**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**BÁO CÁO BTL THUỘC HỌC PHẦN:**

**CÔNG CỤ PHÁT TRIÊN PHẦN MỀM**

**Nghiên cứu xây dựng hệ thống chấm công tự động**

**bằng công cụ Emgu, OpenCV trên Visual Studio**

|  |  |
| --- | --- |
| GVHD: | Ths Nguyễn Thái Cường |
| Nhóm - Lớp: | 2 – 20222IT6003004 |
| Thành viên: | Nguyễn Phúc Hưng |
|  | Nguyễn Khánh Thọ  Nguyễn Quang Hưng  Đặng Việt Anh Đinh Anh Quân |
|  |  |
|  |  |
|  |  |

Hà Nội, Năm 2023

# **MỤC LỤC**

[**MỤC LỤC 2**](#_Toc136937696)

[**DANH MỤC BẢNG BIỂU 6**](#_Toc136937697)

[**DANH MỤC BẢNG BIỂU 7**](#_Toc136937698)

[**LỜI NÓI ĐẦU 8**](#_Toc136937699)

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 10**](#_Toc136937700)

[**1.1 Đặt vấn đề 10**](#_Toc136937701)

[**1.2 Một số nghiên cứu trước đây 10**](#_Toc136937702)

[**1.3 Phạm vi đề tài. 12**](#_Toc136937703)

[**1.4 Phương pháp nghiên cứu 13**](#_Toc136937704)

[**1.5 Bố Cục và nội dung 13**](#_Toc136937705)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 14**](#_Toc136937706)

[**2.1 Giới thiệu về bài toán dò tìm khuôn mặt 14**](#_Toc136937707)

[**2.2 Đặc trưng Haar-Like Features 14**](#_Toc136937708)

[**2.3 Trích đặc trưng Haar-Like từ ảnh 16**](#_Toc136937709)

[**2.4 Thuật toán AdaBoost 17**](#_Toc136937710)

[**2.5 Bộ phân tầng Cascade of Boosted Classifier 18**](#_Toc136937711)

[**2.6 Định vị đối tượng 20**](#_Toc136937712)

[**2.7 Tổng quan về thư viện OpenCV để nghiên cứu về nhận dạng mặt người 21**](#_Toc136937713)

[**2.8 Thư viện EmguCV 22**](#_Toc136937714)

[**CHƯƠNG 3: YÊU CẦU CHỨC NĂNG VÀ MÔ HÌNH HÓA USECASE 23**](#_Toc136937715)

[**3.1 Yêu cầu chức năng 23**](#_Toc136937716)

[**3.2 Mô hình hóa CSDL 23**](#_Toc136937717)

[**3.2.1 Các yêu cầu về dữ liệu 23**](#_Toc136937718)

[**3.2.2 Biểu đồ thực thể liên kết 24**](#_Toc136937719)

[**3.2.3 Mô tả chi tiết các bảng CSDL 25**](#_Toc136937720)

[**CHƯƠNG 4: MÔ TẢ USECASE 28**](#_Toc136937721)

[**4.1 Danh sách Usecase 28**](#_Toc136937722)

[**4.2 Sơ đồ UC 30**](#_Toc136937723)

[**4.3 Mô tả chi tiết UC 31**](#_Toc136937724)

[**UC01: Đăng nhập 31**](#_Toc136937725)

[**UC02: Thêm tài khoản 32**](#_Toc136937726)

[**UC03: Cập nhật tài khoản 33**](#_Toc136937727)

[**UC04: Xóa tài khoản 34**](#_Toc136937728)

[**UC05: Tìm kiếm tài khoản 36**](#_Toc136937729)

[**UC06: Thêm nhân viên 37**](#_Toc136937730)

[**UC07: Cập nhật thông tin nhân viên 38**](#_Toc136937731)

[**UC08: Xóa nhân viên 40**](#_Toc136937732)

[**UC09: Tìm kiếm nhân viên 41**](#_Toc136937733)

[**UC10: Chấm công nhân viên 42**](#_Toc136937734)

[**UC11: Xem báo cáo theo ngày 44**](#_Toc136937735)

[**UC12: Xuất báo cáo theo ngày 46**](#_Toc136937736)

[**UC13: Xem báo cáo theo tháng 47**](#_Toc136937737)

[**UC14: Xuất báo cáo theo tháng 49**](#_Toc136937738)

[**UC15: Lấy lại mật khẩu 51**](#_Toc136937739)

[**UC16: Xem bảng chấm công 52**](#_Toc136937740)

[**UC17: Xuất bảng chấm công 54**](#_Toc136937741)

[**UC18: : Gửi báo cáo theo ngày qua email 55**](#_Toc136937742)

[**UC19: Gửi báo cáo theo tháng qua email 57**](#_Toc136937743)

[**UC20: Gửi bảng chấm công qua Email 59**](#_Toc136937744)

[**UC21: Thêm nhân viên đi muộn 61**](#_Toc136937745)

[**UC22: Sửa thông tin nhân viên đi muộn 63**](#_Toc136937746)

[**UC23: Xóa thông tin nhân viên đi muộn 64**](#_Toc136937747)

[**UC24: Tìm thông tin nhân viên đi muộn 65**](#_Toc136937748)

[**UC25: Gửi yêu cầu 67**](#_Toc136937749)

[**UC26: Xử lý yêu cầu 70**](#_Toc136937750)

[**CHƯƠNG 5: GIAO DIỆN NGƯỜI DÙNG 72**](#_Toc136937751)

[**5.1 Login 72**](#_Toc136937752)

[**5.2 Recovery 72**](#_Toc136937753)

[**5.3 Account manage 73**](#_Toc136937754)

[**5.4 Add account 73**](#_Toc136937755)

[**5.5 Edit account 74**](#_Toc136937756)

[**5.5 Request(Admin) 74**](#_Toc136937757)

[**5.6 Accountant 75**](#_Toc136937758)

[**5.7 Attendance 75**](#_Toc136937759)

[**5.8 Late manage 76**](#_Toc136937760)

[**5.9 Add late 76**](#_Toc136937761)

[**5.10 Edit late 77**](#_Toc136937762)

[**5.11 Report day 77**](#_Toc136937763)

[**5.12 Report month 78**](#_Toc136937764)

[**5.13 Request(Staff) 78**](#_Toc136937765)

[**5.14 Worker manage 79**](#_Toc136937766)

[**5.15 Add worker 79**](#_Toc136937767)

[**5.16 Edit worker 80**](#_Toc136937768)

[**CHƯƠNG 6: THỰC HIỆN BÀI TOÁN 81**](#_Toc136937769)

[**Màn hình Attendance: 81**](#_Toc136937770)

[**Màn hình Add worker: 83**](#_Toc136937771)

[**Màn hình Edit worker: 84**](#_Toc136937772)

[**CHƯƠNG 7: KIỂM THỬ 86**](#_Toc136937773)

[**7.1 Test case chấm công nhân viên 86**](#_Toc136937774)

[**7.2 Test case thêm tài khoản 87**](#_Toc136937775)

[**7.3 Test case sửa tài khoản 87**](#_Toc136937776)

[**7.4 Test case thêm nhân viên 88**](#_Toc136937777)

[**7.5 Test case sửa nhân viên 89**](#_Toc136937778)

[**7.6 Test case thêm thông tin nhân viên đi muộn 90**](#_Toc136937779)

[**7.5 Test case sửa thông tin nhân viên đi muộn 90**](#_Toc136937780)

[**7.5 Test case Report 91**](#_Toc136937781)

[**CHƯƠNG 8: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM** 92](#_Toc136937782)

[**8.1. Nội dung đã thực hiện** 92](#_Toc136937783)

[**8.2. Hướng phát triển** 92](#_Toc136937784)

[**TÀI LIỆU THAM KHẢO** 94](#_Toc136937785)

# **DANH MỤC BẢNG BIỂU**

[Bảng 1 Bảng Account 24](#_Toc136677282)

[Bảng 2 Bảng Faculty 24](#_Toc136677283)

[Bảng 3 Bảng WorkerList 24](#_Toc136677284)

[Bảng 4 Bảng Attendance 25](#_Toc136677285)

[Bảng 5 Bảng ListEmail 25](#_Toc136677286)

[Bảng 6 Bảng LateList 25](#_Toc136677287)

[Bảng 7 Bảng Request 26](#_Toc136677288)

[Bảng 8 Danh sách Usecase 28](#_Toc136677289)

[Bảng 9 UseCase đăng nhập 30](#_Toc136677290)

[Bảng 10 UseCase thêm tài khoản 31](#_Toc136677291)

[Bảng 11 UseCase cập nhật tài khoản 32](#_Toc136677292)

[Bảng 12 UseCase Xóa tài khoản 34](#_Toc136677293)

[Bảng 13 UseCase Tìm kiếm tài khoản 35](#_Toc136677294)

[Bảng 14 UseCase Thêm nhân viên 36](#_Toc136677295)

[Bảng 15 UseCase cập nhật thông tin nhân viên 38](#_Toc136677296)

[Bảng 16 UseCase xóa nhân viên 39](#_Toc136677297)

[Bảng 17 UseCase tìm kiếm nhân viên 40](#_Toc136677298)

[Bảng 18 UseCase chấm công nhân viên 42](#_Toc136677299)

[Bảng 19 UseCase xem báo cáo theo ngày 43](#_Toc136677300)

[Bảng 20 UseCase xuất báo cáo theo ngày 45](#_Toc136677301)

[Bảng 21 UseCase xem báo cáo theo tháng 47](#_Toc136677302)

[Bảng 22 UseCase xuất báo cáo theo tháng 48](#_Toc136677303)

[Bảng 23 UseCase lấy lại mật khẩu 50](#_Toc136677304)

[Bảng 24 UseCase xem bảng chấm công 51](#_Toc136677305)

[Bảng 25 UseCase xuất bảng chấm công 53](#_Toc136677306)

[Bảng 26 UseCase gửi báo cáo theo ngày qua email 55](#_Toc136677307)

[Bảng 27 Gửi báo cáo theo tháng qua email 57](#_Toc136677308)

[Bảng 28 UseCase gửi bảng chấm công qua email 59](#_Toc136677309)

[Bảng 29 UseCase thêm nhân viên đi muộn 61](#_Toc136677310)

[Bảng 30 UseCase sửa thông tin nhân viên đi muộn 62](#_Toc136677311)

[Bảng 31 UseCase xóa thông tin nhân viên đi muộn 64](#_Toc136677312)

[Bảng 32 UseCase tìm thông tin nhân viên đi muộn 65](#_Toc136677313)

[Bảng 33 UseCase gửi yêu cầu 66](#_Toc136677314)

[Bảng 34 UseCase xử lý yêu cầu 69](#_Toc136677315)

[Bảng 35 Tài liệu tham khảo 120](#_Toc136677316)

# **DANH MỤC BẢNG BIỂU**

[Hình 1Dò tìm khuôn mặt 15](#_Toc136677317)

[Hình 2 Đặc trưng Haar-Like 16](#_Toc136677318)

[Hình 3 Ví dụ 16](#_Toc136677319)

[Hình 4 Hình ảnh minh họa tích phân ảnh và công thức tính tích phân ảnh 17](#_Toc136677320)

[Hình 5 Tính nhanh tổng điểm ảnh trong vùng D 18](#_Toc136677321)

[Hình 6 Kỹ thuật Adaboost 19](#_Toc136677322)

[Hình 7 Bộ phận tầng Haar Cascade 20](#_Toc136677323)

[Hình 8 Dò tìm đối tượng với Haar-Like Features và Cascade of Boosted Classifiers 21](#_Toc136677324)

[Hình 9 Thư viện OpenCV 22](#_Toc136677325)

[Hình 10 Thư viện EmguCV 23](#_Toc136677326)

[Hình 11 Biểu đồ thực thể liên kết 25](#_Toc136677327)

[Hình 12 Sơ đồ UseCase 31](#_Toc136677328)

[Hình 13 Giao diện màn hình Login 73](#_Toc136677329)

[Hình 14 Giao diện màn hình Recovery 73](#_Toc136677330)

[Hình 15 Giao diện màn hình Account manage 74](#_Toc136677331)

[Hình 16 Giao diện màn hình Add account 74](#_Toc136677332)

[Hình 17 Giao diện màn hình Edit account 75](#_Toc136677333)

[Hình 18 Giao diện màn hình Request(Admin) 75](#_Toc136677334)

[Hình 19 Giao diện màn hình Accountant 76](#_Toc136677335)

[Hình 20 Giao diện màn hình Attendance 76](#_Toc136677336)

[Hình 21 Giao diện màn hình Late manage 77](#_Toc136677337)

[Hình 22 Giao diện màn hình Add late 77](#_Toc136677338)

[Hình 23 Giao diện màn hình Edit late 78](#_Toc136677339)

[Hình 24 Giao diện màn hình Report day 78](#_Toc136677340)

[Hình 25 Giao diện màn hình Report month 79](#_Toc136677341)

[Hình 26 Giao diện màn hình Request(Staff) 79](#_Toc136677342)

[Hình 27 Giao diện màn hình worker manage 80](#_Toc136677343)

[Hình 28 Giao diện màn hình Add worker 80](#_Toc136677344)

[Hình 29 Giao diện màn hình edit worker 81](#_Toc136677345)

# 

# **LỜI NÓI ĐẦU**

Với xu hướng công nghệ ngày càng phát triển, hệ thống tự động càng được ưu chuộng vì sự tiện lợi của nó. Thay vì sử dụng nhân lực để thực hiện các công việc đơn giản thì nay chúng ta đã có thể sử dụng những phần mềm, công cụ hỗ trợ để xử lý những công việc này nhằm tăng năng xuất và tránh lãng phí nhân sự, tiền bạc. Nắm bắt được xu thế, với sự hướng dẫn tận tình của thầy Ths.Nguyễn Thái Cường, nhóm 2 chúng em quyết định nghiên cứu và phát triển phần mềm chấm công tự động bằng nhận diện khuôn mặt.

Ở bài tập lớn lần này, chúng em nghiên cứu và áp dụng EmguCV (một thư viện hỗ trợ thị giác máy tính) để có thể nhận dạng khuôn mặt của các nhân viên trong phòng ban với ngôn ngữ lập trình C# trên nền tảng WPF và công cụ Visual Studio 2022.

Nhóm đã cố gắng hoàn thành bài tập lớn, mong được sự giúp đỡ của thầy.

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

* 1. **Đặt vấn đề**

Trong thời kỳ phát triển mạnh mẽ của công nghệ thông tin, con người đang dần áp dụng các ứng dụng khoa học máy tính để phục vụ các công việc hằng ngày. Trong đó, những phần mềm bảo mật an ninh được thiết kế dựa trên hệ thống thị giác máy tính để thực hiện việc phát hiện, nhận dạng và sàng lọc xuất hiện ngày càng nhiều với những đối tượng ngày càng phong phú, đa dạng. Hiện nay, các ứng dụng nhận dạng khuôn mặt được xây dựng và phát triển trên rất nhiều thiết bị cũng như hệ thống nhằm giúp con người thuận tiện hơn trong việc quản lý, bảo mật, giám sát hoặc tìm kiếm. Các ứng dụng nhận diện khuôn mặt, nụ cười trong máy ảnh hoặc smartphone dần trở nên quen thuộc với mọi người.

Nhận thấy sự cần thiết trong việc nghiên cứu công nghệ nhận diện giúp con người có thể thuận tiện hơn trong nhiều công việc trong cuộc sống,kèm theo đó là tiền đề để nâng cao và phát triển mạnh về hệ thống thị giác máy tính, các ứng dụng về nhận dạng hình ảnh khuôn mặt nhằm mục đích tăng cáo tính bảo mật của hệ thống.

Đề tài với tên gọi “ nghiên cứu về thư viện OpenCV – nghiên cứu về hệ thống tự động chấm công bằng cách nhận dạng khuôn mặt” được thực hiện với mong muốn khai thác những khía cạnh về nhận dạng tự động và tự động vận hành nhằm giảm gánh nặng cho con người cũng như tiết kiệm thời gian. Có thể nói đây là phương hướng nghiên cứu có thể mang lại nhiều thuận lợi cũng nhợi ích cho xã hội.

* 1. **Một số nghiên cứu trước đây**
* Có rất nhiều nghiên cứu sử dụng các phương pháp khác nhau để giải quyết bài toán nhận dạng khuôn mặt, hai cách phổ biến nhất là thường được sử dụng là haarCascade và nhận dạng dựa trên xét tổng thể của khuôn mặt (Appearance based face recognition). Nhìn chung, các phương pháp đều có ưu nhược điểm nhằm giải quyết bài toán nhận dạng mặt người
  + Những năm 1960, Woody Bledsoe, Helen Chan Wolf, và Charles Bisson đã cho ra đời hệ thống nhận dạng khuôn mặt bán tự động đầu tiên. Hệ thống yêu cầu người quản trị xác định vị trí các đặc trưng như: mắt, tai, mũi và miệng trên các tấm ảnh. Sau đó, hệ thống sẽ tính khoảng cách và tỷ lệ đến một điểm tham chiếu chung. Cuối cùng là so sánh với kho dữ liệu lưu trữ.
  + Năm 1970, Goldstein, Harmon và Lesk sử dụng 21 dấu hiệu chủ quan như: màu tóc và độ dày môi để nhận dạng tự động. Giống như ở giải pháp trước, các phép đo và các vị trí được tính bằng tay nên đòi hỏi nhiều thời gian.
  + Năm 1988, Kirby và Sirovich áp dụng phương pháp phân tích thành phần chính (PCA), phương pháp này sử dụng chuẩn đại số tuyến tính cho vấn đề nhận dạng Nguyễn Hoàng Phúc – 1111540 Nghiên cứu thư viện OpenCv - Ứng dụng nhận dạng khuôn mặt người Trang 2 khuôn mặt. Đây được coi là cột mốc quan trọng vì nó cho thấy ít hơn một trăm giá trị cần thiết để nhận dạng một khuôn mặt được chuẩn hoá.
  + Năm 1991, Turk và Pentland sử dụng thuật toán eigenfaces để nhận dạng khuôn mặt tự động trên thời gian thực. Dù phương pháp này một phần nào bi hạn chế bởi yếu tố môi trường nhưng nó cũng góp phần đáng kể cho sự phát triển của công nghệ nhận dạng khuôn mặt.
  + Năm 2001, Paul Viola và Michael J. Jones đã cho ra đời đặc trưng Haar-Like, đó là những đặc trưng ảnh số cũng với phương pháp tích phân ảnh (integral images) để nâng cao khả năng rút trích đặc trưng trên thời gian thực.
  + Tháng 7-2003, Trần Phước Long và Nguyễn Văn Lượng dùng mạng Neural để dò tìm khuôn mặt trên ảnh, kết hợp phương pháp phân tích thành phần chính và biến đổi Cosine rời rạc để rút ra Vector đặc trưng làm đầu vào cho hai bộ nhận dạng SVM và HMM. Hệ thống tiếp cận từ các phương pháp học mạnh, nhưng nhiều trường hợp tối ưu cần thử nghiệm nhiều lần với nhiều bộ tham số khác nhau.
  + Năm 2005, Trần Lê Hồng Dũ sử dụng các đặc trưng lòi lõm để phát hiện khuôn mặt trên ảnh nhưng còn hạn chế về độ sáng, quan hệ hình học và các đặc trưng cùng mức.
  + Năm 2007, Lê Hồng Chuyên đã sử dụng mạng Neural để phát hiện khuôn mặt trên ảnh. Hệ thống đã nhận dạng được những khuôn mặt ở các tư thế: thẳng đứng, không thẳng đứng hoặc bị che mất một phần. Nhưng kết quả dò tìm không cao trong môi trường ảnh có nhiều người.
  + Năm 2013, Mã Trường Thành thực hiện đề tài “Điều khiển Robot Pioneer P3- DX bám sát đối tượng” đã sử dụng đặc trưng Haar-Like, bộ phân tầng Cascades of Boosted Classifiers và thuật toán Adaboost để nhận dạng đối tượng.
  + Năm 2013, Châu Ngân Khánh đã sử dụng đặc trưng Haar-Like, bộ phân tầng Cascades of Boosted Classifiers và thuật toán Adaboost để nhận dạng đối tượng bằng cách so khớp SIFT. Nhưng cơ sở dữ liệu không bao quát được hết sự thay đổi của khuôn mặt con người trên thời gian thực…

## **1.3 Phạm vi đề tài.**

* Đề tài tập trung nghiên cứu gói thư viện mã nguồn mở OpenCV để giải quyết bài toán quan sát, phát hiện và nhận dạng đối tượng theo thời gian thực bằng đặc trưng HaarLike Features – Cascades of Boosted Classifiers.
* Triển khai nghiên cứu bằng việc thu thập ảnh, huấn luyện và giải quyết bài toán thực tế về nhận dạng khuôn mặt người bằng cách lưu các hoạt động của đối tượng dưới dạng ảnh và video thông qua Webcam.

