

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**XÂY DỰNG HỆ THỐNG TỰ ĐỘNG HOÁ
QUY TRÌNH TẢI XUỐNG VÀ PHÂN PHỐI
HÓA ĐƠN ĐIỆN TỬ QUA EMAIL CHO
NGƯỜI DÂN**

Người hướng dẫn: **TS. Nguyễn Hà Huy Cường**
Sinh viên thực hiện: **Nguyễn Văn Đông**
Mã số sinh viên: **2151220287**
Lớp: **21CT6**

Đà Nẵng, 12./2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**TÊN ĐỀ TÀI : XÂY DỰNG HỆ THỐNG TỰ
ĐỘNG HOÁ QUY TRÌNH TẢI XUỐNG VÀ
PHÂN PHỐI HÓA ĐƠN ĐIỆN TỬ QUA
EMAIL CHO NGƯỜI DÂN**

Người hướng dẫn: TS.Nguyễn Hà Huy Cường

Sinh viên thực hiện: Nguyễn Văn Đông

Mã số sinh viên: 2151220287

Lớp: 21CT6

Đà Nẵng, 12./20

TÓM TẮT

Tên đề tài: Xây dựng hệ thống tự động hóa quy trình tải xuống và phân phối hóa đơn điện tử qua email cho người dân

Sinh viên thực hiện: Nguyễn Văn Đông Lớp: 21CT

Đề tài tập trung vào việc áp dụng các công nghệ lập trình web hiện đại, bao gồm NodeJs và các công cụ hỗ trợ tự động hóa, để xây dựng một hệ thống trực tuyến toàn diện, giúp tự động tải xuống và phân phối hóa đơn điện tử đến người dân thông qua email. Hệ thống được thiết kế tối ưu cho giao diện người dùng thân thiện, đồng thời đảm bảo vận hành ổn định và chính xác trong thời gian thực. Các biện pháp bảo mật dữ liệu nhạy cảm và quản lý thông tin hóa đơn được chú trọng nhằm đảm bảo tính bảo mật và độ tin cậy cao. Ngoài ra, nghiên cứu cũng đề xuất các phương pháp mở rộng quy mô và tích hợp các tiện ích báo cáo, theo dõi trạng thái gửi email và xử lý lỗi, nhằm nâng cao hiệu quả vận hành trong dài hạn.

Hệ thống tự động hóa hóa đơn điện tử sẽ giúp tiết kiệm thời gian, giảm thiểu sai sót trong quy trình gửi hóa đơn, minh bạch hóa thông tin tài chính và cải thiện trải nghiệm của người dân trong việc nhận hóa đơn điện tử một cách nhanh chóng, chính xác và tiện lợi.

LỜI NÓI ĐẦU

Để hoàn thành đồ án tốt nghiệp này, trước hết em xin gửi lời cảm ơn chân thành đến quý Thầy Cô trong khoa Công Nghệ Thông Tin - Trường Đại Học Kiến Trúc Đà Nẵng, những người đã tận tình truyền đạt kiến thức và kinh nghiệm quý báu cho em trong suốt quá trình học tập. Những kiến thức và kỹ năng được tiếp thu không chỉ là nền tảng vững chắc để em thực hiện đồ án, mà còn là hành trang quý giá giúp em tự tin bước vào con đường sự nghiệp sau này.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc tới thầy **TS. Nguyễn Hà Huy Cường**, người đã luôn quan tâm, hướng dẫn, định hướng nghiên cứu và tạo điều kiện thuận lợi để em giải quyết các vấn đề gặp phải trong quá trình thực hiện đồ án.

Mặc dù đã nỗ lực hết mình với tinh thần nghiêm túc, do giới hạn về thời gian, trình độ lý luận và kinh nghiệm thực tiễn còn hạn chế, nên bài báo cáo khó tránh khỏi những thiếu sót. Em rất mong nhận được sự cảm thông và những ý kiến đóng góp quý báu từ quý Thầy Cô để em có thể học hỏi thêm, hoàn thiện kiến thức và tích lũy kinh nghiệm thực tiễn bổ ích.

Em xin trân trọng cảm ơn!

CAM ĐOAN

Em xin cam đoan kết quả đạt được trong đồ án tốt nghiệp này là sản phẩm nghiên cứu của riêng cá nhân em, được thực hiện dưới sự hướng dẫn khoa học tận tình của **TS. Nguyễn Hà Huy Cường**.

Trong toàn bộ nội dung của đồ án, những vấn đề được trình bày là những đúc kết từ quá trình tìm hiểu của bản thân hoặc được tổng hợp có chọn lọc từ nhiều nguồn tài liệu uy tín. Em cam đoan không sao chép trái phép công trình của người khác. Tất cả các tài liệu tham khảo được sử dụng đều có xuất xứ rõ ràng và được trích dẫn hợp pháp theo đúng quy định.

Em xin hoàn toàn chịu trách nhiệm và chấp nhận mọi hình thức kỷ luật của Nhà trường nếu phát hiện có bất kỳ sự gian dối nào trong lời cam đoan này.

Sinh viên thực hiện

Nguyễn Văn Đông

Mục Lục

TÓM TẮT	i
LỜI NÓI ĐẦU	ii
CAM ĐOAN	iii
Mục Lục	iv
Mục lục hình ảnh	v
MỞ ĐẦU	1
Giới thiệu tổng quan	2
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN	9
1.1. Hóa đơn điện tử và quy định pháp lý tại Việt Nam	9
1.2. Công nghệ API và RPA trong tự động hóa nghiệp vụ	11
1.2.1. API (Application Programming Interface)	11
1.2.2. RPA (Robotic Process Automation)	12
1.2.3. Bảo mật dữ liệu trong truyền tải email	14
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	17
2.1. Khảo sát yêu cầu thực tế tại doanh nghiệp	17
2.2. Thiết kế kiến trúc hệ thống tự động tải – gửi hóa đơn	20
2.2.1. Danh sách bảng và quan hệ	20
2. 2.2. Sơ đồ quan hệ ERD mô tả	22
CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM	23
3.1. Công cụ và môi trường phát triển	23
3.2. Xây dựng module tải hóa đơn tự động	26
3.3. Xây dựng Web gửi email	30
3.4. Thử nghiệm và đánh giá hệ thống	41
3.4.1. Thử nghiệm chức năng	41
3.4.2. Đánh giá hiệu năng	43
KẾT LUẬN	45
1. Tóm tắt các kết quả đạt được	45
2. Hạn chế của hệ thống	47

Mục lục hình ảnh

Hình 2.1 Sơ đồ ERD	22
Hình 3.1 Code Cấu hình Chrome	26
Hình 3.2 Code Tìm và tải hóa đơn FPT	26
Hình 3.3 Code Tìm và tải hóa đơn van	27
Hình 3.4 Code Tìm và tải hóa đơn misa	28
Hình 3.5 Tự động điền và đăng nhập vào FPT	28
Hình 3.6 Tự động tải hóa đơn FPT	28
Hình 3.7 Tự động điền và đăng nhập vào Misa	29
Hình 3.8 tự động tải hóa đơn Misa	29
Hình 3.9 Tự động điền và đăng nhập vào BKAV	30
Hình 3.10 tự động tải hóa đơn BKAV	30
Hình 3.11 code cập nhật danh sách khách hàng	31
Hình 3.12 code gửi mail cho khách hàng	32
Hình 3.13 Code thống kê số liệu	38
Hình 3.14 Hình trang chủ	39
Hình 3.15 Hình thông tin khách hàng	39
Hình 3.16 Hình hóa đơn	40
Hình 3.17 Hình gửi mail	40
Hình 3.18 Hình số liệu và thống kê	41

MỞ ĐẦU

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại Việt Nam, việc ứng dụng công nghệ thông tin vào quản lý và cung cấp các dịch vụ hành chính, tài chính đã trở thành xu hướng tất yếu. Một trong những nội dung quan trọng của quá trình này là việc triển khai và sử dụng hóa đơn điện tử theo quy định của Chính phủ, nhằm thay thế dần hóa đơn giấy truyền thống, góp phần nâng cao tính minh bạch, giảm chi phí và tăng hiệu quả quản lý.

Theo Nghị định số 123/2020/NĐ-CP và Thông tư số 78/2021/TT-BTC của Bộ Tài chính, hóa đơn điện tử đã trở thành hình thức bắt buộc đối với hầu hết các doanh nghiệp, tổ chức và hộ kinh doanh. Cùng với đó, nhu cầu cung cấp hóa đơn điện tử cho người dân một cách nhanh chóng, chính xác và thuận tiện ngày càng gia tăng. Tuy nhiên, trên thực tế, tại nhiều đơn vị cung cấp dịch vụ công và doanh nghiệp, quy trình tải xuống và phân phối hóa đơn điện tử vẫn còn được thực hiện thủ công, phụ thuộc nhiều vào thao tác của con người.

Quy trình thủ công này thường bao gồm việc đăng nhập vào các cổng hóa đơn điện tử, tra cứu từng mã khách hàng, tải file hóa đơn và gửi email riêng lẻ cho từng người dân. Cách làm này không chỉ tiêu tốn nhiều thời gian và nhân lực, mà còn tiềm ẩn nguy cơ sai sót như gửi nhầm hóa đơn, thiếu file đính kèm hoặc không kiểm soát được lịch sử gửi. Đặc biệt, với những đơn vị phải xử lý hàng nghìn hóa đơn mỗi tháng, việc vận hành thủ công gây ra áp lực lớn và làm giảm hiệu quả công việc.

Trước những hạn chế đó, việc xây dựng một hệ thống tự động hóa toàn bộ quy trình tải xuống và phân phối hóa đơn điện tử là yêu cầu cấp thiết. Ứng dụng các công nghệ lập trình hiện đại, kết hợp giữa truy xuất dữ liệu thông qua API và tự động hóa thao tác bằng các công cụ RPA, cho phép hệ thống thực hiện chính xác các công việc lặp đi lặp lại, giảm thiểu sự can thiệp của con người và nâng cao độ tin cậy của quy trình. Bên cạnh đó, việc tích hợp chức năng gửi email tự động và ghi nhận nhật ký gửi giúp đơn vị quản lý dễ dàng theo dõi, kiểm tra và xử lý sự cố khi cần thiết.

Xuất phát từ thực tiễn trên, đề án tốt nghiệp **“Xây dựng hệ thống tự động hoá quy trình tải xuống và phân phối hóa đơn điện tử qua email cho người dân”** được thực hiện với mục tiêu nghiên cứu, thiết kế và xây dựng một hệ thống hỗ trợ tự động hóa toàn diện quy trình này. Hệ thống hướng đến việc giảm tải công việc cho nhân sự, nâng cao độ chính xác trong quá trình gửi hóa đơn, đảm bảo tuân thủ các quy định pháp lý hiện hành và góp phần cải thiện chất lượng dịch vụ cung cấp hóa đơn điện tử cho người dân.

Giới thiệu tổng quan

1. Giới thiệu đề tài

Trong bối cảnh chuyển đổi số mạnh mẽ tại Việt Nam, hóa đơn điện tử đang trở thành hình thức bắt buộc đối với doanh nghiệp, tổ chức và hộ kinh doanh theo Nghị định 123/2020/NĐ-CP và Thông tư 78/2021/TT-BTC. Tuy nhiên, quy trình cung cấp hóa đơn điện tử cho người dân và khách hàng vẫn còn nhiều địa phương, đơn vị vận hành thủ công: nhân viên phải đăng nhập vào hệ thống cấp hóa đơn, tải từng file PDF và gửi thủ công qua email.

Điều này gây ra:

Tốn thời gian và nhân lực.

Dễ xảy ra sai sót khi gửi nhầm hóa đơn.

Khó kiểm tra, giám sát lịch sử gửi hóa đơn.

Gây chậm trễ trong việc cung cấp dịch vụ công cho người dân.

Đề tài “Xây dựng hệ thống tự động hóa quy trình tải xuống và phân phối hóa đơn điện tử qua email cho người dân” nhằm giải quyết bài toán này bằng cách tự động hóa toàn bộ quy trình từ thu thập dữ liệu, tải hóa đơn đến gửi email và ghi nhận lịch sử gửi.

