

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



Semester: 241

Programming Intergration Project

(CO3015)

Advisor: Phan Trọng Nhàn

Students: Nguyễn Hoàng Hữu Nghiên - 2111949
Nguyễn Tiến Phát - 2114381
Đặng Minh Nhật - 2212387
Vũ Trần Duy Nguyên - 2212338
Trần Mạnh Tuấn - 2213807



Contents

1	Danh sách thành viên	3
2	Nội dung báo cáo	3
2.1	Tìm hiểu đề tài và dữ liệu	3
2.1.1	Tìm hiểu đề tài	3
2.1.2	Tìm hiểu dữ liệu	4
2.2	Xác định yêu cầu đề tài	5
2.3	Thiết kế hệ cơ sở dữ liệu	5
2.3.1	Hệ cơ sở dữ liệu production	5
2.3.2	Hệ cơ sở dữ liệu analysis	10
2.3.2.a	Lý thuyết	10
2.3.2.b	Thiết kế	11
2.4	Hiện thực ứng dụng	12
2.4.1	Yêu cầu ứng dụng	12
2.4.1.a	Functional requirement	12
2.4.1.b	Non-functional requirement	12
2.4.2	Architecture và Repositories	13
2.4.3	Usecase diagram	13
2.4.3.a	Xác thực người dùng (User authentication)	14
2.4.3.b	Các thao tác trên hệ cơ sở dữ liệu (CRUD)	16
2.4.3.c	Truy vấn và phân tích dữ liệu	19
2.4.4	Activity diagram	20
2.4.5	Sequence diagram	25
2.4.6	Class diagram	31
2.4.7	Component diagram	32
2.4.8	Demo	37
2.4.8.a	Đăng nhập vào hệ thống	37
2.4.8.b	Các tính năng CRUD	38
2.4.8.c	Dashboard	40
2.4.8.d	Tính năng phân tích	40
2.5	Trích xuất, chuyển đổi, tải (ETL)	41
2.5.1	Giới thiệu	41



2.5.2	Các kỹ thuật ETL từ cơ bản đến nâng cao	42
2.5.3	Intial Load ETL	43
2.5.4	Streaming ETL	49
2.6	Business Intelligence Dashboard	52
2.6.1	Giới thiệu	52
2.6.2	Các nội dung hiển thị	52
2.6.3	Thiết kế dashboard	53
2.7	Phân tích dữ liệu	55
2.7.1	Giới thiệu	55
2.7.2	Thiết kế tính năng phân tích dữ liệu	56
2.7.2.a	Thiết kế câu trả lời	56
2.7.2.b	Quy trình phân tích và truy vấn dữ liệu	57
2.7.3	Demo	60
2.7.3.a	Chat completion	60
2.7.3.b	Truy vấn dữ liệu	61
2.7.3.c	Phân tích dữ liệu	62
2.7.4	Các hạn chế và cách khắc phục	62



1 Danh sách thành viên

No.	Fullname	Student ID	Problems	Percentage of work
1	Nguyễn Hoàng Hữu Nhiên	2111949	- Backend - Viết báo cáo	100%
2	Nguyễn Tiến Phát	2114381	- Backend - Viết báo cáo	100%
3	Đặng Minh Nhật	2212387	- Frontend - Viết báo cáo	100%
4	Vũ Trần Duy Nguyên	2212338	- Database - Tạo slide thuyết trình	100%
5	Trần Mạnh Tuấn	2213807	- Frontend - Viết báo cáo	100%

2 Nội dung báo cáo

2.1 Tìm hiểu đề tài và dữ liệu

2.1.1 Tìm hiểu đề tài

Trong hầu hết mọi doanh nghiệp, sales là hoạt động đóng vai trò quan trọng trong việc thúc đẩy doanh số và lợi nhuận trực tiếp cho công ty. Cụ thể, sales là quá trình tiếp thị và bán hàng để đạt được doanh số, lợi nhuận cho một doanh nghiệp. Bao gồm các hoạt động như tìm kiếm, gặp gỡ, tiếp cận, thuyết phục và ký kết hợp đồng với khách hàng để bán sản phẩm hoặc dịch vụ. Công việc của một nhân viên sales thường bao gồm nhiều hoạt động khác nhau nhằm tiếp cận khách hàng, tạo mối quan hệ và thực hiện quá trình bán hàng. Một số công việc cơ bản của một nhân viên sales bao gồm:

- Tìm kiếm khách hàng tiềm năng:** Sử dụng các công cụ tìm kiếm, khai thác cơ sở dữ liệu để tiếp cận với những khách hàng có khả năng mua sản phẩm/dịch vụ của doanh nghiệp.
- Xây dựng mối quan hệ với khách hàng:** Gặp gỡ khách hàng, lắng nghe và thấu hiểu nhu cầu của họ. Bằng cách tương tác chuyên nghiệp và tạo lòng tin, nhân viên sales cần xây dựng mối quan hệ đáng tin cậy và bền vững với khách hàng.
- Thăm dò nhu cầu và tư vấn:** Đặt câu hỏi, lắng nghe và hiểu rõ về những vấn đề cũng như mong muốn của khách hàng. Dựa trên thông tin đó, tư vấn các giải pháp phù hợp để đáp ứng nhu cầu của khách hàng thông qua sản phẩm hoặc dịch vụ được cung cấp bởi doanh nghiệp.
- Quản lý quá trình bán hàng:** Nhân viên sales quản lý quá trình bán hàng từ việc tạo báo giá, xử lý đơn hàng, theo dõi tiến trình sản xuất và giao hàng. Đảm bảo các yêu cầu của khách hàng được đáp ứng đúng thời hạn và chất lượng.

Business Intelligence (BI) hoặc còn được gọi là trí tuệ kinh doanh là quá trình đưa các thông tin cần thiết đến đúng người và đúng thời điểm giúp hỗ trợ đưa ra các quyết định đúng



dẫn. Hệ thống Business Intelligence là một tập hợp các quy trình, kiến trúc và công nghệ lưu trữ và chuyển đổi dữ liệu thô thành thông tin có ý nghĩa. Một hệ thống trí tuệ kinh doanh có thể giúp trong lĩnh vực sales như sau:

- **Thống nhất dữ liệu:** Hệ thống BI sẽ kéo các dữ liệu từ nhiều nguồn liên quan đến việc kinh doanh của công ty, chuyển đổi dữ liệu thô và lưu trong kho dữ liệu để có thể dùng trong các quá trình phân tích sau đó.
- **Nhận diện các điểm nổi bật:** Quá trình phân tích của hệ thống trí tuệ kinh doanh sẽ giúp nhân viên sales đánh giá các hoạt động và quyết định của công ty: chiến dịch quảng bá, chiến dịch giảm giá,... Để từ đó, các nhu cầu của khách hàng hoặc xu hướng tiêu dùng có thể được tận dụng tốt hơn trong tương lai.
- **Tạo các báo cáo:** Hệ thống BI còn có các công cụ giúp người dùng tạo các báo cáo và mô tả dữ liệu thông qua các biểu đồ giúp nhận diện khuynh hướng của thị trường hoặc làm tài liệu để trình bày các ý tưởng của mình.