## **1.4 Phương pháp nghiên cứu**

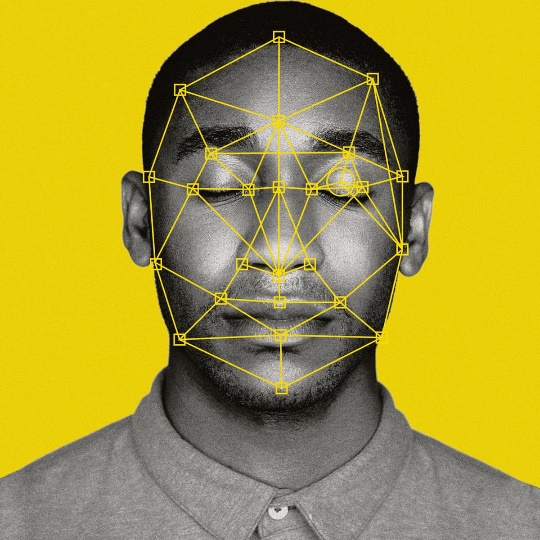
* + Nghiên cứu gói thư viện OpenCv – cụ thể là gói thư viện Emgu với ngôn ngữ được sử dụng là C# (C Sharp).
  + Sử dụng các hàm trong thư viện OpenCV bằng việc kết hợp đặc trưng HaarLike với thuật toán Adaboost và mô hình Cascades of Boosted Classifiers để thực hiện dò tìm đối tượng trên mặt phẳng ảnh theo thời gian thực để triển khai giải quyết bài toán đặt ra.

## **1.5 Bố Cục và nội dung**

* Bài báo cáo nghiên cứu về việc nhận dạng khuôn mặt người với OpenCV với những nội dung sau:
  + **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**: Giới thiệu tổng quan về đề tài, xác định vấn đề, bối cảnh và nội dung bài toán cần giải quyết. Trình bày về lịch sử giải quyết vấn đề, các nghiên cứu và thành quả của những người đi trước trong cùng lĩnh vực, phạm vi và phương pháp nghiên cứu đề tài.
  + **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT:** Giới thiệu về bài toán dò tìm khuôn mặt, đặc trưng Haar-Like Features, thuật toán AdaBoost và bộ phân tầng Cascade of Boosted Classifier. Tổng quan về gói thư viện OpenCV để nghiên cứu về nhận dạng mặt người, thư viện Emgu nền tảng của OpenCV trên nền ngôn ngữ C# và hệ thống gửi tin nhắn GSMComm.
  + **CHƯƠNG 3,4,5: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU:** Thể hiện cách giải quyết bài toán thông qua hệ thống thực nghiệm. Xây dựng một bộ huấn luyện nhằm nhận dạng khuôn mặt người trên mặt phẳng ảnh. Nêu lên các thiết kế về chương trình và trình bày về thực nghiệm chương trình kèm theo hướng dẫn

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1 Giới thiệu về bài toán dò tìm khuôn mặt**



Hình Dò tìm khuôn mặt

Bài toán dò tìm khuôn mặt (face detection) là một trong những bài toán quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Nó đặt ra mục tiêu nhận diện và phát hiện khuôn mặt trong một hình ảnh hoặc một đoạn video.

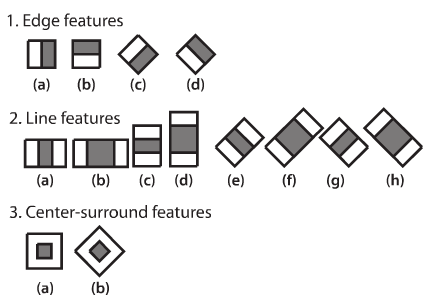
Bài toán này thường được giải quyết thông qua việc sử dụng các thuật toán học máy và deep learning để học và tìm kiếm các đặc trưng của khuôn mặt, ví dụ như hình dạng, màu sắc, kích thước và đặc điểm hình học khác.

Các ứng dụng của bài toán dò tìm khuôn mặt rất đa dạng, từ nhận dạng khuôn mặt để mở khóa điện thoại, nhận diện khuôn mặt để đăng nhập vào các ứng dụng, đến giám sát an ninh và phát hiện tội phạm. Tuy nhiên, bài toán này còn đối mặt với nhiều thách thức như sự biến đổi về ánh sáng, góc nhìn và vật liệu, cũng như sự đa dạng của các khuôn mặt trong thực tế.

**2.2 Đặc trưng Haar-Like Features**

Đặc trưng Haar-Like Features là một trong những phương pháp được sử dụng trong bài toán dò tìm khuôn mặt. Phương pháp này được đề xuất bởi Viola và Jones vào năm 2001 và đã trở thành một trong những phương pháp phổ biến trong lĩnh vực nhận dạng khuôn mặt.

Trong bài báo của tác giả *Paul Viola và Michael Jones* đã trình bày một phương pháp mới và nhanh hơn để xử lý hình ảnh và phát hiện khuôn mặt bằng cách sử dụng các đặc điểm hình chữ nhật như hình dưới đây. Các **đặc điểm hình chữ nhật** tương tự như nhân được sử dụng để phát hiện các đặc điểm khác nhau của khuôn mặt như mắt và các nốt như trong hình minh họa.



Hình Đặc trưng Haar-Like

Các tính năng hình chữ nhật được chạy lần lượt trên hình ảnh và tổng số pixel nằm trong phần màu trắng được trừ cho tổng số pixel nằm trong phần màu đen.

Trong đó:

Edge features: Đặc trưng cạnh

Line features: Đặc trưng đường

Center-suround features: Đặc trưng trung tâm

Trong hình minh họa bên dưới, đặc điểm hình chữ nhật đầu tiên đang tính toán cho sự khác biệt về cường độ giữa vùng mắt và vùng má trên khuôn mặt. Và đặc điểm hình chữ nhật thứ hai là đo sự chênh lệchvề cường độ giữa hai vùng mắt và sống mũi. Bộ lọc Haar chỉ có thể nhìn cụ thể vào một vùng trong cửa sổ để nhận diện.



Hình Ví dụ

* 1. **Trích đặc trưng Haar-Like từ ảnh**

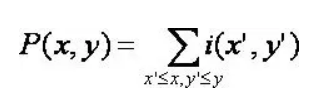
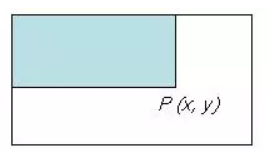
Đặc trưng Haar-like sẽ diễn đạt được tri thức về các đối tượng trong ảnh (bởi vì nó biểu diễn mối liên hệ giữa các bộ phận của đối tượng), điều mà bản thân từng điểm ảnh không diễn đạt được – dựa trên việc co giãn của khung cửa sổ tìm kiếm. Để tính giá trị các đặc trưng Haar-Like, ta tính sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng như trong công thức:

****

Vì vậy chúng ta sẽ thấy rằng, với tổng các pixel trên ảnh *(pixel vùng trắng và pixel vùng đen)* sẽ cho ta các giá trị của đặc trưng Haar – Like. Nhưng để tính toán những giá trị của các đặc trưng Haar-like cho tất cả các vị trí trên ảnh đòi hỏi chi phí tính toán khá lớn, dẫn đến không thể đáp ứng được cho các ứng dụng đòi hỏi tính thời gian thực*(run-time)* trực tiếp từ Camera *(hay Webcam trên Laptop)*. Do đó Viola và Jones đề xuất một khái niệm gọi là ***“tich phân ảnh”*** *(Integral Image)* để tính toán nhanh cho các đặc trưng cơ bản của Haar-like.

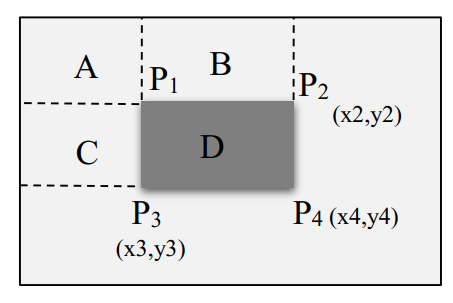
Tích phân ảnh được định nghĩa là 1 mảng 2 chiều có kích thước bằng kích thước ảnh cần tìm.

Mỗi phần tử trong mảng có giá trị bằng tổng các phần từ ở phái trên và bên trái của nó. Bắt đầu từ vị trí trên bên trái và kết thúc tại vị trí dưới bên phải của ảnh, tính lần lượt từ trái qua phải và từ trên xuống dưới.



Hình Hình ảnh minh họa tích phân ảnh và công thức tính tích phân ảnh

Sau khi tính tích phân ảnh, ta tính tổng điểm ảnh theo các sau:



Hình Tính nhanh tổng điểm ảnh trong vùng D

Tính điểm D:

Tổng = A + B + C + D

D =Tổng – (A+B) – (A+C) + A

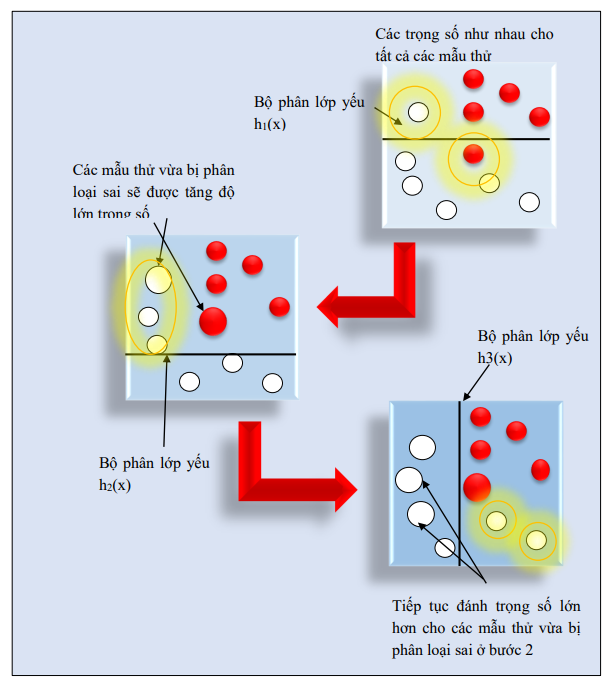
Với A + B + C + D chính là giá trị tại điểm P4 trên đảo hàm ảnh , tương tự như vậy A+B là giá trị tại điểm P2, A+C là giá trị tại điểm P3, và A là giá trị tại điểm P1.  
 Vậy ta có thể viết lại biểu thức tính D ở trên như sau:

D= P4-P2-P3+P1

D= (x4,y4)-(x2,y2)-(x3,y3)+(x1,y1)

## **2.4 Thuật toán AdaBoost**

Thuật toán AdaBoost (Adaptive Boosting) là một trong những thuật toán quan trọng và phổ biến nhất được sử dụng trong bài toán dò tìm khuôn mặt. AdaBoost kết hợp nhiều bộ phân loại yếu thành một bộ phân loại mạnh. Mỗi bộ phân loại yếu được huấn luyện để phân loại một tập con của dữ liệu và nhận được một trọng số ứng với độ chính xác của nó. AdaBoost sẽ tăng trọng số của các dữ liệu bị phân loại sai và giảm trọng số của các dữ liệu được phân loại đúng. Sau đó, các bộ phân loại yếu sẽ được sử dụng để tạo ra một bộ phân loại mạnh bằng cách kết hợp dự đoán của chúng theo một quy tắc đa số.



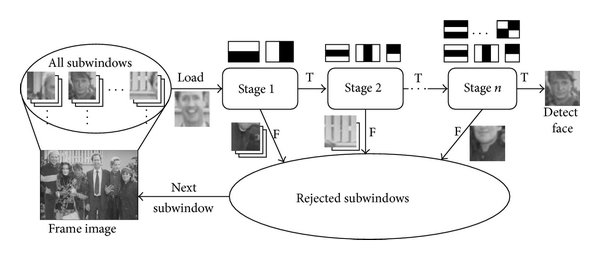
Hình Kỹ thuật Adaboost

## **2.5 Bộ phân tầng Cascade of Boosted Classifier**

Bộ phân tầng Cascade of Boosted Classifier là một phương pháp dò tìm khuôn mặt cải tiến dựa trên AdaBoost. Nó bao gồm nhiều giai đoạn, mỗi giai đoạn sử dụng một bộ phân loại được huấn luyện để loại bỏ các vùng không phải là khuôn mặt. Ở mỗi giai đoạn, hình ảnh được chia thành nhiều vùng nhỏ hơn, các vùng này được xem xét bởi các bộ phân loại với độ chính xác khác nhau. Các vùng bị loại bỏ không phải là khuôn mặt sẽ không được xem xét trong các giai đoạn tiếp theo. Các khuôn mặt được phát hiện trong các giai đoạn sau sẽ được xác định chính xác hơn vì chỉ cần xem xét các vùng nhỏ hơn và đặt ưu tiên cho các vùng có khả năng cao hơn chứ không cần phân tích toàn bộ hình ảnh.

Bộ phân tầng Cascade of Boosted Classifier có nhiều ưu điểm, bao gồm tăng tốc độ dò tìm, cải thiện độ chính xác và giảm độ phức tạp tính toán. Tuy nhiên, cần phải cân nhắc kỹ lưỡng trong việc lựa chọn các tham số và bộ phân loại để đạt được hiệu quả cao nhất.

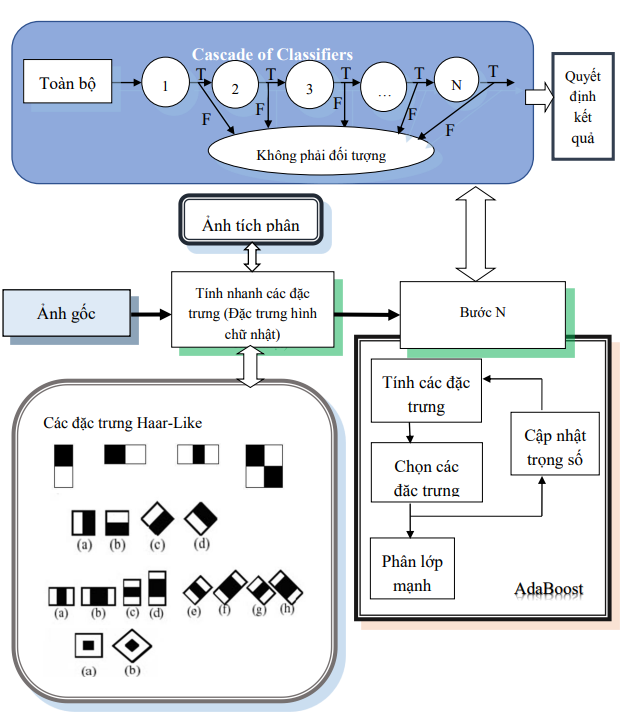
Để được nhận dạng là khuôn mặt, các cửa sổ ảnh nhỏ sẽ phải đi qua hết các stage của bộ phân loại. Các bộ lọc ở mỗi stage đằng sau sẽ được huấn luyện bằng các mẫu không có đối tượng mà các stage trước nó nhận định sai, do đó sự kết hợp các lớp lọc này sẽ giúp hệ thống giảm thiếu khả năng nhận định sai.



Hình Bộ phận tầng Haar Cascade

* **Bước 1:** Hình ảnh (đã được gửi đến bộ phân loại) được **chia thành các phần nhỏ**(hoặc các cửa sổ con như trong hình minh họa).
* **Bước 2:** Chúng tôi đặt N bộ dò theo cách xếp tầng trong đó mỗi bộ phát hiện sự kết hợp của các loại **đặc trưng khác nhau** từ các hình ảnh (ví dụ: đường thẳng, cạnh, hình tròn, hình vuông) được truyền qua. Giả sử khi việc trích xuất đối tượng địa lý được thực hiện, mỗi phần phụ được gán một **giá trị tin cậy**.
* **Bước 3:** Hình ảnh (hoặc hình ảnh phụ) có độ tin cậy cao nhất được phát hiện dưới dạng khuôn mặt và được gửi đến **bộ tích lũy** trong khi phần còn lại bị từ chối. Do đó, Cascade tìm nạp khung hình / hình ảnh tiếp theo nếu còn lại và bắt đầu lại quá trình.

## **2.6 Định vị đối tượng**

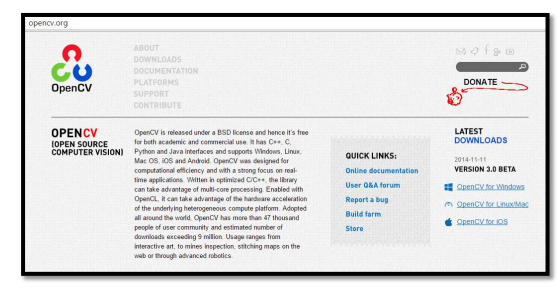


Hình Dò tìm đối tượng với Haar-Like Features và Cascade of Boosted Classifiers

* Image Kernels là 1 ma trận nhỏ dùng để sử dụng các hiệu ứng chỉnh sửa. Trong mỗi ô của ma trận sẽ chứa giá trị tờ 0 đến 255 để biểu thị giá trị sang tối của từng pixel ảnh trong ảnh chính. Từ đó, người dùng sẽ áp dụng các công thức tính toán để làm mờ, làm rõ nét bằng các phép nhân ma trận. Hơn thế nữa, chúng ta có thể dùng phép tích phân ảnh để sử dụng trong việc nhận dạng khuôn mặt.
* Trình tự hoạt động: Sau khi quét ảnh từ camera, hệ thống sẽ chuyển hình ảnh quét được thành ảnh đen trắng. Sau đó chuyển hình ảnh đen trắng đã có thành dạng Image Kernels. Sau đó hệ thống trích các đặc trưng Haar-Like từ ảnh thông qua tích phân ảnh và sử dụng các hình chữ nhật đặc trưng để phát hiện gương mặt trong khung hình. Sau đó được đưa qua bộ điều chỉnh AdaBoost để loại bỏ bớt các đặc trưng không cần thiết. Các đặc trưng còn lại sẽ được đưa qua bộ phân lớp để quyết định xem có phải mặt người hay không.

## **2.7 Tổng quan về thư viện OpenCV để nghiên cứu về nhận dạng mặt người**

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở cung cấp nhiều công cụ và thuật toán để xử lý ảnh và thị giác máy tính. OpenCV được viết bằng C++, nhưng cũng có thể sử dụng các ngôn ngữ khác như Python và Java thông qua các giao diện lập trình ứng dụng (API).



Hình Thư viện OpenCV

OpenCV cung cấp nhiều tính năng để hỗ trợ nhận dạng mặt người, bao gồm:

1. Xử lý ảnh: OpenCV cung cấp các tính năng để xử lý ảnh như đọc, ghi, chuyển đổi, cắt, phóng to, thu nhỏ ảnh, làm mịn ảnh, tăng độ tương phản và cân bằng sáng tối.
2. Dò tìm khuôn mặt: OpenCV cung cấp các thuật toán để dò tìm khuôn mặt như Haar Cascades, LBP (Local Binary Pattern), và HOG (Histogram of Oriented Gradients).
3. Nhận dạng khuôn mặt: OpenCV cung cấp các thuật toán để nhận dạng khuôn mặt như Eigenfaces, Fisherfaces, và Local Binary Patterns Histograms (LBPH).
4. Ghi chú và vẽ hình trên ảnh: OpenCV cung cấp các công cụ để ghi chú và vẽ hình trên ảnh để đánh dấu khuôn mặt đã được tìm thấy hoặc hiển thị kết quả của thuật toán.

OpenCV cũng có thể được sử dụng để kết hợp nhiều thuật toán nhận dạng khuôn mặt và tạo ra một hệ thống hoàn chỉnh để nhận dạng khuôn mặt trong thời gian thực.

Việc sử dụng OpenCV để nghiên cứu về nhận dạng khuôn mặt là rất phổ biến vì thư viện này cung cấp nhiều tính năng và thuật toán cho bài toán này và có thể được sử dụng trên nhiều nền tảng khác nhau như Windows, Linux, MacOS, Android và iOS.

**2.8 Thư viện EmguCV**

Emgu CV là gói thư viện được viết hoàn toàn bằng ngôn ngữ C#. Nó bao gồm các chức năng của OpenCV, cho phép gọi các hàm của OpenCV từ Visual Studio bằng các ngôn ngữ .Net như: C#, VB, VC++, v.v... Vì Visual Studio/.Net không thể gọi trực tiếp các hàm của OpenCV nên Emgu là giải pháp cho vấn đề này. Emgu có thể chạy trên Windows, Linux, iOS, Mac OS X, Android và Windows Phone.



Hình Thư viện EmguCV

**CHƯƠNG 3: YÊU CẦU CHỨC NĂNG VÀ MÔ HÌNH HÓA USECASE**

## **3.1 Yêu cầu chức năng**

**-Quản trị viên:**

**+** Đăng nhập

+ Thêm tài khoản

+ Sửa tài khoản

+ Xóa tài khoản

+ Lấy lại mật khẩu

+ Xử lý yêu cầu

**-Quản lý:**

+ Đăng nhập

+ Lấy lại mật khẩu

+ Quản lý nhân viên (thêm, sửa, xóa, tìm kiếm)

+ Xem báo cáo theo ngày và theo tháng

+ Gửi báo cáo qua email

+ Xuất báo cáo

+ Chấm công

+ Xem bảng lương

+ Gửi bảng lương qua email

+ Xuất bảng lương

+ Gửi yêu cầu

+ Quản lý nhân viên đi làm muộn (thêm, sửa, xóa, tìm kiếm)

## **3.2 Mô hình hóa CSDL**

### 3.2.1 Các yêu cầu về dữ liệu

Để xây dựng 1 hệ thống chấm công tự động cần có các yêu cầu sau:

- Hệ thống lưu trữ thông tin các tài khoản admin và nhân viên bao gồm: tên đăng nhập, mật khẩu, email, ảnh đại diện, chức vụ. phòng ban.