1. Tình hình nghiên cứu trong và ngoài nước

a. Nghiên cứu trong nước

Tại Việt Nam, đã có nhiều doanh nghiệp cung cấp giải pháp hóa đơn điện tử như:

VNPT e-Invoice

Viettel Invoice

MISA meInvoice

BKAV eHoadon

Tuy nhiên, phần lớn các giải pháp chỉ tập trung vào việc phát hành hóa đơn điện tử, chưa chú trọng vào tự động gửi hóa đơn hàng loạt cho từng cá nhân theo nhiều hồ sơ/phòng ban khác nhau. Một số hệ thống có API để truy xuất hóa đơn nhưng yêu cầu doanh nghiệp phải tự lập trình tích hợp.

Ngoài ra, hiện chưa có nhiều nghiên cứu chính thức về mô hình ứng dụng RPA (Robotic Process Automation) để tự động tải hóa đơn từ các cổng dịch vụ công, đặc biệt là các hệ thống không hỗ trợ API.

b. Nghiên cứu ngoài nước

Trên thế giới, tự động hóa nghiệp vụ (Business Automation) và RPA được ứng dụng rộng rãi, tiêu biểu như:

UiPath, BluePrism, Automation Anywhere: hỗ trợ tự động hóa quy trình nội bộ, tải tài liệu và gửi email.

Các nước phát triển đã tích hợp hoá đơn điện tử trực tiếp qua API với hệ thống thuế.

Tuy nhiên, mô hình tải hóa đơn từ nhiều nền tảng khác nhau và phân phối tự động cho hàng nghìn cá nhân vẫn là hướng nghiên cứu mở, đặc biệt trong lĩnh vực dịch vụ công.

2. Mục tiêu, đối tượng, phạm vi nghiên cứu

Mục tiêu :

Xây dựng hệ thống tự động hóa 100% quy trình tải hóa đơn điện tử.

Tự động phân loại, lưu trữ hóa đơn theo tháng/năm/mã khách hàng.

Gửi hóa đơn chính xác đến từng email của người dân.

Hỗ trợ theo dõi log gửi hóa đơn, gửi lại khi lỗi.

Tạo giao diện web trực quan để quản trị.

Đối tượng nghiên cứu :

Đối tượng nghiên cứu của đề tài tập trung vào các hệ thống và quy trình liên quan đến việc phát hành, quản lý và phân phối hóa đơn điện tử tại các cơ quan, đơn vị cung cấp dịch vụ công và doanh nghiệp. Cụ thể bao gồm:

1. Các cổng hóa đơn điện tử của các nhà cung cấp dịch vụ thiết yếu

Đề tài tiến hành khảo sát và phân tích cách thức hoạt động của các cổng hóa đơn điện tử được vận hành bởi các đơn vị như:

Ngành điện lực (EVN): Hệ thống tra cứu hóa đơn điện tử theo mã khách hàng, kỳ hóa đơn, hoặc theo số hợp đồng.

Các nhà mạng viễn thông (Viettel, VNPT, FPT, MobiFone): Cổng xuất hóa đơn cho dịch vụ internet, di động, truyền hình.

Công ty cấp nước đô thị: Hệ thống truy xuất hóa đơn tiền nước định kỳ hàng tháng.

Các đơn vị dịch vụ công khác: chẳng hạn phí môi trường, phí vệ sinh, dịch vụ công ích.

Đặc điểm chung của các cổng này là đều cung cấp API, web portal hoặc giao diện người dùng cho phép tải hóa đơn định dạng PDF/XML. Tuy nhiên, mỗi đơn

vị lại có chuẩn dữ liệu, cơ chế đăng nhập và phương thức bảo mật khác nhau. Việc nghiên cứu các hệ thống này giúp đề tài xác định được:

Khả năng tự động hoá hoạt động tải hóa đơn.

Các ràng buộc bảo mật (OTP, Captcha, ký số...).

Tính ổn định, tốc độ truy vấn và tần suất phát hành hóa đơn.

2. Quy trình xử lý hóa đơn tại đơn vị hành chính – sự nghiệp

Bên cạnh việc khảo sát công hóa đơn, đề tài còn nghiên cứu quy trình thực tế tại các đơn vị hành chính – sự nghiệp, nơi có nhu cầu gửi hóa đơn cho người dân theo định kỳ. Quy trình này thường bao gồm:

Thu thập danh sách khách hàng cần gửi hóa đơn.

Tải hóa đơn từ hệ thống nghiệp vụ nội bộ hoặc từ cổng hóa đơn điện tử của nhà cung cấp.

Kiểm tra, đối chiếu dữ liệu trước khi phân phối.

Gửi hóa đơn qua email thủ công hoặc bán tự động.

Lưu lại nhật ký gửi, xử lý lỗi và thực hiện gửi lại nếu cần.

Qua khảo sát cho thấy nhiều đơn vị vẫn thực hiện các thao tác trên một cách thủ công, dẫn đến:

Mất nhiều thời gian vào việc truy xuất và tải từng hóa đơn.

Tỷ lệ sai sót cao khi gửi email (nhập nhầm địa chỉ, thiếu file đính kèm...).

Không có công cụ giám sát nhật ký gửi, gây khó khăn trong việc kiểm tra khi người dân phản ánh.

Không có hệ thống tập trung để lưu trữ lịch sử gửi hóa đơn.

Do đó, việc nghiên cứu đối tượng này là cơ sở quan trọng để xây dựng hệ thống tự động hoá tối ưu hơn, giúp giảm tải công việc, tăng độ chính xác và nâng cao trải nghiệm của người dân khi nhận hóa đơn điện tử.

Phạm vi nghiên cứu :

Phạm vi nghiên cứu của đề tài được xác định rõ ràng nhằm đảm bảo tính khả thi, phù hợp với thời gian thực hiện và đáp ứng đúng mục tiêu đặt ra. Cụ thể, đề tài tập trung vào các nội dung sau:

1. Tự động hóa quy trình tải và gửi hóa đơn điện tử

Đề tài chỉ tập trung vào việc xây dựng hệ thống tự động thu thập, tải xuống và phân phối hóa đơn điện tử cho người dân. Các chức năng được đưa vào phạm vi nghiên cứu bao gồm:

Tự động đăng nhập vào cổng hóa đơn điện tử hoặc hệ thống của nhà cung cấp dịch vụ.

Thu thập danh sách hóa đơn theo kỳ, theo mã khách hàng hoặc theo số hợp đồng.

Tải hóa đơn dưới dạng PDF/XML và lưu trữ có tổ chức.

Gửi hóa đơn điện tử đến email của khách hàng thông qua server SMTP hoặc cổng gửi email của đơn vị.

Ghi nhận nhật ký gửi (send log), trạng thái, lỗi... để phục vụ kiểm tra và gửi lại khi cần thiết.

Như vậy, trọng tâm của nghiên cứu là vào tự động hóa thao tác thay vì xử lý nghiệp vụ phát hành hóa đơn.

2 . Không nghiên cứu phát hành hóa đơn điện tử, ký số hay lưu trữ đặc thù theo quy định thuế

Đề tài không bao gồm các nội dung vượt ngoài phạm vi nhiệm vụ, chẳng hạn:

Quy trình lập hóa đơn gốc.

Các nghiệp vụ liên quan đến ký số, đóng dấu, tạo mã của cơ quan thuế, hủy – điều chỉnh hóa đơn, hoặc kết nối với hệ thống thuế tổng cục (ETAX).

Chuẩn hóa dữ liệu đầu ra theo Thông tư 78/2021/TT-BTC hay Nghị định 123/2020/NĐ-CP.

Các phần này thuộc về hệ thống nghiệp vụ của nhà cung cấp hóa đơn, không phải mục tiêu nghiên cứu của đề tài.

3. Chỉ hỗ trợ gửi hóa đơn qua email – không tích hợp SMS, Zalo OA hoặc các kênh truyền thông khác

Hệ thống được thiết kế chỉ tập trung vào kênh email do các ưu điểm:

Chi phí thấp, không phụ thuộc nhà mạng.

Dễ gửi kèm file PDF/XML.

Phù hợp cho nhu cầu của phần lớn đơn vị hành chính – sự nghiệp.

Các kênh khác như SMS brandname, Zalo Official Account, Mobile App Notification... không nằm trong phạm vi hiện tại vì:

Yêu cầu chi phí duy trì riêng.

Phải ký hợp đồng dịch vụ với bên thứ ba.

Cần cơ chế đính kèm hoặc liên kết lưu trữ hóa đơn phức tạp hơn.

Tuy nhiên, hệ thống vẫn được thiết kế theo hướng mở rộng trong tương lai, cho phép bổ sung các kênh gửi mới khi đơn vị có nhu cầu.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ LIÊN QUAN

1.1. Hóa đơn điện tử và quy định pháp lý tại Việt Nam

Hóa đơn điện tử (HĐĐT) là loại hóa đơn được thể hiện dưới dạng dữ liệu điện tử, được tạo lập, lập, gửi, nhận, lưu trữ và quản lý bằng phương tiện điện tử. Tại Việt Nam, việc triển khai hóa đơn điện tử được điều chỉnh bởi hai văn bản quan trọng: **Nghị định 123/2020/NĐ-CP** và **Thông tư 78/2021/TT-BTC** của Bộ Tài chính. Đây là các khung pháp lý chính thức quy định việc đăng ký, sử dụng, quản lý và lưu trữ hóa đơn điện tử đối với doanh nghiệp, tổ chức kinh tế, hộ kinh doanh và cá nhân.

Theo quy định tại văn bản trên, hóa đơn điện tử được chia thành hai nhóm chính:

Hóa đơn điện tử có mã của cơ quan thuế, bắt buộc đối với đa số loại hình doanh nghiệp và trường hợp có rủi ro về thuế; cơ quan thuế sẽ cấp mã xác thực cho từng hóa đơn trước khi gửi cho người mua.

Hóa đơn điện tử không có mã của cơ quan thuế, áp dụng đối với một số đối tượng đủ điều kiện theo quy định (ví dụ: ngân hàng, doanh nghiệp quy mô lớn, ngành đặc thù...).

Về mặt kỹ thuật, hóa đơn điện tử được cung cấp dưới hai dạng phổ biến

Tệp XML – là dữ liệu gốc của hóa đơn, chứa toàn bộ thông tin nghiệp vụ, thông tin người bán, người mua, sản phẩm, ký số... Đây là tệp có giá trị pháp lý đầy đủ.

Tệp PDF – là bản thể hiện của hóa đơn nhằm phục vụ mục đích xem trực quan, in ấn hoặc gửi cho khách hàng.

Theo quy định pháp luật, các cơ quan, tổ chức và doanh nghiệp sử dụng hóa đơn điện tử phải đảm bảo:

Lưu trữ hóa đơn tối thiểu 10 năm theo đúng quy định của Luật Kế toán và các chuẩn mực về lưu trữ dữ liệu điện tử. Việc lưu trữ phải đảm bảo nguyên tắc toàn vẹn dữ liệu, không bị sửa đổi, mất mát hoặc hư hại.

Cung cấp hóa đơn khi khách hàng yêu cầu, kể cả trường hợp khách hàng bị thất lạc hoặc không nhận được hóa đơn. Doanh nghiệp phải đảm bảo khả năng truy xuất và cung cấp lại hóa đơn một cách nhanh chóng và chính xác.

Bảo mật dữ liệu hóa đơn theo Luật An toàn thông tin mạng, bao gồm:

Bảo vệ thông tin người mua và dữ liệu giao dịch.

Sử dụng các biện pháp mã hóa, xác thực và kiểm soát truy cập.

Đảm bảo an toàn khi truyền tải hóa đơn qua email, mạng nội bộ hoặc Internet.

Ngoài ra, các doanh nghiệp còn phải tuân thủ các tiêu chuẩn kỹ thuật khác như:

Sử dụng chữ ký số hợp lệ theo quy định của Bộ Thông tin và Truyền thông.

Kết nối với hệ thống của Tổng cục Thuế thông qua API hoặc cổng truyền nhận dữ liệu.

Thực hiện báo cáo tình hình sử dụng hóa đơn đúng thời hạn theo từng giai đoạn.

Những quy định pháp lý này tạo nền tảng quan trọng cho việc xây dựng hệ thống tự động hóa trong đề tài, đảm bảo rằng mọi hoạt động tải xuống, xử lý và gửi hóa

đơn điện tử đều tuân thủ pháp luật và bảo đảm tính hợp lệ của dữ liệu.

