

ĐẠI HỌC QUỐC GIA, THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



TÊN MÔN (MÃ MÔN)

BÁO CÁO XXX

Chủ đề

GVHD:

SV thực hiện:	Student1	MSSV
	SV2	MSSV
	SV3	MSSV
	SV4	MSSV



## Mục lục

<b>1</b>	<b>Tổng quan</b>	<b>2</b>
1.1	Đặt vấn đề . . . . .	2
1.2	Nghiên cứu các giải pháp hiện có . . . . .	3
1.2.1	Nhóm 1: Các ứng dụng tập trung vào Email (Email-centric) . . . . .	3
1.2.2	Nhóm 2: Các nền tảng Quản lý Quan hệ Khách hàng (CRM) đa kênh . . . . .	3
1.2.3	Nhóm 3: Các giải pháp mã nguồn mở (Open-source) . . . . .	3
1.3	Xác định "lỗ hổng" thị trường và đề xuất giải pháp của luận án . . . . .	4
1.4	Mục tiêu và phạm vi luận án . . . . .	4
1.4.1	Mục tiêu . . . . .	4
1.4.2	Phạm vi của PoC . . . . .	5

# 1 Tổng quan

Bối cảnh của cuộc cách mạng công nghiệp lần thứ tư và sự thâm nhập sâu rộng của công nghệ số đã làm thay đổi căn bản phương thức trao đổi thông tin trong môi trường doanh nghiệp và tổ chức. Bên cạnh kênh giao tiếp truyền thống và mang tính chính thức là Email, sự phổ biến ngày càng tăng của các nền tảng nhắn tin tức thời (Instant Messaging - IM) như Zalo, WhatsApp, Telegram hay Viber đã tạo ra một hệ sinh thái giao tiếp đa kênh phức tạp.

Sự đa dạng hóa này, mặc dù mang lại tính tiện lợi và tức thời, nhưng cũng đồng thời làm nảy sinh một thách thức cốt lõi: đó là **\*\*sự phân mảnh nghiêm trọng của dữ liệu liên lạc\*\***. Lịch sử trao đổi với một đối tác, khách hàng hay ứng viên không còn tồn tại ở một nơi duy nhất mà bị phân tán trên nhiều nền tảng khác nhau.

Luận án này tập trung nghiên cứu và đề xuất giải pháp cho vấn đề nêu trên. Mục tiêu chính là trình bày cơ sở lý luận, phân tích thực trạng và đề xuất một mô hình kiến trúc cho một **\*\*Nền tảng Quản lý Trao đổi Tập trung\*\***. Nền tảng này được thiết kế với Email là thành phần trung tâm (Email-centric) và sở hữu kiến trúc module linh hoạt, cho phép tích hợp tùy chọn các kênh giao tiếp khác nhằm giải quyết bài toán phân mảnh dữ liệu một cách hiệu quả.

## 1.1 Đặt vấn đề

Trong các quy trình nghiệp vụ chuyên nghiệp, chẳng hạn như tuyển dụng, quản lý quan hệ khách hàng (CRM), hay hỗ trợ kỹ thuật, Email thường đóng vai trò là kênh khởi đầu, mang tính chính thức. Tuy nhiên, để tăng tốc độ phản hồi và sự thuận tiện, các trao đổi tiếp theo thường được chuyển dịch sang các ứng dụng IM. Chính sự chuyển dịch không đồng nhất này đã dẫn đến nhiều bất cập trong công tác quản lý và vận hành. Những bất cập này có thể được nhận diện cụ thể như sau:

- **Sự phân mảnh dữ liệu (Data Fragmentation):** Đây là thách thức lớn nhất. Lịch sử trao đổi bị phân tán trên nhiều ứng dụng (Email, Zalo, Telegram...) khiến việc tổng hợp một bức tranh toàn cảnh về lịch sử tương tác của một đối tác trở nên khó khăn, tốn thời gian và dễ sai sót.
- **Sự đứt gãy ngữ cảnh (Contextual Discontinuity):** Khi bắt đầu một tương tác mới, người dùng buộc phải truy xuất thông tin từ nhiều nguồn riêng lẻ để nắm bắt lại bối cảnh của cuộc hội thoại. Quá trình này làm giảm hiệu suất và khả năng phản hồi một cách chính xác và kịp thời.
- **Suy giảm hiệu suất và gia tăng rủi ro nghiệp vụ:** Việc chuyển đổi liên tục giữa các ứng dụng không chỉ gây lãng phí thời gian mà còn tiềm ẩn nguy cơ bỏ sót thông tin quan trọng (ví dụ: một yêu cầu khẩn cấp qua tin nhắn). Điều này có thể dẫn đến các quyết định thiếu cơ sở, phản hồi trùng lặp, hoặc mang lại trải nghiệm tiêu cực cho đối tác.
- **Hiện tượng quá tải thông tin (Information Overload):** Ngay cả khi dữ liệu được thu thập thủ công, khối lượng email và tin nhắn đồ sộ trong một luồng lịch sử dài vẫn là một rào cản. Tồn tại nhu cầu cấp thiết về một cơ chế thông minh có khả năng tự động "tiêu hóa" lượng thông tin đó, chắt lọc và trích xuất những thông tin cốt lõi.