- Danh sách nhân viên: Tên nhân viên, id, ngày sinh,lương cơ bản của 1 ca làm việc, ảnh hổ sơ.

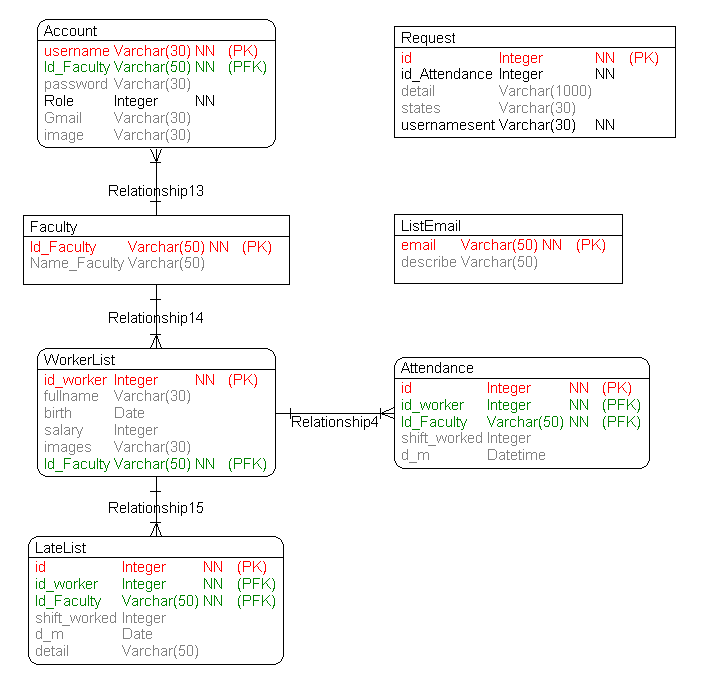
- Danh sách phòng ban: id, tên phòng ban.

-Danh sách email: tài khoản email, mô tả tài khoản.

-Danh sách điểm danh:id, id công nhân, id phòng ban, ca, ngày tháng.

-Danh sách đi muộn : id, id công nhân , id phòng ban , ca , ngày , lý do đi muộn

### 3.2.2 Biểu đồ thực thể liên kết

****

Hình Biểu đồ thực thể liên kết

### 3.2.3 Mô tả chi tiết các bảng CSDL

-Bảng Account

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Username | Varchar(30) | Tên đăng nhập |
| Password | Varchar(30) | Mật khẩu |
| Id\_faculty | Varchar(50) | Id của phòng ban |
| Role | Int | Chức vụ |
| Gmail | Varchar(30) | Gmail cá nhân |
| Image | Varchar(30) | Ảnh đại diện |

Bảng Bảng Account

-Bảng Faculty

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Id\_Faculty | Varchar(50) | Id của phòng ban |
| Name\_Falculty | Varchar(50) | Tên phòng ban |

Bảng Bảng Faculty

-Bảng WorkerList

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Int | Id nhân viên |
| Fullname | Varchar(30) | Tên nhân viên |
| Birth | Date | Ngày sinh |
| Salary | Int | Lương cơ bản của 1ca làm việc |
| Image | Varchar(30) | Ảnh hồ sơ |
| Id-Faculty | Varchar(50) | Id phòng ban |

Bảng Bảng WorkerList

-Bảng Attendance

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Int | Id điểm danh |
| Id\_worker | Int | Id nhân viên |
| Id\_Faculty | Varchar(50) | Id phòng ban |
| Shift\_worked | Int | Ca |
| D\_m | DateTime | Ngày, giờ điểm danh |

Bảng Bảng Attendance

-Bảng ListEmail

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Email | Varchar(50) | Email |
| Describe | Varchar(50) | Mô tả của email |

Bảng Bảng ListEmail

-Bảng LateList

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Int | Id điểm danh |
| Id\_worker | Int | Id nhân viên |
| Id\_Faculty | Varchar(50) | Id phòng ban |
| Shift\_worked | Int | Ca |
| D\_m | Date | Ngày đi muộn |
| Detail | Varchar(50) | Lý do đi muộn |

Bảng Bảng LateList

-Bảng Request

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| Id | Int | Id yêu cầu |
| Id\_Attendance | Int | Id điểm danh |
| Detail | Varchar(1000) | Chi tiết yêu cầu |
| States | Varchar(30) | Trạng thái yêu cầu |
| Usernamesent | Varchar(30) | Người gửi yêu cầu |

Bảng Bảng Request

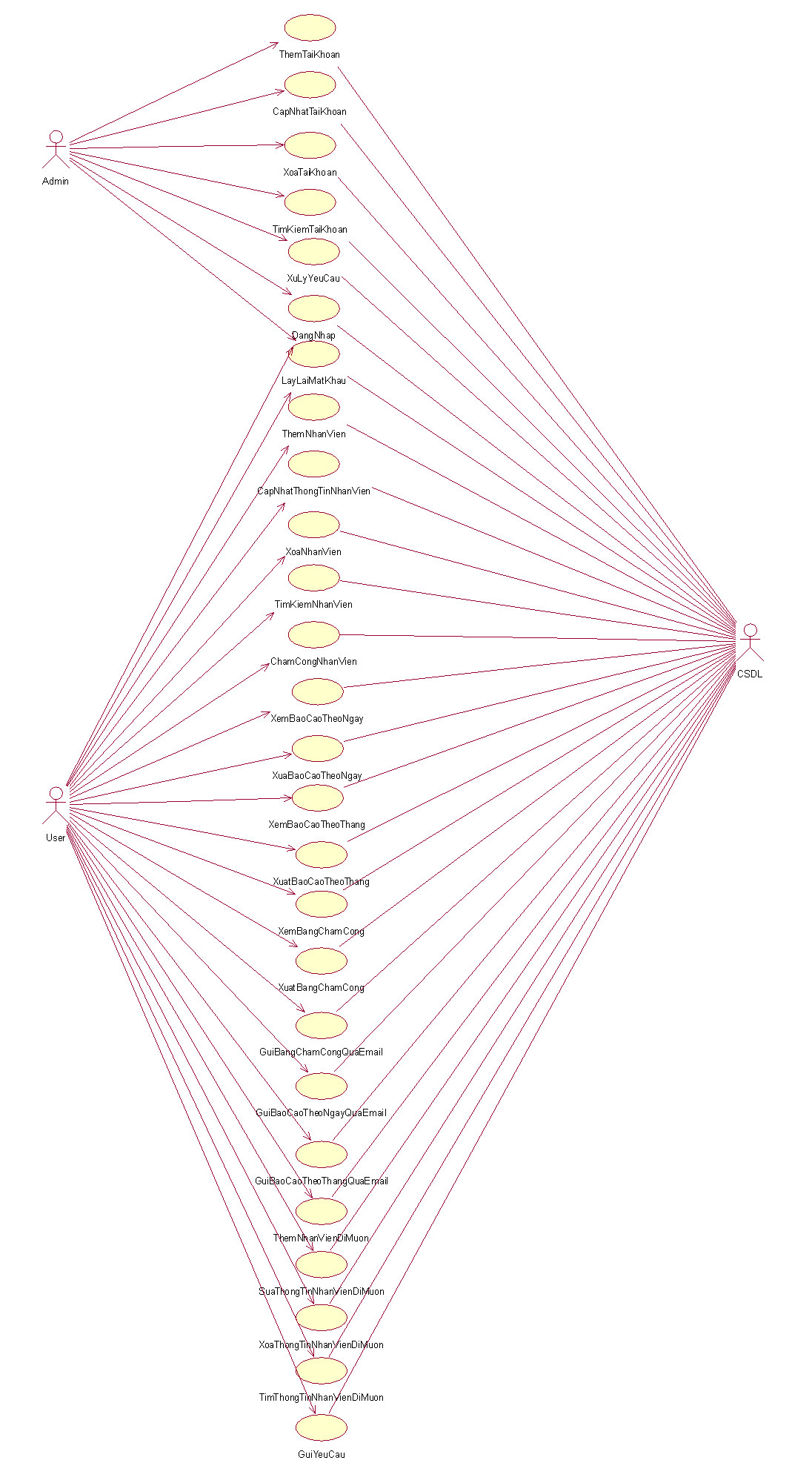
**CHƯƠNG 4: MÔ TẢ USECASE**

## **4.1 Danh sách Usecase**

|  |  |  |
| --- | --- | --- |
| **Tên UC** | **Mã UC** | **Chức năng** |
| Đăng nhập | UC01 | Đăng nhập vào hệ thống |
| Thêm tài khoản | UC02 | Thêm tài khoản |
| Cập nhật tài khoản | UC03 | Sửa thông tin tài khoản |
| Xóa tài khoản | UC04 | Xóa tài khoản |
| Tìm tài khoản | UC05 | Tìm kiếm thông tin tài khoản |
| Thêm nhân viên | UC06 | Thêm thông tin nhân viên |
| Cập nhật nhân viên | UC07 | Sửa thông tin nhân viên |
| Xóa nhân viên | UC08 | Xóa thông tin nhân viên |
| Tìm kiếm nhân viên | UC09 | Tìm kiếm thông tin nhân viên |
| Chấm công | UC10 | Chấm công nhân viên |
| Xem báo cáo theo ngày | UC11 | Xem danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt theo ngày |
| Xuất báo cáo theo ngày | UC12 | Xuất danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt theo ngày |
| Xem báo cáo theo tháng | UC13 | Xem danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt theo tháng |
| Xuất báo cáo theo tháng | UC14 | Xuất file excel danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt theo tháng |
| Lấy lại mật khẩu | UC15 | Lấy lại mật khẩu |
| Xem bảng lương | UC16 | Xem bảng lương |
| Xuất bảng chấm công | UC17 | Xuất file excel chứa bảng lương |
| Gửi báo cáo theo ngày | UC18 | Gửi file danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt theo ngày |
| Gửi báo cáo theo tháng | UC19 | Gửi file danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt theo tháng |
| Gửi bảng lương | UC20 | Gửi file bảng lương qua email |
| Thêm nhân viên đi muộn | UC21 | Thêm thông tin nhân viên đi làm muộn |
| Sửa thông tin nhân viên đi muộn | UC22 | Sửa thông tin nhân viên đi làm muộn |
| Xóa thông tin nhân viên đi muộn | UC23 | Xóa thông tin nhân viên đi làm muộn |
| Tìm thông tin nhân viên đi muộn | UC24 | Tìm kiếm thông tin nhân viên đi muộn |
| Gửi yêu cầu | UC25 | Gửi yêu cầu đến admin để xóa điểm danh |
| Xử lý yêu cầu | UC26 | Xử lý các yêu cầu được gửi đến |

Bảng Danh sách Usecase

## **4.2 Sơ đồ UC**

****

Hình Sơ đồ UseCase

## **4.3 Mô tả chi tiết UC**

### UC01: Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Login | **Code** | UC01 |
| **Description** | Cho phép actor đăng nhập vào hệ thống | | |
| **Actor** | Admin/Staff | **Trigger** | Actor bấm nút login |
| **Pre-condition** | Điền đủ vào ô tài khoản và mật khẩu | | |
| **Post condition** | Chuyển tới trang menu với role tương ứng | | |

Bảng UseCase đăng nhập

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor nhập tên đăng nhập/mật khẩu và click vào button Login trên trang đăng nhập ([SC01](#_SC01:_My_Profile)). |  |  |
|  |  | 2 | Kiểm tra tên đăng nhập/mật khẩu là chính xác sau đó chuyển tới trang default role tương ứng ([SC03](#_SC03:_My_Group), [SC02](#_SC02:_List_users)...) |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

|  |  |
| --- | --- |
| MS01 | “Tên Đăng Nhập và/hoặc Mật Khẩu của bạn không chính xác. Vui lòng kiểm tra và thử lại.”  Message thông báo khi actor nhập sai tên đăng nhập/mật khẩu |

### UC02: Thêm tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Thêm tài khoản | **Code** | UC02 |
| **Description** | Cho phép actor thêm tài khoản staff/admin trong công ty | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Add]** trên màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Thêm thành công tài khoản | | |

Bảng UseCase thêm tài khoản

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Thêm tài khoản thành công** | | | |
| 1 | Actor click nút **[Add]** trên màn hình Account Manage |  |  |
|  |  | 2 | Hệ thống Load trang Thêm mới thông tin tài khoản |
| 3 | Nhập vào các thông tin được yêu cầu. |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Lưu thông tin người dùng mới vào bảng Account trên CSDL, thông báo thành công và chuyển sang trang Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message: N/A**

### UC03: Cập nhật tài khoản

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Cập nhật thông tin tài khoản | **Code** | UC03 |
| **Description** | Cho phép actor cập nhật thông tin tài khoản trong hệ thống. | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Edit]** ở màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Cập nhật tài khoản thành công | | |

Bảng UseCase cập nhật tài khoản

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Cập nhật thông tin tài khoản thành công** | | | |
| 1 | Actor chọn 1 tài khoản rồi click button **[Edit]** ở màn hình Quản lý tài khoản |  |  |
|  |  | 2 | Hệ thống lấy thông tin của tài khoản cần sửa trong bảng Account trên CSDL rồi hiển thị lên trang Thông tin tài khoản |
| 3 | Cập nhật những thông tin cần thiết rồi ấn button **[Save]** |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Hệ thống update thông tin tài khoản trong bảng WorkerList trên CSDL. Hệ thống thông báo thành công và chuyển về màn hình Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC04: Xóa tài khoản

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xóa tài khoản | **Code** | UC04 |
| **Description** | Cho phép actor xóa tài khoản trong công ty | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Del]** ở màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xóa tài khoản thành công | | |

Bảng UseCase Xóa tài khoản

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xóa tài khoản thành công** | | | |
| 1 | Actor chọn 1 tài khoản rồi click button **[Del]** ở màn hình Quản lý tài khoản |  |  |
|  |  | 2 | Hệ thống hiển thị thông báo “Are you sure ?” |
| 3 | Actor click **[Yes]** |  |  |
|  |  | 4 | Hệ thống xóa thông tin tài khoản đó ra khỏi bảng Account và những bảng liên quan trên CSDL. Thông báo thành công và chuyển sang trang Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

|  |  |
| --- | --- |
| MS02 | “Xác nhận xóa  Message thông báo khi actor nhập xóa |

### UC05: Tìm kiếm tài khoản

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Tìm kiếm tài khoản | **Code** | UC05 |
| **Description** | Cho phép actor tìm kiếm tài khoản bằng tên đăng nhập hoặc email | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Search]** bên cạnh thanh tìm kiếmtrên màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Tìm kiếm tài khoản thành công | | |

Bảng UseCase Tìm kiếm tài khoản

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Tìm kiếm tài khoản thành công** | | | |
| 1 | Actor nhập tên đăng nhập hoặc email của tài khoản cần tìm vào thanh tìm kiếm rồi click button **[Search]** trênmàn hình Quản lý tài khoản |  |  |
|  |  | 2 | Hệ thống tìm tài khoản có tên đăng nhập/ email trùng với tên đăng nhập/ email actor nhập trong bảng Account trên CSDL rồi hiển thị lên list tài khoản trên màn hình Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC06: Thêm nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Quản lý nhân viên | **Code** | UC06 |
| **Description** | Cho phép actor quản lý nhân viên trong công ty | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Add]** trên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Thêm thành công nhân viên | | |

Bảng UseCase Thêm nhân viên

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Thêm nhân viên thành công** | | | |
| 1 | Actor chọn nút **[Add]** màn hình Worker Manage |  |  |
|  |  | 2 | Hệ thống Load trang Thêm mới thông tin người dùng |
| 3 | Nhập vào các thông tin được yêu cầu. |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Lưu thông tin người dùng mới vào bảng WorkerList trên CSDL, thông báo thành công và chuyển sang trang Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC07: Cập nhật thông tin nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Cập nhật thông tin nhân viên | **Code** | UC07 |
| **Description** | Cho phép actor cập nhật thông nhân viên trong phòng ban. | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Edit]** trên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Cập nhật thông tin nhân viên thành công | | |

Bảng UseCase cập nhật thông tin nhân viên

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Cập nhật thông tin nhân viên thành công** | | | |
| 1 | Actor chọn 1 nhân viên rồi click button **[Edit]** ở hiển thị trên màn hình Quản lý nhân viên |  |  |
|  |  | 2 | Hệ thống lấy thông tin của nhân viên cần sửa trong bảng WorkerList trên CSDL rồi hiển thị lên trang Thông tin nhân viên |
| 3 | Cập nhật những thông tin cần thiết rồi ấn button **[Save]** |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Hệ thống update thông tin nhân viên trong bảng WorkerList trên CSDL. Hệ thống thông báo thành công và chuyển về màn hình Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC08: Xóa nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xóa nhân viên | **Code** | UC08 |
| **Description** | Cho phép actor xóa nhân viên trong công ty | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Del]** trên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xóa nhân viên thành công | | |

Bảng UseCase xóa nhân viên

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xóa nhân viên thành công** | | | |
| 1 | Actor chọn 1 nhân viên và click button **[Del]** ở trên màn hình Quản lý nhân viên |  |  |
|  |  | 2 | Hệ thống hiển thị thông báo “Are you sure ?” |
| 3 | Actor click **[Yes]** |  |  |
|  |  | 4 | Hệ thống xóa thông tin nhân viên đó ra khỏi bảng WorkerList và những bảng liên quan trên CSDL. Thông báo thành công và chuyển sang trang Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC09: Tìm kiếm nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Tìm kiếm nhân viên | **Code** | UC09 |
| **Description** | Cho phép actor tìm kiếm nhân viên trong phòng ban bằng mã ID | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Search]** bên cạnh thanh tìm kiếmtrên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Tìm kiếm nhân viên thành công | | |

Bảng UseCase tìm kiếm nhân viên

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Tìm kiếm nhân viên thành công** | | | |
| 1 | Actor nhập mã Id của nhân viên cần tìm vào thanh tìm kiếm rồi click button **[Search]** trênmàn hình Quản lý nhân viên |  |  |
|  |  | 2 | Hệ thống tìm nhân viên có mã ID trùng với ID actor nhập trong bảng WorkerList trên CSDL rồi hiển thị lên list nhân viên trên màn hình Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC10: Chấm công nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Chấm công nhân viên | **Code** | UC10 |
| **Description** | Cho phép actor chấm công nhân viên trong phòng ban. | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Attandance]** |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Actor có được danh sách nhân viên đã chấm công trong ngày | | |

Bảng UseCase chấm công nhân viên

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Danh sách nhân viên đã điểm danh trong ngày** | | | |
| 11 | Actor bấm “Switch” |  |  |
|  |  | 12 | Hệ thống chuyển từ đợt điểm danh ca 1 sang điểm danh ca 2 (ca sáng sang ca chiều) |
| 1 | Actor bấm “Start” |  |  |
|  |  | 2 | Hệ thống lấy list dữ liệu “image” trong bảng WorkerList trong CSDL. |
|  |  | 3 | Hệ thống quét khuôn mặt rồi so sánh trong dữ liệu vừa lấy. |
|  |  | 4 | Hệ thống thêm tên nhân viên và thời gian điểm danh vào danh sách nhân viên đã điểm danh được hiển thị trên màn hình. |
| 5 | Actor bấm “End” |  |  |
|  |  | 7 | Hệ thống hiển thị thông báo kết thúc điểm danh. |
|  |  | 8 | Hệ thống lọc id nhân viên trong danh sách nhân viên đã điểm danh trên màn hình và đưa ra danh sách nhân viên chưa chấm công. |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC11: Xem báo cáo theo ngày

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xem báo cáo theo ngày | **Code** | UC05 |
| **Description** | Cho phép actor tìm kiếm và xem danh sách thống kê nhân viên trong phòng ban đi làm đúng giờ, đi làm muộn và vắng mặt trong 2 ca theo ngày | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Search]** ở màn hình Report\_Day |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xem thông tin danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt thành công | | |

Bảng UseCase xem báo cáo theo ngày

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xem thông tin danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt thành công** | | | |
| 1 | Từ màn hình bất kỳ, actor lựa chọn ở left menu “Report” |  |  |
|  |  | 2 | Hệ thống hiển thị giao diện màn hình Report\_Day |
| 3 | Tại màn hình Report\_Day, Actor chọn ngày rồi bấm nút “Search” |  |  |
|  |  | 4 | Hệ thống đọc dữ liệu trong bảng WorkerList, Attendance, LateList trong CSDL có thuộc tính id phòng ban trùng với id phòng ban của actor, d\_m trùng với ngày mà actor đã chọn, chia theo từng ca, ở mỗi ca lấy được 2 danh sách nhân viên đi làm đúng giờ, nhân viên đi làm muộn, từ 2 danh sách trên lọc ra những nhân viên không có trong 2 danh sách đó (danh sách nhân viên vắng mặt). |
|  |  | 5 | Hiển thị ra màn hình 3 danh sách của ca 1, ẩn các danh sách của ca 2 |
| 6 | Actor chọn ca 2 ở Combobox |  |  |
|  |  | 7 | Ẩn các danh sách của ca 1 và hiển thị các danh sách của ca 2 (và ngược lại nếu actor chọn lại ca ở combobox) |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC12: Xuất báo cáo theo ngày

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xuất danh sách theo ngày | **Code** | UC11 |
| **Description** | Cho phép actor có thông tin danh sách nhân viên nghỉ/ đi làm theo ngày | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Export]** trong trang Report\_Day |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền thông tin ngày-tháng-năm | | |
| **Post condition** | Xuất báo cáo theo ngày thành công | | |