Qua đó, ta có thể thấy, một hệ thống trí tuệ kinh doanh là giải pháp để giải quyết các vấn đề về thông tin trong lĩnh vực sales và là cơ sở để các doanh nghiệp lớn có thể quản lý và tận dụng triệt để nguồn thông tin hiện có để phát triển nhanh chóng và bền vững.

2.1.2 Tìm hiểu dữ liệu

File dữ liệu CompanyX.bak chứa dữ liệu tổng quát trong một công ty sản xuất và phân phối nhiều loại mặt hàng. File dữ liệu được chia làm nhiều phần như sau:

- **Human Resources:** Chứa các thông tin về nguồn nhân lực trong tổ chức. Cụ thể là các thông tin về các bộ phận trong tổ chức, nhân viên trong tổ chức, lương của nhân viên và các ca làm việc của nhân viên.
- **Person:** Bao gồm các thông tin về từng thành viên trong tổ chức hoặc có liên quan. Các thông tin bao gồm: địa chỉ vật lý, địa chỉ email, số điện thoại, nơi ở,...
- **Production:** Bao gồm các thông tin về hàng hóa và các hoạt động sản xuất trong công ty.
- **Purchasing:** Chứa các thông tin liên quan đến việc mua hàng hóa của công ty như: các đơn nguyên liệu đặt mua bởi công ty, các nguồn bán nguyên liệu cho công ty,...
- **Sales:** Chứa các thông tin liên quan đến hoạt động sales của công ty như: tỉ giá hối đoái, thông tin về khách hàng (các cửa hàng hoặc các cá nhân), những nơi tập trung khách hàng của công ty và các khuyến mãi đặc biệt được áp dụng cho hàng hóa được bán.

Ngoài ra, qua việc tìm hiểu từ các nguồn, các dữ liệu cần thiết trong việc sales bao gồm:

- **Nhân viên:** Thông tin về nhân viên trong bộ phận tiếp thị và bán hàng của công ty.
- **Sản phẩm:** Thông tin về các sản phẩm hiện đang được sản xuất và cung cấp bởi công ty để phục vụ những hoạt động như: tìm kiếm khách hàng tiềm năng, quảng cáo sản phẩm trên thị trường và giới thiệu hoặc gợi ý các sản phẩm thích hợp cho khách hàng của công ty.



- **Khuyến mãi:** Thông tin về các khuyến mãi đặc biệt và các sản phẩm nào đang được áp dụng các khuyến mãi đó để hỗ trợ việc quảng cáo và lôi kéo người tiêu dùng.
- **Khách hàng:** Thông tin về khách hàng, bao gồm cửa hàng hoặc người đại diện. Về người đại diện, các thông tin sẽ bao gồm các mục cơ bản như: tên, tuổi, số điện thoại, địa chỉ liên lạc, email, thành phố. Về cửa hàng, các thông tin sẽ gồm có: tên cửa hàng, thu nhập hàng năm, loại hình kinh doanh, năm bắt đầu, diện tích, số lượng nhân viên,... Ngoài ra, cần phải theo dõi vùng địa lý của khách hàng để phân loại.
- **Đơn hàng:** Thông tin về hoạt động đặt hàng của khách hàng. Các trường của đơn hàng gồm có: thời gian đặt hàng, thời gian giao hàng, loại hình giao hàng, thuế, tổng tiền hàng, tổng tiền giao hàng, tổng tiền của đơn hàng và bình luận về đơn hàng. Về danh mục này, các thông tin cần thiết khác về các sản phẩm khác nhau, số lượng của từng loại sản phẩm trong đơn hàng và khuyến mãi được áp dụng cũng cần được xem xét.

2.2 Xác định yêu cầu đề tài

Qua hướng dẫn của giáo viên và tìm hiểu, nhóm rút ra được những yêu cầu chính của đề tài đồ án như sau:

- **Production database:** Tìm hiểu, chọn lọc các dữ liệu cần thiết từ file dữ liệu và xây dựng hệ cơ sở dữ liệu vận hành (production database) dựa trên đề tài sales cho hệ thống.
- **Analysis database:** Xây dựng hệ cơ sở dữ liệu phân tích (analysis database) sử dụng các mô hình data warehouse.
- **Extract, Transform and Load:** Xây dựng pipeline ETL di chuyển dữ liệu vận hành sang hệ cơ sở dữ liệu phân tích.
- **Ứng dụng CRUD:** Hiện thực ứng dụng có các tính năng CRUD cho hệ thống.
- **Dashboard:** Xây dựng dashboard hiển thị thông tin. Dashboard phải có các biểu đồ, thông tin được trình bày rõ ràng, dễ hiểu và liên quan đến usecase của hệ thống.
- **Phân tích:** Áp dụng mô hình phân tích để rút ra các đánh giá hoặc dự đoán trên các thông tin hay dữ liệu hiện có.

2.3 Thiết kế hệ cơ sở dữ liệu

2.3.1 Hệ cơ sở dữ liệu production

Cơ sở dữ liệu được thiết kế với các bảng và thuộc tính cụ thể để hỗ trợ quản lý và tối ưu hóa hoạt động bán hàng, đồng thời cung cấp thông tin chi tiết để hỗ trợ ra quyết định kinh doanh.

Bảng **Customer** giúp quản lý thông tin khách hàng, gồm hai loại: cửa hàng và cá nhân.

Customer Store

- StoreID: Định danh duy nhất cho từng cửa hàng để phân biệt dữ liệu.



