ liệu phòng thi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Diễn giải | Tên trường | Kiểu | Mặc định | Ràng buộc | Ghi chú |
| 1 | MongoDB tự tạo | \_id | ObjectId  (auto) |  |  | Đối tượng MongoDb tự sinh |
| 2 | Mã phòng | Room\_id | String |  | PK | Khoá chính |
| 3 | Số lượng sinh viên tối đa | Capacity | String |  |  |  |
| 4 | Mã bài thi | Exam\_id | Date |  | FK |  |
| 5 | Mã giám thị | Invigilator\_id | String |  | FK |  |
| 6 | Danh sách mã sinh viên | Students | String |  |  | Chứa mã sinh viên các sinh viên có trong phòng thi |
| 7 | Danh sách camera | Cameras | String |  |  | Chứa mã camera có trong phòng |
| 8 | Danh sách máy tính | Computers | String |  |  | Chứa mã máy tính có trong phòng |

Bảng 2.x: Dữ liệu lưu trữ thông tin phòng thi

* + 1. Quy trình đối chiếu sinh viên với danh sách phòng thi

Cụ thể các bước thực hiện:

Bước 1: Tạo trang hiển thị phòng thi và danh sách sinh viên trong phòng thi

Đây là bước khá đơn giản, khi chúng ta chỉ cần tạo ra các dữ liệu bao gồm các phòng thi, danh sách sinh viên, và sinh viên và sử dụng mô hình MVC để cho phép hiển thị nó lên giao diện Web của hệ thống.

* Khởi tạo các dữ liệu liên quan đến phòng thi, danh sách sinh viên.
* Sử dụng mô hình MVC của Java Spring Boot để có thể tạo các câu truy vấn và xử lý dữ liệu.
* Hiển thị nó trên giao diện.

Bước 2: Trích xuất danh sách sinh viên thành file Excel

Đây là bước quan trọng, hệ thống sẽ cần phải trích xuất file Excel để phục vụ cho các bước tiếp theo.

* Sử dụng thư viện hỗ trợ để có thể trích xuất được file Excel (XSSWorkbook).
* Tạo các câu lệnh để có thể trích xuất đúng các trường vào file Excel.
* Đặt tên và trích xuất.

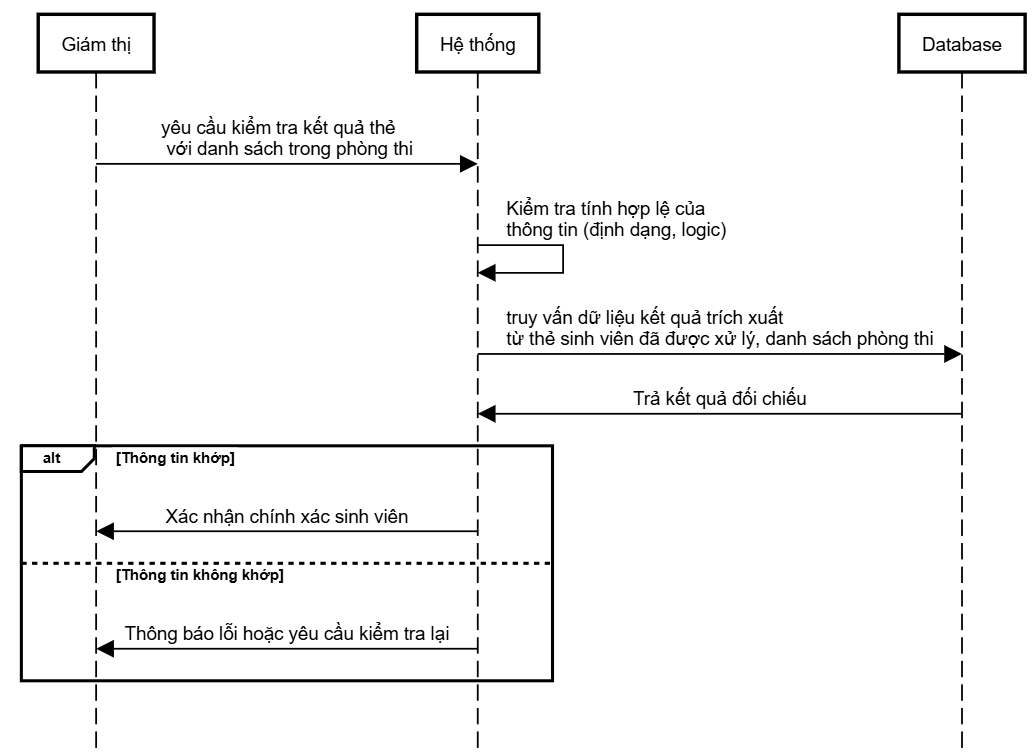
Bước 3: Nhập file Excel và hiển thị vào giao diện của “Kiểm tra khuôn mặt và trích xuất thông tin thẻ sinh viên”

Sau khi đã có được file Excel danh sách sinh viên, các giám thị có thể tùy ý linh hoạt đưa nó vào trong giao diện “Kiểm tra khuôn mặt và trích xuất thông tin thẻ sinh viên” để có thể dễ dàng đối chiếu với từng sinh viên 1 mà phòng thi mình trực.

* Nhận file Excel từ giao diện bằng cách xử lý Multipart Request.
* Đọc nội dung file Excel bằng thư viện (Apache POI hoặc EasyExcel).
* Đưa lên giao diện với các trường trong file Excel.
  + 1. Thiết kế quy trình nghiệp vụ

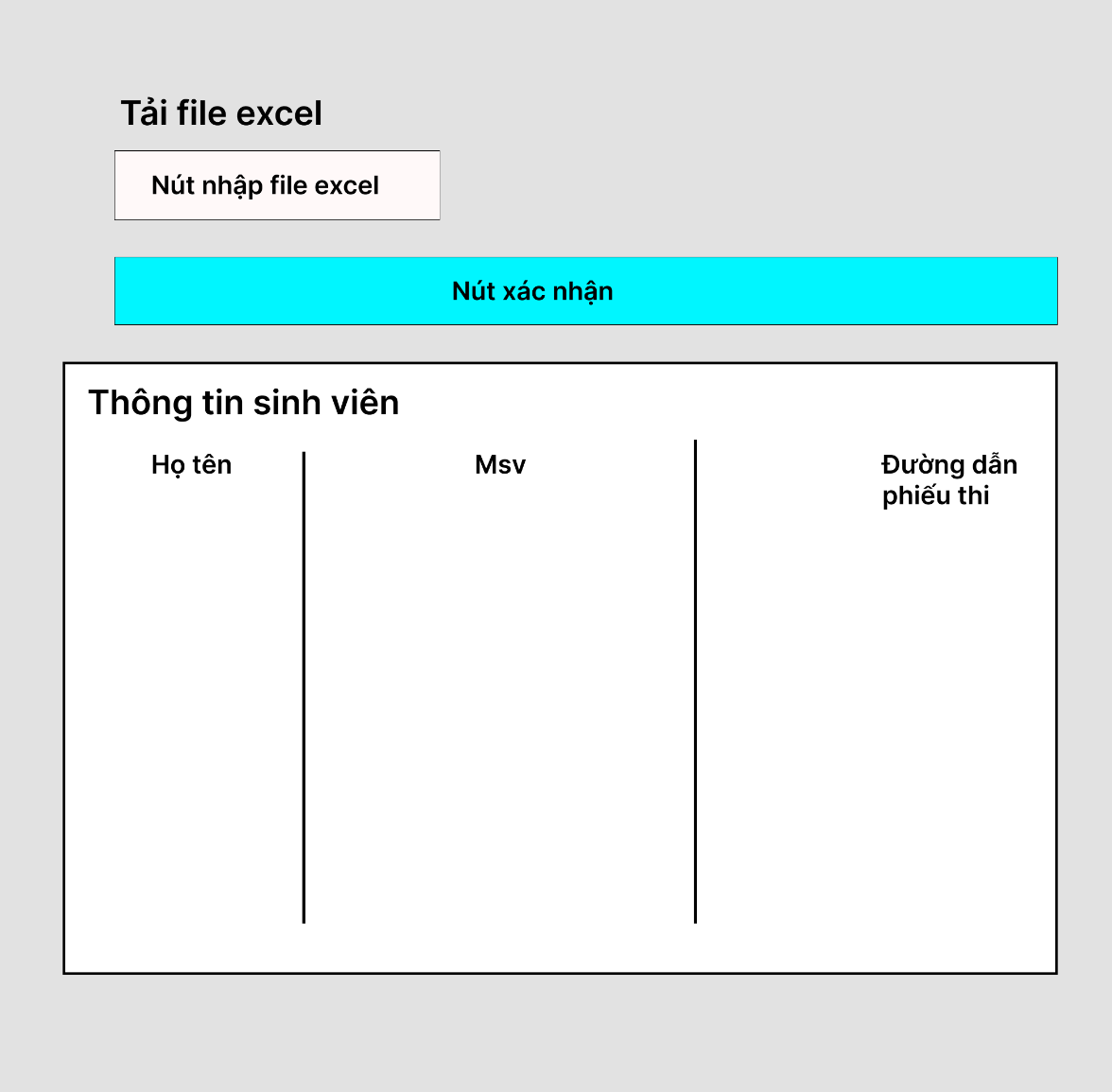
|  |  |
| --- | --- |
| **Mục đích** | Đảm bảo có thể giúp giám thị dễ dàng kiểm tra sinh viên |
| **Các bước thực hiện** | **Bước 1:** Tạo trang hiển thị phòng thi và danh sách sinh viên trong phòng thi đó.  **Bước 2:** Trích xuất danh sách sinh viên thành file Excel.  **Bước 3:** Nhập file Excel và hiển thị nó vào giao diện của “Kiểm tra khuôn mặt và trích xuất thông tin thẻ sinh viên”. |
| **Các bước**  **bổ sung** | * Áp dụng các công nghệ của Java Spring Boot để có thể dễ dàng quản lý dữ liệu, trích xuất danh sách sinh viên thành file Excel/ |
| **Các ngoại lệ** | * Thông tin danh sách sinh viên bị sai. * Không thể trích xuất được danh sách sinh viên thành file Excel. |
| **Tài liệu liên quan** | Java Spring Boot, thư viện XSSWorkbook |
| **Tham chiếu** | Mục 1.6.1.2: Chức năng đối chiếu sinh viên với danh sách phòng thi  Mục 3.3.1.2: Mô-đun đối chiếu sinh viên với danh sách phòng thi. |