Bảng UseCase xuất báo cáo theo ngày

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xuất báo cáo theo ngày thành công** | | | |
| 1 | Ở cửa sổ Report\_Day, sau khi dữ liệu mà actor muốn lưu đã có ở trên màn hình, actor bấm nút “Export” |  |  |
|  |  | 2 | Hệ thống hiện cửa sổ chọn vị trí lưu file báo cáo. |
| 3 | Người dùng chọn vị trí lưu file và tên file được để mặc định: “Report” (được phép chỉnh sửa), rồi bấm lưu |  |  |
|  |  | 4 | Hệ thống xuất file excel báo cáo vào đúng vị trí người dùng đã chọn. Trong file excel, thông tin của 2 ca được chia làm 2 sheet tương ứng, ở mỗi sheet chứa dữ liệu 3 bảng các danh sách nhân viên đi làm đúng giờ, đi làm muộn và vắng mặt. |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

##### **N/A**

### UC13: Xem báo cáo theo tháng

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xem báo cáo theo tháng | **Code** | UC08 |
| **Description** | Cho phép actor tìm kiếm và xem danh sách nhân viên trong phòng ban về số ca đi làm, đi làm muộn và vắng mặt theo ngày | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Search]** ở màn hình Report­­­­­­­­­­\_Month |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xem thông tin danh sách nhân viên trong phòng ban gồm các thông tin về số ca đi làm, đi làm muộn, vắng mặt thành công | | |

Bảng UseCase xem báo cáo theo tháng

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xem thông tin danh sách nhân viên trong phòng ban gồm các thông tin về số ca đi làm, đi làm muộn, vắng mặt thành công** | | | |
| 1 | Từ màn hình bất kỳ, actor lựa chọn ở left menu “Report”/ “Month” |  |  |
|  |  | 2 | Hệ thống hiển thị giao diện màn hình Report\_Month |
| 3 | Tại màn hình Report\_Month, Actor nhập tháng, năm rồi bấm nút “Search” |  |  |
|  |  | 4 | Hệ thống đọc dữ liệu trong bảngWorkerList, Attendance, LateList trong CSDL có thuộc tính id phòng ban trùng với id phòng ban của actor, thuộc tính d\_m trùng với tháng và năm mà actor đã chọn, từ đó đếm số ca đi làm (là số ca đi làm đúng giờ lấy từ bảng Attendance và số ca đi làm muộn lấy từ bảng LateList), số ca đi làm muộn và số ca ko đi làm (bằng tổng số ca của cả tháng – số ca đi làm đã tính trước đó) của từng nhân viên trong phòng ban, rồi hiển thị danh sách nhân viên với thống kê vừa tính được ra màn hình |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC14: Xuất báo cáo theo tháng

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xuất danh sách theo ngày | **Code** | UC12 |
| **Description** | Cho phép actor có thông tin danh sách nhân viên và số ca đi làm theo tháng | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Export]** trong trang Report\_Month |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền thông tin tháng muốn xem | | |
| **Post condition** | Xuất báo cáo tho tháng thành công | | |

Bảng UseCase xuất báo cáo theo tháng

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xuất báo cáo theo tháng thành công** | | | |
| 1 | Ở cửa sổ Report\_Month, sau khi dữ liệu mà actor muốn lưu đã có ở trên màn hình, actor bấm nút “Export” |  |  |
|  |  | 2 | Hệ thống hiện cửa sổ chọn vị trí lưu file báo cáo. |
| 3 | Người dùng chọn vị trí lưu file và tên file được để mặc định: “Report” (được phép chỉnh sửa), rồi bấm lưu |  |  |
|  |  | 4 | Hệ thống xuất file excel báo cáo vào đúng vị trí người dùng đã chọn. Trong file excel là thông tin danh sách thống kê các ca đi làm của nhân viên mà actor đã tìm kiếm trước đó |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

##### **N/A**

### UC15: Lấy lại mật khẩu

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Register | **Code** | UC13 |
| **Description** | Cho phép actor đăng ký tài khoản vào hệ thống | | |
| **Actor** | Staff | **Trigger** | Actor bấm nút Cant login |
| **Pre-condition** | Điền đủ vào ô tài khoản và mật khẩu, email | | |
| **Post condition** | Chuyển tới trang đăng nhập | | |

Bảng UseCase lấy lại mật khẩu

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Đăng ký thành công** | | | |
| 1 | Người dùng truy cập vào trang đăng ký trên hệ thống. |  |  |
| 2 | Người dùng nhập thông tin email, tên đăng nhập và họ và tên, và bấm nút Recovery |  |  |
|  |  | 3 | Hệ thống kiểm tra tính hợp lệ của thông tin người dùng. Nếu thông tin hợp lệ, hệ thống sẽ gửi 1 mã mật khẩu mới qua email |
|  |  | 4 | Hệ thống chuyển đến trang Đăng nhập. |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

|  |  |
| --- | --- |
| MS05 | “Vui lòng điền đủ thông tin cá nhân để lấy lại tài khoản.”  Message thông báo khi actor không nhập đủ thông tin được yêu cầu |
| MS06 | “Lấy mật khẩu thành công”  Message thông báo khi actor điền đủ thông tin và hợp lệ sau khi hệ thống kiểm tra |

### UC16: Xem bảng chấm công

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xem bảng chấm công | **Code** | UC11 |
| **Description** | Cho phép actor xem bảng chấm công của phòng ban theo tháng, năm | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Accountant]** ở left menu |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xem bảng chấm công thành công | | |

Bảng UseCase xem bảng chấm công

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xem bảng lương thành công** | | | |
| 1 | Từ màn hình bất kỳ, actor lựa chọn ở left menu “Accountant” |  |  |
|  |  | 2 | Hệ thống hiển thị giao diện màn hình Accountant với dữ liệu được chọn, nhập từ trước là tháng và năm của hiện tại. |
|  |  | 4 | Hệ thống lấy dữ liệu từ bảng WorkerList, Accountant và LateList trong CSDL để tạo 1 bảng lương của phòng ban gồm id nhân viên, tên nhân viên, số ca của các ngày trong tháng, tổng số ca đi làm, tổng số ca đi muộn, lương của nhân viên tương ứng với tháng, năm đã được chọn, nhập từ trước đó rồi hiển thị bảng lương lên màn hình |
| 3 | Tại màn hình Accountant, Actor thay đổi tháng và năm |  |  |
|  |  | 4 | Khi người thay đổi, hệ thống lấy bảng lương tương ứng với tháng, năm mà actor đã thay đổi rồi hiển thị ra màn hình. |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC17: Xuất bảng chấm công

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xuất bảng chấm công | **Code** | UC14 |
| **Description** | Cho phép actor lấy bảng chấm công theo phòng ban và tháng | | |
| **Actor** | Accoutant | **Trigger** | Actor click button **[Export Excel]** trong trang Accountant |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền tên của phòng ban và tháng muốn lấy thông tin | | |
| **Post condition** | Không có | | |

Bảng UseCase xuất bảng chấm công

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Gửi danh sách theo ngày qua email** | | | |
| 1 | Ở màn hình Accountant, sau khi dữ liệu mà actor muốn lưu đã có ở trên màn hình, actor bấm nút “Export” |  |  |
|  |  | 2 | Hệ thống hiện cửa sổ chọn vị trí lưu file báo cáo. |
| 3 | Người dùng chọn vị trí lưu file và tên file được để mặc định: “Report” (được phép chỉnh sửa), rồi bấm lưu |  |  |
|  |  | 4 | Hệ thống xuất file excel báo cáo vào đúng vị trí người dùng đã chọn. Trong file excel là thông tin của bảng lương |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

**N/A**

### UC18: : Gửi báo cáo theo ngày qua email

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Gửi báo cáo theo ngày qua email | **Code** | UC15 |
| **Description** | Cho phép actor gửi báo cáo danh sách nhân viên nghỉ/ đi làm theo ngày qua email | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Send]** trong trang Report\_Day |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền thông tin ngày-tháng-năm | | |
| **Post condition** | Gửi báo báo theo ngày qua email thành công | | |

Bảng UseCase gửi báo cáo theo ngày qua email

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Gửi báo cáo theo ngày qua email thành công** | | | |
| 1 | Ở cửa sổ Report\_Day, sau khi đã xuất file báo cáo actor bấm nút “Send” |  |  |
|  |  | 2 | Hệ thống hiển thị cửa số Send lên màn hình |
| 3 | Tại cửa sổ Send, actor chọn email trong danh sách muốn gửi (nếu không có thì nhập email vào thanh input), nhập các thông tin cần thiết |  |  |
| 4 | Actor bấm vào nút “chọn file” |  |  |
|  |  | 5 | Hệ thống mở cửa sổ chọn file |
| 6 | Tại của sổ chọn file, actor chọn file báo cáo đã xuất từ trước (có thể chọn nhiều file) rồi bấm “Open” |  |  |
|  |  | 7 | Hệ thống lấy được danh sách vị trí file muốn gửi, hiển thị lên cửa sổ Send |
| 8 | Tại của sổ Send, actor bấm “Send” |  |  |
|  |  | 9 | Hệ thống sẽ truy cập vào email của công ty sau đó gửi mail đã được nhập/chọn trước chứa những thông tin và file mà actor đã điền và chọn. |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

|  |  |
| --- | --- |
| MS07 | “Gửi thành công”  Message thông báo khi hệ thống đã gửi thành công tới các email |
| MS08 | “Gửi thất bại”  Message thông báo khi hệ thống không gửi được file tới các email |

### UC19: Gửi báo cáo theo tháng qua email

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Gửi báo cáo theo tháng qua email | **Code** | UC16 |
| **Description** | Cho phép actor gửi báo cáo danh sách nhân viên và số ca đi làm theo tháng qua email | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Send]** trong trang Report\_Month |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền thông tin tháng muốn xem | | |
| **Post condition** | Gửi báo cáo theo tháng qua email thành công | | |

Bảng Gửi báo cáo theo tháng qua email

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Gửi báo cáo theo tháng qua emai thành công** | | | |
| 1 | Ở cửa sổ Report\_Month, sau khi đã xuất file báo cáo actor bấm nút “Send” |  |  |
|  |  | 2 | Hệ thống hiển thị cửa số Send lên màn hình |
| 3 | Tại cửa sổ Send, actor chọn email trong danh sách muốn gửi (nếu không có thì nhập email vào thanh input), nhập các thông tin cần thiết |  |  |
| 4 | Actor bấm vào nút “chọn file” |  |  |
|  |  | 5 | Hệ thống mở cửa sổ chọn file |
| 6 | Tại của sổ chọn file, actor chọn file báo cáo đã xuất từ trước (có thể chọn nhiều file) rồi bấm “Open” |  |  |
|  |  | 7 | Hệ thống lấy được danh sách vị trí file muốn gửi, hiển thị lên cửa sổ Send |
| 8 | Tại của sổ Send, actor bấm “Send” |  |  |
|  |  | 9 | Hệ thống sẽ truy cập vào email của công ty sau đó gửi mail đã được nhập/chọn trước chứa những thông tin và file mà actor đã điền và chọn. |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

|  |  |
| --- | --- |
| MS09 | “Gửi thành công”  Message thông báo khi hệ thống đã gửi thành công tới các email |
| MS10 | “Gửi thất bại”  Message thông báo khi hệ thống không gửi được file tới các email |

### UC20: Gửi bảng chấm công qua Email

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xuất bảng chấm công | **Code** | UC17 |
| **Description** | Cho phép actor gửi file bảng chấm công theo phòng ban cho email được chọn | | |
| **Actor** | Accoutant | **Trigger** | Actor click button **[Send]** trong trang Accountant |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền tên của phòng ban và tháng muốn lấy thông tin | | |
| **Post condition** | Gửi bảng chấm công qua email thành công | | |

Bảng UseCase gửi bảng chấm công qua email

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Gửi bảng chấm lương qua email thành công** | | | |
| 1 | Ở màn hình Accountant, sau khi đã xuất file báo cáo actor bấm nút “Send” |  |  |
|  |  | 2 | Hệ thống hiển thị cửa số Send lên màn hình |
| 3 | Tại cửa sổ Send, actor chọn email trong danh sách muốn gửi (nếu không có thì nhập email vào thanh input), nhập các thông tin cần thiết |  |  |
| 4 | Actor bấm vào nút “chọn file” |  |  |
|  |  | 5 | Hệ thống mở cửa sổ chọn file |
| 6 | Tại của sổ chọn file, actor chọn file báo cáo đã xuất từ trước (có thể chọn nhiều file) rồi bấm “Open” |  |  |
|  |  | 7 | Hệ thống lấy được danh sách vị trí file muốn gửi, hiển thị lên cửa sổ Send |
| 8 | Tại của sổ Send, actor bấm “Send” |  |  |
|  |  | 9 | Hệ thống sẽ truy cập vào email của công ty sau đó gửi mail đã được nhập/chọn trước chứa những thông tin và file mà actor đã điền và chọn. |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

**N/A**

### UC21: Thêm nhân viên đi muộn

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Thêm nhân viên đi muộn | **Code** | UC01 |
| **Description** | Cho phép actor thêmthông tin nhân viên đi làm muộn trong phòng ban | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Add]** trên màn hình Late Manage |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Thêm thành công thông tin nhân viên đi làm muộn | | |

Bảng UseCase thêm nhân viên đi muộn

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Thêm thành công nhân viên đi làm muộn** | | | |
| 1 | Từ màn hình bất kỳ, actor lựa chọn ở left menu “Late Manage”/ “Add” |  |  |
|  |  | 2 | Hệ thống hiển thị màn hình AddLate và hiển thị form nhập thông tin cần thiết |
| 3 | Actor nhập các thông tin cần thiết |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Lưu thông tin nhân viên đó đi làm muộn vào bảng LateList trên CSDL, thông báo thành công. |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC22: Sửa thông tin nhân viên đi muộn

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Sửa thông tin nhân viên đi muộn | **Code** | UC02 |
| **Description** | Cho phép actor sửa thông nhân viên đi làm muộn trong phòng ban. | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Edit]** ở màn hình LateManage |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Sửa thông tin nhân viên đi muộn thành công | | |

Bảng UseCase sửa thông tin nhân viên đi muộn

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Sửa thông tin nhân viên đi làm muộn thành công** | | | |
| 1 | Từ màn hình LateManage, actor chọn thông tin nhân viên đi muộn trong phòng ban trên DataGrid rồi bấm nút “Edit” |  |  |
|  |  | 2 | Hệ thống hiển thị màn hình EditLate và hiển thị form nhập thông tin đã được điền sẵn thông tin của nhân viên đi muộn đó |
| 3 | Actor sửa lại các thông tin cần thiết |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Sửa lại thông tin nhân viên đó đi làm muộn ở bảng LateList trên CSDL, thông báo thành công. |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC23: Xóa thông tin nhân viên đi muộn

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xóa nhân viên đi làm muộn | **Code** | UC03 |
| **Description** | Cho phép actor xóa thông tin nhân viên đi làm muộn trong phòng ban | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Delete]** ở màn hình LateManage |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xóa thông tin nhân viên đi làm muộn thành công | | |

Bảng UseCase xóa thông tin nhân viên đi muộn

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xóa thông tin nhân viên đi làm muộn thành công** | | | |
| 1 | Từ màn hình LateManage, actor chọn thông tin nhân viên đi muộn trong phòng ban trên DataGrid rồi bấm nút “Delete” |  |  |
|  |  | 2 | Hệ thống hiển thị cửa sổ thống báo xác nhận xóa |
| 3 | Actor click button “OK” |  |  |
|  |  | 4 | Hệ thống xóa dữ liệu thông tin nhân viên đó đi làm muộn trong bảng LateList trên CSDL, rồi thông báo thành công |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC24: Tìm thông tin nhân viên đi muộn

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Tìm thông tin nhân viên đi muộn | **Code** | UC04 |
| **Description** | Cho phép actor tìm kiếm thông tin nhân viên hoặc các nhân viên trong phòng ban đi muộn bằng tên nhân viên , tháng , năm hoặc ngày cụ thể | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Search]** ở màn hình LateManage |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Tìm kiếm thông tin thành công | | |

Bảng UseCase tìm thông tin nhân viên đi muộn

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Tìm kiếm nhân viên thành công** | | | |
| 1 | Ở màn hình LateManage, actor nhập tên của nhân viên cần tìm vào thanh tìm kiếm, chọn tháng trong comboBox và nhập năm hoặc chọn ngày rồi click button **[Search]** ở trên màn hình |  |  |
|  |  | 2 | Hệ thống tìm thông tin nhân viên hoặc các nhân viên trong phòng ban đi muộn có thông tin trùng khớp với thông tin mà actor đã nhập trong bảng LateList trên CSDL rồi hiển thị lên bảng danh sách trên màn hình LateList |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC25: Gửi yêu cầu

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Gửi yêu cầu | **Code** |  |
| **Description** | Cho phép actor gửi yêu cầu xóa điểm danh của nhân viên | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Send]** trong trang Request của Staff |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền thông tin nhân viên bị điểm danh sai | | |
| **Post condition** | Gửi thành công yêu cầu cho admin | | |

Bảng UseCase gửi yêu cầu

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Gửi bảng chấm công qua email thành công** | | | |
| 1 | Người dùng truy cập vào hệ thống và chọn chức năng “Request". |  |  |
|  |  | 2 | Hệ thống hiển thị trang Request |
| 3 | Người dùng nhập thông tin được yêu cầu gồm tên công nhân, id, ngày điểm danh, ca điểm danh |  |  |
| 4 | Người dùng bấm Send |  |  |
|  |  | 5 | Hệ thống truy xuất dữ liệu của công nhân bị điểm danh nhầm, và ghi vào csdl để gửi cho admin |
|  |  |  | Hệ thống hiển thị thông báo gửi thành công nếu tìm được công nhân đó nếu không, hệ thống hiển thị thông báo yêu cầu nhập lại |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

|  |  |
| --- | --- |
| MS | “Request sent successful”  Message thông báo khi hệ thống truy xuất được đúng dữ liệu của nhân viên bị điểm danh sai |
| MS | “Request sent failure”  Message thông báo khi hệ thống không truy xuất được dữ liệu của nhân viên bị điểm danh sai |

### UC26: Xử lý yêu cầu

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xử lý yêu cầu | **Code** |  |
| **Description** | Cho phép actor xử lý các yêu cầu từ staff | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Accept] hoặc[Deny]** trong trang Request của Admin |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, chọn 1 yêu cầu | | |
| **Post condition** | Xử lý thành công yêu cầu | | |

Bảng UseCase xử lý yêu cầu

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Gửi bảng chấm công qua email thành công** | | | |
| 1 | Người dùng truy cập vào hệ thống và chọn chức năng “Request". |  |  |
|  |  | 2 | Hệ thống hiển thị trang Request |
| 3 | Người dùng chọn 1 yêu cầu đang ở trạng thái”Idle” và chọn Accept hoặc Deny |  |  |
|  |  | 4 | Nếu người dùng chọn Accept, hệ thống sẽ xóa chấm công của nhân viên được gửi đến trong yêu cầu từ staff và chuyển trạng thái của yêu cầu thành Accepted, hiển thị thông báo yêu cầu được chấp thuật. Nếu chọn deny, hệ thống sẽ chuyển trạng thái của yêu cầu thành denied và hiển thị thông báo yêu cầu bị từ chối |
|  |  | 5 | Hệ thống hiển thị lại bảng yêu cầu |