Từ những phân tích trên, tính cấp thiết của một giải pháp hợp nhất, thông minh được đặt ra.

## 1.2 Nghiên cứu các giải pháp hiện có

Để làm rõ tính mới và sự cần thiết của đề tài, luận án tiến hành khảo sát và phân tích các giải pháp quản lý liên lạc hiện có trên thị trường. Các giải pháp này có thể được phân loại thành ba nhóm chính, mỗi nhóm có những ưu điểm và hạn chế riêng biệt.

### 1.2.1 Nhóm 1: Các ứng dụng tập trung vào Email (Email-centric)

Nhóm này bao gồm các ứng dụng được thiết kế nhằm mục đích cách mạng hóa trải nghiệm email truyền thống, tập trung vào tốc độ, giao diện người dùng và các tính năng thông minh cơ bản.

- **Các đại diện tiêu biểu:** Superhuman, Front, Missive.
- **Phân tích ưu điểm:** Các giải pháp này cung cấp một trải nghiệm người dùng (UX) vượt trội so với các ứng dụng email tiêu chuẩn. Chúng mạnh về các thao tác nhanh (phím tắt), tính năng cộng tác nội bộ (shared inbox), và gần đây là tích hợp AI ở mức độ cơ bản (ví dụ: tóm tắt email, hỗ trợ soạn thảo).
- **Phân tích hạn chế:** Điểm yếu cốt lõi của nhóm này là sự tập trung gần như tuyệt đối vào Email. Khả năng tích hợp các kênh nhắn tin (IM) khác như WhatsApp, Telegram, Zalo là rất yếu, không phải là cốt lõi thiết kế, hoặc hoàn toàn không có. Điều này khiến chúng không thể giải quyết triệt để bài toán phân mảnh dữ liệu đa kênh.

### 1.2.2 Nhóm 2: Các nền tảng Quản lý Quan hệ Khách hàng (CRM) đa kênh

Đây là các hệ thống phần mềm dịch vụ (SaaS) quy mô lớn, toàn diện, được thiết kế để quản lý mọi điểm chạm của khách hàng, trong đó email và tin nhắn chỉ là một phần.

- **Các đại diện tiêu biểu:** Zendesk, HubSpot Service Hub, Intercom.
- **Phân tích ưu điểm:** Thế mạnh tuyệt đối của nhóm này là khả năng hợp nhất dữ liệu đa kênh mạnh mẽ. Chúng có thể thu thập tương tác từ email, live chat, mạng xã hội, SMS... và hợp nhất vào một hồ sơ khách hàng 360 độ.
- **Phân tích hạn chế:** Các giải pháp này bộc lộ ba nhược điểm lớn: (1) *Sự cồng kềnh*: Chúng là các hệ thống lớn, phức tạp, đòi hỏi thời gian thiết lập và đào tạo đáng kể. (2) *Chi phí cao*: Mô hình định giá thường dựa trên mỗi người dùng (per-seat) và đắt đỏ. (3) *Tính chất "overkill" (vượt quá nhu cầu)*: Đối với các đội nhóm nhỏ hoặc cá nhân chỉ cần quản lý email và một vài kênh IM, các tính năng CRM toàn diện của nhóm này là không cần thiết.

### 1.2.3 Nhóm 3: Các giải pháp mã nguồn mở (Open-source)

Nhóm này bao gồm các nền tảng mà người dùng có thể tự triển khai (self-host), cung cấp khả năng kiểm soát hoàn toàn dữ liệu và tùy biến.

- **Các đại diện tiêu biểu:** Chatwoot, Rocket.Chat.
- **Phân tích ưu điểm:** Lợi thế lớn nhất là *quyền riêng tư và kiểm soát dữ liệu* (data sovereignty), do dữ liệu được lưu trữ nội bộ, không phụ thuộc vào bên thứ ba. Thêm vào đó là khả năng tùy biến cao và chi phí bản quyền bằng không.
- **Phân tích hạn chế:** Các giải pháp này đòi hỏi năng lực kỹ thuật cao để triển khai, bảo trì và nâng cấp. Mặc dù có khả năng tích hợp đa kênh, trải nghiệm người dùng với Email thường không được tối ưu và mượt mà như các giải pháp ở Nhóm 1.