- Name: Tên cửa hàng giúp dễ dàng nhận diện.
- AnnualRevenue: Doanh thu hàng năm dùng để phân tích khả năng sinh lời của cửa hàng.
- BusinessType: Loại hình kinh doanh để phân loại cửa hàng (bán lẻ, bán buôn, dịch vụ,...).
- YearOpened: Năm thành lập để theo dõi tuổi đời và xu hướng kinh doanh.
- AddressLine1, AddressLine2: Địa chỉ giúp quản lý vị trí của cửa hàng.
- AnnualSales: Dữ liệu về doanh số hàng năm để phân tích hiệu quả bán hàng.
- CountryRegionName: Quốc gia hoặc khu vực, giúp phân tích doanh số theo vùng.
- Specialty: Chuyên môn của cửa hàng (ví dụ: xe đạp thể thao, xe đạp địa hình,...).
- SquareFeet: Diện tích cửa hàng để đánh giá quy mô.
- NumberOfEmployee: Số lượng nhân viên giúp đo lường hiệu quả quản lý nhân sự.
- City: Thành phố để phân tích theo địa phương.

Customer Individual

- IndividualID: Định danh duy nhất của từng khách hàng cá nhân.
- FirstName, MiddleName, LastName: Tên khách hàng để giao dịch và cá nhân hóa dịch vụ.
- Title: Danh xưng (Mr., Ms., Dr.) giúp thể hiện sự chuyên nghiệp.
- EmailAddress, PhoneNumber: Thông tin liên lạc để hỗ trợ khách hàng và quảng bá.
- CountryRegionName, AddressLine1, AddressLine2: Địa chỉ giúp quản lý vị trí khách hàng.

Bảng **Employee** tập trung vào quản lý nhân sự

- EmployeeID: Định danh duy nhất của nhân viên để quản lý.
- Name: Tên nhân viên để nhận diện.
- JobTitle: Chức danh để phân chia trách nhiệm.
- PhoneNumber, EmailAddress: Liên lạc nội bộ hoặc với khách hàng.
- AddressLine1, AddressLine2, CountryRegionName: Thông tin địa chỉ để quản lý nhân viên.
- PasswordHash: Dùng để bảo vệ tài khoản và quản lý bảo mật.
- City: Địa điểm hoạt động để phân tích nguồn nhân lực theo khu vực.

Bảng **Territory** cung cấp thông tin về khu vực bán hàng

- TerritoryID: Định danh khu vực bán hàng để quản lý vùng lãnh thổ.
- Name: Tên khu vực để phân biệt.



- Group: Nhóm khu vực (ví dụ: miền Bắc, miền Nam) giúp phân tích doanh số theo cụm.
- SalesYTD: Doanh số năm nay để theo dõi hiệu quả bán hàng.
- SalesLastYear: Doanh số năm trước để so sánh và phân tích xu hướng.

Bảng **SaleOrderHeader** quản lý thông tin về các đơn hàng

- SalesOrderID: Định danh đơn hàng để quản lý.
- OrderDate, DueDate, ShipDate: Theo dõi tiến trình đơn hàng từ lúc đặt đến khi giao.
- Freight: Chi phí vận chuyển để tính tổng chi phí.
- Subtotal, TaxAmt, TotalDue: Tính toán tài chính của đơn hàng.
- Comment: Lưu ý hoặc phản hồi từ khách hàng.
- ShipMethod: Phương thức vận chuyển để tối ưu hóa logistics.
- SalesReason: Lý do bán hàng (quảng cáo, khuyến mãi) để đo lường hiệu quả marketing.

Bảng **SaleOrderDetail** quản lý thông tin chi tiết cho từng đơn hàng

- SaleOrderDetailID: Định danh chi tiết đơn hàng.
- OrderCity: Thành phố nơi đặt hàng để phân tích địa phương.
- UnitPriceDiscount: Chiết khấu đơn giá để theo dõi ưu đãi.
- UnitPrice, LineTotal: Giá và tổng giá trị để tính doanh thu.

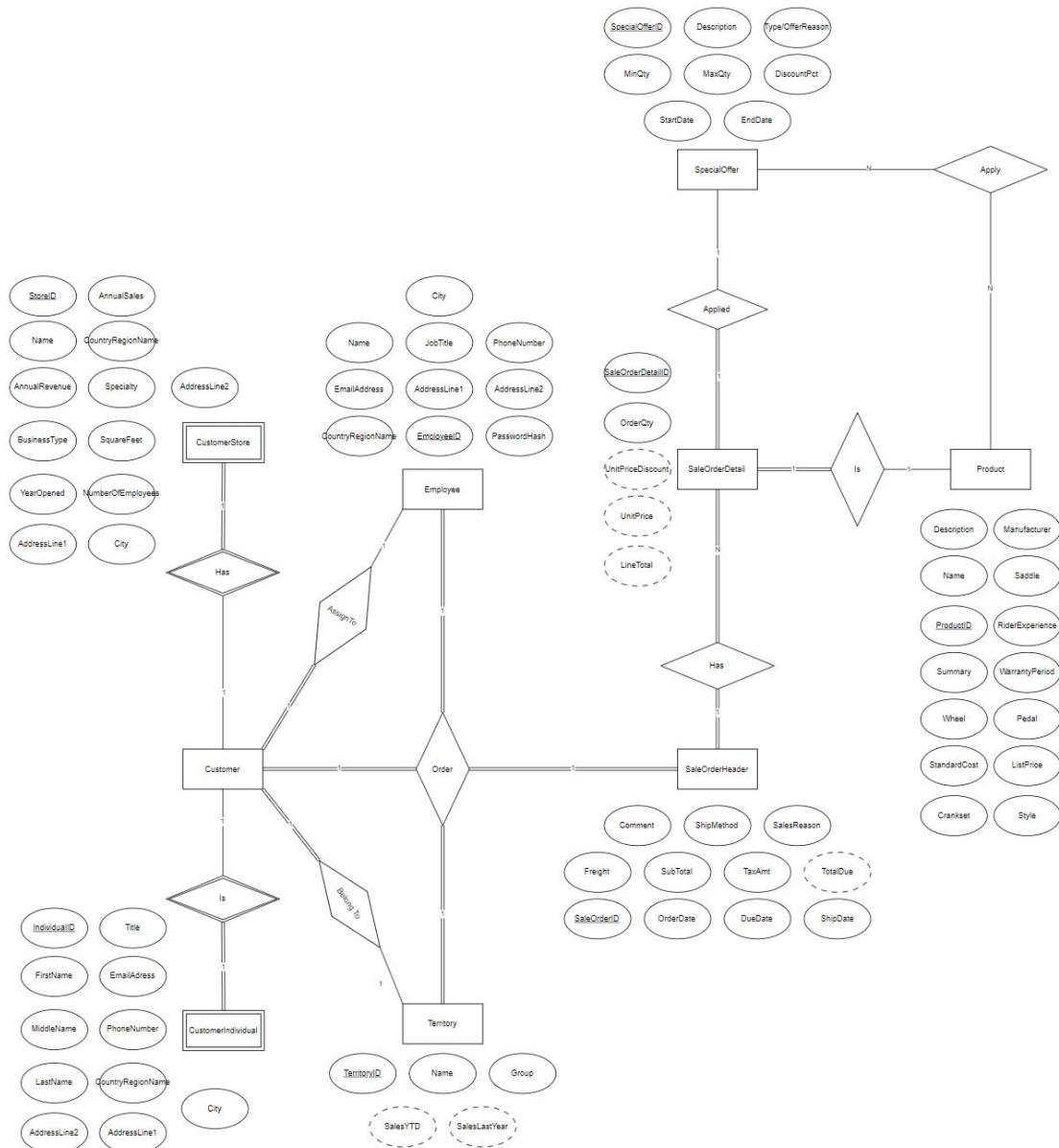
Bảng **SpecialOffer** lưu trữ thông tin về các chương trình khuyến mãi