##### **Business Rules**

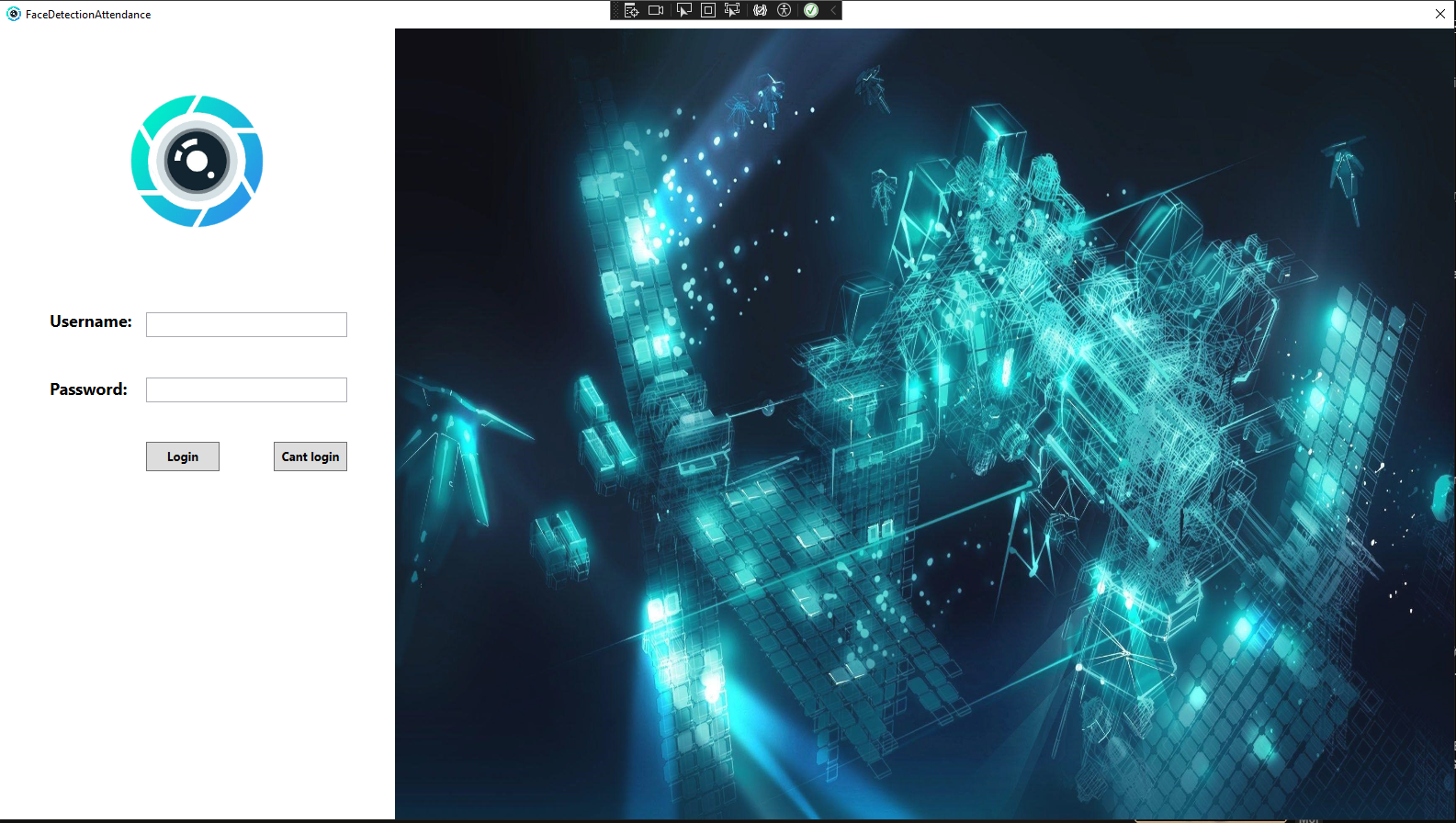
|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

|  |  |
| --- | --- |
| MS | “Request accepted”  Message thông báo khi hệ thống xóa thành công nhân viên bị điểm danh sai |
| MS | “Request sent failure”  Message thông báo khi hệ thống chuyển thành công trạng thái của request thành Denied |