Bảng 2.x. Bảng chức năng đối chiếu với danh sách sinh viên trong phòng thi



Hình 2.x: Biểu đồ tuần tự sprint 2

* + 1. Thiết kế giao diện



Hình 2.x: Thiết kế giao diện của chức năng đối chiếu với các sinh viên trong phòng thi

* 1. Sprint 3: Đối chiếu khuôn mặt trên thẻ với ảnh thực tế
     1. Giới thiệu Sprint 3

Sprint 3 của đề tài nhằm mục đích thực hiện chức năng đối chiếu khuôn mặt trên thẻ sinh viên với khuôn mặt thực tế của sinh viên trong quá trình xác minh danh tính. Đây là một bước quan trọng trong quy trình xác minh, giúp tăng tính chính xác và độ tin cậy của hệ thống trong việc nhận diện và xác thực sinh viên tham gia kỳ thi.

Thời gian thực hiện trong vòng 12 ngày từ 27/10/2024 – 8/11/2024 với các công cần làm:

* B1: Thu nhập và chuẩn bị dữ liệu mẫu (Đặng Ngọc Anh phụ trách – 2 ngày)
* B2: Xây dựng chức năng lõi đối chiếu khuôn mặt thẻ với khuôn mặt thực tế (Đặng Ngọc Anh phụ trách – 5 ngày)
* B3: Tạo giao diện và kết hợp chức năng đối chiếu vào giao diện thông qua backend, truy vấn với cơ sở dữ liệu (Đặng Ngọc Anh và Thái Văn Tuấn phụ trách – 3 ngày)
* B4: Kiểm thử chức năng đối chiếu khuôn mặt với thẻ sinh viên (Đặng Ngọc Anh và Thái Văn Tuấn phụ trách – 2 ngày)

A diagram of a person with a circle with text

Description automatically generated with medium confidence

Hình 2.x. Biểu đồ ca sử dụng của giám thị với sprint 3

A diagram of a person with text

Description automatically generated

Hình 2.x. Biểu đồ ca sử dụng của sinh viên trong sprint 3

* + 1. Thiết kế dữ liệu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Diễn giải | Tên trường | Kiểu | Mặc định | Ràng buộc | Ghi chú |
| 1 | MongoDB tự tạo | \_id | ObjectId  (auto) |  |  | Đối tượng MongoDb tự sinh |
| 2 | Mã sinh viên | Std\_id | String |  | PK | Khoá chính |
| 3 | Mã phòng thi | Room\_id | String |  | FK |  |
| 4 | Đường dẫn ảnh thẻ | Card\_image | String |  |  |  |
| 5 | Đường dẫn ảnh khuôn mặt đã cắt | Face\_image | String |  |  |  |
| 6 | Đường dẫn ảnh qua camera | Face\_camera | String |  |  |  |
| 7 | Kết quả so sánh khuôn mặt | Face\_result | Boolean |  |  |  |
| 8 | Thời gian kiểm tra sinh viên | Verification\_time | date |  |  |  |
| 9 | Phiếu thi của sinh viên | Exam\_ticket | Object |  |  |  |

Bảng 2.x: Collection lưu trữ thông tin kết quả kiểm tra trước khi thi

* + 1. Quy trình đối chiếu khuôn mặt

Các bước thực hiện:

1) Bước 1: Chuẩn bị dữ liệu đầu vào

Đây là bước để đảm bảo có thể lấy được 2 hình ảnh quan trọng đó là *ảnh sinh viên trên thẻ và ảnh sinh viên được chụp thực tế*. phải đảm bảo các ảnh phải được cắt đúng khuôn mặt và chính xác để chuẩn cho các bước tiếp theo.

2) Bước 2: Xây dựng chương trình đối chiếu khuôn mặt

Bằng sự hỗ trợ của ngôn ngữ Python và các thư viện hỗ trợ, chúng tôi đã sử dụng thuật toán khoảng cách Euculidđể có thể tiền xử lý việc so sánh và cho kết quả.

3) Bước 3: Đẩy kết quả lên giao diện

Sau khi xử lý xong chương trình, kết quả sẽ được lưu và được đẩy lên giao diện để giám thị có thể xác nhận và thông báo kết quả.

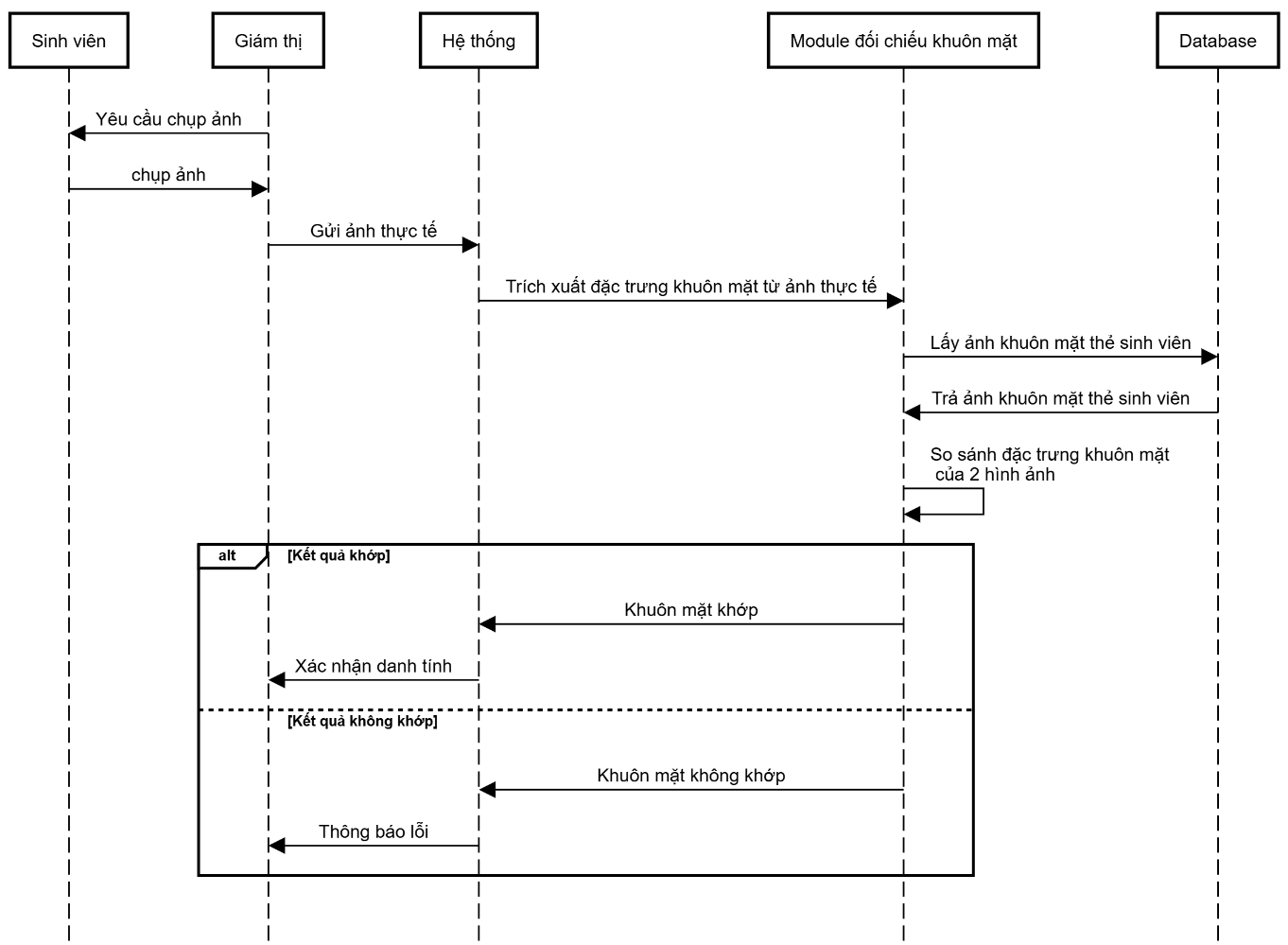
* + 1. Thiết kế quy trình nghiệp vụ

Quy trình đối chiếu khuôn mặt trong Sprint 3 sẽ bao gồm các bước sau để đảm bảo tính chính xác và hiệu quả trong việc xác minh danh tính của sinh viên qua việc so khớp khuôn mặt trên thẻ sinh viên với khuôn mặt thực tế.