- SpecialOfferID: Định danh chương trình khuyến mãi.
- Description: Mô tả chi tiết về chương trình.
- TypeOfferReason: Loại khuyến mãi và lý do để phân loại chiến lược marketing.
- MinQty, MaxQty: Số lượng tối thiểu và tối đa để kích thích tiêu thụ.
- DiscountPct: Tỷ lệ chiết khấu để đo lường mức độ ưu đãi.
- StartDate, EndDate: Khoảng thời gian áp dụng để quản lý chiến dịch.

Cuối cùng, bảng **Product** quản lý thông tin sản phẩm

- ProductID: Định danh sản phẩm.
- Description: Mô tả để khách hàng hiểu rõ sản phẩm.
- Manufacturer: Nhà sản xuất để theo dõi nguồn cung ứng.
- Name: Tên sản phẩm để nhận diện.

- Saddle, RiderExperience: Các đặc điểm chuyên biệt để đáp ứng nhu cầu khách hàng.
- Summary: Tóm tắt thông tin sản phẩm để dễ dàng quảng bá.
- WarrantyPeriod: Thời gian bảo hành để xây dựng niềm tin.
- Wheel, Pedal, Crankset, Style: Các đặc điểm chi tiết giúp phân loại và tùy chỉnh sản phẩm.
- StandardCost, ListPrice: Giá vốn và giá bán để quản lý lợi nhuận.



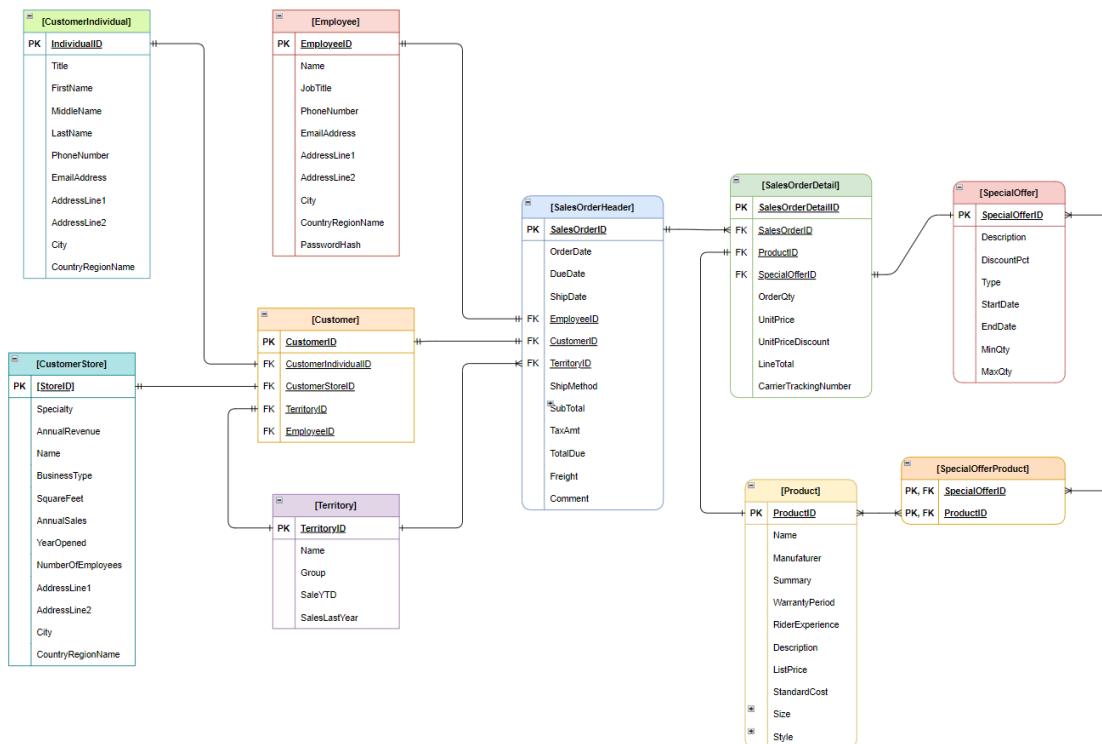
Hình 1: Enhanced Entity Relationship Diagram (EERD)



Xem chi tiết [tại đây](#) Mô hình EERD bao gồm:

- **Customer Store:** StoreID, Name, AnnualRevenue, BusinessType, YearOpened, AddressLine1, AddressLine2, AnnualSales, CountryRegionName, Specialty, SquareFeet, NumberOfEmployee, City
- **Customer Individual:** IndividualID, FirstName, MiddleName, LastName, Title, EmailAddress, PhoneNumber, CountryRegionName, AddressLine1, AddressLine2
- **Employee:** EmployeeID, Name, JobTitle, PhoneNumber, EmailAddress, AddressLine1, AddressLine2, CountryRegionName, PasswordHash, City
- **Territory:** TerritoryID, Name, Group, SalesYTD, SalesLastYear
- **SaleOrderHeader:** SalesOrderID, OrderDate, DueDate, ShipDate, Freight, Subtotal, TaxAmt, TotalDue, Comment, Shipmethod, SalesReason
- **SaleOrderDetail:** SaleOrderDetailID, OrderCity, UnitPriceDiscount, UnitPrice, LineTotal
- **SpecialOffer:** SpecialOfferID, Description, TypeOfferResson, MinQty, MaxQty, DiscountPct, StartDate, EndDate
- **Product:** ProductID, Description, Manufacturer, Name, Saddle, RiderExperience, Summary, WarrantyPeriod, Wheel, Pedal, StandardCost, ListPrice, Crankset, Style

Từ mô hình EERD, nhóm em đã thực hiện thiết kế sơ đồ ánh xạ cơ sở dữ liệu, với PK là khóa chính và FK là khóa ngoại được liên kết từ khóa chính của một thực thể khác.