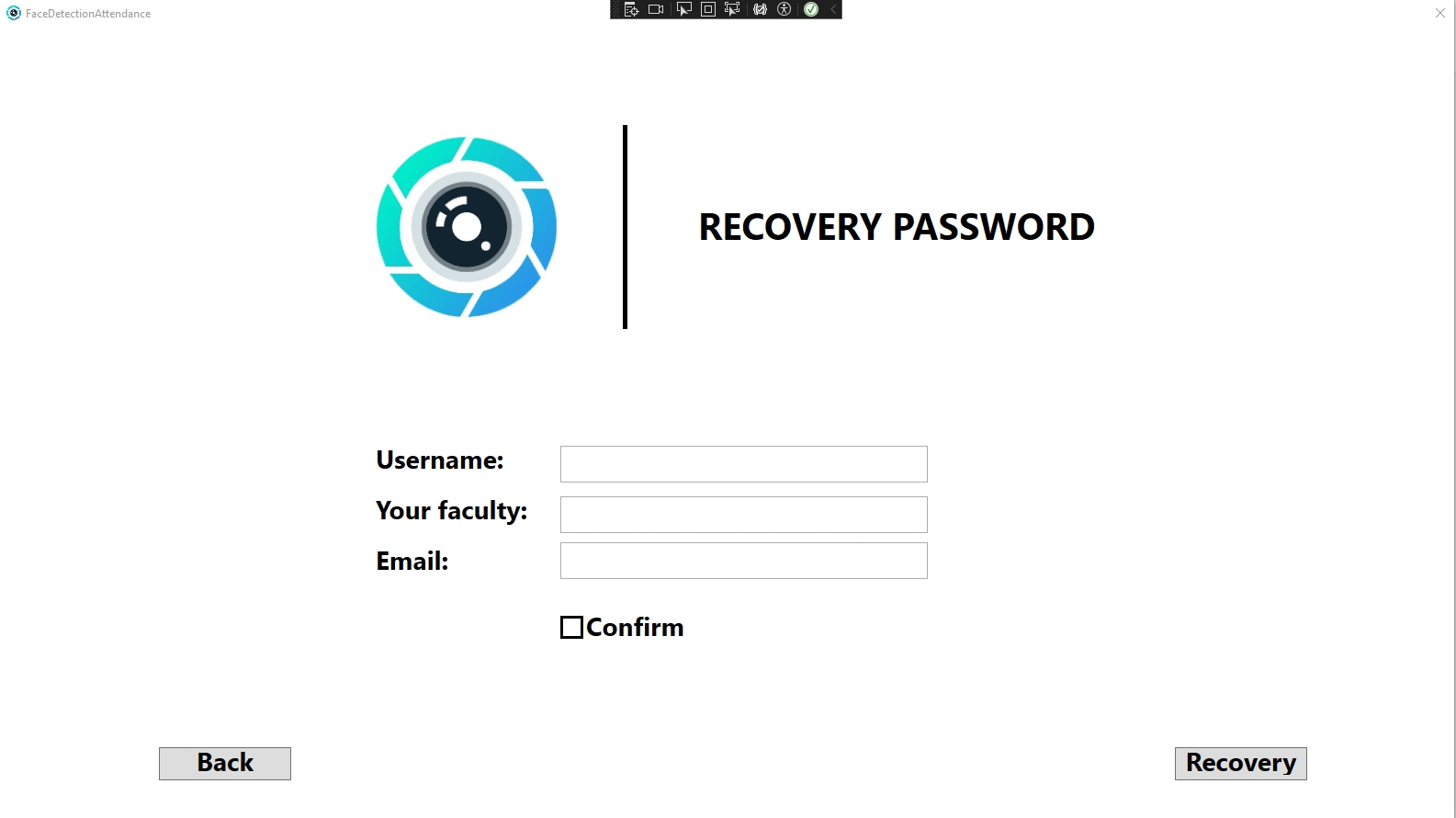
**CHƯƠNG 5: GIAO DIỆN NGƯỜI DÙNG**

**5.1 Login**

****

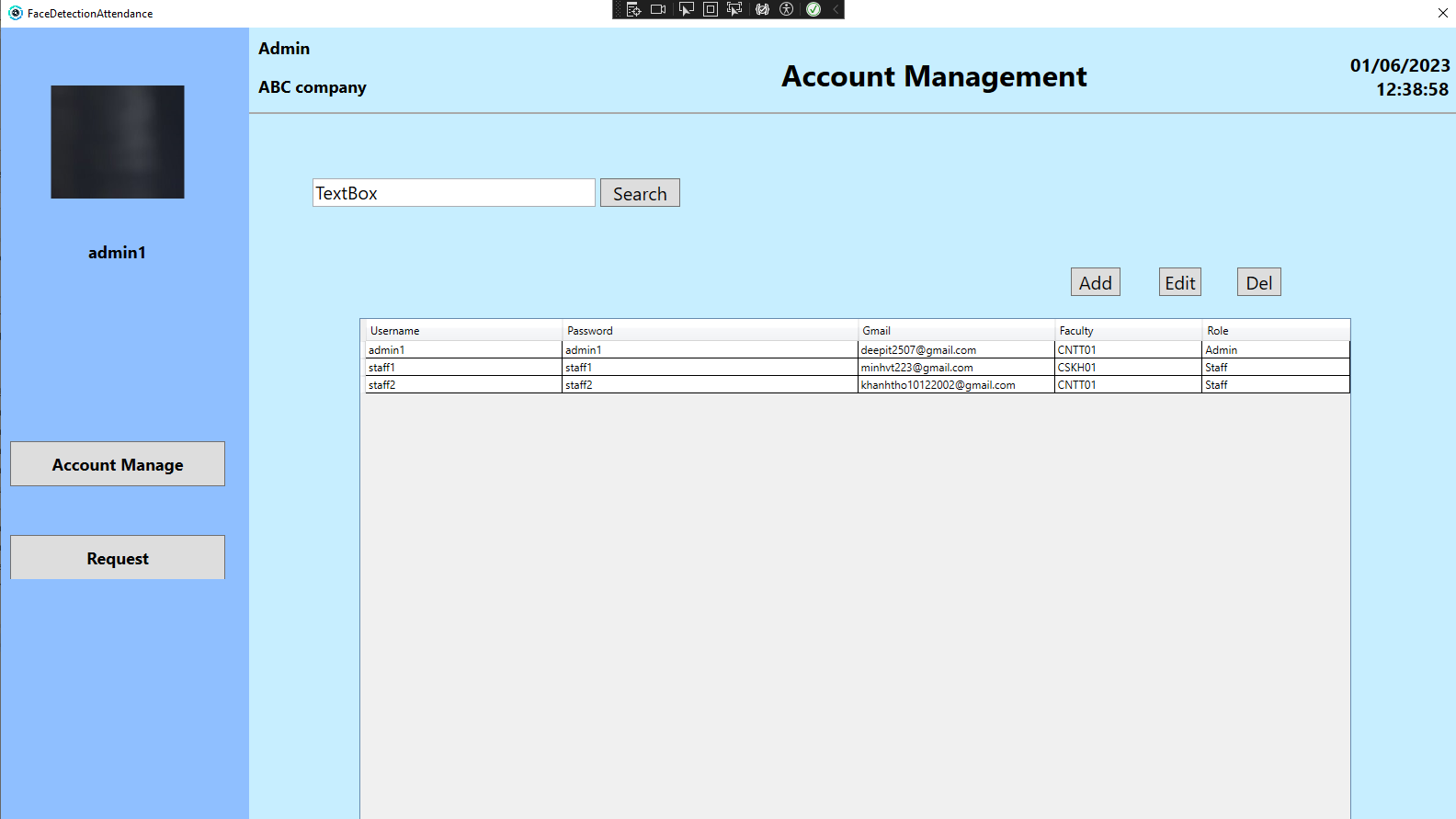
Hình Giao diện màn hình Login

**5.2 Recovery**

****

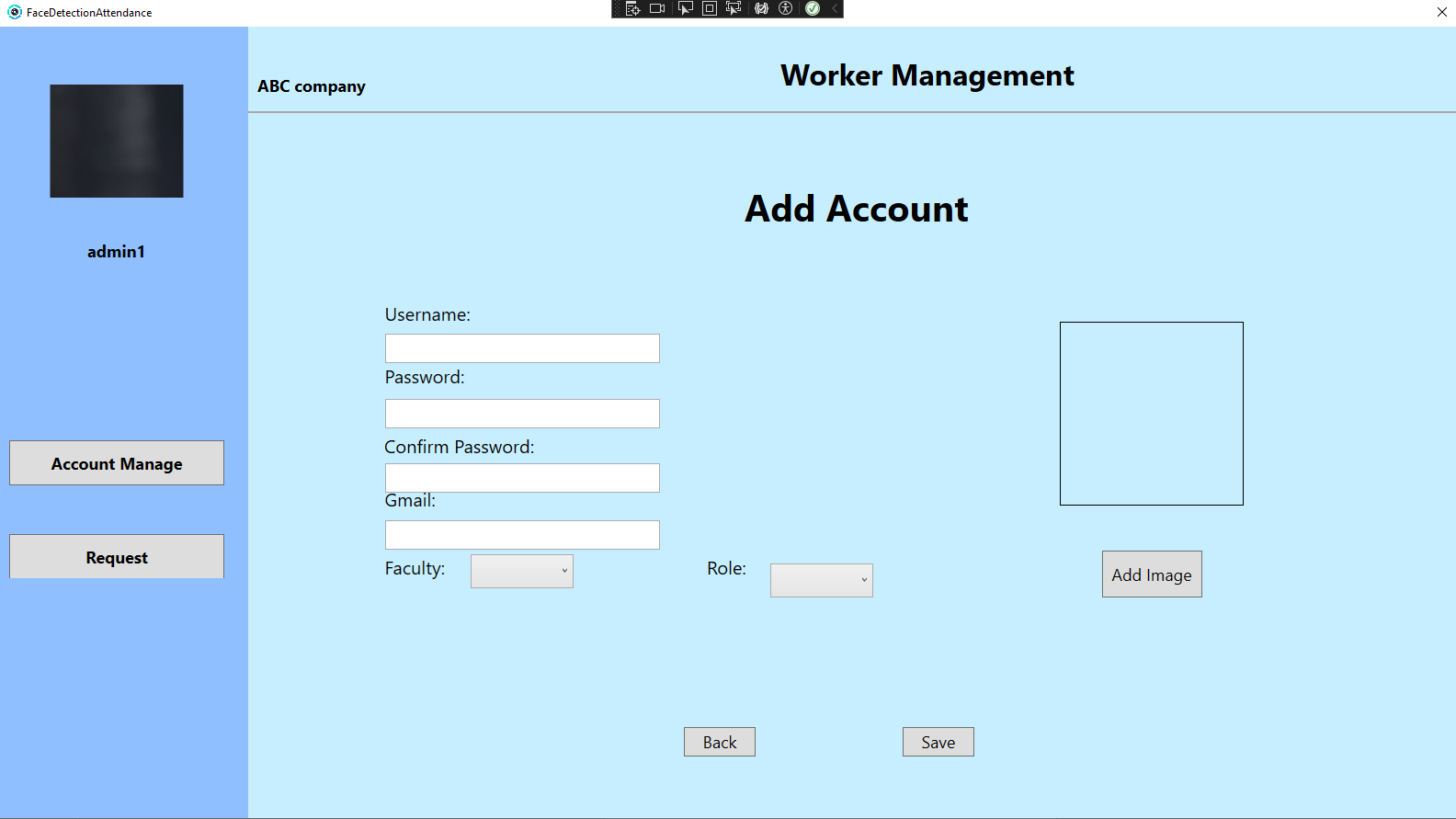
Hình Giao diện màn hình Recovery

**5.3 Account manage**

****

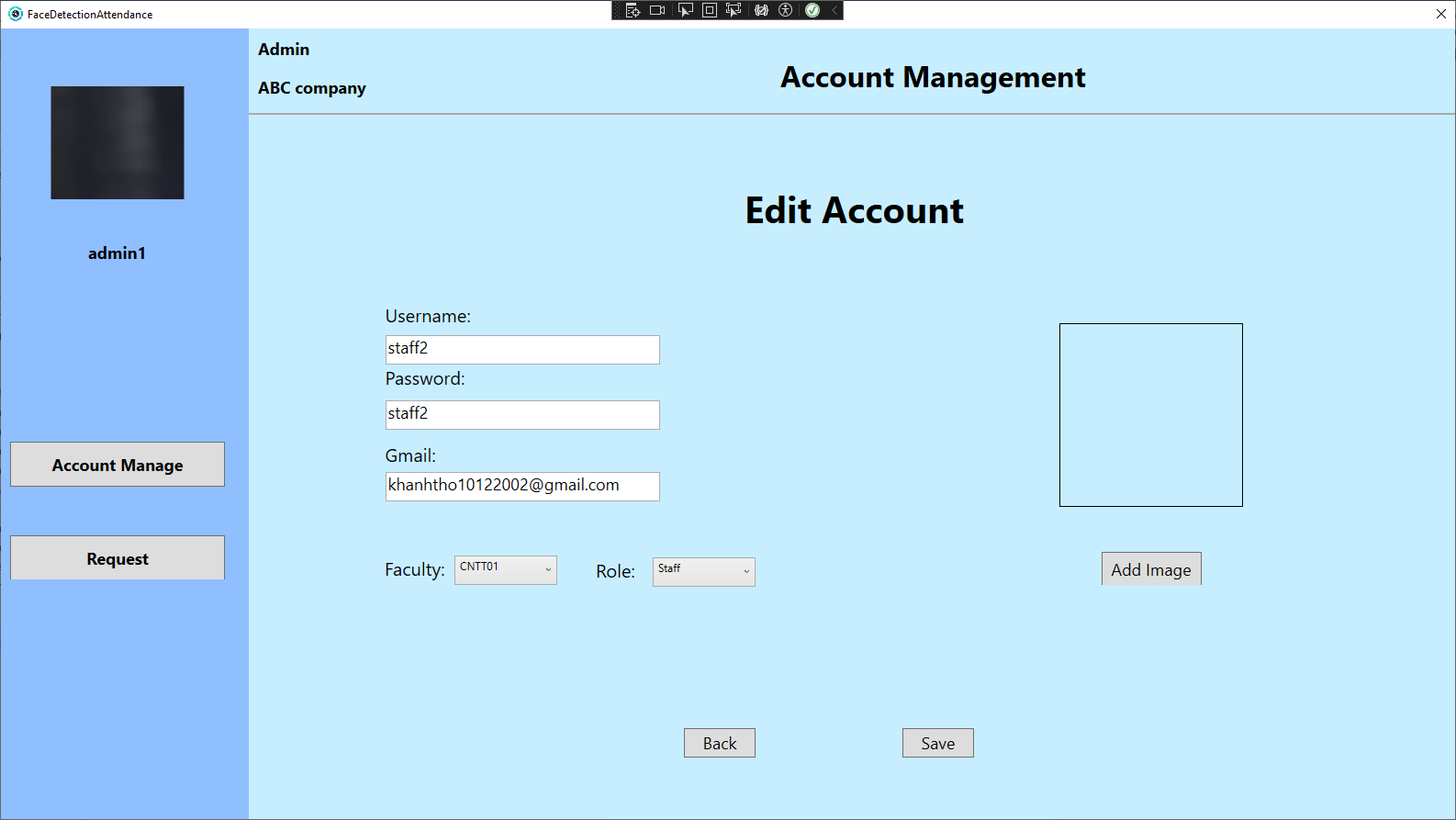
Hình Giao diện màn hình Account manage

**5.4 Add account**

****

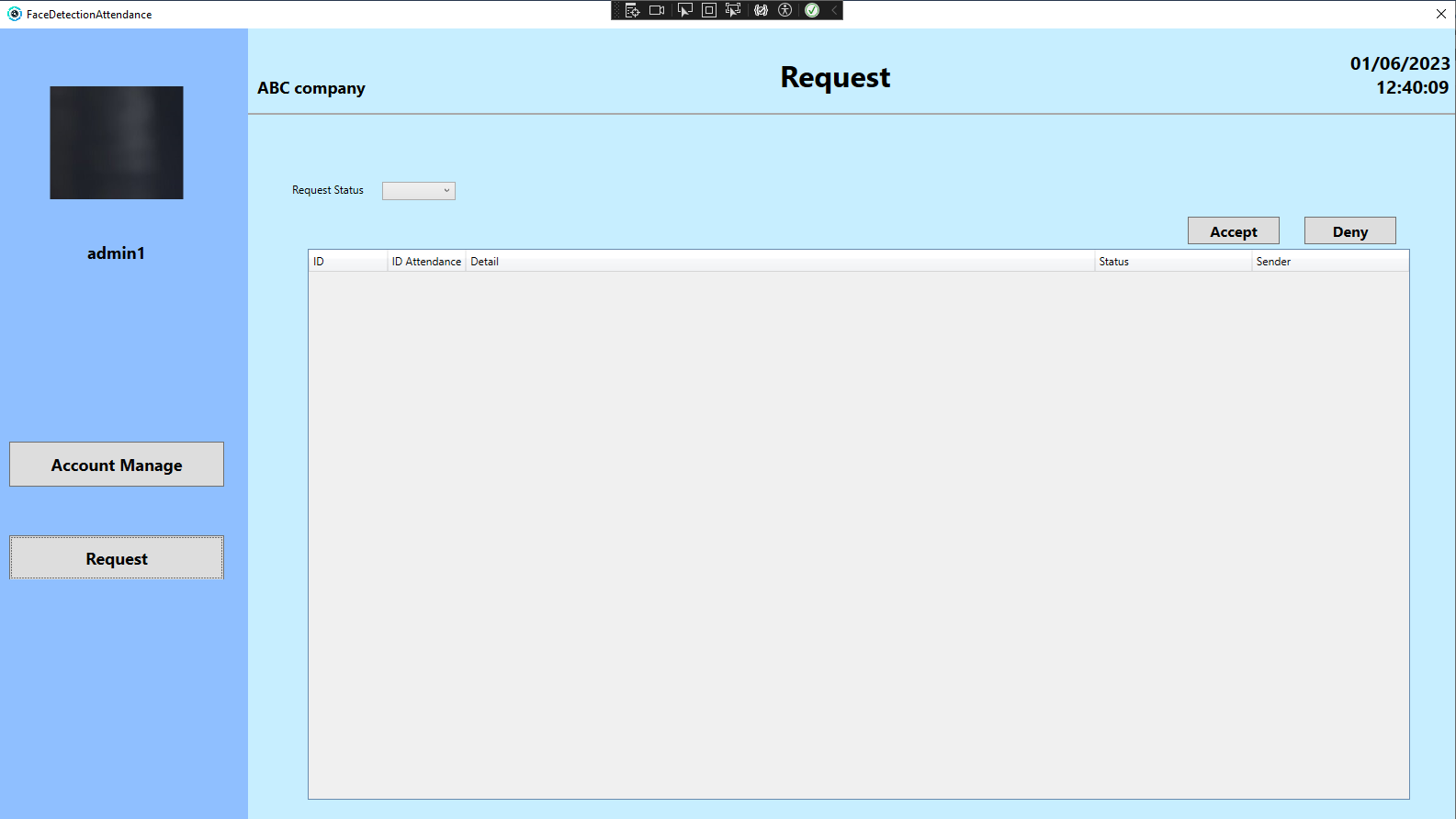
Hình Giao diện màn hình Add account

**5.5 Edit account**

****

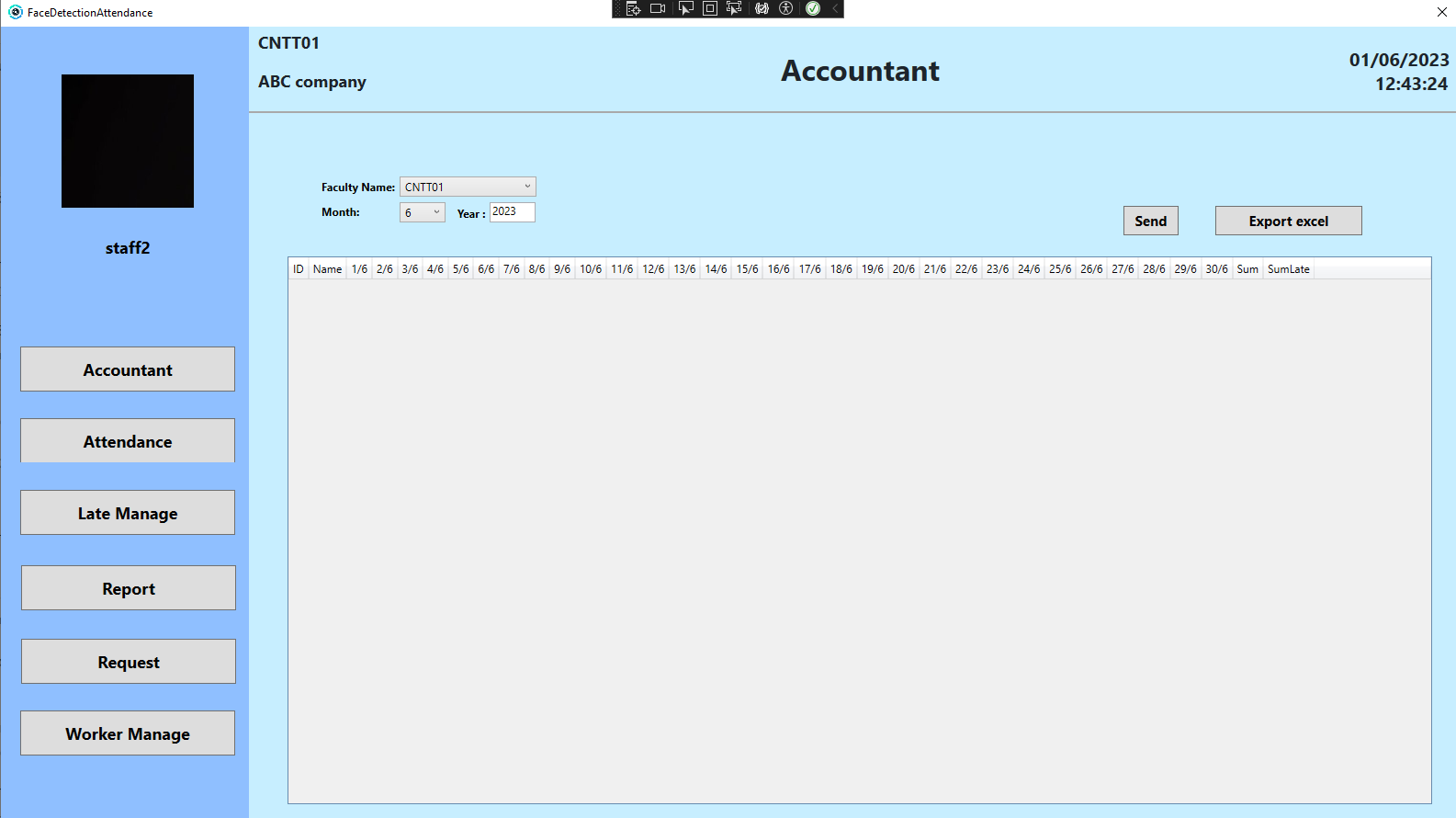
Hình Giao diện màn hình Edit account

**5.5 Request(Admin)**

****

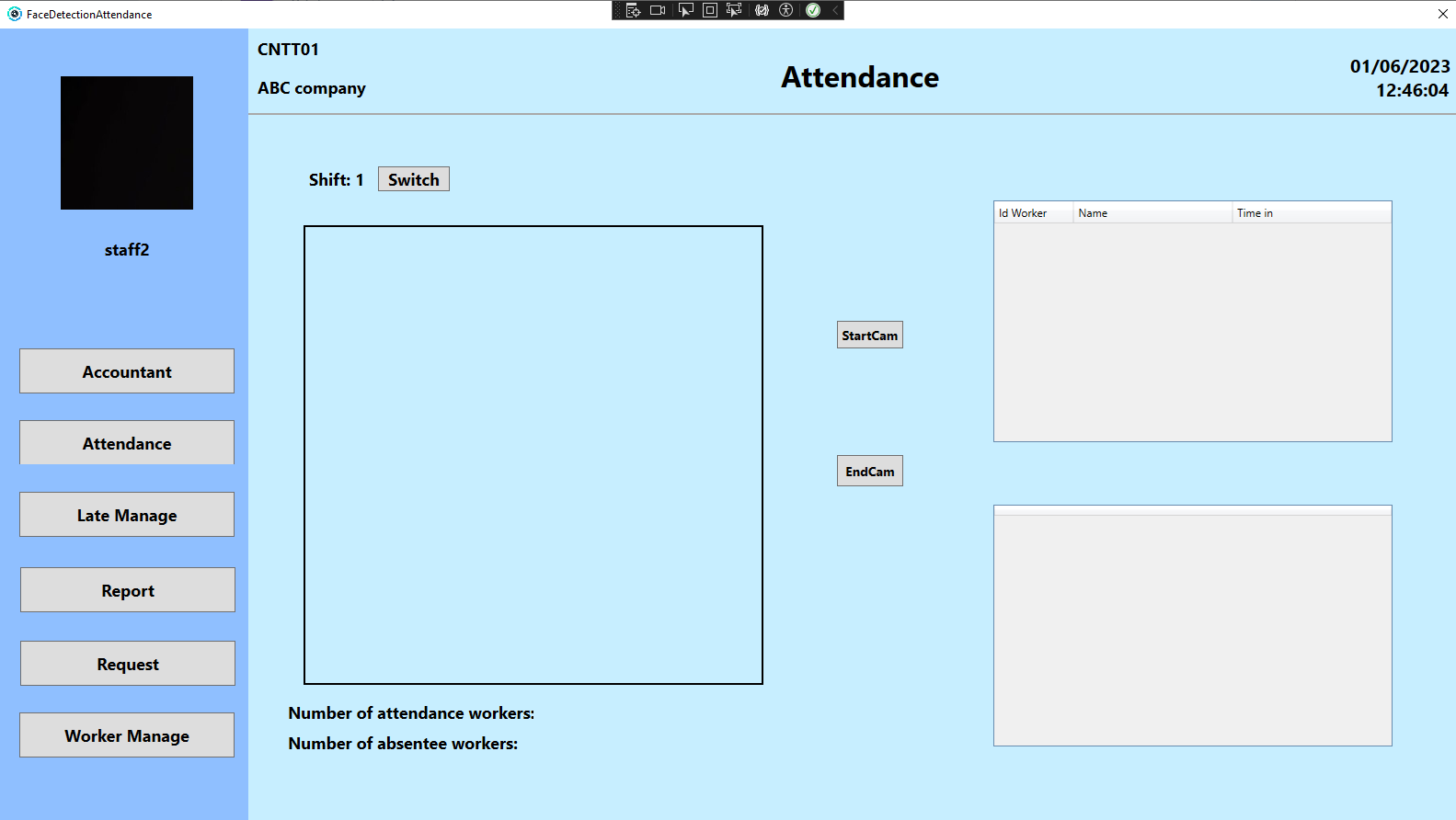
Hình Giao diện màn hình Request(Admin)