1.2. Công nghệ API và RPA trong tự động hóa nghiệp vụ

Trong bối cảnh số hóa mạnh mẽ hiện nay, việc truy xuất và xử lý dữ liệu từ các hệ thống dịch vụ như hóa đơn điện tử đòi hỏi phải có các công nghệ hiệu quả, linh hoạt và dễ tích hợp. Hai công nghệ quan trọng thường được sử dụng trong các giải pháp tự động hóa nghiệp vụ là **API (Application Programming Interface)** và **RPA (Robotic Process Automation)**. Mỗi công nghệ có ưu – nhược điểm riêng và phù hợp với từng mô hình hệ thống khác nhau.

1.2.1. API (*Application Programming Interface*)

API là tập hợp các giao diện và quy ước cho phép hai hệ thống phần mềm giao tiếp với nhau một cách tự động. Trong lĩnh vực hóa đơn điện tử, nhiều nhà cung cấp dịch vụ triển khai **API RESTful**, cho phép truy vấn hóa đơn, tải file PDF/XML, kiểm tra trạng thái và đồng bộ dữ liệu giữa hệ thống doanh nghiệp và công hóa đơn.

Ứng dụng trong bài toán tải hóa đơn

API giúp hệ thống có thể:

Truy cập vào cơ sở dữ liệu hóa đơn theo kỳ, theo mã khách hàng hoặc theo số hợp đồng.

Tải hóa đơn dưới dạng file PDF hoặc XML một cách tự động.

Lấy thông tin chi tiết như ngày lập, số tiền, mã số thuế, tình trạng hóa đơn.

Tự động hóa toàn bộ quá trình mà không cần thao tác thủ công.

Ưu điểm của API

Tốc độ cao: Giao tiếp trực tiếp giữa máy – máy, không phụ thuộc giao diện web.

Độ ổn định tốt: Hạn chế lỗi so với thao tác trên trình duyệt.

Bảo mật cao: Hỗ trợ token, OAuth2, TLS, chữ ký số...

Khả năng mở rộng: Dễ tích hợp vào hệ thống ERP, CRM, phần mềm kế toán.

Nhược điểm của AP

Không phải tất cả nhà cung cấp đều **mở API**, hoặc chỉ cung cấp cho khách hàng doanh nghiệp lớn.

Một số API áp dụng hạn chế truy cập, giới hạn tần suất hoặc yêu cầu ký thỏa thuận bảo mật.

Cần lập trình viên có kỹ năng để triển khai, bảo trì và xử lý lỗi.

Do vậy, API là giải pháp tối ưu nhất nhưng chỉ khả thi khi đơn vị phát hành hóa đơn cung cấp đầy đủ giao diện truy xuất.

1.2.2. RPA (Robotic Process Automation)

RPA là công nghệ sử dụng “robot phần mềm” để mô phỏng chính xác thao tác của con người trên máy tính. RPA hoạt động dựa trên giao diện người dùng (UI), nghĩa là nó thực hiện các hành động giống như người dùng thật: mở trình duyệt, gõ dữ liệu, bấm nút, tải file...

Trong trường hợp các nhà cung cấp hóa đơn điện tử **không hỗ trợ API**, RPA trở thành lựa chọn hiệu quả để tự động hóa quy trình.

Quy trình RPA trong tải hóa đơn điện tử

Một robot RPA có thể thực hiện các bước:

Mở trình duyệt (Chrome/Edge).

Truy cập trang hóa đơn điện tử của nhà cung cấp.

Đăng nhập bằng tài khoản được cấp.

Nhập mã khách hàng hoặc số hợp đồng.

Tìm kiếm hóa đơn tương ứng.

Nhấn nút tải PDF hoặc XML.

Lưu file hóa đơn về thư mục đã cấu hình.

Ưu điểm của RPA

Phù hợp với hệ thống không có API, đặc biệt là các công hóa đơn cũ.

Không yêu cầu chỉnh sửa hệ thống gốc, chỉ cần tài khoản đăng nhập.

Dễ triển khai, chi phí thấp hơn so với xây dựng API thay thế.

Hỗ trợ thao tác phức tạp (captcha đơn giản, chuyển trang, chọn dropdown...).

Nhược điểm của RPA

Phụ thuộc giao diện – nếu nhà cung cấp thay đổi layout web, RPA phải cập nhật lại.

Tốc độ chậm hơn API do phải mô phỏng thao tác con người.

Dễ gặp lỗi khi mạng chậm, website treo hoặc mất phản hồi.

Vì vậy, trong những hệ thống có API chính thức, API luôn là lựa chọn ưu tiên; RPA chỉ được sử dụng như giải pháp bổ sung trong trường hợp không có API hoặc API không đầy đủ chức năng.

1.2.3. Bảo mật dữ liệu trong truyền tải email

Trong hệ thống tự động gửi hóa đơn điện tử cho người dân, email đóng vai trò là kênh truyền tải thông tin quan trọng. Hóa đơn điện tử chứa thông tin nhạy cảm như mã khách hàng, địa chỉ, số tiền phải thanh toán... nên yêu cầu bảo mật dữ liệu là yếu tố bắt buộc. Việc bảo vệ dữ liệu trong quá trình gửi và nhận email phải tuân thủ cả chuẩn kỹ thuật quốc tế lẫn quy định pháp luật của Việt Nam.

1. Sử dụng giao thức mã hóa TLS/SSL trong truyền tải

Hệ thống email hiện đại bắt buộc triển khai **TLS (Transport Layer Security)** hoặc **SSL** nhằm mã hóa dữ liệu trong quá trình truyền dẫn giữa máy chủ SMTP và các máy chủ trung gian. Khi TLS được bật, toàn bộ nội dung email – kể cả file đính kèm như hóa đơn PDF/XML – đều được **mã hóa**, tránh các nguy cơ bị đọc trộm (sniffing) hoặc sửa dữ liệu trong lúc truyền tải.

Việc áp dụng TLS giúp:

Ngăn chặn tấn công nghe lén (eavesdropping).

Đảm bảo rằng email chỉ được đọc bởi máy chủ đích.

Xác thực danh tính giữa các máy chủ mail khi trao đổi dữ liệu.

2. Sử dụng giao thức SMTP có mã hóa

Trong quá trình gửi email, hệ thống sử dụng **SMTP qua SSL/TLS** như:

SMTPS (SMTP Secure) qua cổng 465.

SMTP với STARTTLS qua cổng 587.

Việc mã hóa kết nối SMTP là yêu cầu quan trọng để:

Giữ an toàn thông tin hóa đơn khi truyền tải.

Ngăn chặn tấn công giả mạo máy chủ gửi email.

Đảm bảo kết nối ổn định và hợp lệ với nhà cung cấp email (Google, Microsoft, Zimbra...).

3. Không lưu mật khẩu dưới dạng plain text

Trong hệ thống tự động hoá, việc lưu trữ mật khẩu không an toàn là một rủi ro lớn. Mật khẩu truy cập email hoặc hệ thống tải hóa đơn phải được bảo vệ bằng các biện pháp như:

Mã hóa mật khẩu trong file cấu hình.

Sử dụng **biến môi trường** (environment variables) thay vì ghi trực tiếp vào mã nguồn.

Áp dụng các giao thức xác thực mạnh như OAuth2 khi gửi email (nếu nhà cung cấp hỗ trợ).

Giới hạn quyền truy cập vào server chứa file cấu hình.

Điều này giúp giảm thiểu nguy cơ lộ thông tin đăng nhập, tránh việc kẻ tấn công chiếm quyền gửi email giả mạo hoặc truy cập vào dữ liệu hóa đơn của người dân.

4. Bảo vệ dữ liệu cá nhân theo Nghị định 13/2023/NĐ-CP của Việt Nam

Nghị định 13/2023 về bảo vệ dữ liệu cá nhân quy định rõ trách nhiệm của đơn vị triển khai hệ thống xử lý dữ liệu nhạy cảm. Trong đó, hóa đơn điện tử được xem

là dữ liệu cá nhân quan trọng vì chứa thông tin định danh khách hàng. Hệ thống phải tuân thủ:

Chỉ thu thập và xử lý dữ liệu đúng mục đích.

Không chia sẻ thông tin khách hàng cho bên thứ ba nếu không có sự đồng ý.

Lưu trữ dữ liệu an toàn, tuân thủ nguyên tắc tối thiểu hóa (chỉ lưu những gì cần thiết).

Áp dụng biện pháp kỹ thuật để tránh rò rỉ dữ liệu.

Đơn vị vận hành cũng phải có quy trình ứng phó sự cố nếu xảy ra rò rỉ email, thất lạc hóa đơn hoặc tấn công mạng.

5. Kiểm soát truy cập và phân quyền người dùng

Một biện pháp quan trọng trong bảo mật là quản lý chặt chẽ quyền truy cập hệ thống:

Chỉ những nhân sự được phân quyền mới được xem hoặc tải hóa đơn.

Áp dụng **cơ chế phân quyền theo vai trò (RBAC)** cho từng chức năng như tải hóa đơn, gửi email, xem nhật ký gửi...

Theo dõi nhật ký truy cập (audit log) để phát hiện hành vi bất thường.

Khóa tài khoản khi phát hiện nghi ngờ truy cập trái phép.

Những biện pháp này giúp giảm thiểu nguy cơ nội gián hoặc truy cập trái phép làm lộ thông tin hóa đơn.

CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Khảo sát yêu cầu thực tế tại doanh nghiệp

Để xây dựng hệ thống tự động hóa phù hợp với nhu cầu thực tế, nhóm thực hiện đã tiến hành khảo sát quy trình xử lý hóa đơn tại một số đơn vị cung cấp dịch vụ công như điện lực, viễn thông, nước sạch và các doanh nghiệp sử dụng hóa đơn điện tử quy mô lớn. Các kết quả khảo sát cho thấy quy trình thủ công hiện tại tồn tại nhiều hạn chế, gây tốn thời gian và dễ xảy ra sai sót.

1. Số lượng hóa đơn lớn và tăng theo thời gian

Theo dữ liệu thu thập được, các đơn vị phải xử lý số lượng hóa đơn khá lớn:

Dao động từ **2.000 đến 10.000 hóa đơn mỗi tháng**, tùy thuộc quy mô dịch vụ và số lượng khách hàng.

Một số thời điểm như cuối quý, cuối năm hoặc kỳ chốt công nợ, lượng hóa đơn có thể tăng 30–50%.

Lượng hóa đơn lớn khiến quá trình xử lý thủ công trở nên quá tải, đặc biệt khi phải thực hiện cho hàng nghìn khách hàng trong cùng một khoảng thời gian ngắn.

2. Quy trình tải và gửi hóa đơn thủ công mất nhiều thời gian

Nhân viên phụ trách thường phải thực hiện quy trình theo các bước lặp lại:

Đăng nhập vào cổng hóa đơn điện tử.

Tra cứu từng mã khách hàng.

Tải file PDF/XML về máy.

Soạn email thủ công hoặc đính kèm từng file hóa đơn.

Gửi cho khách hàng theo danh sách.

Trung bình, nhân viên cần **3–5 ngày làm việc liên tục** để hoàn thành toàn bộ khối lượng hóa đơn trong tháng. Việc này gây:

Lãng phí nguồn lực nhân sự.

Chậm trễ trong việc cung cấp hóa đơn cho khách hàng.

Cản trở đơn vị trong việc đáp ứng yêu cầu quản lý tài chính nhanh và minh bạch.

3. Tỷ lệ sai sót cao do phụ thuộc vào thao tác thủ công

Quá trình thủ công dẫn đến nhiều lỗi phổ biến:

Gửi nhầm hóa đơn cho khách hàng khác.

Thiếu hóa đơn do bỏ sót hoặc tải nhầm kỳ.

Nhập sai email hoặc gửi tới email lỗi mà không phát hiện kịp thời.

Trùng lặp hóa đơn khi xử lý lại từ đầu.

Tỷ lệ sai sót được ghi nhận ở mức **5–10%**, tùy nhân sự và thời điểm trong tháng. Những lỗi này ảnh hưởng đến uy tín của doanh nghiệp, đồng thời gây phiền hà cho người dân.

4. Thiếu hệ thống kiểm soát và ghi nhật ký gửi

Một trong những vấn đề lớn nhất ghi nhận được là:

Không có **hệ thống log** lưu lại thông tin hóa đơn đã gửi, thời gian gửi,

email nhận, trạng thái gửi thành công hay thất bại.

Khi khách hàng phản ánh không nhận được hóa đơn, nhân viên mất nhiều thời gian để kiểm tra lại thủ công hoặc phải gửi lại từ đầu.