Hình 2: Relational Mapping

Xem chi tiết [tại đây](#).

2.3.2 Hệ cơ sở dữ liệu analysis

2.3.2.a Lý thuyết

Data warehouse hay còn gọi là kho dữ liệu là hệ thống được thiết kế để lưu trữ và quản lý dữ liệu từ nhiều nguồn khác nhau. Kho dữ liệu này cung cấp một góc nhìn toàn diện cho các lĩnh vực của doanh nghiệp. Ngoài ra, data warehouse còn cho phép người dùng truy vấn dữ liệu dễ dàng và hiệu quả để thực hiện các thao tác tổng hợp hoặc phân tích trên kết quả truy vấn đó. Data warehouse có các đặc điểm chính như sau:

- Có chủ đề nhất định:** Mô hình của kho dữ liệu được thiết kế dựa vào nhu cầu hoặc lĩnh vực cụ thể của doanh nghiệp. Data warehouse sẽ cung cấp các thông tin liên quan đến chủ đề này để thực hiện các quá trình khác.
- Được tích hợp:** Dữ liệu trong data warehouse được tổng hợp từ nhiều nguồn khác nhau. Một quy trình trích xuất, chuyển đổi và tải được áp dụng để cung cấp dữ liệu cho kho dữ liệu.
- Bất biến:** Cấu trúc của kho dữ liệu tuân theo một số các quy luật và mô hình nhất định để đảm bảo hiệu suất truy vấn và tính mạch lạc của dữ liệu. Cấu trúc của data warehouse

không được thay đổi thường xuyên. Quá trình tập hợp và thao tác trên data warehouse phải đảm bảo các thuộc tính ACID.

Các loại lược đồ thông dụng của kho dữ liệu bao gồm:

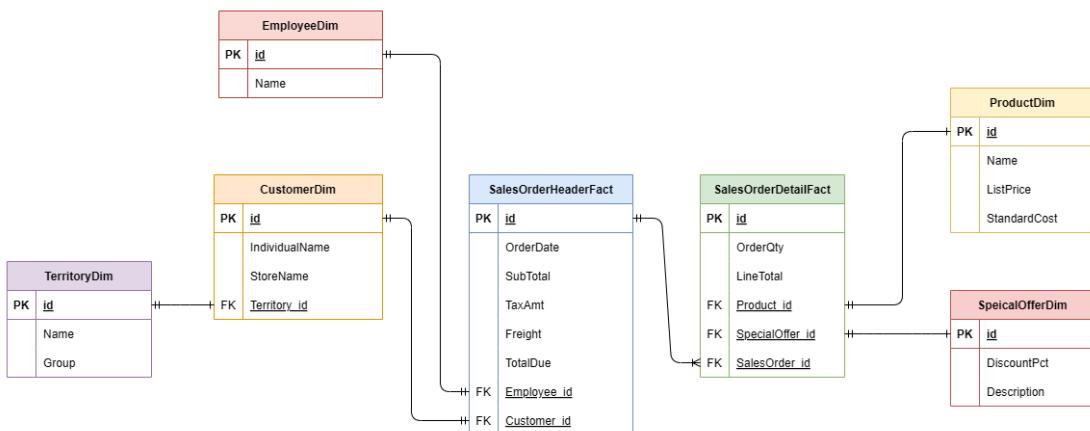
- **Star Schema (hình sao):** Gồm 1 bảng Fact (bảng sự kiện) nằm ở trung tâm và được bao quanh bởi những bảng Dimension (bảng chiều). Dữ liệu của lược đồ hình sao không được chuẩn hóa. Các câu hỏi nhắm vào bảng Fact và được cấu trúc bởi các bảng Dimension.
- **Snow Flade Schema (hình bông tuyết):** Là dạng mở rộng của lược đồ hình sao bằng các bổ sung các Dim. Bảng Fact như lược đồ hình sao, bảng Dim được chuẩn hóa. Các chiều được cấu trúc rõ ràng. Bảng Dim được chia thành chiều chính hay chiều phụ.
- **Galaxy Schema (hình thiên hà):** Chứa nhiều bảng Fact sử dụng chung một số bảng Dim. Lược đồ là sự kết hợp của nhiều data mart.

2.3.2.b Thiết kế

Sau nghiên cứu và cân nhắc kỹ lưỡng, nhóm đã chọn lược đồ hình sao (star schema) để thiết kế hệ cơ sở dữ liệu phân tích (data warehouse) cho hệ thống vì các lý do sau:

- **Dễ hiểu:** Lược đồ hình sao có cấu trúc các bảng (table) dễ hiểu với một hay nhiều bảng dữ kiện (fact table) và nhiều bảng chiều của dữ kiện xung quanh (dimension table)
- **Dễ thao tác:** Nhờ cấu trúc đơn giản của star schema, các câu truy vấn dữ liệu trên hệ cơ sở dữ liệu cũng đơn giản và dễ hiểu. Điều này hạn chế việc sử dụng các câu truy vấn dữ liệu JOIN phức tạp. Ngoài ra, các thao tác truy vấn như lọc (filtering), lấy các tập con (dicing) và tổng hợp dữ liệu (aggregating), lý tưởng cho các hệ thống trí tuệ kinh doanh.
- **Dễ mở rộng:** Lược đồ hình sao có thể được mở rộng để thêm các bảng dữ liệu hay chiều của dữ kiện mà không làm phức tạp hóa lược đồ tổng quát.

Data warehouse của hệ thống sẽ được thiết kế theo lược đồ hình sao như sau:



Hình 3: Data Warehouse Schema



Xem chi tiết [tại đây](#).