**5.6 Accountant**

****

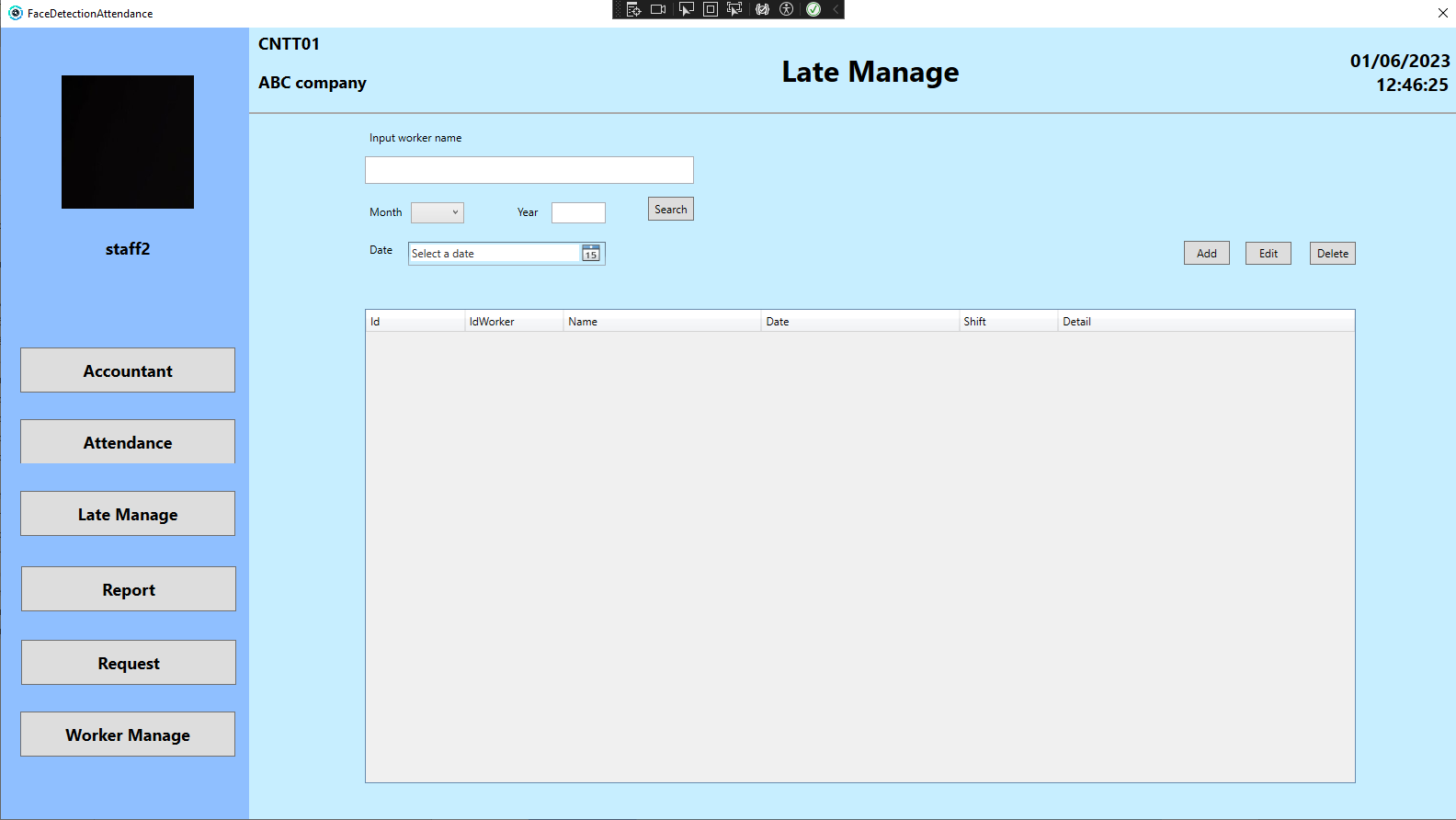
Hình Giao diện màn hình Accountant

**5.7 Attendance**

****

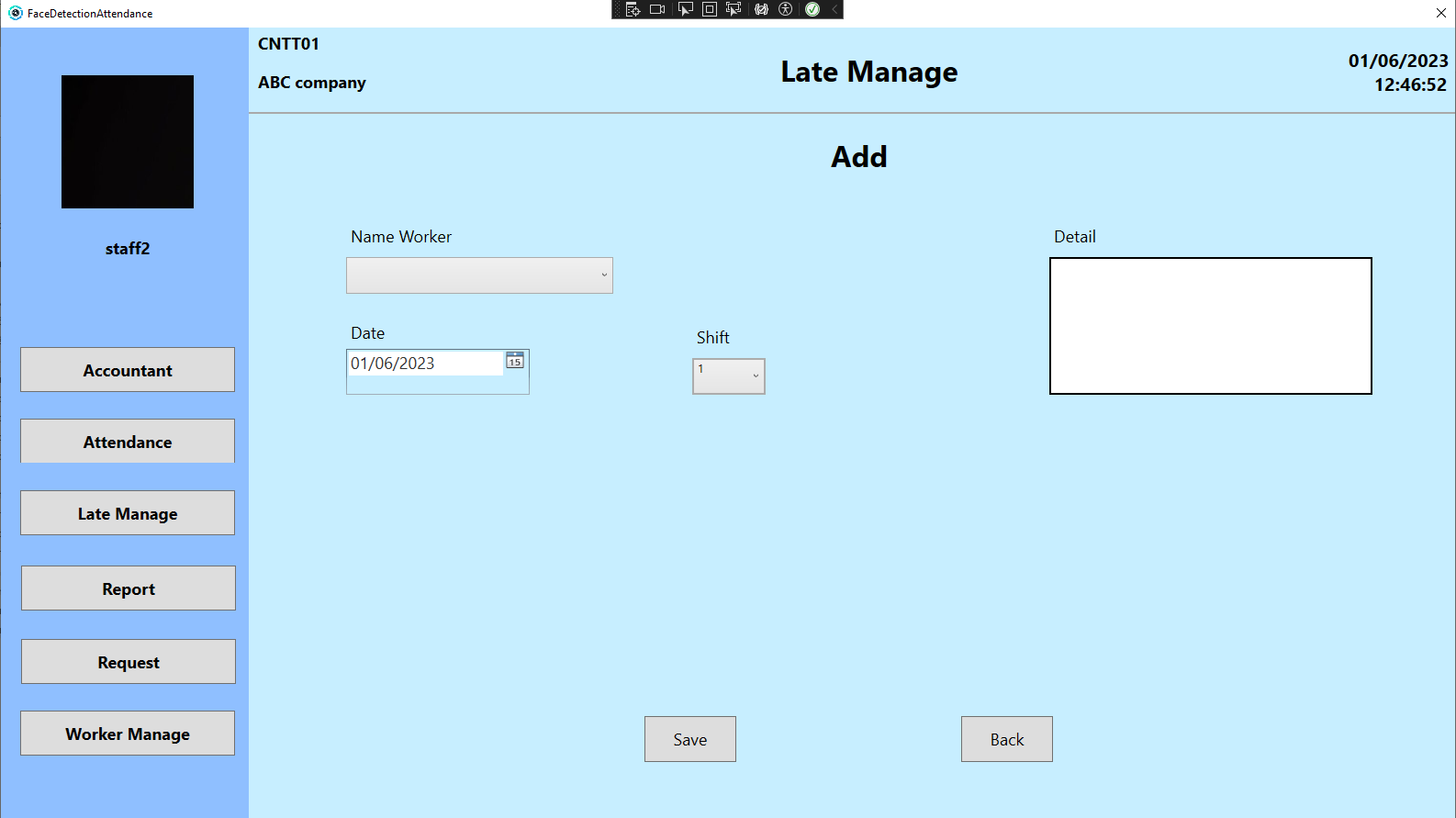
Hình Giao diện màn hình Attendance

**5.8 Late manage**

****

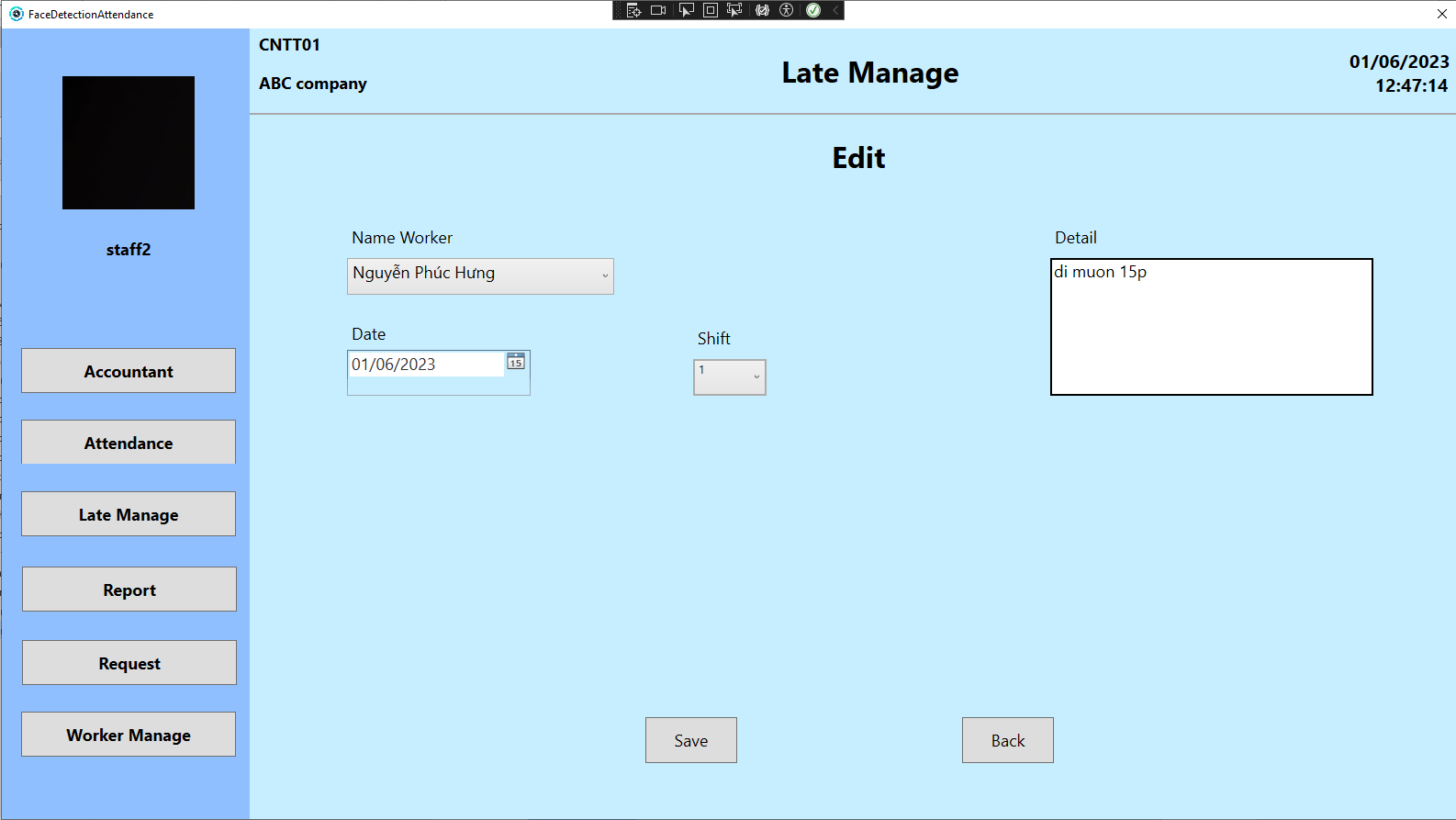
Hình Giao diện màn hình Late manage

**5.9 Add late**

****

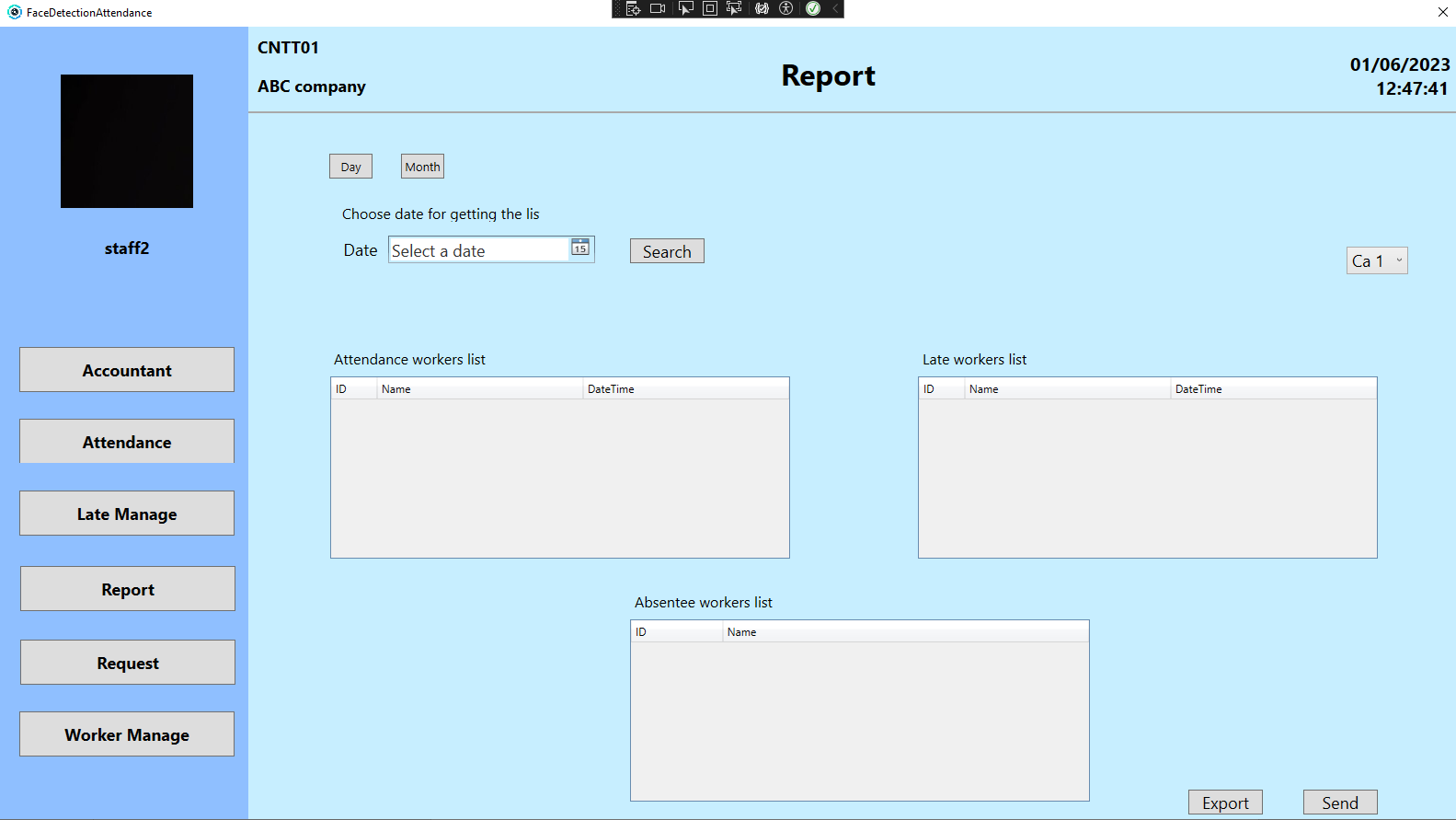
Hình Giao diện màn hình Add late

**5.10 Edit late**

****

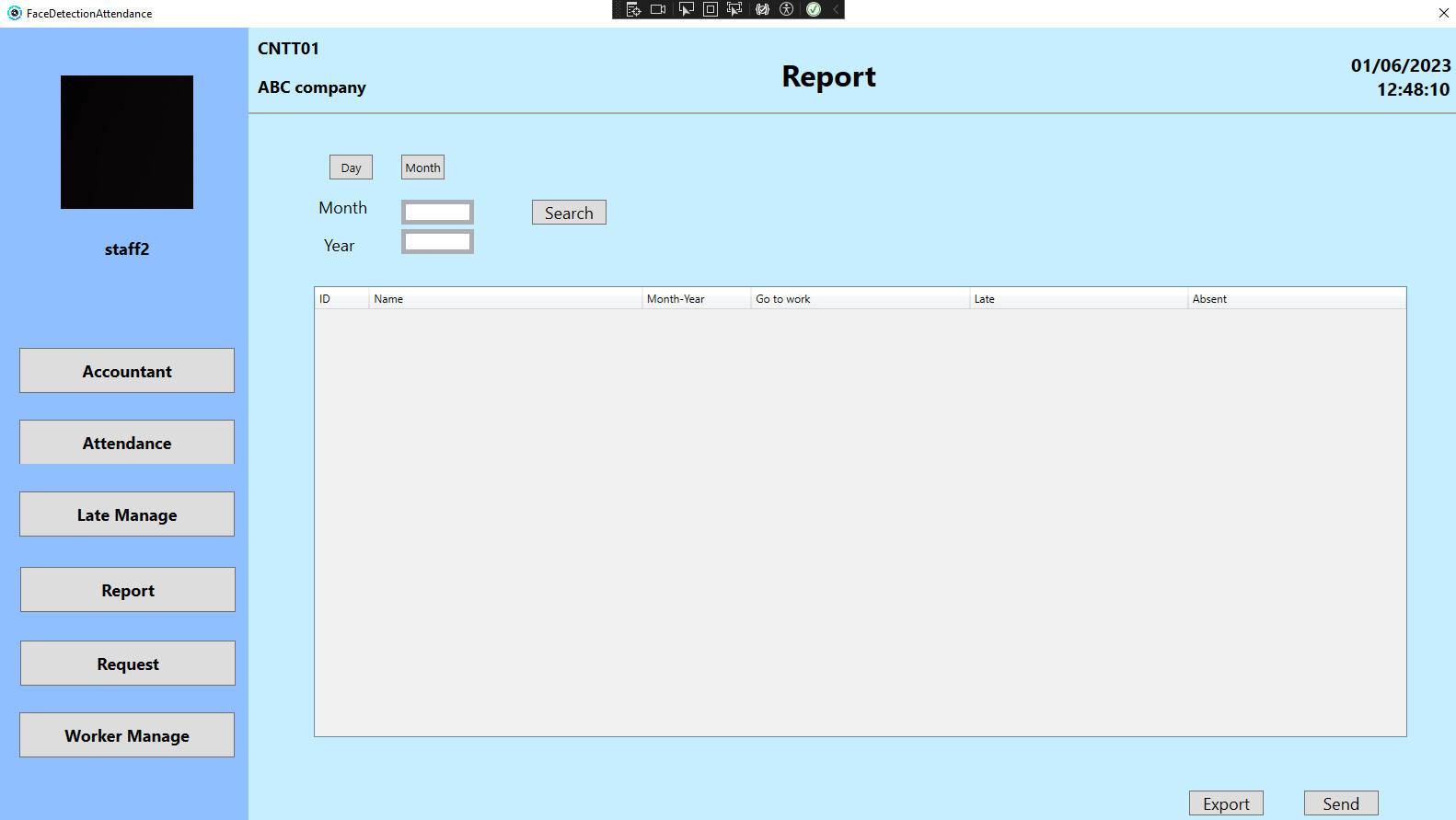
Hình Giao diện màn hình Edit late

**5.11 Report day**

****

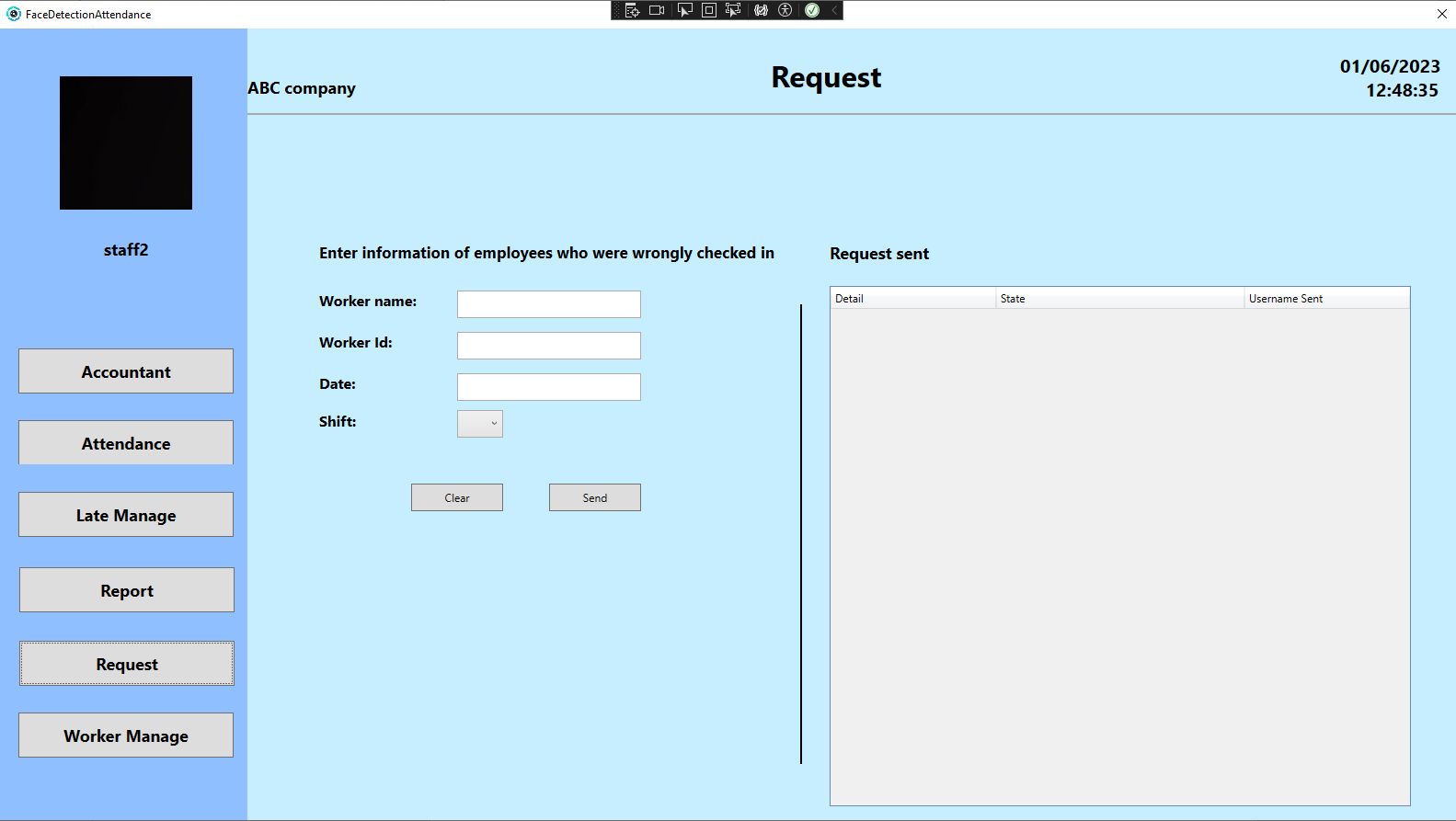
Hình Giao diện màn hình Report day

**5.12 Report month**

****

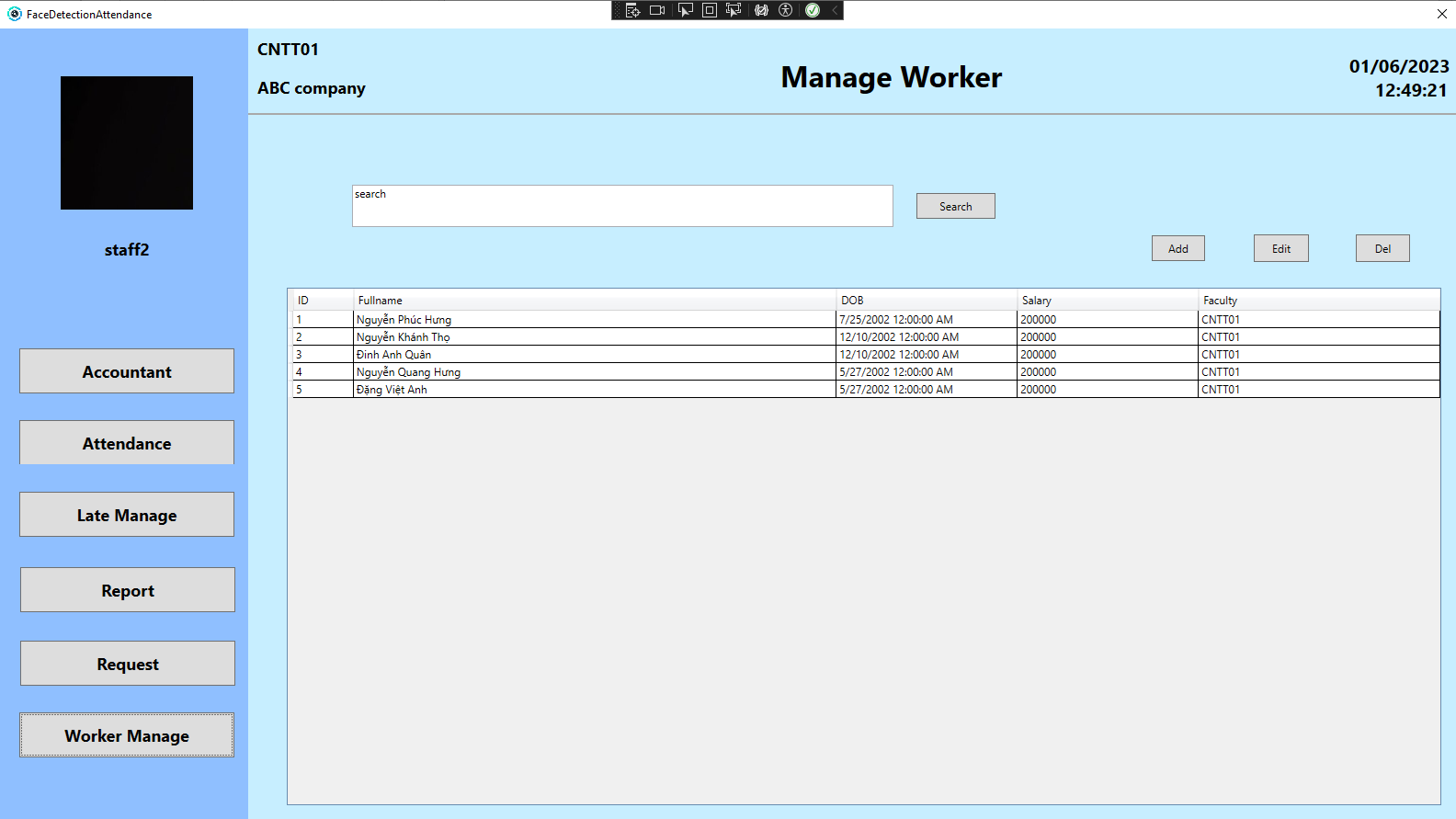
Hình Giao diện màn hình Report month

**5.13 Request(Staff)**

****

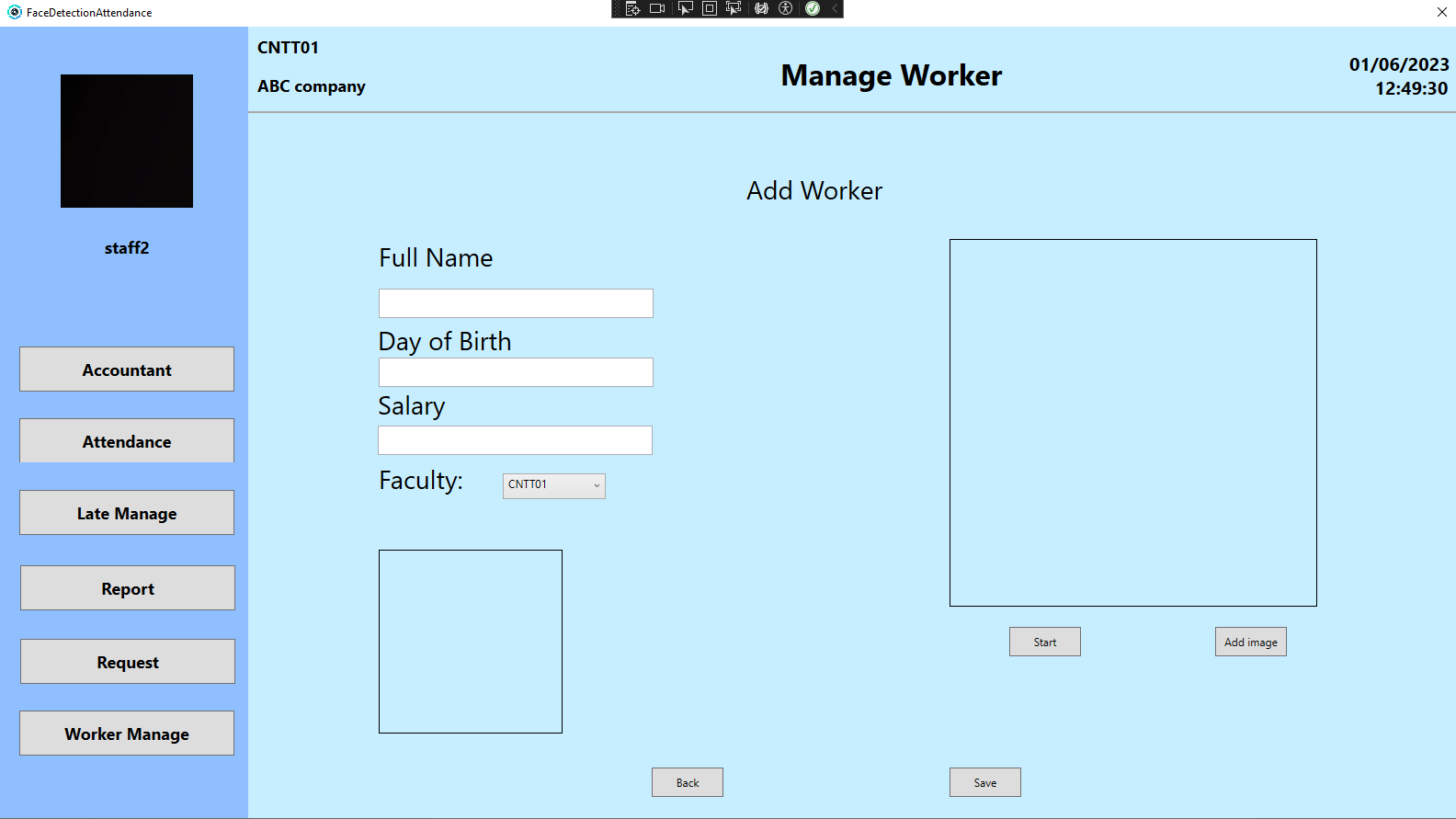
Hình Giao diện màn hình Request(Staff)

**5.14 Worker manage**

****

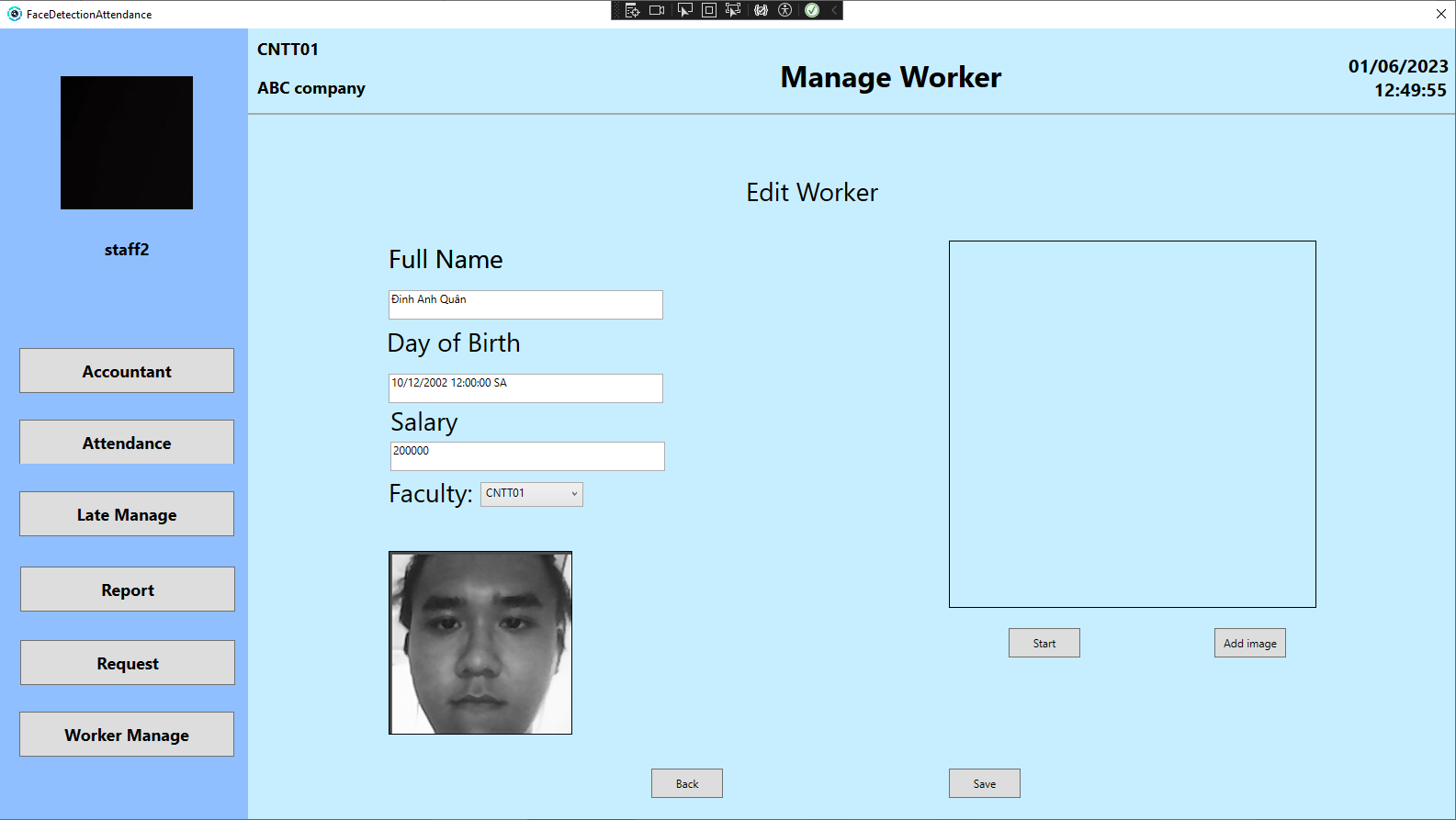
Hình Giao diện màn hình worker manage

**5.15 Add worker**

****

Hình Giao diện màn hình Add worker

**5.16 Edit worker**

****

Hình Giao diện màn hình edit worker

**CHƯƠNG 6: THỰC HIỆN BÀI TOÁN**

**Màn hình Attendance:**

+ Giao diện: Hiển thị thông tin nhân viên đi làm và thông tin nhân viên vắng mặt và số lượng nhân viên đi làm và vắng mặt. Nhận vào hình ảnh của nhân viên thông qua webcam.