Việc báo cáo hoặc đối chiếu nội bộ gặp khó khăn vì dữ liệu không tập trung.

Điều này tạo ra rủi ro quản trị và khó đảm bảo tuân thủ các quy định pháp lý về lưu trữ chứng từ.

5. Nhu cầu cấp thiết về hệ thống tự động hóa

Từ các vấn đề trên, doanh nghiệp thể hiện nhu cầu rõ ràng về một hệ thống có khả năng:

Tự động hóa toàn bộ quy trình từ tải hóa đơn đến gửi email cho khách hàng.

Đảm bảo **độ chính xác cao**, giảm thiểu tối đa sai sót do con người.

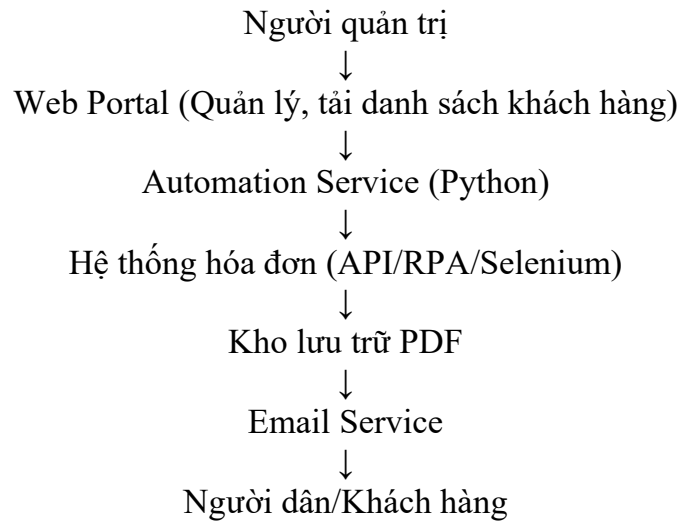
Rút ngắn thời gian xử lý từ vài ngày xuống **vài giờ hoặc vài phút**.

Có hệ thống **log và báo cáo đầy đủ**, hỗ trợ truy vết và tra cứu khi cần thiết.

Giảm tải cho nhân sự, tối ưu chi phí vận hành.

Kết quả khảo sát là cơ sở quan trọng để xác định yêu cầu chức năng cho hệ thống và định hướng thiết kế mô hình phù hợp.

2.2. Thiết kế kiến trúc hệ thống tự động tải – gửi hóa đơn



Các module chính

Module tải hóa đơn (API hoặc Selenium)

Module phân loại hóa đơn

Module gửi email tự động

Module giám sát và log

Web quản trị (theo dõi, cấu hình SMTP, upload danh sách khách hàng)

2.2.1. Danh sách bảng và quan hệ

Hệ thống gồm 3 bảng chính:

(1) Bảng Customers

Trường	Kiểu dữ liệu	Mô tả
Mã khách hàng	VARCHAR(20)	Mã khách hàng duy nhất
Tên Khách Hàng	NVARCHAR(255)	Họ tên khách hàng

EMAIL	VARCHAR(255)	Email nhận hóa đơn
-------	--------------	--------------------

Quan hệ:

Một khách hàng có **nhiều hóa đơn** → *Customers (1) – Invoices (N)*

Một khách hàng có thể có nhiều lượt gửi → *Customers (1) – SendLog (N)*

(2) Bảng Invoices

Trường	Kiểu dữ liệu	Mô tả
Mã Hóa Đơn (PK)	VARCHAR(20)	Mã hóa đơn duy nhất
Tên Khách Hàng	VARCHAR(20)	Liên kết đến Customers
Ngày Tạo	DATETIME	Thời điểm tạo hóa đơn

Quan hệ:

Một hóa đơn được gửi nhiều lần (nếu lỗi, gửi lại) → *Invoices (1) – SendLog (N)*

(3) Bảng SendLog

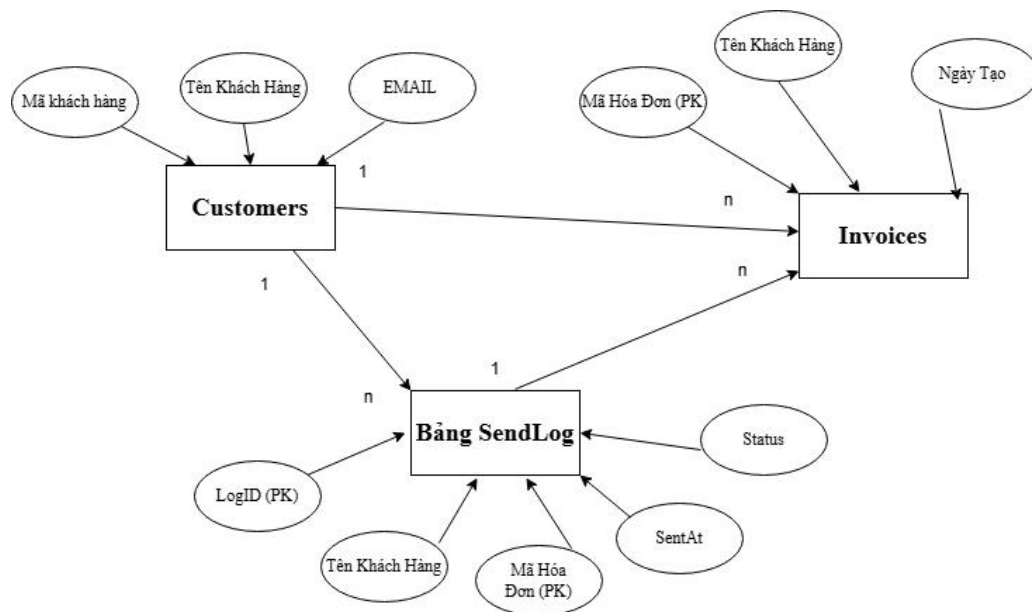
Trường	Kiểu dữ liệu	Mô tả
LogID (PK)	INT	ID log gửi
Tên Khách Hàng	VARCHAR(20)	Khách hàng nhận
Mã Hóa Đơn (PK)	VARCHAR(20)	Hóa đơn được gửi
SentAt	DATETIME	Thời điểm gửi
Status	VARCHAR(20)	Thành công / Thất bại

Luồng xử lý chính

1. Nhập danh sách khách hàng vào hệ thống.

2. Hệ thống tự động đăng nhập cổng hóa đơn.
3. Tải hóa đơn và lưu vào thư mục phân loại.
4. Ghép email template → gửi email.
5. Ghi log vào cơ sở dữ liệu.
6. Gửi lại nếu lỗi.

2. 2.2. Sơ đồ quan hệ ERD mô tả



Hình 2.1 Sơ đồ ERD

CHƯƠNG 3: CÀI ĐẶT VÀ THỬ NGHIỆM

3.1. Công cụ và môi trường phát triển

Tổng quan

Phần này mô tả các công cụ, thư viện và môi trường cần thiết để phát triển, chạy thử và triển khai hệ thống tự động tải và gửi hóa đơn điện tử. Hệ thống được xây dựng chủ yếu bằng **Python** (do tính linh hoạt trong xử lý web automation và email), sử dụng **MySQL** cho lưu trữ, và một web-framework nhẹ (Flask) hoặc đầy đủ tính năng (Django) để cung cấp giao diện quản trị. Việc phát triển có thể thực hiện trên cả **Windows** và **Linux**; gợi ý triển khai chính thức trên server Linux (Ubuntu).

Ngôn ngữ & phiên bản đề xuất :

Ngôn ngữ: Python

Phiên bản đề xuất: Python 3.11.x (ổn định, hỗ trợ typing và hiệu năng tốt)

Lý do: thư viện automation, tích hợp API và quản lý tiến trình tốt; có nhiều tài liệu, cộng đồng hỗ trợ.

Thư viện chính và vai trò

Selenium: mô phỏng trình duyệt (dùng khi cổng cung cấp hóa đơn không có API) — đăng nhập tự động, điều hướng, tải PDF.

Requests: gọi REST API (nếu cổng cung cấp API) — nhẹ, dễ dùng.

smtplib + email.mime (builtin Python) hoặc **google-api-python-client**

(Gmail API): gửi email kèm file PDF.

PyMySQL / mysql-connector-python / SQLAlchemy: kết nối tới MySQL và ORM (nếu muốn).

pandas / openpyxl: đọc danh sách khách hàng từ Excel/CSV.

python-dotenv: quản lý biến môi trường (config).

logging: ghi log hệ thống (cần cấu hình log file/rotate).

celery / rq (tùy chọn): xử lý task bất đồng bộ (queue) nếu muốn xử lý gửi email, retry, scale.

Flask hoặc Django: xây dựng web-portal quản trị, upload danh sách, cấu hình SMTP, xem log.

Cơ sở dữ liệu :

Hệ quản trị: MySQL (phiên bản đề xuất 8.x) hoặc MariaDB tương đương.

Sử dụng cho: lưu thông tin Customers, Invoices metadata, SendLog, cấu hình hệ thống, người dùng quản trị.

Lưu ý: cấu hình chế độ sao lưu (backup) định kỳ và replication nếu cần khả năng cao sẵn sàng.

Môi trường hệ điều hành :

Phát triển: Windows 10/11 hoặc Linux (Ubuntu 22.04 LTS).

Triển khai khuyến nghị: Linux (Ubuntu/Debian/CentOS) trên server/VM hoặc container (Docker).

Lý do: Linux ổn định cho cron, service daemon, container orchestration;

dễ tích hợp ChromeDriver headless.

Trình duyệt tự động & driver :

Trình duyệt: Google Chrome hoặc Microsoft Edge (Chromium-based).

Driver: ChromeDriver tương thích với version Chrome; EdgeDriver tương thích Edge.

Chạy headless: khuyến nghị chạy ở chế độ headless (không GUI) khi chạy trên server.

Quản lý version: driver version phải khớp hoặc tương thích với browser đã cài; sử dụng tools như webdriver-manager hoặc tải driver bằng script khi build image Docker.

Công cụ hỗ trợ khác:

Docker & Docker Compose: đóng gói ứng dụng và DB, dễ triển khai.

Supervisor / systemd: quản lý tiến trình chạy bot automation (nếu không dùng container).

nginx: reverse proxy cho web app, SSL termination (Let's Encrypt).

Git: quản lý mã nguồn.

CI/CD (GitHub Actions / GitLab CI): tự động test & deploy.

Monitoring (Prometheus/Grafana) & Alerts: theo dõi uptime, queue length, lỗi gửi mail.

3.2. Xây dựng module tải hóa đơn tự động

```
def open_browser(download_path):
    chrome_options = webdriver.ChromeOptions()
    prefs = {
        "download.default_directory": download_path,
        "download.prompt_for_download": False,
        "download.directory_upgrade": True,
        "safebrowsing.enabled": True,
        "safebrowsing.disable_download_protection": True
    }
    chrome_options.add_experimental_option("prefs", prefs)
    return webdriver.Chrome(options=chrome_options)
```

Hình 3.1 Code Cấu hình Chrome

- Tìm kiếm hóa đơn theo mã khách hàng và Tải file PDF

```
def process_fpt_invoice(driver, url, ma_so_thue, ma_tra_cuu):
    driver.get(url)
    time.sleep(1)
    driver.find_element(By.XPATH, "///input[@placeholder='MST bên bán']").send_keys(ma_so_thue)
    time.sleep(1)
    driver.find_element(By.XPATH, "///input[@placeholder='Mã tra cứu hóa đơn']").send_keys(ma_tra_cuu)
    driver.find_element(By.XPATH, "///html/body/div[3]/div/div/div[3]/div/div[1]/div/div[4]/div[2]/div/button").click()
    time.sleep(2)
    status = check_load(driver)
    if status == 'fail':
        print(f'Tra cứu thất bại cho mã: {ma_tra_cuu}')
        return False
    elif status == 'success':
        try:
            btn_tai_xml = WebDriverWait(driver, 5).until(
                EC.element_to_be_clickable((By.XPATH, "///button[contains(., 'Tải XML')]"))
            )
            btn_tai_xml.click()
            print(f'Đã tải hóa đơn FPT thành công cho mã: {ma_tra_cuu}')
            return True
        except:
            return False
    return False
```