### 1.3 Xác định "lỗ hổng" thị trường và đề xuất giải pháp của luận án

Phân tích ba nhóm giải pháp trên cho thấy một "lỗ hổng" thị trường rõ rệt:

- Người dùng cần một giải pháp có **trải nghiệm Email xuất sắc** (như Nhóm 1).
- Nhưng giải pháp đó phải có **khả năng mở rộng đa kênh linh hoạt** (như Nhóm 2), mà không bị **cồng kênh và đắt đỏ**.
- Đồng thời, giải pháp cần đảm bảo **quyền kiểm soát dữ liệu** và khả năng **tích hợp AI sâu** (vượt ra ngoài tóm tắt đơn thuần, tiến tới gợi ý ngữ cảnh, trích xuất vấn đề).

Luận án này đề xuất một hướng giải quyết nhằm lấp đầy "lỗ hổng" đó thông qua việc **thiết kế và hiện thực một hệ thống Chứng minh Khái niệm (Proof of Concept - PoC)** với các đặc điểm kiến trúc sau:

1. **Kiến trúc lấy Email làm lõi (Email-centric Core):** Tập trung giải quyết tốt nhất bài toán Email trước tiên. Thay vì xem email là "một trong nhiều kênh", chúng ta xem Email là kênh bắt buộc, là xương sống của giao tiếp nghiệp vụ.
2. **Thiết kế Module và Microservice:** Hệ thống được thiết kế theo kiến trúc monorepo nhưng với các thành phần dịch vụ độc lập (backend, frontend, AI-service). Cách tiếp cận này cho phép hệ thống (1) linh hoạt: dễ dàng phát triển, nâng cấp từng phần; và (2) mở rộng: sẵn sàng các "adapter" để kết nối với các kênh IM khác (Zalo, Telegram...) trong tương lai mà không ảnh...
3. **Tích hợp AI như một dịch vụ độc lập (AI as a Service):** Dịch vụ AI (phát triển bằng FastAPI) được tách rời khỏi logic nghiệp vụ của backend. Điều này cho phép:
  - Dễ dàng chuyển đổi giữa các nhà cung cấp AI (ví dụ: từ OpenAI sang AI local) mà không cần thay đổi mã nguồn backend.
  - Cho phép AI thực hiện các tác vụ phức tạp (như tóm tắt, gợi ý, và xa hơn là AI agent) một cách độc lập và bất đồng bộ (async).
4. **Khả năng tự triển khai (Self-hostable):** Toàn bộ hệ thống được đóng gói bằng Docker Compose, cho phép người dùng toàn quyền kiểm soát dữ liệu của mình (giải quyết vấn đề của Nhóm 1 và 2) mà không quá phức tạp về kỹ thuật (như Nhóm 3).

PoC được xây dựng trong luận án này sẽ tập trung vào việc hiện thực hóa kiến trúc trên. Nó sẽ giải quyết các vấn đề đã đặt ra bằng cách: (1) Đồng bộ và hợp nhất Email vào một giao diện timeline; (2) Tự động tóm tắt và trích xuất vấn đề từ các luồng email dài (giải quyết "quá tải thông tin"); và (3) Cung cấp kiến trúc nền tảng sẵn sàng cho việc mở rộng đa kênh sau này.

### 1.4 Mục tiêu và phạm vi luận án

#### 1.4.1 Mục tiêu

Luận án đặt ra các mục tiêu kỹ thuật cụ thể sau:

- Phân tích và thiết kế kiến trúc hệ thống quản lý giao tiếp tập trung theo mô hình microservice linh hoạt.
- Hiện thực PoC với các module chính: Backend (Next.js), Frontend (Next.js/React) và AI Service (FastAPI).

- Xây dựng cơ chế đồng bộ hóa email và hợp nhất dữ liệu liên lạc vào cơ sở dữ liệu MongoDB.
- Phát triển chức năng AI (sử dụng OpenAI API làm giải pháp ban đầu) để tóm tắt luồng email và gợi ý phản hồi.
- Thử nghiệm và đánh giá PoC thông qua các kịch bản dữ liệu giả lập.

#### 1.4.2 Phạm vi của PoC

Để đảm bảo tính khả thi trong khuôn khổ thời gian của luận án, PoC sẽ tập trung vào các phạm vi sau:

- **Trong phạm vi (In-scope):**
  - Hệ thống chỉ tập trung vào tích hợp và xử lý kênh **\*\*Email\*\***.
  - Chức năng AI sử dụng API của bên thứ ba (OpenAI) để xử lý tóm tắt.
  - Giao diện người dùng (Frontend) tập trung vào việc hiển thị timeline trao đổi và kết quả tóm tắt của AI.
  - Thử nghiệm trên dữ liệu giả lập (synthesized data).
- **Ngoài phạm vi (Out-of-scope):**
  - Chưa tích hợp trực tiếp với các API của kênh nhắn tin (Zalo, WhatsApp, Telegram...). Luận án chỉ dừng ở mức thiết kế kiến trúc sẵn sàng cho việc này.
  - Chưa triển khai các mô hình AI local (như Ollama).
  - Chưa phát triển các tính năng nâng cao như AI agent hay quản lý xác thực người dùng (authentication).