Trong hệ cơ sở dữ liệu production, hai bảng SalesOrderHeader và SalesOrderDetail chứa các thông tin quan trọng như: ngày đặt hàng của đơn hàng, các chi phí trong một đơn hàng, khách hàng của đơn hàng, vùng, các sản phẩm được đặt trong đơn hàng và các khuyến mãi được áp dụng. Qua đó, ta có thể thấy hai bảng này diễn tả hoạt động kinh doanh cốt lõi của doanh nghiệp cần được theo dõi và phân tích. Chính vì thế, hai bảng SalesOrderHeader và SalesOrderDetail được chọn làm hai bảng dữ kiện (fact table) cho lược đồ của data warehouse: SalesOrderHeaderFact và SalesOrderDetailFact tương ứng. Sau đó, các bảng được chọn làm bảng chiều dữ liệu trong hệ cơ sở dữ liệu production là Territory, CustomerIndividual, CustomerStore, Customer, Employee, Product và SpecialOffer. Ngoài ra, các bảng trong kho dữ liệu chỉ bao gồm các trường quan trọng đã được chọn lọc từ các bảng của hệ cơ sở dữ liệu production.

2.4 Hiện thực ứng dụng

2.4.1 Yêu cầu ứng dụng

2.4.1.a Functional requirement

Các functional requirement của ứng dụng gồm có:

- **Xác thực người dùng:** Người dùng trong tổ chức có thể thực hiện các thao tác xác nhận danh tính (authentication) như: đăng ký, xác thực email, đổi mật khẩu, đăng nhập vào ứng dụng.
- **CRUD:** Có các tính năng CRUD để xem, thêm, sửa và xóa các record trong hệ cơ sở dữ liệu của hệ thống.
- **Hệ cơ sở dữ liệu và ETL:** Có kết nối đến hệ cơ sở dữ liệu của hệ thống. Có quy trình ETL được áp dụng để di chuyển dữ liệu từ hệ cơ sở dữ liệu vận hành sang hệ cơ sở dữ liệu phân tích.
- **Dashboard:** Có dashboard với giao diện trực quan, dễ hiểu, hiển thị các thông tin cần thiết về các hoạt động sales của tổ chức.
- **Phân tích:** Có tính năng phân tích dữ liệu hiện có và rút ra các thông tin mới từ dữ liệu. Hệ thống sẽ đưa ra cho người dùng các báo cáo thích hợp với nhu cầu phân tích.

2.4.1.b Non-functional requirement

Các non-functional requirement của ứng dụng bao gồm:

- **Độ trễ (Latency):** Hệ thống phải đáp ứng được lượng lớn người dùng truy cập và lượng lớn các thao tác thực hiện bởi người dùng. Độ trễ cho các thao tác trên hệ thống tương đối thấp (5-10s). Độ trễ cho các thao tác trên hệ cơ sở dữ liệu như CRUD và ETL phải nằm trong khoảng từ 5 đến 10 giây.
- **Có sẵn (Availability):** Hệ thống phải hoạt động 24/7 để người dùng có thể sử dụng bất cứ lúc nào.

- **Giao diện (Interface):** Giao diện đơn giản và dễ hiểu giúp người dùng dễ làm quen và sử dụng. Người dùng mới có thể sử dụng hệ thống sau 10 phút.
- **Bảo mật (Security):** Thông tin cá nhân của người dùng và thông tin liên quan đến quá trình kinh doanh trong hệ cơ sở dữ liệu phải được bảo mật.
- **Bảo trì và phát triển (Maintenance and Expandability):** Hệ thống được thiết kế để dễ dàng bảo trì và thêm các tính năng mới.

2.4.2 Architecture và Repositories

Sau khi thảo luận và cân nhắc, nhóm đã quyết định xây dựng ứng dụng web với các phần mềm, dịch vụ và công nghệ sau

- **Next.js:** Frontend của hệ thống (website) sẽ được xây dựng sử dụng framework Next.js vì tính dễ sử dụng và hiệu suất.
- **Django:** Django sẽ được sử dụng để hiện thực backend của hệ thống. Django là framework giúp phát triển hệ thống nhanh và dễ dàng với các công cụ tích hợp các công nghệ khác được cung cấp sẵn. Ngoài ra, Python là ngôn ngữ được framework Django sử dụng và cũng là ngôn ngữ với nhiều thư viện thao tác và phân tích trên dữ liệu.
- **Aiven:** Aiven là dịch vụ triển khai và host hệ cơ sở dữ liệu trên đám mây. Aiven được sử dụng để host hệ cơ sở dữ liệu Postgres (production và analysis) của hệ thống.
- **Google AI Studio:** Nền tảng phát triển trí tuệ nhân tạo của Google cung cấp các mô hình AI để người dùng có thể triển khai trên ứng dụng, hệ thống của mình. API của Google AI Studio được sử dụng trong hệ thống để giúp tạo và thực hiện các truy vấn hoặc phân tích theo yêu cầu của người dùng. Ngoài ra, API còn giúp tổng quát thông tin và tạo các báo cáo cho người dùng.

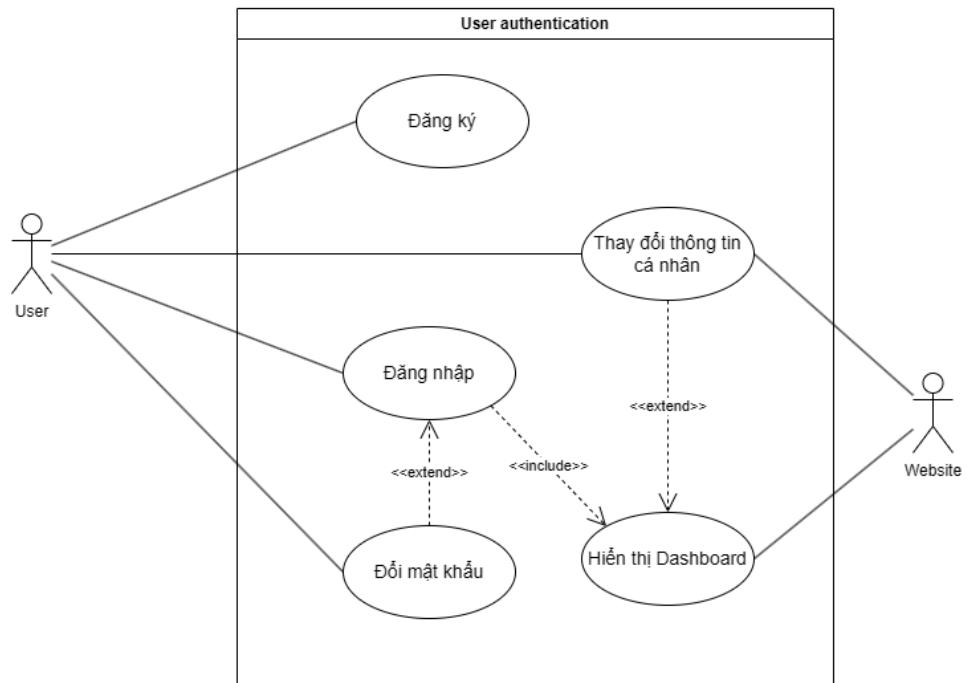
Nhóm đã sử dụng nền tảng Github để quản lý mã nguồn dự án. Cụ thể, frontend và backend của dự án được lưu trong 2 repository sau:

- **Frontend:** https://github.com/MinzNhat/DA_HTTT
- **Backend:** https://github.com/Marky303/DA_HTTT_E

2.4.3 Usecase diagram

Xem chi tiết [tại đây](#).