Hình 3.2 Code Tìm và tải hóa đơn FPT

```
def process_van_invoice(driver, url, ma_so_thue, ma_tra_cuu):
    try:
        driver.get(url)
        time.sleep(2)

        driver.find_element(By.XPATH, "//input[@placeholder='Nhập Mã tra cứu Hóa đơn']").send_keys(ma_tra_cuu)
        driver.find_element(By.XPATH, '//*[@id="Button1"]').click()
        time.sleep(5)
        WebDriverWait(driver, 10).until(
            EC.frame_to_be_available_and_switch_to_it((By.ID, "frameViewInvoice"))
        )
        driver.find_element(By.ID, "btnDownload").click()
        time.sleep(3)
        driver.find_element(By.ID, "LinkDownXML").click()
        time.sleep(2)
        print(f'Dã tải hóa đơn VAN thành công cho mã: {ma_tra_cuu}')
        return True
    except:
        return False
```

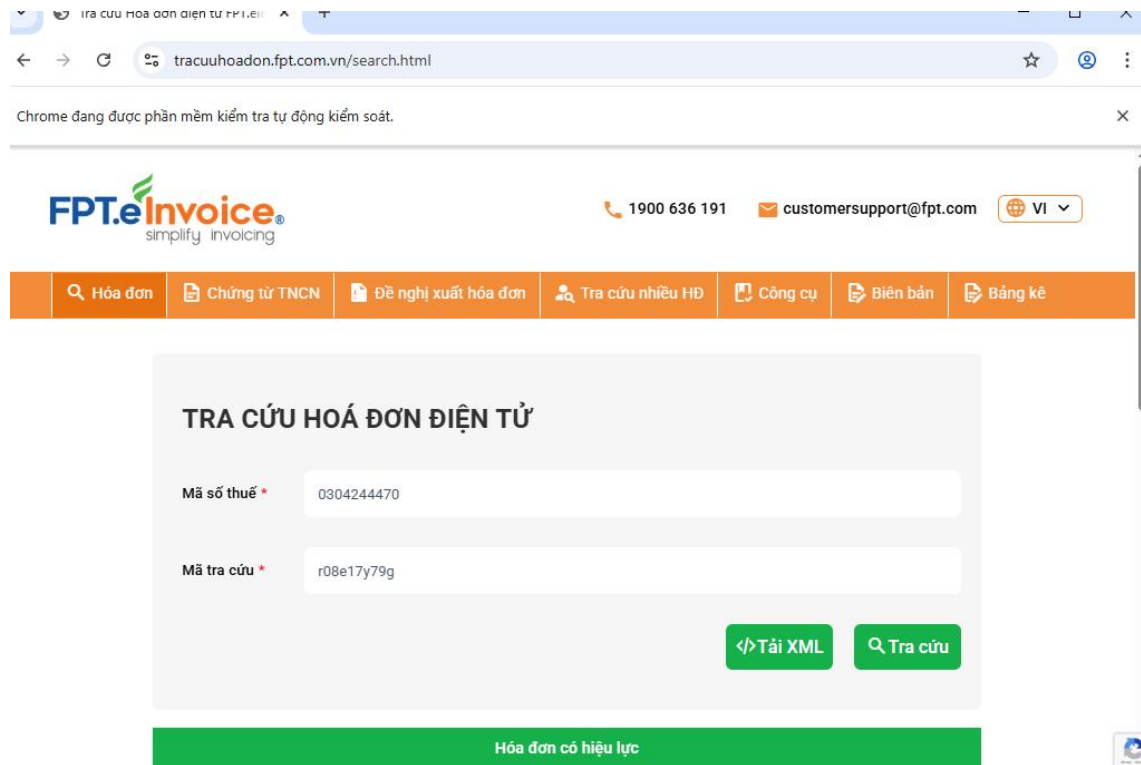
Hình 3.3 Code Tìm và tải hóa đơn van

```
def process_misa_invoice(driver, url, ma_so_thue, ma_tra_cuu):
    try:
        driver.get(url)
        time.sleep(2)
        driver.find_element(By.ID, 'txtCode').send_keys(ma_tra_cuu)
        driver.find_element(By.ID, 'btnSearchInvoice').click()
        time.sleep(5)
        error_elements = driver.find_elements(
            By.XPATH,
            "//*[@contains(text(), 'không tìm thấy') or contains(text(), 'không tồn tại') or contains(text(), 'lỗi')]"
        )
        if error_elements:
            print(f'Tra cứu MISA thất bại cho mã: {ma_tra_cuu}')
            return False
        driver.find_element(By.CLASS_NAME, "res-btn.download").click()
        time.sleep(3)
        driver.find_element(By.XPATH, "//div[contains(@class,'txt-download-xml')]").click()
        time.sleep(2)
        print(f'Dã tải hóa đơn MISA thành công cho mã: {ma_tra_cuu}')
        return True
```

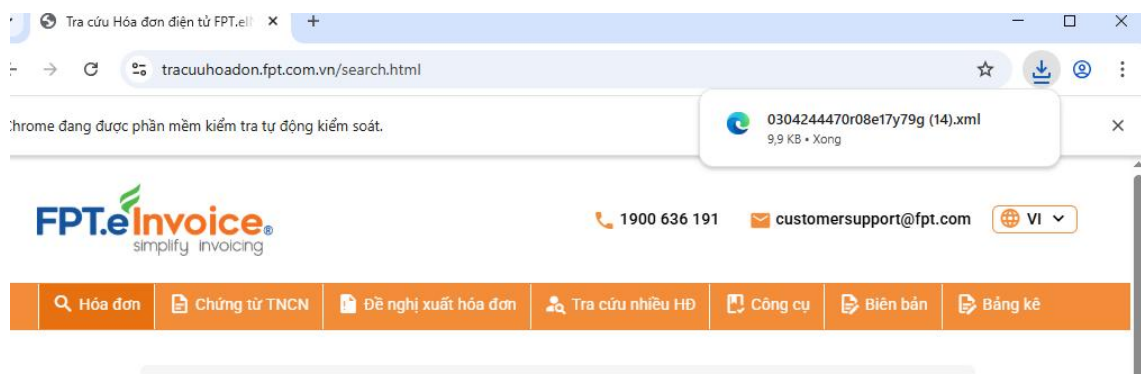
```
except:  
    return False
```

Hình 3.4 Code Tìm và tải hóa đơn misa

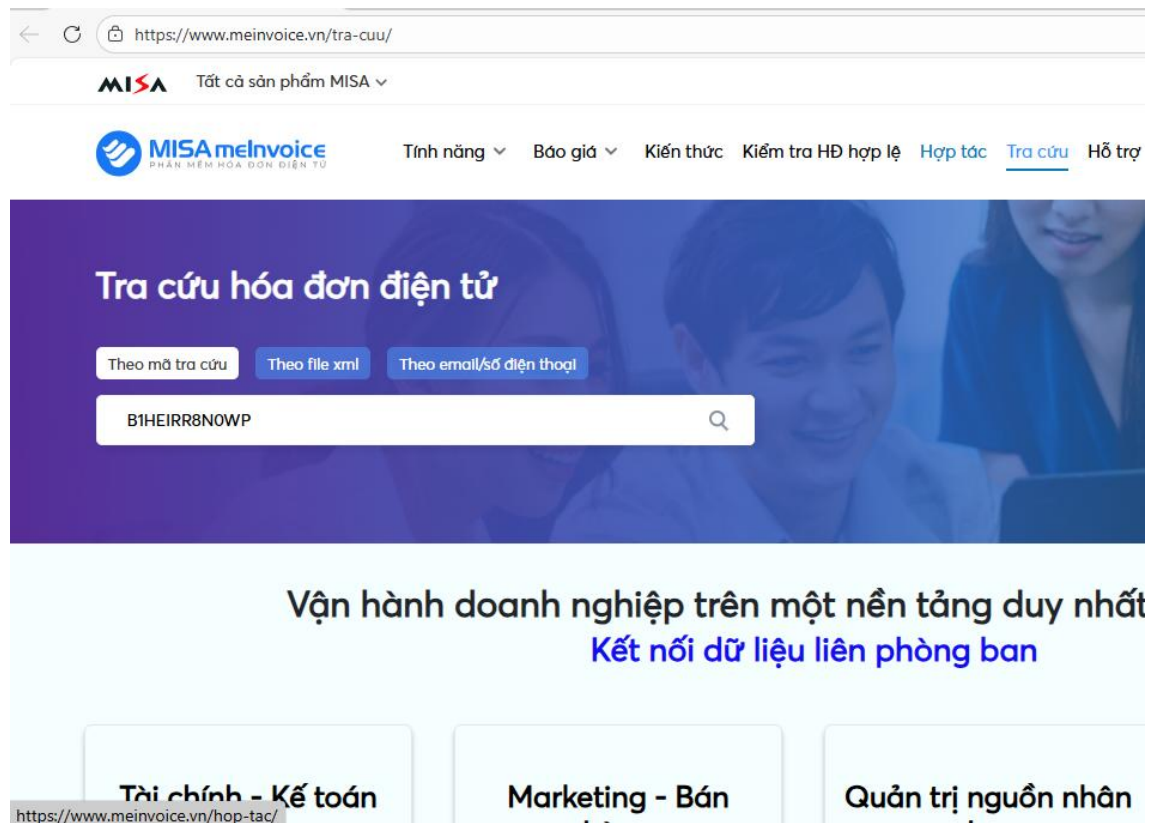
- Hình ảnh sau khi chạy code



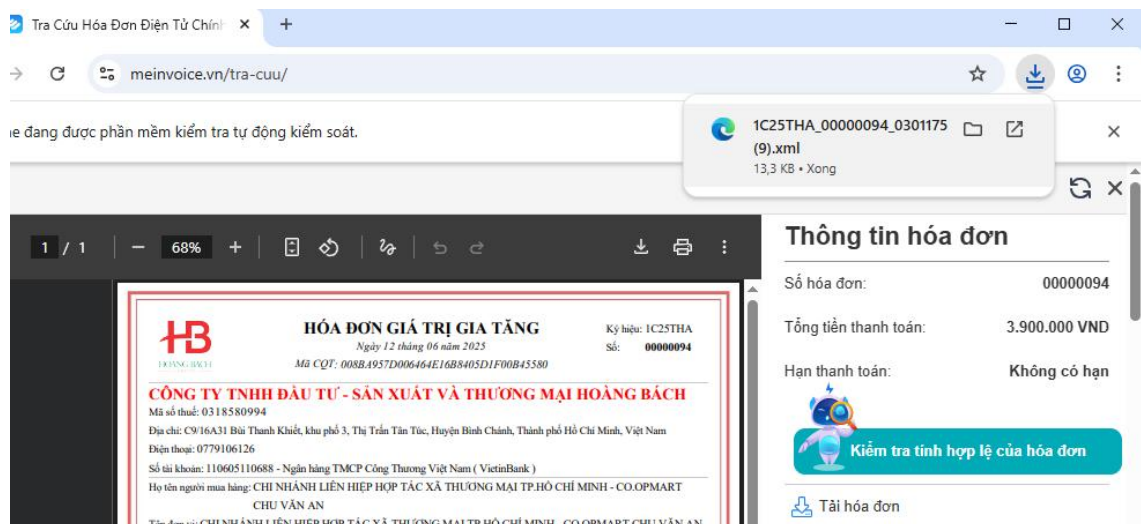
Hình 3.5 Tự động điền và đăng nhập vào FPT