1) So sánh khuôn mặt: Sau khi trích xuất khuôn mặt từ cả hai ảnh, hệ thống sử dụng phương pháp so khớp khuôn mặt, như tính toán độ tương đồng giữa các đặc trưng khuôn mặt (sử dụng Face Recognition, Deep Learning hoặc các mô hình học máy khác). Nếu độ tương đồng giữa hai khuôn mặt đạt mức chấp nhận được, hệ thống sẽ xác nhận danh tính của sinh viên. Nếu không khớp, hệ thống sẽ thông báo và yêu cầu kiểm tra lại.

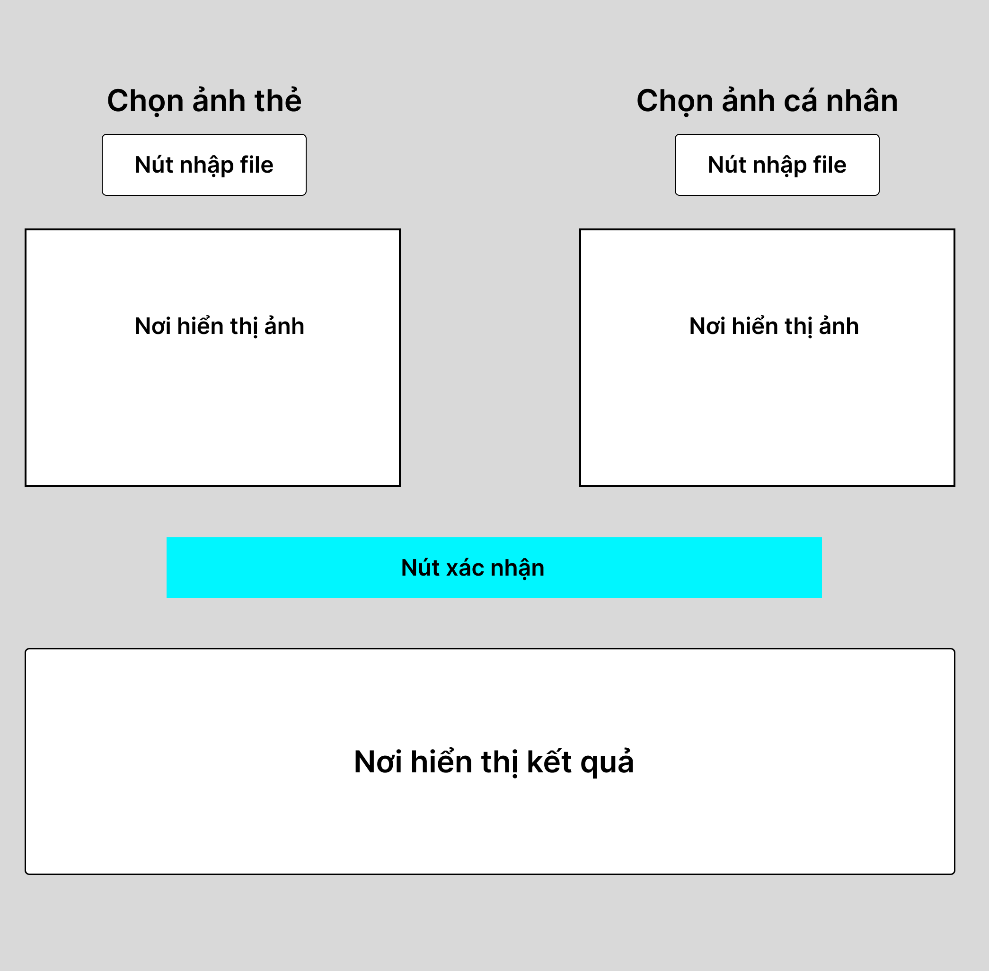
2) Cập nhật kết quả và lưu trữ: Nếu khuôn mặt khớp, hệ thống sẽ lưu lại kết quả và cập nhật vào cơ sở dữ liệu, xác nhận danh tính sinh viên. Nếu không khớp, hệ thống sẽ ghi lại kết quả và gửi thông báo cho giám thị để kiểm tra lại thủ công.

3) Thông báo kết quả: Sau khi quá trình đối chiếu hoàn tất, hệ thống sẽ thông báo kết quả cho sinh viên và giám thị qua giao diện người dùng, xác nhận thành công hoặc thất bại trong việc đối chiếu khuôn mặt.



Hình 2.x: Biểu đồ tuần tự sprint 3

* + 1. Thiết kế giao diện



Hình 2.x: Giao diện của đối chiếu khuôn mặt

* 1. Sprint 4: Xuất phiếu thi và giám sát quy trình thi
     1. Giới thiệu Sprint 4

Sprint 4 tập trung vào hai nhiệm vụ chính:

Xuất phiếu thi: Hệ thống sẽ tạo phiếu thi cá nhân cho sinh viên dựa trên thông tin đã lưu trong cơ sở dữ liệu, bao gồm thông tin cá nhân và lịch thi cụ thể. Phiếu thi sẽ được xuất dưới dạng PDF và có thể in trực tiếp hoặc gửi qua email cho sinh viên.

Giám sát quy trình thi: Phát triển Mô-đun giám sát quá trình thi nhằm phát hiện các hành vi gian lận, xác minh danh tính sinh viên và theo dõi vị trí thi theo thời gian thực. Mô-đun này sẽ kết hợp nhận diện khuôn mặt và giám sát từ camera để đảm bảo minh bạch và công bằng trong kỳ thi.

* + 1. Thiết kế dữ liệu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Diễn giải | Tên trường | Kiểu | Mặc định | Ràng buộc | Ghi chú |
| 1 | MongoDB tự tạo | \_id | ObjectId  (auto) |  |  | Đối tượng MongoDb tự sinh |
| 2 | Mã sinh viên | Std\_id | String |  | PK | Khoá chính |
| 3 | Mã phòng thi | Room\_id | String |  | FK |  |
| 4 | Đường dẫn ảnh thẻ | Card\_image | String |  |  |  |
| 5 | Đường dẫn ảnh khuôn mặt đã cắt | Face\_image | String |  |  |  |
| 6 | Đường dẫn ảnh qua camera | Face\_camera | String |  |  |  |
| 7 | Kết quả so sánh khuôn mặt | Face\_result | Boolean |  |  |  |
| 8 | Thời gian kiểm tra sinh viên | Verification\_time | date |  |  |  |

* + 1. Quy trình xuất phiếu thi

Quy trình xuất phiếu thi được thực hiện tự động từ cơ sở dữ liệu và thông tin sinh viên. Các bước chính:

Truy xuất dữ liệu sinh viên: Hệ thống truy vấn thông tin từ collection Students trong MongoDB.

Các trường thông tin gồm: std\_id, std\_name, std\_class, std\_faculty, std\_course, và lịch thi từ collection liên quan (ví dụ: ExamSchedules).

Tạo phiếu thi:

* Dữ liệu sẽ được định dạng vào mẫu phiếu thi chuẩn.
* Bao gồm mã sinh viên, họ tên, lớp, ngày thi, giờ thi, địa điểm và mã phòng thi.

Xuất file PDF: Phiếu thi được tạo dưới dạng file PDF có mã QR chứa thông tin mã sinh viên và lịch thi để hỗ trợ kiểm tra nhanh tại phòng thi.

Gửi phiếu thi: Hệ thống gửi phiếu thi qua email sinh viên hoặc cung cấp link tải trực tiếp từ cổng thông tin sinh viên.

Kiểm tra và xác nhận: Sinh viên kiểm tra phiếu thi và xác nhận thông tin chính xác. Nếu có sai sót, gửi yêu cầu chỉnh sửa trực tiếp từ hệ thống.

* + 1. Thiết kế quy trình nghiệp vụ
    2. Mô hình quy trình xuất phiếu thi

Bắt đầu: Kích hoạt quy trình bởi cán bộ quản lý hoặc hệ thống tự động trước kỳ thi.

Truy vấn dữ liệu sinh viên:

* Dữ liệu lấy từ MongoDB thông qua Java Spring Boot.
* Kiểm tra tính toàn vẹn và độ chính xác của thông tin sinh viên.

Tạo phiếu thi: OpenCV hoặc các thư viện xử lý PDF để tạo file PDF có mã QR.

Gửi phiếu: Gửi email thông báo hoặc tải phiếu thi từ hệ thống.

Phản hồi: Tiếp nhận phản hồi từ sinh viên về phiếu thi qua giao diện web.