2.4.3.a Xác thực người dùng (User authentication)



Hình 4: Usecase Diagram for user authentication

Use Case	Đăng ký
Đối tượng	User
Tiền điều kiện	User đã truy cập vào trang web
Hậu điều kiện	User tạo được tài khoản mới để sử dụng dịch vụ
Luồng chính	<ol style="list-style-type: none"> Người dùng chọn nút "Đăng ký tài khoản" Người dùng điền các thông tin cần thiết vào form đăng ký Người dùng bấm nút "Đăng ký tài khoản" Hệ thống gửi email xác nhận đăng ký tài khoản đến người dùng Người dùng truy cập đường link trong email để xác thực tài khoản
Luồng thay thế	Không có
Mở rộng	Không có
Ngoại lệ	Tại bước 3: Nếu người dùng điền thông tin không hợp lệ, người dùng sẽ được thông báo đăng ký thất bại

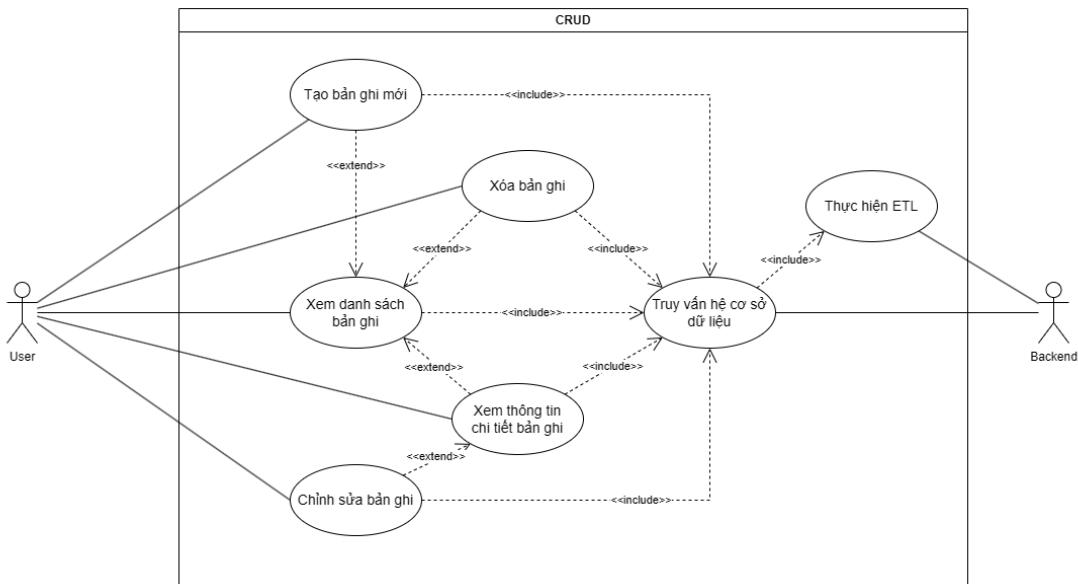


Use Case	Dăng nhập
Đối tượng	User
Tiền điều kiện	User đã truy cập vào trang web và đã tạo tài khoản
Hậu điều kiện	User đăng nhập được và có thể sử dụng dịch vụ
Luồng chính	1. Người dùng điền tài khoản và mật khẩu 2. Người dùng bấm nút "Đăng nhập" 3. Người dùng được chuyển đến trang Dashboard của dịch vụ
Luồng thay thế	Không có
Mở rộng	Tại bước 1: Người dùng có thể nhấn nút "Lưu đăng nhập" để tự động đăng nhập lần sau
Ngoại lệ	Tại bước 3: Nếu người dùng điền sai thông tin, người dùng sẽ được thông báo đăng nhập thất bại

Use Case	Đổi mật khẩu
Đối tượng	User
Tiền điều kiện	User đã truy cập vào trang web và đã tạo tài khoản
Hậu điều kiện	User có thể thay đổi mật khẩu của tài khoản đã tạo
Luồng chính	1. Người dùng nhấn nút "Quên mật khẩu?" 2. Người dùng điền email của tài khoản đã tạo 3. Hệ thống sẽ gửi email thực hiện thay đổi mật khẩu đến người dùng 4. Người dùng nhấn vào đường link trong email để đổi mật khẩu 5. Người dùng điền mật khẩu mới và bấm "Xác nhận"
Luồng thay thế	Không có
Mở rộng	Không có
Ngoại lệ	Không có

Use Case	Thay đổi thông tin cá nhân
Đối tượng	User
Tiền điều kiện	User đã đăng nhập vào dịch vụ
Hậu điều kiện	User có thể thay đổi thông tin của tài khoản
Luồng chính	1. Người dùng nhấn nút "Cài đặt hệ thống" 2. Người dùng điền các thông tin cần chỉnh sửa vào form 3. Người dùng nhấn nút "Cập nhật"
Luồng thay thế	Không có
Mở rộng	Không có
Ngoại lệ	Ở bước 3: Nếu thông tin chỉnh sửa không hợp lệ, người dùng sẽ được thông báo cập nhật thông tin thất bại

2.4.3.b Các thao tác trên hệ cơ sở dữ liệu (CRUD)



Hình 5: Usecase Diagram for CRUD

Use Case	Xem danh sách bản ghi
Đối tượng	User
Tiền điều kiện	User đã đăng nhập vào dịch vụ
Hậu điều kiện	User có thể xem danh sách bản ghi của một danh mục
Luồng chính	1. Người dùng chọn danh mục mà mình muốn xem các bản ghi 2. Trang web hiển thị danh sách các bản ghi của danh mục đó
Luồng thay thế	Không có
Mở rộng	Không có
Ngoại lệ	Không có



Use Case	Tạo bản ghi mới
Đối tượng	User
Tiền điều kiện	User đã đăng nhập vào dịch vụ
Hậu điều kiện	User có thể tạo bản ghi mới trong danh mục mà mình muốn
Luồng chính	<ol style="list-style-type: none">Người dùng chọn danh mục mà mình muốn xem các bản ghiTrang web hiển thị danh sách các bản ghi của danh mục đóNgười dùng nhấn nút "Thêm mới"Người dùng điền các thông tin của bản ghiNgười dùng nhấn nút "Tạo"
Luồng thay thế	Không có
Mở rộng	Ở bước 4: Người dùng có thể nhấn nút "Hủy" để hủy tạo bản ghi mới
Ngoại lệ	Ở bước 5: Nếu các thông tin của bản ghi không hợp lệ, hệ thống sẽ thông báo cho người dùng nhập lại và quay lại bước 4.