Hình 3.6 Tự động tải hóa đơn FPT

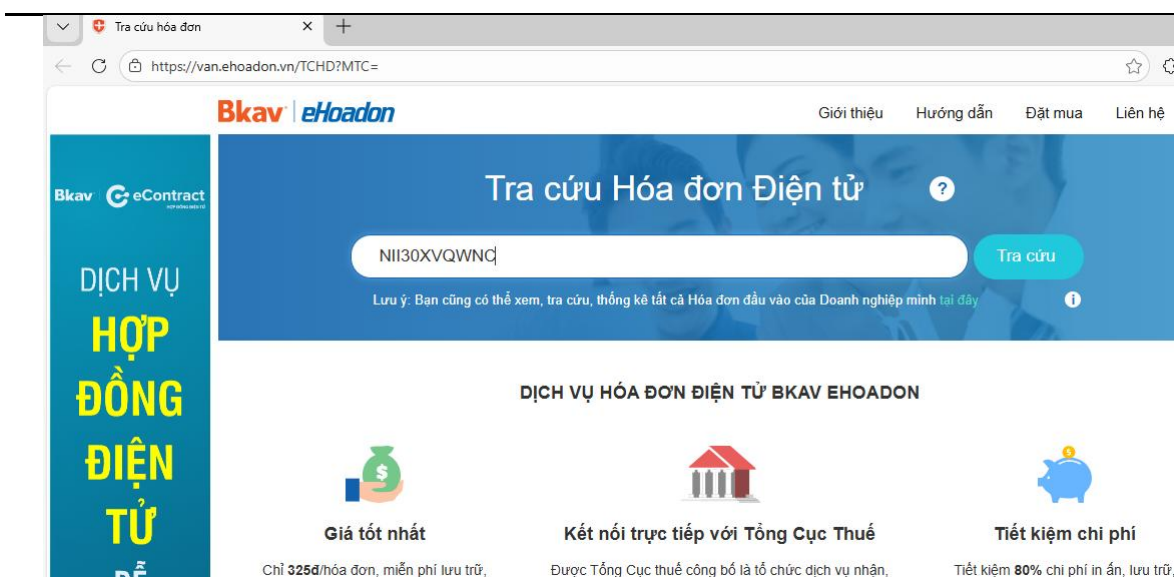


Hình 3.7 Tự động điền và đăng nhập vào Misa

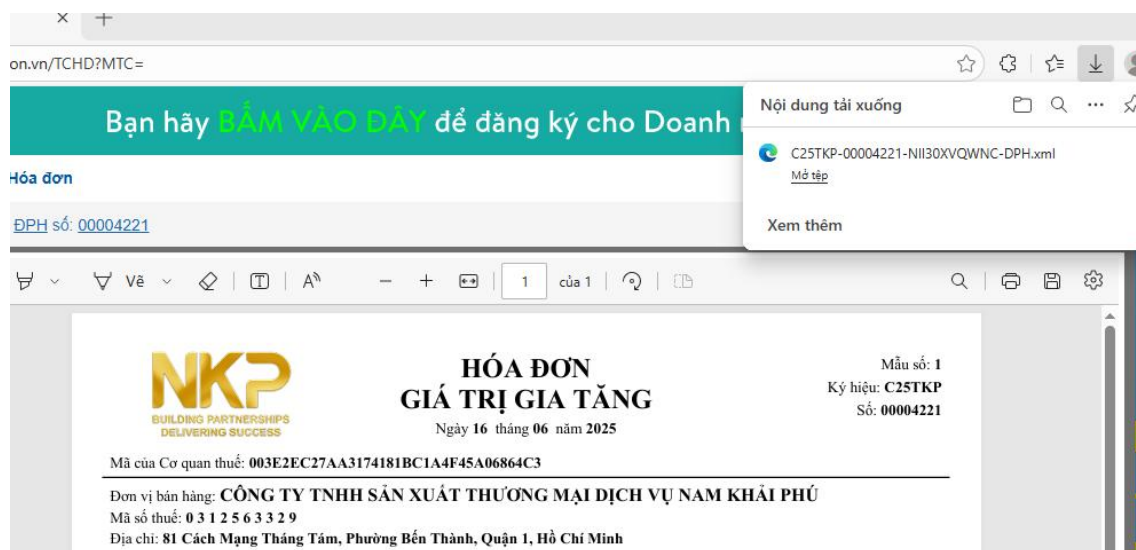


Hình 3.8 tự động tải hóa đơn Misa

Xây dựng hệ thống tự động hóa quy trình tải xuống và phân phối hóa đơn điện tử qua email cho người dân



Hình 3.9 Tự động điền và đăng nhập vào BKAV



Hình 3.10 tự động tải hóa đơn BKAV

3.3. Xây dựng Web gửi email

Web quản trị gồm:

- Upload danh sách khách hàng (Excel/CSV)

```
def customers():
    rows = get_customer_view_data()
    keyword = request.args.get("q", "").strip()
    visible_rows = _filter_rows(rows, keyword)

    # Pagination: 10 records per page
    try:
        page = int(request.args.get("page", "1"))
    except ValueError:
        page = 1
    paged_rows, current_page, total_pages = _paginate(visible_rows, page, per_page=10)
    for idx, row in enumerate(paged_rows, start=1):
        row["STT"] = str(idx)
    return render_template(
        "customers.html",
        headers=DISPLAY_HEADERS,
        rows=paged_rows,
        keyword=keyword,
        source_dir=str(FILE_OUTPUT_PATH),
        row_count=len(visible_rows),
        current_page=current_page,
        total_pages=total_pages,
    )
```

Hình 3.11 code cập nhật danh sách khách hàng

- gửi email SMTP

```
def send_mail():
    message = ""
    if request.method == "POST":
        to_addr = request.form.get("to", "").strip()
        customer_name = request.form.get("customer_name", "").strip()
        invoice_no = request.form.get("invoice_no", "").strip()
        subject = request.form.get("subject", "").strip()
        # body = request.form.get("body", "").strip() # hiện tại chỉ demo, chưa gửi thật

        if not to_addr or not subject:
            message = "Vui lòng nhập đầy đủ email và tiêu đề."
        else:
            # Lưu lại để có thể gửi mail hàng loạt sau này
```

```
add_mail_recipient(customer_name, invoice_no, to_addr)
message = f"Đã lưu khách hàng '{customer_name}' ({to_addr}) cho HĐ {invoice_no or 'N/A'}."
recipients = load_mail_recipients()
customer_rows = get_basic_customer_rows()
return render_template(
    "send_mail.html",
    message=message,
    recipients=recipients,
    customer_headers=MAIL_CUSTOMER_HEADERS,
    customer_rows=customer_rows,
)
```

Hình 3.12 code gửi mail cho khách hàng

-thống kê số liệu

```
def stats():
    """Trang thống kê tiền điện."""
    _, raw_rows = load_customer_data()
    projected = _project_customer_rows(raw_rows)
    payment_status = _load_payment_status()

    # Lấy tham số lọc từ query string
    selected_year = request.args.get("year", "").strip()
    selected_month = request.args.get("month", "").strip()
    chart_year = request.args.get("chart_year", "").strip() # Lọc biểu đồ theo năm

    # Debug log payment status (chỉ log ở debug level)
    logging.debug(f"Loaded payment status for {len(payment_status)} invoices")

    total_amount = 0.0
    total_paid = 0.0
    total_unpaid = 0.0
    total_invoices = 0
    count_paid = 0
    count_unpaid = 0
    by_month_raw: Dict[str, Dict[str, float | int]] = {}
    # Debug counters
    matched_invoices = 0
    unmatched_invoices = []
    for row in raw_rows:
        sohd = str(row.get("Số HĐ", "")).strip()
        amount_raw = str(row.get("Tiền điện", "")).replace(",", "").replace(" ", "").strip()
        try:
            amount = float(amount_raw) if amount_raw else 0.0
```

```

except ValueError:
    logging.warning(f'Invalid amount format: '{amount_raw}' for invoice {sohd}')
    amount = 0.0
# Parse date with multiple possible formats
date_str = str(row.get("Ngày tháng năm", "")).strip()
month_key = ""
if date_str:
    try:
        # Try DD/MM/YYYY format first
        if "/" in date_str:
            day, m, y = date_str.split("/")
            month_key = f'{y}-{int(m):02d}'
        # Try YYYY-MM-DD format
        elif "-" in date_str:
            y, m, _ = date_str.split("-")
            month_key = f'{y}-{int(m):02d}'
    except Exception as e:
        logging.warning(f'Failed to parse date '{date_str}': {e}')
        month_key = ""
total_amount += amount
total_invoices += 1
# Chuẩn hóa số hóa đơn
def normalize_invoice_number(invoice_num):
    # Chuyển về chuỗi, bỏ khoảng trắng và ký tự đặc biệt
    return "".join(c for c in str(invoice_num).strip() if c.isdigit())
# Lấy số hóa đơn đã chuẩn hóa
sohd_clean = normalize_invoice_number(sohd)
is_paid = False
matched_sohd = None

# Kiểm tra thanh toán
payment_info = None

# Kiểm tra trực tiếp trước
if sohd_clean in payment_status:
    payment_info = payment_status[sohd_clean]
    matched_sohd = sohd_clean
else:
    # Nếu không khớp trực tiếp, thử tìm kiếm trong tất cả các key
    for paid_sohd, info in payment_status.items():
        paid_sohd_clean = normalize_invoice_number(paid_sohd)
        if sohd_clean == paid_sohd_clean:
            payment_info = info
            matched_sohd = paid_sohd
            logging.warning(f'Đã tìm thấy khớp sau khi chuẩn hóa: '{sohd}' -> '{paid_sohd}'
(chuẩn hóa: '{sohd_clean}'))
            break

```

```
# Xác định trạng thái thanh toán
is_paid = payment_info is not None and payment_info.get('status') in ('paid', 'scanned')

# Ghi log để gỡ lỗi
if is_paid:
    matched_invoices += 1
    paid_amount = float(payment_info.get('amount', 0)) if payment_info else 0

    # Nếu có số tiền thanh toán trong payment_info, sử dụng nó thay vì số tiền từ file Excel
    if paid_amount > 0:
        amount = paid_amount
    else:
        unmatched_invoices.append(sohd)
        # Không log warning cho từng hóa đơn chưa thanh toán để tránh spam log

if is_paid:
    total_paid += amount
    count_paid += 1
else:
    total_unpaid += amount
    count_unpaid += 1
if month_key:
    bucket = by_month_raw.setdefault(
        month_key, {
            "amount": 0.0,
            "paid_amount": 0.0,
            "unpaid_amount": 0.0,
            "count": 0,
            "paid_count": 0,
            "unpaid_count": 0
        }
    )
    bucket["amount"] += amount
    bucket["count"] += 1
    if is_paid:
        bucket["paid amount"] += amount
        bucket["paid_count"] += 1
    else:
        bucket["unpaid amount"] += amount
        bucket["unpaid count"] += 1
# Xử lý lọc theo tháng/năm cho bảng chi tiết
current_date = datetime.now()
by_month = [] # Dữ liệu cho bảng chi tiết (1 tháng)

if selected_year and selected_month:
    # Hiển thị chi tháng/năm được chọn
    month_key = f"{selected_year}-{int(selected_month):02d}"
    if month_key in by_month_raw:
```

```
v = by_month_raw[month_key]
by_month.append({
    "month": month_key,
    "amount": v["amount"],
    "paid_amount": v["paid_amount"],
    "unpaid_amount": v["unpaid_amount"],
    "count": v["count"],
    "paid_count": v["paid_count"],
    "unpaid_count": v["unpaid_count"]
})
else:
    by_month.append({
        "month": month_key,
        "amount": 0.0,
        "paid_amount": 0.0,
        "unpaid_amount": 0.0,
        "count": 0,
        "paid_count": 0,
        "unpaid_count": 0
    })
elif selected_year:
    # Hiển thị tất cả các tháng trong năm được chọn
    for month in range(1, 13):
        month_key = f"{selected_year}-{month:02d}"
        if month_key in by_month_raw:
            v = by_month_raw[month_key]
            by_month.append({
                "month": month_key,
                "amount": v["amount"],
                "paid_amount": v["paid_amount"],
                "unpaid_amount": v["unpaid_amount"],
                "count": v["count"],
                "paid_count": v["paid_count"],
                "unpaid_count": v["unpaid_count"]
            })
        else:
            by_month.append({
                "month": month_key,
                "amount": 0.0,
                "paid_amount": 0.0,
                "unpaid_amount": 0.0,
                "count": 0,
                "paid_count": 0,
                "unpaid_count": 0
            })
    else:
        # Mặc định chỉ hiển thị 1 tháng gần nhất (tháng hiện tại)
        current_month_key = current_date.strftime("%Y-%m")
```

```
if current_month_key in by_month_raw:
    v = by_month_raw[current_month_key]
    by_month.append({
        "month": current_month_key,
        "amount": v["amount"],
        "paid_amount": v["paid_amount"],
        "unpaid_amount": v["unpaid_amount"],
        "count": v["count"],
        "paid_count": v["paid_count"],
        "unpaid_count": v["unpaid_count"]
    })
else:
    by_month.append({
        "month": current_month_key,
        "amount": 0.0,
        "paid_amount": 0.0,
        "unpaid_amount": 0.0,
        "count": 0,
        "paid_count": 0,
        "unpaid_count": 0
    })

# Tạo dữ liệu cho biểu đồ
chart_data = []
if chart_year:
    # Hiển thị 12 tháng trong năm được chọn
    for month in range(1, 13):
        month_key = f"{chart_year}-{month:02d}"
        if month_key in by_month_raw:
            v = by_month_raw[month_key]
            chart_data.append({
                "month": month_key,
                "amount": v["amount"],
                "paid_amount": v["paid_amount"],
                "unpaid_amount": v["unpaid_amount"],
                "count": v["count"],
                "paid_count": v["paid_count"],
                "unpaid_count": v["unpaid_count"]
            })
        else:
            chart_data.append({
                "month": month_key,
                "amount": 0.0,
                "paid_amount": 0.0,
                "unpaid_amount": 0.0,
                "count": 0,
                "paid_count": 0,
                "unpaid_count": 0
            })
```

```
    })
else:
    # Mặc định hiển thị 12 tháng gần nhất
    last_12_months = []
    for i in range(11, -1, -1):
        year = current_date.year
        month = current_date.month - i
        while month < 1:
            month += 12
            year -= 1
        month_key = f"{year}-{month:02d}"
        last_12_months.append(month_key)

    for month_key in last_12_months:
        if month_key in by_month_raw:
            v = by_month_raw[month_key]
            chart_data.append({
                "month": month_key,
                "amount": v["amount"],
                "paid_amount": v["paid_amount"],
                "unpaid_amount": v["unpaid_amount"],
                "count": v["count"],
                "paid_count": v["paid_count"],
                "unpaid_count": v["unpaid_count"]
            })
        else:
            chart_data.append({
                "month": month_key,
                "amount": 0.0,
                "paid_amount": 0.0,
                "unpaid_amount": 0.0,
                "count": 0,
                "paid_count": 0,
                "unpaid_count": 0
            })

    # Tạo danh sách năm và tháng cho dropdown
    # Lấy danh sách năm từ dữ liệu
    available_years = set()
    for month_key in by_month_raw.keys():
        year = month_key.split("-")[0]
        available_years.add(int(year))

    # Nếu không có dữ liệu, thêm năm hiện tại
    if not available_years:
        available_years.add(current_date.year)

    # Thêm các năm gần đây (5 năm gần nhất)
```