Kết thúc: Phiếu thi được lưu trữ vào hệ thống, sẵn sàng cho quá trình kiểm tra và giám sát thi

* + 1. Thiết kế giao diện

Text

* 1. Sprint 5: Các chức năng quản lý của quản trị viên

Text

* + 1. Giới thiệu Sprint 5

Text

* + 1. Thiết kế dữ liệu

Text

* + 1. Quy trình đối chiếu khuôn mặt

Text

* + 1. Thiết kế quy trình nghiệp vụ

Text

* + 1. Thiết kế giao diện

Text

* 1. Biều đồ ca sử dụng

A diagram of a diagram

Description automatically generatedHình 2.x. Biểu đồ ca sử dụng

1. XÂY DỰNG HỆ THỐNG VẬN HÀNH VÀ QUẢN LÝ QUY TRÌNH THI TRẮC NGHIỆM KHÁCH QUAN
   1. Môi trường phát triển
      1. Giới thiệu công nghệ và công cụ sử dụng
         1. Python FastAPI

Python FastAPI là một framework hiện đại, nhanh và mạnh mẽ dùng để xây dựng các API với Python. FastAPI được đánh giá cao nhờ hiệu suất cao, dễ sử dụng, và hỗ trợ tạo tài liệu tự động cho các API. Framework này được thiết kế để tận dụng khả năng đồng bộ và bất đồng bộ của Python, giúp tối ưu hóa tốc độ xử lý và cải thiện hiệu quả của ứng dụng. Với cú pháp rõ ràng và tài liệu dễ hiểu, FastAPI rất phù hợp cho các ứng dụng yêu cầu tốc độ phản hồi nhanh và khả năng mở rộng linh hoạt, chẳng hạn như hệ thống quản lý và vận hành quy trình thi trắc nghiệm khách quan.

* + - 1. Java Spring Boot

Java Spring Boot là một framework mạnh mẽ và phổ biến để phát triển các ứng dụng web và API backend. Được xây dựng trên nền tảng Spring Framework, Spring Boot giúp đơn giản hóa quá trình thiết lập và cấu hình đề tài Java, giúp cho việc phát triển ứng dụng trở nên nhanh chóng và hiệu quả. Spring Boot cung cấp các công cụ tự động hóa, tạo cấu trúc đề tài sẵn sàng sử dụng, và hỗ trợ xử lý dữ liệu linh hoạt. Với khả năng tích hợp tốt với cơ sở dữ liệu như MongoDB và các công nghệ bảo mật mạnh mẽ, Spring Boot là lựa chọn lý tưởng để xây dựng backend của hệ thống quản lý quy trình thi trắc nghiệm, đảm bảo tốc độ, tính mở rộng, và bảo mật cho hệ thống.

* + - 1. Hệ quản trị cơ sở dữ liệu MongoDB Atlas

MongoDB Atlas là một dịch vụ cơ sở dữ liệu đám mây được quản lý hoàn toàn, được thiết kế để làm việc với các ứng dụng hiện đại đòi hỏi khả năng mở rộng và linh hoạt. Atlas cung cấp một môi trường đáng tin cậy và an toàn để lưu trữ, quản lý và truy vấn dữ liệu NoSQL, đồng thời hỗ trợ tích hợp dễ dàng với các công cụ khác trong hệ sinh thái phát triển.

Việc sử dụng MongoDB Atlas cho đề tài giúp quản lý dữ liệu sinh viên và kết quả thi một cách hiệu quả, đảm bảo hệ thống có thể mở rộng quy mô và xử lý các yêu cầu cao về tốc độ cũng như tính sẵn sàng. Atlas cung cấp các tính năng như sao lưu tự động, quản lý bảo mật nâng cao, và khả năng giám sát hệ thống theo thời gian thực, giúp tối ưu hóa hiệu suất và bảo vệ dữ liệu quan trọng.

* + 1. Quản lý mã nguồn bằng Github

GitHub là một nền tảng quản lý mã nguồn và hợp tác phát triển dựa trên hệ thống kiểm soát phiên bản Git. Việc sử dụng GitHub trong đề tài giúp nhóm phát triển theo dõi sự thay đổi của mã nguồn, quản lý các phiên bản và hợp tác hiệu quả hơn khi làm việc nhóm.

GitHub hỗ trợ việc phân nhánh (branching) và hợp nhất (merging) dễ dàng, giúp phát triển và thử nghiệm các tính năng mới mà không làm gián đoạn phiên bản chính của hệ thống. Ngoài ra, GitHub cung cấp công cụ để quản lý vấn đề (issues), theo dõi tiến độ, và tích hợp CI/CD để tự động hoá việc kiểm thử và triển khai ứng dụng, đảm bảo quy trình phát triển liên tục và linh hoạt.

* + 1. Phân chia công việc, quản lý tiến độ đề tài bằng Trello

Trello là một công cụ quản lý đề tài trực tuyến dựa trên phương pháp Kanban, giúp theo dõi tiến độ công việc một cách trực quan và hiệu quả. Trong đề tài này, Trello được sử dụng để phân chia công việc giữa các thành viên trong nhóm, từ đó đảm bảo mọi người có thể theo dõi tiến độ công việc và điều chỉnh kế hoạch khi cần thiết.

Mỗi công việc sẽ được đại diện dưới dạng các thẻ (cards) trong các cột (lists) như "To Do", "In Progress" và "Completed", giúp dễ dàng xác định tình trạng hiện tại của từng nhiệm vụ. Các nhiệm vụ được phân chia theo các giai đoạn của đề tài, ví dụ như phân tích yêu cầu, phát triển tính năng, kiểm thử và triển khai.

Trello cũng cho phép thêm các deadline, bình luận, đính kèm tài liệu và phân công nhiệm vụ cho các thành viên cụ thể, tạo nên một không gian làm việc hợp tác và có tổ chức. Bằng cách này, Trello giúp nhóm theo dõi tiến độ đề tài, đảm bảo đúng thời hạn và tối ưu hóa hiệu quả công việc.

A screenshot of a computer

Description automatically generated

Hình 3.x: Công cụ quản lý Trello

* 1. Chức năng xử lý ảnh đầu vào và trích xuất thông tin trên thẻ sinh viên

Trong quá trình phát triển hệ thống, một trong những chức năng quan trọng nhất là xử lý ảnh đầu vào và trích xuất thông tin từ thẻ sinh viên. Để đảm bảo quá trình nhận diện và trích xuất thông tin chính xác, hệ thống sẽ được chia thành các Mô-đun chuyên biệt. Dưới đây là quy trình hoàn thiện chức năng

* + 1. Xây dựng các thành phần liên quan
       1. Mô-đun xử lý ảnh đầu vào

Mô-đun xử lý ảnh đầu vào sẽ chịu trách nhiệm tiếp nhận ảnh từ nhiều nguồn khác nhau (chẳng hạn như ảnh tải lên từ người dùng hoặc ảnh chụp trực tiếp qua camera) và thực hiện các bước tiền xử lý để chuẩn bị ảnh cho quá trình nhận diện.

**Các chức năng chính của mô-đun này bao gồm:**

Tiếp nhận ảnh đầu vào:

* Hệ thống có thể nhận ảnh thẻ sinh viên từ nhiều nguồn như ảnh tải lên hoặc ảnh chụp trực tiếp qua camera của thiết bị di động hoặc máy tính.
* Mô-đun cần phải xác định định dạng ảnh hợp lệ và kích thước ảnh để xử lý.

Tiền xử lý ảnh:

* Chuyển đổi màu sắc: Hệ thống có thể chuyển ảnh về ảnh xám để giảm độ phức tạp cho các bước xử lý sau này.
* Làm sạch ảnh: Bao gồm việc loại bỏ các nhiễu nền, cải thiện độ sáng tối của ảnh, và làm sắc nét ảnh nếu cần thiết.
* Cắt ảnh thẻ sinh viên: Nếu ảnh đầu vào là ảnh lớn hoặc có nhiều yếu tố khác, hệ thống sẽ cần phải cắt phần ảnh chứa thẻ sinh viên.
* Chuyển hướng ảnh: Nếu ảnh thẻ sinh viên được chụp nghiêng, hệ thống sẽ cần quay ảnh sao cho thẻ sinh viên trở về vị trí ngang.

Giảm thiểu độ nhiễu và tối ưu hóa cho nhận diện:

* Sử dụng các kỹ thuật như lọc Gaussian, làm mịn ảnh, để giảm thiểu các nhiễu nền và cải thiện độ chính xác của các thuật toán nhận diện.

import cv2

import numpy as np

import os

PROCESSED\_FOLDER = "processed\_images"

def preprocess\_image(image\_path: str) -> str:

image = cv2.imread(image\_path)

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

\_, thresh = cv2.threshold(gray, 150, 255, cv2.THRESH\_BINARY)

processed\_image\_path = os.path.join(PROCESSED\_FOLDER, f"processed\_{os.path.basename(image\_path)}")

cv2.imwrite(processed\_image\_path, thresh)

return processed\_image\_path

Hình 3.x: Phần lập trình mô-đun xử lý ảnh đầu vào

* + - 1. Mô-đun trích xuất thông tin trên thẻ sinh viên

Sau khi ảnh đã được tiền xử lý, Mô-đun trích xuất thông tin sẽ chịu trách nhiệm nhận diện và trích xuất các dữ liệu quan trọng từ thẻ sinh viên. Mục tiêu của Mô-đun này là thu thập các thông tin như mã số sinh viên, họ tên, ngày sinh, ngành học, và các thông tin khác từ ảnh thẻ sinh viên.

**Các chức năng chính của mô-đun này bao gồm:**

Nhận diện và trích xuất thông tin văn bản (OCR): Sử dụng công nghệ nhận dạng văn bản quang học (OCR) để nhận diện và trích xuất các trường thông tin trên thẻ sinh viên. Các trường thông tin này có thể bao gồm: Mã số sinh viên, họ và tên sinh viên ngành học, khoá học, đơn vị

Sử dụng công cụ OCR: Mô-đun này sẽ tích hợp công cụ OCR để thực hiện quá trình nhận diện và chuyển đổi hình ảnh văn bản thành văn bản số. Các thư viện OCR phổ biến như Tesseract hoặc PaddleOCR có thể được sử dụng.