+ Mã lệnh:

var frame = \_videoSource.QueryFrame().ToImage<Bgr, byte>().Resize(640, 480, Inter.Cubic);

var grayFrame = frame.Convert<Gray, byte>();

// Detect faces in current frame

var faces = \_faceClassifier.DetectMultiScale(grayFrame, 1.2, 5);

foreach (var face in faces)

{

// Extract face region of interest

var faceRect = new Rectangle(face.X, face.Y, face.Width, face.Height);

var faceImage = grayFrame.Copy(faceRect).Resize(200, 200, Inter.Cubic);

// Recognize face

var result = \_recognizer.Predict(faceImage);

// Display result

var label = result.Label.ToString();

if (checkId(Int32.Parse(label)))

{

//Display worker name if recognized

var worker = workerLabels.FirstOrDefault(w => w.Id == Int32.Parse(label));// take worker id

if (worker != null)

{

Dispatcher.Invoke(() =>

{

//KHi nhan dien ra cong nhan

if (CheckAttandance(Int32.Parse(label),\_faculty,shift))

{

string querry = "INSERT INTO Attendance VALUES (@id\_worker, @date, @fid, @shift)";

if (dataconnecttion.GetConnection().State == System.Data.ConnectionState.Closed)

dataconnecttion.GetConnection().Open();

command = new SqlCommand(querry, dataconnecttion.GetConnection());

//command.Parameters.AddWithValue("@id", id);

command.Parameters.AddWithValue("@id\_worker", Int32.Parse(label));

command.Parameters.AddWithValue("@date", DateTime.Now);

command.Parameters.AddWithValue("@fid", \_faculty);

command.Parameters.AddWithValue("@shift", shift);

command.ExecuteNonQuery();

dataconnecttion.GetConnection().Close();

}

AttendanceWorker temp = new AttendanceWorker();

temp.name = worker.Name;

temp.id = Int32.Parse(label);

temp.date = DateTime.Now;

Attendance.Items.Add(new { id\_worker = worker.Id, Name = worker.Name, TimeIn = DateTime.Now });

workerLabels.Remove(worker);

AttendanceTxt.Text = Attendance.Items.Count.ToString();

});

}

}

// Draw a rectangle around the face

frame.Draw(faceRect, new Bgr(0, 0, 255), 2);

}

// Display the processed frame in the image control

var bitmap = frame.ToBitmap();

var bitmapSource = Imaging.CreateBitmapSourceFromHBitmap(

bitmap.GetHbitmap(), IntPtr.Zero, Int32Rect.Empty, BitmapSizeOptions.FromEmptyOptions());

VideoDisplay.Source = bitmapSource;

}

Đoạn mã trên được sử dụng để nhận diện nhân viên và hiển thị họ lên bảng. Khi quản lý bật vào màn hình thì hệ thống sẽ load dữ liệu của nhân viên bao gồm id và ảnh, sau đó hệ thống sẽ dùng 2 List để lưu trữ 2 mảng giá trị id và ảnh của nhân viên theo 1 cặp trị số giống nhau. Ở đây, ta dùng List để dễ dàng thêm và xóa phần tử cũng như không phải cấp phát bộ nhớ trước khi lưu dữ liệu. Khi hệ thống nhận diện được mặt người sẽ ghi dữ liệu của nhân viên đó vào cơ sở dữ liệu để tránh mất mát dữ liệu và xóa id và ảnh của nhân viên đó ra khỏi 2 List đã tạo để tránh trùng lặp nhân viên khi chấm công. Ngoài ra, để nhận được dữ liệu từ webcam, ta sử dụng lớp VideoCapture để có thể kết nối đến webcam và bật/tắt tùy ý.

**Màn hình Add worker:**

+Giao diện: Nhận các thông tin nhân viền mà người dùng nhập vào gồm: tên, ngày sinh, lương, phòng ban, ảnh. Hiển thị các ô trống để người dùng điền thông tin và nút lưu, chụp ảnh, bật cam. Hiển thị thêm các khung webcam và khung ảnh kết quả.

+ Mã lệnh:

if (iscapturing == true)

{

try

{

// Crop the face image

Rectangle face = faces[0];

Image<Gray, byte> faceImage = image.Convert<Gray, byte>().Copy(face);

string nameimg = FullNametxt.Text;

int idNext = getIdWorker\_Next();

string imagePath = $"{resourceFolderPath}\\WorkerImage\\{Falcutybox.SelectedItem.ToString()}\\{nameimg}{idNext}.png";

if (File.Exists(imagePath))

{

ResultImg.Source = null;

File.Delete(imagePath);

}

Resize(faceImage).Save(imagePath);

BitmapImage bitmap = new BitmapImage();

bitmap.BeginInit();

bitmap.CacheOption = BitmapCacheOption.OnLoad;

bitmap.UriSource = new Uri(imagePath, UriKind.Absolute);

bitmap.EndInit();

ResultImg.Source = bitmap;

MessageBox.Show("Add worker picture done.");

}

catch(Exception ex)

{

MessageBox.Show("Please, try add picture again", "Error add picture", MessageBoxButton.OK);

}

Đoạn mã trên để chụp ảnh nhân viên bằng webcam. Khi người dùng bấm chụp thì hệ thống sẽ nhận diện khuôn mặt của nhân viên và cắt ảnh chỉ còn mặt của nhân viên và hiện thị thông báo khi chụp ảnh thành công. Bắt lỗi khi trong anh chụp không có khuôn mặt nào.

string querry = "Insert into WorkerList Values(@fullname, @birth,@salary ,@images, @fid)";

if (dtc.GetConnection().State == System.Data.ConnectionState.Closed)

dtc.GetConnection().Open();

int id\_worker = getIdWorker\_Next();

try

{

SQLcommand = new SqlCommand(querry, dtc.GetConnection());

SQLcommand.Parameters.Add("@fullname", SqlDbType.NVarChar).Value = fullname;

SQLcommand.Parameters.Add("@birth", SqlDbType.Date).Value = DoB.Date.ToString();

SQLcommand.Parameters.Add("@fid", SqlDbType.NVarChar).Value = faculty;

SQLcommand.Parameters.Add("@images", SqlDbType.NVarChar).Value = fullname+id\_worker;

SQLcommand.Parameters.AddWithValue("@salary", Int32.Parse(Salarytxt.Text));

SQLcommand.ExecuteNonQuery();

MessageBox.Show("Worker added successfully.");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

Đoạn mã trên được dùng để ghi dữ liệu của nhân viên vừa được thêm vào cơ sở dữ liệu và lưu ảnh chụp vào đúng folder của hệ thống. Bắt lỗi liên kết đến cơ sở dữ liệu, sai định dạng dữ liệu.

**Màn hình Edit worker:**

+ Giao diện: Nhận vào thông tin người dùng muốn sửa, thông tin người dùng đã sửa. Hiển thị thông tin người dùng muốn sửa và thông báo khi sửa thành công.

+ Mã lệnh:

FullNametxt.Text = worker.Fullname;

Dobtxt.Text = worker.Birth.ToString();

Facultycbb.SelectedItem = worker.Fid;

Salarytxt.Text = worker.Salary.ToString();

string query = "Select images, id from WorkerList where fullname =@fullname";

if (dtc.GetConnection().State == System.Data.ConnectionState.Closed)

dtc.GetConnection().Open();

try{

cmd = new SqlCommand(query, dtc.GetConnection());

cmd.Parameters.AddWithValue("@fullname", fullname);

SqlDataReader reader = cmd.ExecuteReader();

string imageName = " ";

int idReader = 0;

while (reader.Read())

{

imageName = reader.GetString(0);

idReader = reader.GetInt32(1);

}

reader.Close();

}catch(Exception ex){

MessageBox.Show(ex.Message);

}

id = idReader;

string currentImage = $"{resourceFolderPath}\\WorkerImage\\{\_faculty}\\{imageName}.png";

BitmapImage bitmap = new BitmapImage();

bitmap.BeginInit();

bitmap.CacheOption = BitmapCacheOption.OnLoad;

bitmap.UriSource = new Uri(currentImage, UriKind.Absolute);

bitmap.EndInit();

ResultImg.Source = bitmap;

dtc.GetConnection().Close();

Đoạn mã trên được dùng để hiển thị các thông tin của tài khoản mà người dùng muốn sửa bao gồm cả ảnh hồ sơ. Dùng DataReader để lấy ra từng dữ liệu sau khi thực thi câu lệnh sql. Bắt lỗi kết nối đến cơ sở dữ liệu và lỗi không tìm được thông tin.

string query = "Update WorkerList set fullname = @fullname, Birth = @dob, fid =@fid, images = @image, salary = @salary where id = @id";

if (dtc.GetConnection().State == System.Data.ConnectionState.Closed)

dtc.GetConnection().Open();

cmd = new SqlCommand(query, dtc.GetConnection());

cmd.Parameters.AddWithValue("@fullname", FullNametxt.Text);

cmd.Parameters.AddWithValue("@dob", DoB);

cmd.Parameters.AddWithValue("@fid", Facultycbb.SelectedItem.ToString());

cmd.Parameters.AddWithValue("@image", FullNametxt.Text + id);

cmd.Parameters.AddWithValue("@id", id);

cmd.Parameters.AddWithValue("@salary", Int32.Parse(Salarytxt.Text.ToString()));

cmd.ExecuteNonQuery();

//Image processing

if (File.Exists($"D:\\{FullNametxt.Text}{id}.png") || FullNametxt.Text != fullname)

{

//change everything -> delete old image and replace

//Delete old image

string oldpath = $"{resourceFolderPath}\\WorkerImage\\{\_faculty}\\{fullname}{id}.png";

File.Delete(oldpath);

//Copy new image to directed folder

string targetpath = $"{resourceFolderPath}\\WorkerImage\\{Facultycbb.SelectedItem.ToString()}\\{FullNametxt.Text}{id}.png";

File.Copy($"D:\\{FullNametxt.Text}.png", targetpath);

File.Delete($"D:\\{FullNametxt.Text}{id}.png");

}

else if (FullNametxt.Text == fullname && Facultycbb.SelectedItem.ToString() == \_faculty)

{

//only change dob ->do not thing

}

else

{

//only change falculty-> Move old iamge to new folder

string oldpath = $"{resourceFolderPath}\\WorkerImage\\{\_faculty}\\{fullname}{id}.png";

string targetpath = $"{resourceFolderPath}\\WorkerImage\\{Facultycbb.SelectedItem.ToString()}\\{FullNametxt.Text}{id}.png";

File.Move(oldpath, targetpath);

}

MessageBox.Show("Successful change worker's information");

Đoạn mã trên được sử dụng để lưu thông tin vừa sửa đổi vào cơ sở dữ liệu. Xử lý file ảnh tùy thuộc vào các thay đổi về nhân viên đó. Khi chỉ thay đổi ngày sinh thì không thay đổi đến file ảnh. Khi thay đổi tên nhân viên nên tên ảnh thay đổi, ta xóa ảnh cũ đi và lưu ảnh mới vào đúng thư mục. Còn khi chỉ thay đổi phòng ban thì ta chỉ cần chuyển file ảnh cũ sang đúng folder. Ta sử dụng File.Delete() để xóa file còn File.Copy() để di chuyển file sang folder khác.

**CHƯƠNG 7: KIỂM THỬ**

### 7.1 Test case chấm công nhân viên

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| CCNV\_01 | Chấm công  ca 1  thành công | Actor đăng nhập vào hệ thống, dữ liệu nhân viên đã có trên CSDL | 1.Tại màn hình Attendance, bấm [Start].  2.Khi kết thúc điểm danh bấm [End] | Hệ thống hiển thị danh sách nhân viên có mặt, vắng mặt và số lượng | Hệ thống hiển thị danh sách nhân viên có mặt, vắng mặt và số lượng | Passed |
| CCNV\_02 | Chấm công ca 2 thành công | Actor đăng nhập vào hệ thống, dữ liệu nhân viên đã có trên CSDL | 1.Tại màn hình Attendance, bấm [Switch]  2. Bấm [Start].  3.Khi kết thúc điểm danh bấm [End] | Hệ thống hiển thị danh sách nhân viên có mặt, vắng mặt và số lượng | Hệ thống hiển thị danh sách nhân viên có mặt, vắng mặt và số lượng | Passed |

### 7.2 Test case thêm tài khoản

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| TTK\_01 | Thêm tài khoản thành công | Admin đăng nhập vào hệ thống | 1.Bấm [Account Manage] – [Add]  2.Nhập chính xác dữ liệu vào các trường, chọn ảnh  3.Bấm [Save] | Hệ thống hiển thị thông báo thêm thành công | Hệ thống hiển thị thông báo thêm thành công | Passed |
| TTK\_02 | Mật khẩu xác nhận khác với mật khẩu | Admin đăng nhập vào hệ thống | 1.Tại cửa sổ Add account, nhập mật khẩu xác nhận khác với mật khẩu  2. Bấm [Save] | Hệ thống thông báo sai thông tin nhập vào | Hệ thống thông báo sai thông tin nhập vào | Passed |

### 7.3 Test case sửa tài khoản

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| STK\_01 | Sửa tài khoản thành công | Admin đăng nhập vào hệ thống | 1.Bấm [Account Manage], chọn tài khoản rồi bấm [Edit]  2.Nhập chính xác dữ liệu vào các trường, chọn ảnh  3.Bấm [Save] | Hệ thống hiển thị thông báo sửa thành công | Hệ thống hiển thị thông báo sửa thành công | Passed |

### 7.4 Test case thêm nhân viên

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| TNV\_01 | Thêm nhân viên thành công | Staff đăng nhập vào hệ thống | 1.Bấm [Worker Manage] – [Add]  2.Nhập chính xác dữ liệu vào các trường, chụp ảnh  3.Bấm [Save] | Hệ thống hiển thị thông báo thêm thành công | Hệ thống hiển thị thông báo thêm thành công | Passed |
| TNV\_02 | Sai định dạng ngày tháng năm | Staff đăng nhập vào hệ thống | 1.Tại cửa sổ Add worker, nhập trường ngày sinh sai định dạng dd/mm/yyyy  2. Bấm [Save] | Hệ thống thông báo sai định dạng ngày | Hệ thống thông báo sai định dạng ngày | Passed |

### 7.5 Test case sửa nhân viên

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| SNV\_01 | Sửa nhân viên thành công | Staff đăng nhập vào hệ thống | 1.Bấm [Worker Manage], chọn nhân viên rồi bấm [Edit]  2.Nhập chính xác dữ liệu vào các trường, chụp ảnh  3.Bấm [Save] | Hệ thống hiển thị thông báo sửa thành công | Hệ thống hiển thị thông báo sửa thành công | Passed |

### 7.6 Test case thêm thông tin nhân viên đi muộn

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| NVDM\_01 | Thêm thông tin nhân viên đi muộn thành công | Staff đăng nhập vào hệ thống | 1.Bấm [Late Manage] – [Add]  2.Nhập chính xác dữ liệu vào các trường  3.Bấm [Save] | Hệ thống hiển thị thông báo thêm thành công | Hệ thống hiển thị thông báo thêm thành công | Passed |

### 7.5 Test case sửa thông tin nhân viên đi muộn

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| SNVDM\_01 | Sửa thông tin nhân viên đi muộn thành công | Staff đăng nhập vào hệ thống | 1.Bấm [Late Manage], chọn nhân viên rồi bấm [Edit]  2.Nhập chính xác dữ liệu vào các trường  3.Bấm [Save] | Hệ thống hiển thị thông báo sửa thành công | Hệ thống hiển thị thông báo sửa thành công | Passed |

### 7.5 Test case Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Mô tả TestCase** | | **Các bước thực hiện** | **Kết quả mong đợi** | **Kết quả thực tế** | **Trạng thái** |
| **Tên test case** | **Tiền điều kiện** |
| RP\_01 | Xem báo cáo thành công | Staff đăng nhập vào hệ thống | 1.Bấm [Report]  2.Chọn ngày  3.Bấm [Search] | Hệ thống hiển thị dữ liệu lên bảng | Hệ thống hiển thị dữ liệu lên bảng | Passed |
| RP\_02 | Gửi báo cáo thành công | Đã xuất dữ liệu file báo cáo trước đó | 1.Tại màn hình bất kì có chức năng Send bấm Send  2.Tại màn hình send, nhập thông tin vào các trường  3.bấm [Send] | Hệ thống hiển thị thông báo gửi thành công | Hệ thống hiển thị thông báo gửi thành công | Passed |
| RP\_03 | Xuất báo cáo thành công | Dữ liệu báo cáo được hiển thị trên bảng | 1.Tại màn hình Report, bấm [Export]  2.Tại cửa sổ lưu file, chọn nơi lưu và sửa tên file  3.Bấm [Save] | Hệ thống hiển thị thông báo xuất thành công | Hệ thống hiển thị thông báo xuất thành công | Passed |

**CHƯƠNG 8: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM**

**8.1. Nội dung đã thực hiện**

Trong quá trình thực hiện Bài tập lớn này, chúng em đã tiếp thu và áp dụng một số kiến thức và kỹ năng quan trọng. Đầu tiên, chúng em đã nắm vững kiến thức về công cụ Emgu, OpenCV, lập trình C# trên Visual Studio và SQL Server, cụ thể là cách xây dựng phần mềm chấm công với công cụ EmguCV, OpenCV trên Visual Studio và quản lý cơ sở dữ liệu SQL Server. Chúng em cũng đã rèn kỹ năng lập trình, sử dụng công cụ và phát triển phần mềm, từ việc thiết kế cấu trúc dữ liệu cho đến việc xử lý logic và tạo giao diện người dùng.

Nhờ việc thực hiện Bài tập lớn này, chúng em đã đạt được các chuẩn đầu ra của học phần. Chúng em đã xây dựng thành công một phần mềm chấm công tự động hoàn chỉnh, với các chức năng quan trọng như nhận diện gương mặt, điểm danh, quản lý tài khoản nhân viên.

Sau quá trình thực hiện Bài tập lớn này, chúng em rút ra được một số bài học kinh nghiệm quan trọng:

* Đầu tiên, việc định rõ yêu cầu và thiết kế cấu trúc dữ liệu từ đầu là rất quan trọng để tránh sự rối loạn và trục trặc trong quá trình phát triển.
* Thứ hai, việc sử dụng công cụ hỗ trợ viết mã như Visual Studio và ứng dụng thư viện Emgu, OpenCV giúp tăng hiệu suất và tiết kiệm thời gian trong quá trình phát triển phần mềm.

**8.2. Hướng phát triển**

Chúng em đề xuất có thể tiếp tục nâng cấp và mở rộng chức năng của phần mềm chấm công tự động. Ví dụ, chúng em có thể cải thiện tính năng nhận diện gương mặt để nhân viên có thể được hệ thống nhận diện chính xác và nhanh chóng hơn. Đồng thời, chúng em cũng có thể tăng cường tính bảo mật và hiệu suất của hệ thống.

Cuối cùng, chúng em muốn đề xuất trao đổi và thảo luận với giảng viên giảng dạy sau khi hoàn thành Bài tập lớn này. Điều này có thể bao gồm xác định tính khả thi của chủ đề nghiên cứu, đánh giá tính phù hợp và khả năng thực hiện dự án với kiến thức, kỹ năng và kinh nghiệm hiện có. Chúng em cũng mong muốn được chia sẻ các thuận lợi và khó khăn mà chúng em đã gặp phải trong quá trình nghiên cứu và phát triển dự án này.

Nhìn chung, việc thực hiện Bài tập lớn này đã giúp chúng em áp dụng và mở rộng kiến thức và kỹ năng đã học được trong lĩnh vực phát triển phần mềm. Chúng em tin rằng kinh nghiệm từ quá trình này sẽ giúp chúng em trở thành những nhà phát triển chuyên nghiệp và đóng góp vào sự phát triển của ngành công nghệ thông tin.

**TÀI LIỆU THAM KHẢO**

|  |  |  |
| --- | --- | --- |
| **Tên nguồn** | **Tác giả** | **Nguồn** |
| HAAR CASCADE LÀ GÌ? | NguyenDuong (người đăng bài) | https://onetech.vn/blog/haar-cascade-la-gi-13561 |
| Lập trình Windows với C#.NET | Phương Lan, Hoàng Đức Hải | http://lib.haui.edu.vn/Opac80/EDetail.aspx?id=371&f=nhaxuatban&v=Lao+%u0111%u1ed9ng+-+X%u00e3+h%u1ed9i |
| Cascade Classifier | OpenCV | https://docs.opencv.org/3.4/db/d28/tutorial\_cascade\_classifier.html |
| Integral Image | mathworks.com | https://www.mathworks.com/help/images/integral-image.html |
| C# Tutorial - Webcam Face Detection for .NET using EMGU.CV in C# | FoxLearn | Fox Learn | https://www.youtube.com/watch?v=FvUdVYEhvHo |

Bảng Tài liệu tham khảo