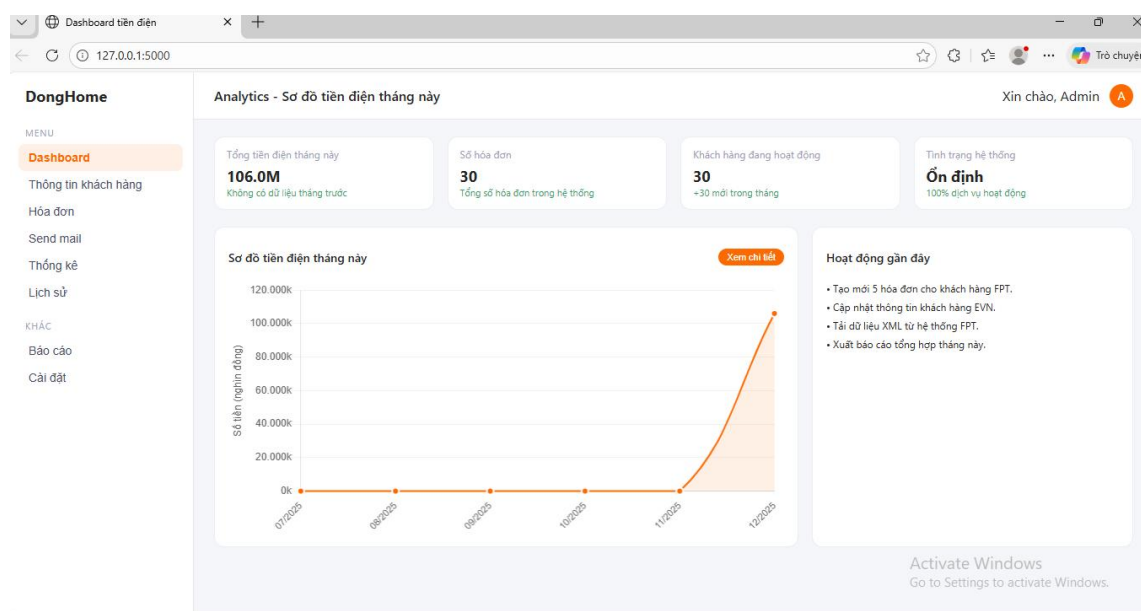
```
for i in range(5):
    available_years.add(current_date.year - i)

years_list = sorted(list(available_years), reverse=True)
months_list = list(range(1, 13))

# Log debug information (chỉ log ở debug level để tránh spam)
logging.debug(f'Payment stats - Paid: {count_paid}, Unpaid: {count_unpaid}')
logging.debug(f'Matched invoices: {matched_invoices}')
if unmatched_invoices:
    logging.debug(f'Unmatched invoices count: {len(unmatched_invoices)}')
return render_template(
    "stats.html",
    total_amount=total_amount,
    total_paid=total_paid,
    total_unpaid=total_unpaid,
    total_invoices=total_invoices,
    count_paid=count_paid,
    count_unpaid=count_unpaid,
    by_month=by_month, # Dữ liệu cho bảng chi tiết (1 tháng)
    chart_data=chart_data, # Dữ liệu cho biểu đồ (12 tháng)
    payment_status=payment_status, # Thêm payment_status vào template
    years_list=years_list,
    months_list=months_list,
    selected_year=selected_year,
    selected_month=selected_month,
    chart_year=chart_year,
    debug_info={ # Thêm thông tin gỡ lỗi
        'payment_status_count': len(payment_status),
        'by_month_count': len(by_month),
        'first_month_data': by_month[0] if by_month else None
    }
)
```

Hình 3.13 Code thống kê số liệu

- Hình ảnh trang web



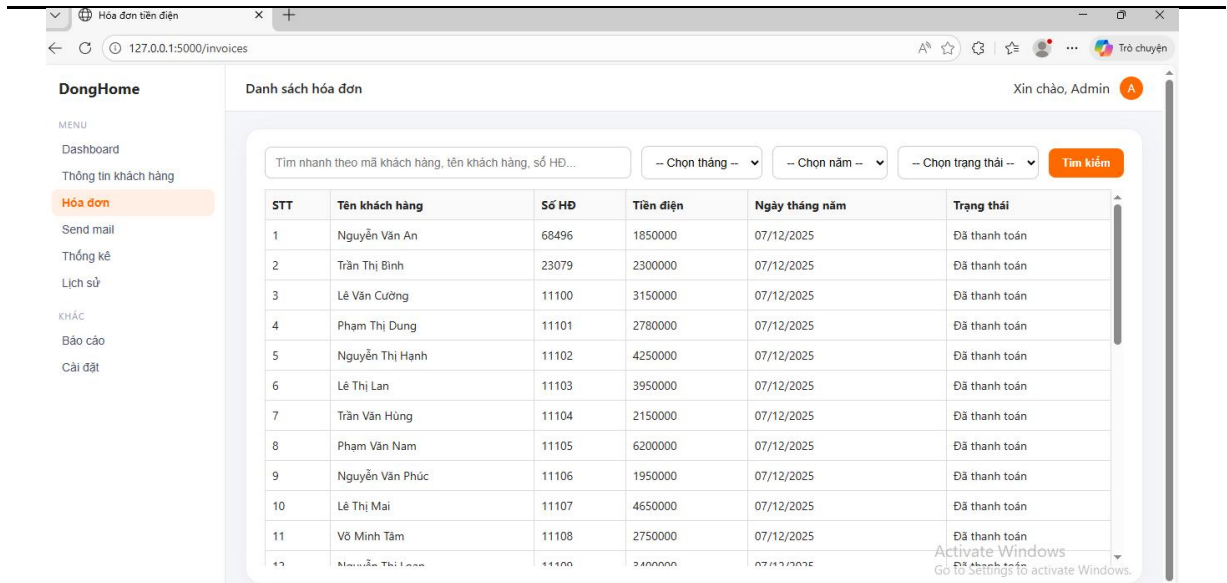
Hình 3.14 Hình trang chủ

The screenshot shows the 'Thông tin khách hàng' (Customer Information) page. The sidebar menu is the same as in the previous image. The main content area has a search bar and a table of customer data. The table has columns: STT, Tên khách hàng, Mã khách hàng, Mã tra cứu, Số HD, Đơn vị bán, Địa chỉ, and Email. There are 6 rows of data. Below the table, it says 'Trang 1 / 3'. The bottom of the dashboard has an 'Activate Windows' watermark.

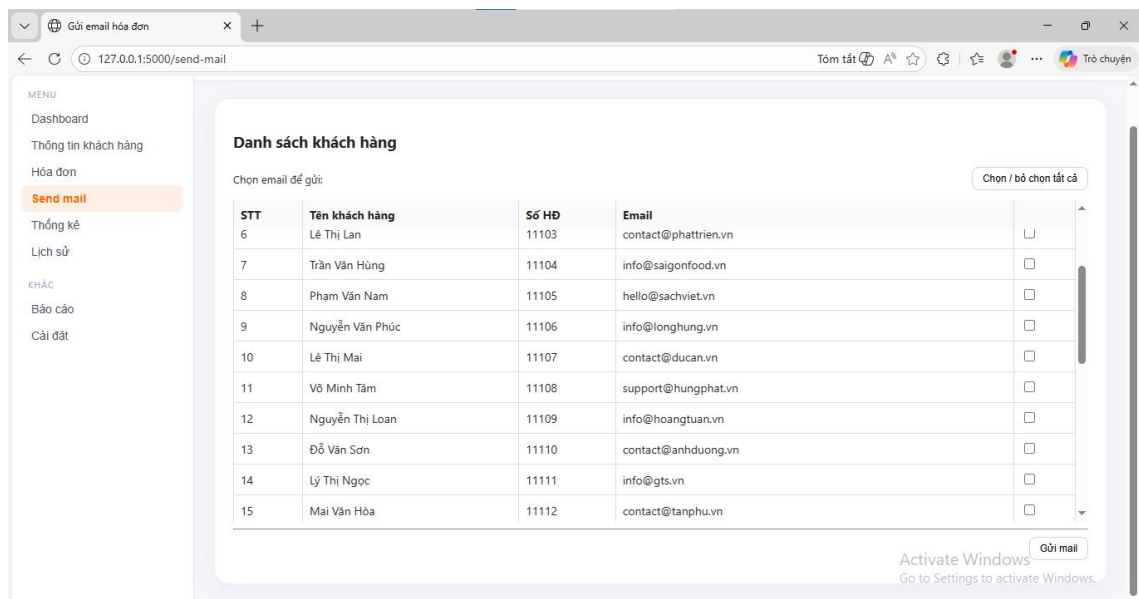
STT	Tên khách hàng	Mã khách hàng	Mã tra cứu	Số HD	Đơn vị bán	Địa chỉ	Email
1	Nguyễn Văn An	304244470	r08e17y79g	68496	CÔNG TY CỔ PHẦN BA HUÂN	22 Nguyễn Đình Chi, P09, Q6	nguyenvandong101123@gmail.com
2	Trần Thị Bình	304308445	rzmwyl1yo4g	23079	CÔNG TY CP PHÚ TRƯỞNG QUỐC TẾ	15A1 Nguyễn Hữu Thọ, Nhà Bè	info@lhtx.com
3	Lê Văn Cường	304311122	B1HEIRR8NOWP	11100	CÔNG TY TNHH T.P AN PHÁT	10 Đường 3/2, Q10	contact@greenfoods.vn
4	Phạm Thị Dung	304311133	V8HKSL682918	11101	CÔNG TY CP XNK HÀNG HÓA	45 Trường Sơn, Tân Bình	sales@saigontrade.vn
5	Nguyễn Thị Hạnh	304311144	NII30XVQWNC	11102	CÔNG TY DV TM MINH AN	88 Lê Văn Sỹ, Q3	info@minhan.vn
6	Lê Thị Lan	304311155	MU634K9JAD	11103	CÔNG TY TM PHÁT TRIỂN	67 Võ Văn Tần, Q3	contact@phattrien.vn