Tiền xử lý văn bản:

* Sau khi trích xuất văn bản từ ảnh, hệ thống sẽ cần xử lý lại văn bản để đảm bảo chính xác. Điều này bao gồm việc loại bỏ các ký tự không cần thiết, chuẩn hóa các ký tự và định dạng ngày tháng
* Loại bỏ các từ ngữ dư thừa, sai chính tả, và làm sạch dữ liệu.

Trích xuất các trường thông tin quan trọng:

* Mã số sinh viên: Cần đảm bảo rằng mã số sinh viên được nhận diện chính xác, tránh nhầm lẫn với các mã số khác.
* Họ tên sinh viên: Hệ thống cần xác định chính xác họ tên sinh viên để đối chiếu với các thông tin khác.
* Các thông tin khác: Các trường thông tin như ngành học, khoá học, đơn vị học, email và số điện thoại cũng cần phải được nhận diện chính xác từ ảnh thẻ sinh viên.

Kiểm tra tính hợp lệ thông tin trích xuất: Sau khi trích xuất, hệ thống sẽ kiểm tra tính hợp lệ của các trường thông tin (ví dụ: kiểm tra định dạng email, số điện thoại, mã số sinh viên) để đảm bảo tính chính xác.

Công cụ/Thư viện sử dụng:

* Tesseract OCR hoặc PaddleOCR: Dùng để nhận diện văn bản từ ảnh.
* Python Regular Expressions (Regex): Dùng để xử lý và chuẩn hóa văn bản đã trích xuất, kiểm tra định dạng email, số điện thoại, v.v.
* Pandas: Dùng để lưu trữ và xử lý các dữ liệu thông tin sinh viên dưới dạng bảng trong cơ sở dữ liệu.

# Hàm chính để xử lý ảnh

def process\_image(image\_path):

    # Lấy kết quả OCR từ PaddleOCR

    result = extract\_text\_regions(image\_path)

    if not result:

        return None

    fields = {}

    # Duyệt qua các bounding boxes và xử lý từng đoạn văn bản

    for line in result:

        points = line[0]

        text = line[1][0]  # Lấy văn bản từ mỗi bounding box

        # Trích xuất và tiền xử lý các đoạn văn bản

        pts = np.array(points, dtype=np.int32)

        rect = cv2.boundingRect(pts)

        x, y, w, h = rect

        cropped\_img = cv2.imread(image\_path)[y:y+h, x:x+w]

        # Tiền xử lý ảnh (phẳng hóa, làm sắc nét, v.v...)

        preprocessed\_img = preprocess\_text\_region(cropped\_img)

        # Nhận diện văn bản với VietOCR

        recognized\_text = recognize\_text\_with\_vietocr(preprocessed\_img)

        logging.info(f"Nhận diện văn bản: {recognized\_text}")

        # Cập nhật kết quả vào các trường

        fields = extract\_info\_from\_ocr(recognized\_text)

    return fields

# Đường dẫn đến ảnh thẻ sinh viên

image\_path = r"D:\Edu\Python\StudentID\_FaceVerification\student-id-face-matching\test\imgTest\ThaiTuanIDCard.jpg"

# Gọi hàm xử lý ảnh

fields = process\_image(image\_path)

if fields:

    print(f"Thông tin đã trích xuất: {fields}")

else:

    print("Không có thông tin trích xuất được.")

Hình

 # Tiền xử lý ảnh (phẳng hóa, làm sắc nét, v.v...)

        preprocessed\_img = preprocess\_text\_region(cropped\_img)

        # Nhận diện văn bản với VietOCR

        recognized\_text = recognize\_text\_with\_vietocr(preprocessed\_img)

        logging.info(f"Nhận diện văn bản: {recognized\_text}")

        # Cập nhật kết quả vào các trường

        fields = extract\_info\_from\_ocr(recognized\_text)

    return fields

# Đường dẫn đến ảnh thẻ sinh viên

image\_path = r"D:\Edu\Python\StudentID\_FaceVerification\student-id-face-matching\test\imgTest\ThaiTuanIDCard.jpg"

# Gọi hàm xử lý ảnh

fields = process\_image(image\_path)

if fields:

    print(f"Thông tin đã trích xuất: {fields}")

else:

    print("Không có thông tin trích xuất được.")

Hình 3.x. Phần lập trình mô-đun trích xuất văn bản

* + 1. Giao diện chức năng xử lý ảnh đầu vào và trích xuất thông tin trên thẻ



* + - 1. Giới thiệu giao diện chính của chức năng sau khi chạy chương trình

Giao diện sẽ gồm các thành phần như sau:

1. Khu vực tải ảnh:

* Nút tải ảnh: Cho phép người dùng tải ảnh lên để hệ thống xử lý.
* Hiển thị ảnh đầu vào: Sau khi tải ảnh, hệ thống sẽ hiển thị ảnh trên giao diện, giúp người dùng thấy rõ bức ảnh cần xử lý.

1. Khu vực kết quả nhận diện:

* Kết quả trích xuất thông tin: Hiển thị các thông tin trích xuất được từ ảnh (ví dụ như tên sinh viên, mã thẻ, ngày sinh, v.v.).
* Khu vực thông báo trạng thái: Thông báo về tình trạng nhận diện, ví dụ như "Nhận diện thành công", "Không nhận diện được thông tin", v.v.

1. Các nút chức năng:

* Xử lý ảnh: Sau khi tải ảnh lên, người dùng có thể nhấn nút này để hệ thống tiến hành nhận diện và trích xuất thông tin.
  + - 1. Hướng dẫn thực hiện với giao diện

Khi người dùng mở giao diện, họ sẽ thực hiện các bước sau:

1) Bước 1: Tải ảnh lên

* Nhấn vào nút "Chọn ảnh" để chọn một ảnh từ máy tính.
* Sau khi ảnh được tải lên, ảnh sẽ được hiển thị ngay lập tức trên giao diện.
* Cuối cùng nhấn nút “Gửi ảnh” để tải lên cho hệ thống xử lý

2) Bước 2: Xử lý ảnh người dùng đã tải lên

* Hệ thống sẽ bắt đầu phân tích và trích xuất các thông tin từ thẻ trong ảnh.
* Quá trình này có thể mất vài giây tùy vào độ phức tạp của ảnh.

3) Bước 3: Xem kết quả hệ thống trả về

* Sau khi xử lý xong, kết quả trích xuất (tên sinh viên, mã số sinh viên, ngày sinh, v.v.) sẽ được hiển thị trong khu vực kết quả.
* Người dùng có thể kiểm tra thông tin.

4) Bước 4: Thực hiện lại hoặc thoát

* Người dùng thực hiện lại bước 1 để tiến hành xử lý ảnh mới.
* Nếu muốn thoát khỏi chương trình, người dùng có thể nhấn nút "Thoát".
  + 1. Kiểm thử chức năng

Dưới đây là bảng trình bày các thử nghiệm chi tiết cho chức năng xử lý ảnh đầu vào và trích xuất thông tin trên thẻ sinh viên. Các ca thử nghiệm này bao gồm mục tiêu kiểm thử, kịch bản kiểm thử, kết quả mong đợi và kết quả thực tế để đảm bảo hệ thống hoạt động đúng như mong đợi trong các tình huống khác nhau.

Bảng dưới đây sẽ giúp bạn kiểm tra từng chức năng của hệ thống một cách cụ thể và có cơ sở để so sánh kết quả thực tế với kết quả mong đợi.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mục tiêu kiểm thử | Kịch bản kiểm thử | Kết quả mong đợi | Kết quả thực tế |
| 1 | Kiểm thử tải ảnh lên | Nhấn nút “Tải ảnh” và chọn ảnh hợp lệ từ máy | Ảnh được tải lên thành công và hiển thị đúng trên giao diện | Như kết quả mong đợi |
| 2 | Kiểm thử tải ảnh với định dạng không hợp lệ | Chọn file có định dạng không phải là ảnh (ví dụ: .txt, .docx) để tải lên | Hệ thống thông báo lỗi về định dạng | Thông báo định dạng không hợp lệ |
| 3 | Kiểm thử xử lý ảnh với chất lượng cao | Sau khi tải ảnh lên. Kiểm tra kết quả trích xuất được từ ảnh rõ nét | Thông tin (họ tên sinh viên, mã số sinh viên, khoá học, ngành học,…) được trích xuất chính xác | Các thông tin được trích xuất đúng với thực tế |
| 4 | Kiểm thử xử lý với ảnh chất lượng kém | Tải ảnh bị mờ, nhiễu lên. Kiểm tra kết quả trích xuất | Hệ thống thông báo lỗi không trích xuất được thông tin | Trả về kết quả sai hoặc chưa đầy đủ. Không thông báo lỗi |
| 5 | Kiểm thử thông báo lỗi khi không thể nhận diện ảnh | Sử dụng ảnh không phải là thẻ sinh viên (ví dụ: ảnh phong cảnh) và tải ảnh lên | Hệ thống hiển thị thông báo lỗi “Không nhận diện được thông tin” | Thông báo lỗi như mong đợi |
| 6 | Kiểm thử với ảnh thẻ có định dạng khác | Sử dụng ảnh của khoá 61 (định dạng thẻ dọc), tải lên và xử lý | Hệ thống thông báo lỗi không đúng định dạng thẻ | Chưa thông báo lỗi |
| 7 | Kiểm thử giao diện người dùng (UI) | Kiểm tra tính năng các nút “Tải ảnh”, “Xử lý” trên giao diện người dùng | Giao diện dễ sử dụng, các nút chức năng hoạt động đúng và không gặp lỗi. | Giao diện dễ sử dụng, không gặp lỗi |
| 8 | Kiểm thử hệ thống phản hồi khi không có ảnh tải lên | Nhấn "Xử lý ảnh" khi chưa tải ảnh nào lên. | Hệ thống hiển thị thông báo lỗi "Chưa tải ảnh lên". | Thông báo lỗi "Chưa tải ảnh lên". |

* 1. Chức năng so khớp thông tin sinh viên với danh sách phòng thi

Chức năng này nhằm mục đích so khớp thông tin sinh viên đã đọc được ở bước xử lý ảnh đầu vào và trích xuất thông tin với danh sách thực tế có trong phòng thi, đảm bảo rằng mỗi sinh viên có thể được phân bổ và có mặt đúng phòng thi giúp việc quản lý và giám sát dễ dàng trong quá trình thi. Dưới đây là các mô-đun cần thiết để thực hiện chức năng này.

* + 1. Xây dựng các mô-đun liên quan
       1. Mô-đun đọc danh sách sinh viên từ file excel

Mục đích của Mô-đun này là đọc danh sách sinh viên từ một file Excel (ví dụ: danh sách thi) để có thể thao tác với dữ liệu và sử dụng trong các chức năng tiếp theo.

Bước thực hiện:

1. Cài đặt các thư viện cần thiết: Để đọc và xử lý file Excel, bạn cần sử dụng thư viện như pandas hoặc openpyxl trong Python. Sử dụng câu lệnh: pip install pandas openpyxl
2. Đọc file Excel và trích xuất dữ liệu: Mô-đun này sẽ đọc các thông tin từ file Excel và trả về dưới dạng một DataFrame hoặc danh sách.
3. Thông tin sinh viên cần trích xuất:

* Mã sinh viên (std\_id)
* Họ tên sinh viên (std\_name)
* Phòng thi của sinh viên (exam\_room)

1. Kết quả mong đợi: Dữ liệu từ file Excel sẽ được trích xuất thành công và trả về dưới dạng bảng, giúp dễ dàng thao tác với các thông tin.

def read\_from\_excel(excel\_path): #hàm đọc tt từ file Excel

    try:

        df = pd.read\_excel(excel\_path)

        name\_col = None

        msv\_col = None

        for col in df.columns:

            if "họ tên" in col.lower():

                name\_col = col

            elif "mã sinh viên" in col.lower():

                msv\_col = col

        if name\_col and msv\_col:

            excel\_data = []

            for index, row in df.iterrows():

                name = row[name\_col]

                msv = row[msv\_col]

                if pd.notna(name) and pd.notna(msv):

                    name = unidecode(str(name)).upper()

                    excel\_data.append(f"{name} - {msv}")

            return excel\_data

        else:

            print("Không tìm thấy cột 'Họ tên' và 'Mã sinh viên'.")

            return []

    except Exception as e:

        print(f"Lỗi khi đọc file Excel: {e}")

        return []

Hình 3.x. Phần lập trình mô-đun đọc danh sách sinh viên từ file Excel

* + - 1. Mô-đun so khớp thông tin sinh viên với danh sách phòng thi

Mô-đun này sẽ giúp đối chiếu thông tin sinh viên với phòng thi mà họ được phân bổ, nhằm kiểm tra tính chính xác của danh sách và phòng thi.

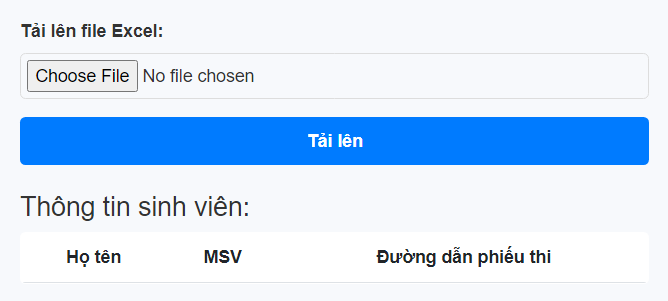
Bước thực hiện:

1. Nhập dữ liệu từ file Excel: Đọc danh sách sinh viên và thông tin phòng thi từ file Excel, như đã mô tả ở trên.
2. Đối chiếu mã sinh viên: So khớp mã sinh viên trong danh sách sinh viên với phòng thi của họ.
3. Kiểm tra các trường hợp:

* Sinh viên đã có phòng thi.
* Sinh viên không có phòng thi.
* Kiểm tra lại thông tin khi có sự thay đổi.

1. Kết quả mong đợi:

* Mô-đun sẽ trả về danh sách các sinh viên cùng với phòng thi của họ, và cho biết có sự khớp hay không.
* Các thông báo lỗi (nếu có) sẽ được đưa ra nếu sinh viên không có phòng thi hoặc có sự sai lệch.
  + 1. Giao diện chức năng so khớp thông tin sinh viên với danh sách



* + - 1. Giới thiệu giao diện chính của chức năng sau khi chạy chương trình

Giao diện sẽ gồm các thành phần như sau:

1. Khu vực tải file excel:

* Nút tải file excel: Cho phép người dùng tải file excel lên để hệ thống xử lý.
* Hiển thị đường dẫn tải lên: Sau khi tải ảnh, hệ thống sẽ hiển thị đường dẫn file trên giao diện, giúp người dùng kiểm tra file cần xử lý.

1. Khu vực kết quả trích xuất:

* Kết quả trích xuất thông tin: Hiển thị các thông tin trích xuất được từ file excel (ví dụ như họ tên sinh viên, mã số sinh viên, thông tin sinh viên, thông tin giám thị của phòng thi đó).
* Khu vực thông báo trạng thái: Thông báo về tình trạng trích xuất, ví dụ như "Trích xuất thành công", "Không nhận diện được thông tin", v.v.

1. Các nút chức năng:

* Trích xuất thông tin: Sau khi tải file lên, người dùng có thể nhấn nút này để hệ thống tiến hành nhận file và trích xuất thông tin.
  + - 1. Hướng dẫn thực hiện với giao diện

1. Khu vực tải file Excel

**Nút tải file Excel:**

Chức năng: Cho phép người dùng tải file Excel chứa danh sách sinh viên và thông tin phòng thi lên hệ thống để tiến hành xử lý.

Cách thực hiện:

* Nhấn vào nút "Tải file Excel" (thường có biểu tượng "Tải lên" hoặc "Chọn file").
* Cửa sổ chọn file sẽ mở ra. Người dùng cần chọn file Excel có định dạng .xlsx (hoặc .xls) từ máy tính của mình.
* Sau khi chọn xong, nhấn "Mở" để tải lên file.

**Hiển thị đường dẫn tải lên:**

Chức năng: Sau khi người dùng tải thành công file Excel, hệ thống sẽ hiển thị đường dẫn của file đã tải lên trên giao diện.

Cách thực hiện:

* Sau khi tải file lên, hệ thống sẽ tự động hiển thị đường dẫn hoặc tên của file trên giao diện.
* Người dùng có thể kiểm tra lại tên và đường dẫn của file để đảm bảo file đã được tải lên chính xác.

1. Khu vực kết quả trích xuất

**Kết quả trích xuất thông tin:**

Chức năng: Hiển thị các thông tin đã được trích xuất từ file Excel sau khi người dùng nhấn nút "Trích xuất thông tin".

Cách thực hiện:

* Sau khi file Excel được tải lên, nhấn nút "Trích xuất thông tin".
* Hệ thống sẽ tiến hành đọc và phân tích nội dung file Excel, trích xuất các thông tin như mã sinh viên, họ tên, thông tin phòng thi (nếu có), và thông tin giám thị.
* Các thông tin này sẽ được hiển thị trên giao diện trong khu vực kết quả trích xuất. Ví dụ: Mã sinh viên, tên sinh viên, phòng thi, giám thị, và trạng thái trích xuất.

**Khu vực thông báo trạng thái:**

Chức năng: Hiển thị trạng thái của quá trình trích xuất thông tin từ file Excel, giúp người dùng biết liệu quá trình này có thành công hay không.

Cách thực hiện: Sau khi nhấn nút "Trích xuất thông tin", hệ thống sẽ thông báo trạng thái trích xuất.

Các thông báo có thể là:

* "Trích xuất thành công": Thông tin đã được trích xuất và hiển thị đúng trên giao diện.
* "Không nhận diện được thông tin": Nếu hệ thống không thể trích xuất thông tin (do lỗi file, dữ liệu không hợp lệ, v.v.), thông báo này sẽ hiển thị.
* "Lỗi trong quá trình xử lý": Thông báo khi có lỗi hệ thống xảy ra trong quá trình đọc file.

1. Các nút chức năng

**Trích xuất thông tin:**

Chức năng: Sau khi tải file Excel lên, người dùng có thể nhấn nút "Trích xuất thông tin" để hệ thống tiến hành trích xuất thông tin từ file.