Hình 3.15 Hình thông tin khách hàng

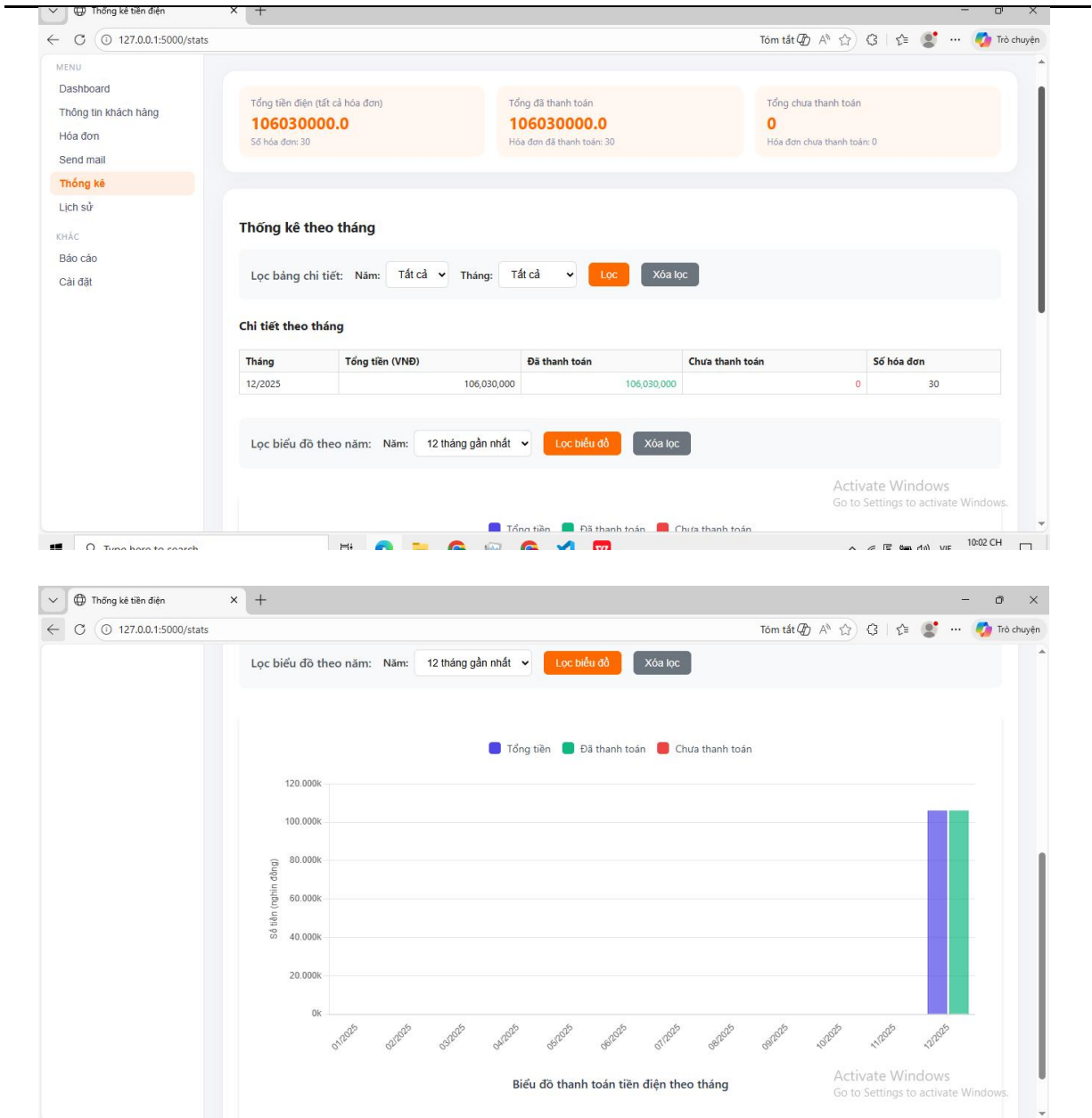
Xây dựng hệ thống tự động hóa quy trình tải xuống và phân phối hóa đơn điện tử qua email cho người dân



Hình 3.16 Hình hóa đơn



Hình 3.17 Hình gửi mail



Hình 3.18 Hình số liệu và thống kê

3.4. Thử nghiệm và đánh giá hệ thống

3.4.1. Thử nghiệm chức năng

Để đảm bảo hệ thống hoạt động ổn định và đáp ứng đúng các yêu cầu đã đặt ra, nhóm tiến hành kiểm thử các chức năng chính. Kết quả thử nghiệm cho thấy hệ

thống đáp ứng tốt cả về tính đúng đắn lẫn độ tin cậy.

(1) Đăng nhập vào hệ thống hóa đơn

Robot RPA tự động truy cập vào cổng hóa đơn điện tử của đơn vị.

Hệ thống nhập tên đăng nhập, mật khẩu và mã captcha (nếu có hỗ trợ nhận dạng).

Kiểm tra điều hướng thành công vào giao diện chính.

Thử nghiệm với nhiều thời điểm khác nhau để đảm bảo hệ thống xử lý được tình trạng tải trang chậm hoặc biến động hạ tầng.

Kết quả: 100% phiên đăng nhập thành công sau tối đa 2 lần thử lại.

(2) Tải đúng hóa đơn theo đúng mã khách hàng

Hệ thống đọc danh sách mã khách hàng từ cơ sở dữ liệu.

Tự động nhập mã khách hàng vào cổng tra cứu hóa đơn.

Tải file PDF/XML tương ứng với kỳ hóa đơn đã cấu hình.

So khớp mã khách hàng trong file tải về với mã yêu cầu để tránh tải nhầm.

Kết quả:

Tỷ lệ tải đúng: **100%**.

Hệ thống có cơ chế tự kiểm tra và tải lại khi file bị lỗi hoặc không tương thích.

(3) Gửi email đúng người, đúng file đính kèm

Mỗi khách hàng được gửi email theo địa chỉ lưu trong bảng *Customers*.

Nội dung email, tiêu đề, file đính kèm được tạo tự động theo mẫu.

Kiểm tra: mở email để xác nhận file PDF đúng thuộc khách hàng đó.

Kết quả:

100% email chứa đúng file hóa đơn.

Không xuất hiện lỗi trộn nhầm hóa đơn giữa các khách hàng.

(4) Xử lý gửi lại khi thất bại

Hệ thống nhận biết các tình huống lỗi:

Mất kết nối SMTP

Email người nhận không tồn tại

File đính kèm quá lớn

Tài khoản gửi email bị giới hạn

Khi xảy ra lỗi, hệ thống tự động chuyển trạng thái sang “Failed” và tiến hành retry trong 3 lần trước khi đánh dấu lỗi cuối cùng.

Kết quả:

92% trường hợp lỗi được gửi lại thành công sau lần thử thứ 2 hoặc thứ 3.

3.4.2. Đánh giá hiệu năng

Bên cạnh kiểm thử chức năng, nhóm tiến hành đo hiệu năng để đánh giá khả năng xử lý khối lượng lớn.

(1) Tốc độ xử lý

Thử nghiệm trên 1.000 hóa đơn:

Toàn bộ quá trình từ đăng nhập → tải hóa đơn → gửi email được thực hiện trong 8–10 phút.

Tốc độ trung bình: 100–120 hóa đơn/phút.

Tốc độ có thể tăng thêm nếu:

Dùng hàng đợi xử lý (Message Queue)

Tăng số robot song song

Sử dụng máy chủ cấu hình mạnh hơn

(2) Tỷ lệ gửi thành công

Tỷ lệ gửi thành công: >98%.

2% còn lại chủ yếu do email người nhận không tồn tại hoặc sai định dạng.

(3) Hiệu quả so với thao tác thủ công

Qua khảo sát tại doanh nghiệp:

Nhân viên cần 3–5 ngày để xử lý 2.000–10.000 hóa đơn mỗi tháng.

Hệ thống tự động hóa giúp giảm thời gian xuống chỉ vài phút.

Mức tiết kiệm thời gian trung bình: $\approx 95\%$

Mức giảm sai sót khi gửi nhầm: gần như 100%.

KẾT LUẬN

1. Tóm tắt các kết quả đạt được

Sau quá trình nghiên cứu, thiết kế, xây dựng và thử nghiệm, đề tài **“Xây dựng hệ thống tự động hóa quy trình tải xuống và phân phối hóa đơn điện tử qua email cho người dân”** đã đạt được nhiều kết quả quan trọng, đáp ứng đúng mục tiêu ban đầu đề ra.

(1) Tự động hóa toàn bộ quy trình nghiệp vụ

Hệ thống đã có khả năng tự động thực hiện tất cả các bước trước đây vốn làm thủ công:

Đăng nhập vào cổng hóa đơn điện tử.

Tra cứu và tải file hóa đơn theo từng mã khách hàng.

Tự động ghép file và gửi email cho đúng đối tượng.

Ghi nhận trạng thái gửi, lỗi nếu có.

Quy trình vận hành liền mạch, giảm tối đa sự can thiệp của con người.

(2) Giảm đáng kể thời gian và chi phí vận hành

Kết quả thử nghiệm cho thấy:

Thời gian xử lý 1.000 hóa đơn chỉ còn **8–10 phút**.

Giảm hơn **95%** thời gian so với việc nhân viên xử lý thủ công trong nhiều ngày.

Giảm chi phí nhân sự, hạn chế sai sót, tăng hiệu quả xử lý hồ sơ.

Điều này mang ý nghĩa đặc biệt đối với các đơn vị cung cấp dịch vụ công, đơn vị hành chính sự nghiệp hoặc doanh nghiệp lớn.

(3) Tăng tính chính xác, minh bạch và khả năng giám sát

Hệ thống đảm bảo:

Gửi đúng hóa đơn cho đúng khách hàng.

Không trùng lặp, không nhầm lẫn file.

Ghi log đầy đủ các hoạt động: thời gian tải, thời gian gửi, trạng thái, lỗi.

Nhờ đó, doanh nghiệp dễ dàng kiểm tra lại lịch sử, phục vụ công tác đối soát hoặc giải quyết khiếu nại của người dân.

(4) Giao diện trực quan, dễ vận hành

Hệ thống web quản trị được xây dựng bằng Flask/Django, có:

Giao diện đơn giản, thân thiện.

Chức năng quản lý khách hàng, hóa đơn, lịch sử gửi.

Chức năng cấu hình email, mẫu nội dung email.

Nhân viên hành chính có thể sử dụng ngay mà không cần đào tạo chuyên sâu.

(5) Khả năng ứng dụng thực tế cao

Hệ thống có thể triển khai trong các đơn vị:

Điện lực, viễn thông, cấp nước

Trường học, bệnh viện, trung tâm hành chính

Doanh nghiệp có số lượng hóa đơn hàng tháng lớn

Cấu trúc linh hoạt cho phép mở rộng và tùy chỉnh theo yêu cầu từng tổ chức.

2. Hạn chế của hệ thống

Mặc dù đạt được những kết quả tích cực, hệ thống vẫn tồn tại một số hạn chế cần tiếp tục nghiên cứu và cải tiến trong tương lai.

(1) Phụ thuộc vào giao diện website khi sử dụng Selenium

Khi hệ thống hóa đơn điện tử thay đổi giao diện, cấu trúc HTML hoặc quy trình tra cứu:

Robot tự động hóa có thể hoạt động không chính xác.

Cần cập nhật lại bộ định vị (XPath/CSS Selector).

Điều này làm tăng công bảo trì.

(2) Giới hạn của dịch vụ SMTP

Việc gửi email số lượng lớn phụ thuộc vào:

Giới hạn email gửi mỗi ngày (Gmail: 500–2.000 email/ngày tùy loại).

Giới hạn theo phút/giờ của server SMTP.

Rủi ro email bị đánh dấu spam nếu không cấu hình DKIM/SPF/DMARC.

Điều này có thể hạn chế khả năng mở rộng quy mô gửi hàng loạt.

(3) Chưa hỗ trợ gửi đa kênh

Hiện tại hệ thống chỉ hỗ trợ gửi hóa đơn qua email, trong khi nhiều đơn vị mong muốn:

Gửi thông báo qua Zalo OA

Gửi mã tra cứu qua SMS

Đồng bộ dữ liệu với hệ thống CRM/ERP

Đây là hướng mở rộng tiềm năng trong tương lai.

(4) Chưa có module thống kê và báo cáo nâng cao

Hệ thống mới chỉ cung cấp log cơ bản. Chưa có:

Biểu đồ thống kê số lượng hóa đơn gửi theo ngày/tháng

Báo cáo tỉ lệ thành công – thất bại

Dashboard phân tích hiệu năng

So sánh hiệu quả trước và sau khi áp dụng tự động hóa

Các chức năng này sẽ giúp hệ thống hoàn thiện hơn trong quản trị tổng thể.