Cách thực hiện:

* Sau khi file Excel đã được tải lên và hiển thị đường dẫn, nhấn vào nút "Trích xuất thông tin".
* Hệ thống sẽ bắt đầu quá trình đọc dữ liệu từ file và trích xuất thông tin như mã sinh viên, tên sinh viên, phòng thi và giám thị.
* Sau khi hoàn thành, các thông tin sẽ được hiển thị trong khu vực kết quả trích xuất.

**Các nút chức năng khác:**

Thoát hoặc Đóng: Để đóng giao diện hoặc thoát khỏi ứng dụng.

Tải lại: Nếu người dùng muốn tải lại file mới hoặc bắt đầu lại quá trình xử lý.

* + 1. Kiểm thử chức năng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mục tiêu kiểm thử | Kịch bản kiểm thử | Kết quả mong đợi | Kết quả thực tế |
| 1 | Đảm bảo hệ thống có thể tải và xử lý file Excel hợp lệ. | 1. Nhấn nút "Tải file Excel". 2. Chọn file Excel hợp lệ từ máy tính. 3. Nhấn "Mở". | 1. Hệ thống hiển thị đường dẫn của file đã tải lên. 2. Đường dẫn file xuất hiện chính xác trên giao diện. | Như kết quả mong đợi |
| 2 | Đảm bảo hệ thống xử lý đúng các file không phải Excel. | 1. Nhấn nút "Tải file Excel". 2. Chọn file không phải file Excel (ví dụ: .txt, .pdf). 3. Nhấn "Mở". | Hệ thống hiển thị thông báo lỗi, yêu cầu người dùng chọn file Excel hợp lệ. | Thông báo định dạng không hợp lệ |
| 3 | Đảm bảo hệ thống trích xuất thông tin chính xác từ file Excel hợp lệ. | 1. Tải file Excel hợp lệ lên hệ thống. 2. Nhấn nút "Trích xuất thông tin". | 1. Hệ thống hiển thị các thông tin trích xuất từ file: mã sinh viên, họ tên, phòng thi, giám thị. 2. Trạng thái trích xuất thành công được hiển thị. | Các thông tin được trích xuất đúng với thực tế |
| 4 | Đảm bảo hệ thống xử lý khi không có dữ liệu trong file Excel. | 1. Tải file Excel không chứa dữ liệu hợp lệ (ví dụ, chỉ có tiêu đề cột mà không có thông tin). 2. Nhấn nút "Trích xuất thông tin". | Hệ thống hiển thị thông báo "Không nhận diện được thông tin". | Lỗi không thông báo, trả về giao diện trống |
| 5 | Đảm bảo hệ thống thông báo lỗi khi trích xuất dữ liệu sai. | 1. Tải file Excel có dữ liệu sai lệch (ví dụ: mã sinh viên sai, thiếu thông tin phòng thi). 2. Nhấn nút "Trích xuất thông tin". | Hệ thống hiển thị thông báo lỗi hoặc thông báo "Lỗi trong quá trình xử lý" nếu không thể trích xuất thông tin chính xác. | Thông báo lỗi như mong đợi |
| 6 | Đảm bảo hệ thống hiển thị thông báo trạng thái chính xác sau khi trích xuất thông tin thành công. | 1. Tải file Excel hợp lệ lên hệ thống. 2. Nhấn nút "Trích xuất thông tin". | Hệ thống hiển thị thông báo "Trích xuất thành công" trong khu vực thông báo trạng thái. | Chưa thông báo lỗi |
| 7 | Kiểm thử giao diện người dùng (UI) | Kiểm tra tính năng các nút “Tải file”, “Xử lý” trên giao diện người dùng | Giao diện dễ sử dụng, các nút chức năng hoạt động đúng và không gặp lỗi. | Giao diện dễ sử dụng, không gặp lỗi |
| 8 | Đảm bảo hệ thống hiển thị thông báo khi không có thông tin phòng thi trong file. | 1. Tải file Excel hợp lệ nhưng không có thông tin phòng thi. 2. Nhấn nút "Trích xuất thông tin". | Hệ thống hiển thị thông báo "Không có thông tin phòng thi" trong khu vực kết quả trích xuất. | Lỗi không thông báo, trả về giao diện trống |

* 1. Chức năng đối chiếu khuôn mặt trên thẻ sinh viên với ảnh thực tế

Chức năng này giúp xác thực danh tính sinh viên thông qua việc đối chiếu khuôn mặt trên thẻ sinh viên với hình ảnh thu được từ camera trong thời gian thực, nhằm đảm bảo tính chính xác và hạn chế gian lận.

* + 1. Xây dựng các mô-đun liên quan
       1. Mô-đun nhận diện và cắt khuôn mặt có trên thẻ sinh viên

Mục tiêu: Tự động phát hiện và cắt khuôn mặt từ ảnh thẻ sinh viên để đưa vào quá trình đối chiếu.

Quy trình:

* Đọc ảnh thẻ sinh viên: Sử dụng OpenCV để đọc ảnh từ nguồn tải lên hoặc camera quét thẻ.
* Phát hiện khuôn mặt: Sử dụng mô hình Haar Cascade, Dlib, hoặc MTCNN để phát hiện khuôn mặt trên ảnh thẻ.
* Cắt và chuẩn hóa khuôn mặt: Cắt vùng chứa khuôn mặt và chuẩn hóa kích thước về cùng tỉ lệ (ví dụ: 112x112 hoặc 224x224).
* Lưu khuôn mặt: Lưu khuôn mặt đã cắt vào bộ nhớ hoặc tạm thời lưu vào cơ sở dữ liệu để phục vụ đối chiếu.

def resize\_image(img, scale\_percent):

    width = int(img.shape[1] \* scale\_percent / 100)

    height = int(img.shape[0] \* scale\_percent / 100)

    dim = (width, height)

    return cv2.resize(img, dim)

def detect\_face(img, scale\_percent=50, expand\_ratio=0.2):

    resized\_img = resize\_image(img, scale\_percent)

    faces = face\_cascade.detectMultiScale(cv2.cvtColor(resized\_img, cv2.COLOR\_BGR2GRAY), scaleFactor=1.1, minNeighbors=5)

    if len(faces) > 0:

        (x, y, w, h) = faces[0]

        x = int(x \* (100 / scale\_percent))

        y = int(y \* (100 / scale\_percent))

        w = int(w \* (100 / scale\_percent))

        h = int(h \* (100 / scale\_percent))

        expand\_w = int(w \* expand\_ratio)

        expand\_h = int(h \* expand\_ratio)

        x = max(x - expand\_w // 2, 0)

        y = max(y - expand\_h // 2, 0)

        w = min(w + expand\_w, img.shape[1] - x)

        h = min(h + expand\_h, img.shape[0] - y)

        return img[y:y+h, x:x+w]

    return None

def resize\_image(img, scale\_percent):

    width = int(img.shape[1] \* scale\_percent / 100)

    height = int(img.shape[0] \* scale\_percent / 100)

    dim = (width, height)

    return cv2.resize(img, dim)

def detect\_face(img, scale\_percent=50, expand\_ratio=0.2):

    resized\_img = resize\_image(img, scale\_percent)

    faces = face\_cascade.detectMultiScale(cv2.cvtColor(resized\_img, cv2.COLOR\_BGR2GRAY), scaleFactor=1.1, minNeighbors=5)

    if len(faces) > 0:

        (x, y, w, h) = faces[0]

        x = int(x \* (100 / scale\_percent))

        y = int(y \* (100 / scale\_percent))

        w = int(w \* (100 / scale\_percent))

        h = int(h \* (100 / scale\_percent))

        expand\_w = int(w \* expand\_ratio)

        expand\_h = int(h \* expand\_ratio)

        x = max(x - expand\_w // 2, 0)

        y = max(y - expand\_h // 2, 0)

        w = min(w + expand\_w, img.shape[1] - x)

        h = min(h + expand\_h, img.shape[0] - y)

        return img[y:y+h, x:x+w]

    return None

Hình 3.x. Phần lập trình mô-đun trích xuất ảnh thẻ sinh viên

* + - 1. Mô-đun đối chiếu khuôn mặt đã cắt trên thẻ với khuôn mặt thực tế

Mục tiêu: So sánh khuôn mặt đã cắt từ thẻ sinh viên với khuôn mặt thu được từ camera theo thời gian thực.

Quy trình:

* Nhận diện khuôn mặt thời gian thực: Sử dụng OpenCV hoặc Dlib để nhận diện khuôn mặt từ luồng camera.
* Mã hóa khuôn mặt: Áp dụng mô hình FaceNet, Dlib (Face Recognition), hoặc DeepFace để tạo mã hóa đặc trưng cho khuôn mặt từ thẻ và ảnh thực tế.
* Đối chiếu khuôn mặt: So sánh các vector mã hóa khuôn mặt bằng cách đo khoảng cách cosine hoặc Euclidean.
* Nếu khoảng cách dưới ngưỡng xác định (ví dụ: 0.6), xác nhận là cùng một người.

Thông báo kết quả: Hiển thị kết quả xác nhận danh tính (thành công/thất bại) kèm theo tỷ lệ chính xác.

def decode\_base64\_image(base64\_string): #hàm giải mã ảnh từ base64

    image\_data = base64.b64decode(base64\_string)

    image = Image.open(BytesIO(image\_data)).convert("RGB")

    return np.array(image)

def get\_face\_embedding(image): #hàm trích xuất vector đặc trưng từ ảnh

    gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

    faces = detector(gray)

    if len(faces) == 0:

        return None

    shape = predictor(gray, faces[0])

    face\_embedding = facerec.compute\_face\_descriptor(image, shape)  #trích xuất vector đặc trưng

    return np.array(face\_embedding)

def detect\_face(img, scale\_percent=50, expand\_ratio=0.2):

    resized\_img = resize\_image(img, scale\_percent)

    faces = face\_cascade.detectMultiScale(cv2.cvtColor(resized\_img, cv2.COLOR\_BGR2GRAY), scaleFactor=1.1, minNeighbors=5)

    if len(faces) > 0:

        (x, y, w, h) = faces[0]

        x = int(x \* (100 / scale\_percent))

        y = int(y \* (100 / scale\_percent))

        w = int(w \* (100 / scale\_percent))

        h = int(h \* (100 / scale\_percent))

        expand\_w = int(w \* expand\_ratio)

        expand\_h = int(h \* expand\_ratio)

        x = max(x - expand\_w // 2, 0)

        y = max(y - expand\_h // 2, 0)

        w = min(w + expand\_w, img.shape[1] - x)

        h = min(h + expand\_h, img.shape[0] - y)

        return img[y:y+h, x:x+w]

    return None

Hình 3.x. Phần lập trình mô-đun đối chiếu khuôn mặt

def compare\_faces(uploaded\_image, face\_image\_base64, threshold=0.6): #hàm so sánh 2 ảnh bằng vector đặc trưng

    """So sánh hai khuôn mặt và trả về kết quả so sánh."""

    image1 = cv2.cvtColor(uploaded\_image, cv2.COLOR\_RGB2BGR)

    image2 = decode\_base64\_image(face\_image\_base64)

    embedding1 = get\_face\_embedding(image1)

    embedding2 = get\_face\_embedding(image2)

    if embedding1 is None:

        return "Ảnh tải lên không tìm thấy khuôn mặt"

    if embedding2 is None:

        return "Ảnh khuôn mặt từ thẻ không tìm thấy khuôn mặt"

    distance = np.linalg.norm(embedding1 - embedding2)

    if distance < threshold:

        return f"Cùng 1 người (Khoảng cách Euclidean: {distance:.2f})"

    else:

        return f"2 người khác nhau (Khoảng cách Euclidean: {distance:.2f})"

* + 1. Giao diện chức năng đối chiếu khuôn mặt trên thẻ với ảnh thực tế

Giao diện người dùng sẽ là công cụ trực quan giúp cán bộ giám sát và sinh viên dễ dàng thực hiện quy trình đối chiếu khuôn mặt nhanh chóng, chính xác trong các kỳ thi.

A screenshot of a computer

Description automatically generated

* + - 1. Giới thiệu giao diện chính chức năng

Giao diện sẽ gồm các thành phần như sau:

1. Khu vực tải ảnh thẻ sinh viên và ảnh cá nhân sinh viên vào hệ thống:

* Nút tải file ảnh: Cho phép người dùng tải file ảnh lên để hệ thống xử lý.
* Hiển thị đường dẫn tải lên: Sau khi tải ảnh, hệ thống sẽ hiển thị đường dẫn file trên giao diện, giúp người dùng kiểm tra file cần xử lý.

1. Khu vực kết quả trích xuất:

* Kết quả trích xuất thông tin: Hiển thị các thông tin trích xuất từ kết quả so sánh 2 khuôn mặt của hệ thống và thông tin trên thẻ.
  + - 1. Hướng dẫn thực hiện với giao diện

Giao diện sẽ bao gồm các thành phần như sau:

1. Khu vực đưa ảnh thẻ sinh viên và ảnh cá nhân sinh viên vào hệ thống

**Nút tải file ảnh:**

Chức năng: cho phép người dùng nhập ảnh và hiển thị 2 hình ảnh thẻ sinh viên và ảnh chụp sinh viên thực tế

Cách thực hiện:

* Nhấn vào nút "Tải file" theo 2 khu vực cho ảnh thẻ sinh viên và ảnh cá nhân (thường có biểu tượng "Tải lên" hoặc "Chọn file").
* Cửa sổ chọn file sẽ mở ra. Người dùng cần chọn file ảnh có định dạng .png (hoặc .jpg) từ máy tính của mình.
* Sau khi chọn xong, nhấn "Mở" để tải lên file.

**Hiển thị đường dẫn tải lên:**

Chức năng: Sau khi người dùng tải thành công 2 file ảnh, hệ thống sẽ hiển thị đường dẫn của file đã tải lên trên giao diện.

Cách thực hiện:

* Sau khi tải file lên, hệ thống sẽ tự động hiển thị đường dẫn hoặc tên của file trên giao diện.
* Người dùng có thể kiểm tra lại tên và đường dẫn của file để đảm bảo file đã được tải lên chính xác.

1. Khu vực kết quả trích xuất

**Kết quả trích xuất thông tin**

Chức năng: Hiển thị các thông tin đã được trích xuất từ file Excel sau khi người dùng nhấn nút "Trích xuất thông tin".

Cách thực hiện:

* Sau khi 2 file ảnh được tải lên, nhấn nút "Gửi".
* Hệ thống sẽ tiến hành đọc và xử lý 2 ảnh, sử dụng các thuật toán và chương trình đã được cài sẵn để tiến hành so sánh.
* Kết quả sẽ được hiển thị xem có phải cùng 1 người không cùng với thông tin thẻ sinh viên bên cạnh.
  + 1. Kiểm thử chức năng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mục tiêu kiểm thử | Kịch bản kiểm thử | Kết quả mong đợi | Kết quả thực tế |
| 1 | Đảm bảo hệ thống có thể tải và xử lý 2 file ảnh hợp lệ. | 1. Nhấn nút "Tải file". 2. Chọn file JPG, PNG, hay các file định dạng hình ảnh hợp lệ từ máy tính. 3. Nhấn "Mở". | 1. Hệ thống hiển thị đường dẫn của file đã tải lên. 2. Đường dẫn file xuất hiện chính xác trên giao diện. | Như kết quả mong đợi |
| 2 | Đảm bảo hệ thống xử lý đúng các file không thuộc định dạng hình ảnh | 1. Nhấn nút "Tải file". 2. Chọn file không phải file ảnh (ví dụ: .txt, .mp3). 3. Nhấn "Mở". | Hệ thống hiển thị thông báo lỗi, yêu cầu người dùng chọn file Excel hợp lệ. | Thông báo định dạng không hợp lệ |
| 3 | Đảm bảo hệ thống hiển thị kết quả chính xác sau khi so sánh thành công. | 1. Tải file ảnh hợp lệ lên hệ thống. 2. Nhấn nút "Gửi ảnh". | Hệ thống hiển thị ngay kết quả so sánh và thông tin thẻ sinh viên. | Chưa thông báo lỗi |
| 4 | Kiểm thử giao diện người dùng (UI) | Kiểm tra tính năng các nút “Tải file”, “Xử lý” trên giao diện người dùng | Giao diện dễ sử dụng, các nút chức năng hoạt động đúng và không gặp lỗi. | Giao diện dễ sử dụng, không gặp lỗi |
| 5 | Đảm bảo hệ thống hiển thị thông báo khi chương trình gặp lỗi | 1. Nhấn nút “Tải file”. 2. Nhấn nút "Gửi ảnh". | Hệ thống hiển thị thông báo "Không xử lý được" trong khu vực kết quả trích xuất. | Lỗi không thông báo, trả về giao diện trống |

* 1. Chức năng xuất phiếu và giám sát quy trình thi
     1. Xây dựng các Mô-đun liên quan
        1. Mô-đun in phiếu thi
        2. Mô-đun giám sát quy trình thi
     2. Giao diện chức năng nhận phiếu và giám sát quy trình thi
        1. Giới thiệu giao diện chính chức năng
        2. Hướng dẫn thực hiện với giao diện
     3. Kiểm thử chức năng
  2. Các chức năng quản lý của quản trị viên

KẾT LUẬN

Kết quả đạt được

Những tồn tại và hạn chế

TÀI LIỆU THAM KHẢO

[1] Lê Văn Phùng, *Kỹ nghệ phần mềm*, NXB Thông tin và Truyền thông, 2014.

[2] Lê Văn Phùng, Lê Hương Giang, *Kỹ nghệ phần mềm nâng cao*, NXB Thông tin và truyền thông, 2015.

[3] Đoàn Văn Ban, Nguyễn Thị Tĩnh, *Giáo trình phân tích thiết kế hệ thống hướng đối tượng bằng UML*, NXB Đại học sư phạm, 2011.

[4] Ian Sommerville*, Software Engineering*, Ninth Edition, Addison-Wesley, 2011.

[5] Scott Tilley, Harry J. Rosenblatt, *Systems Analys and Design*, Shelly Cashman Series, 11th Edition, 2016.

**<Thêm một vài tài liệu liên quan đến công cụ ở đây> theo định dạng sau:**

[#] Tên tác giả*, Tên sách*, phiên bản (nếu có), nhà xuất bản, năm xuất bản.