Use Case	Xem thông tin chi tiết bản ghi
Đối tượng	User
Tiền điều kiện	User đã đăng nhập vào dịch vụ
Hậu điều kiện	User có thể xem thông tin chi tiết của một bản ghi
Luồng chính	<ol style="list-style-type: none">Người dùng chọn danh mục mà mình muốn xem các bản ghiTrang web hiển thị danh sách các bản ghi của danh mục đóNgười dùng nhấn giữ bản ghi mà mình muốn xem thông tin chi tiết và kéo sang tráiTrang web hiển thị trang thông tin chi tiết của bản ghi
Luồng thay thế	Không có
Mở rộng	Ở bước 4: Người dùng có thể nhấn giữ và kéo sang phải để quay lại trang danh sách các bản ghi của danh mục
Ngoại lệ	Không có

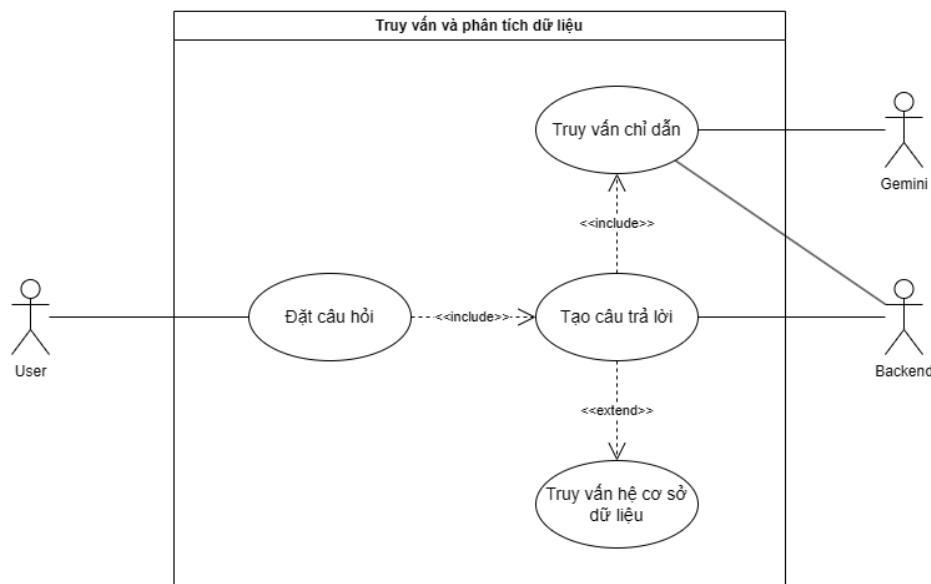


Use Case	Chỉnh sửa bản ghi
Đối tượng	User
Tiền điều kiện	User đã đăng nhập vào dịch vụ
Hậu điều kiện	User có thể chỉnh sửa thông tin chi tiết của một bản ghi
Luồng chính	<ol style="list-style-type: none">Người dùng chọn danh mục mà mình muốn xem các bản ghiTrang web hiển thị danh sách các bản ghi của danh mục đóNgười dùng nhấp giữ bản ghi mà mình muốn xem thông tin chi tiết và kéo sang tráiTrang web hiển thị trang thông tin chi tiết của bản ghiNgười dùng chỉnh sửa các thông tin của bản ghi theo ý muốn.Người dùng nhấp nút "Cập nhật"
Luồng thay thế	Không có
Mở rộng	Ở bước 4: Người dùng có thể nhấn giữ và kéo sang phải để quay lại trang danh sách các bản ghi của danh mục
Ngoại lệ	Ở bước 6: Nếu thông tin đã cập nhật của bản ghi không hợp lệ, trang web sẽ thông báo cập nhật thông tin thất bại

Use Case	Xóa bản ghi
Đối tượng	User
Tiền điều kiện	User đã đăng nhập vào dịch vụ
Hậu điều kiện	User có thể xóa bản ghi của một danh mục
Luồng chính	<ol style="list-style-type: none">Người dùng chọn danh mục mà mình muốn xem các bản ghiTrang web hiển thị danh sách các bản ghi của danh mục đóNgười dùng chọn các bản ghi mà mình muốn xóaNgười dùng nhấp nút "Xóa"
Luồng thay thế	Không có
Mở rộng	Không có
Ngoại lệ	Không có

Use Case	Thực hiện ETL
Đối tượng	Backend
Tiền điều kiện	User thực hiện thao tác CRUD
Hậu điều kiện	Bản ghi được trích xuất, chuyển đổi và tải vào hệ cơ sở dữ liệu phân tích
Luồng chính	<ol style="list-style-type: none">Sau khi người dùng thực hiện thao tác CRUD, backend sẽ thực hiện trích xuất, chuyển đổi và tải tương ứng với thao tác đó trên hệ cơ sở dữ liệu phân tích.
Luồng thay thế	Không có
Mở rộng	Không có
Ngoại lệ	Không có

2.4.3.c Truy vấn và phân tích dữ liệu



Hình 6: Usecase Diagram for data query and analysis

Use Case	Đặt câu hỏi
Đối tượng	User
Tiền điều kiện	User đã đăng nhập vào dịch vụ
Hậu điều kiện	User có thể đặt câu hỏi và nhận câu trả lời phù hợp về các thông tin được lưu trong hệ thống
Luồng chính	1. Người dùng chọn tính năng phân tích. 2. Người dùng nhập câu hỏi của mình và nhấn "Xác nhận". 3. Hệ thống sẽ xử lý câu hỏi của người dùng và trả về câu trả lời thích hợp.
Luồng thay thế	Không
Mở rộng	Không
Ngoại lệ	Ở bước 3: Hệ thống sẽ trả về câu trả lời không phù hợp với yêu cầu người dùng. Người dùng có thể thực hiện lại quy trình.



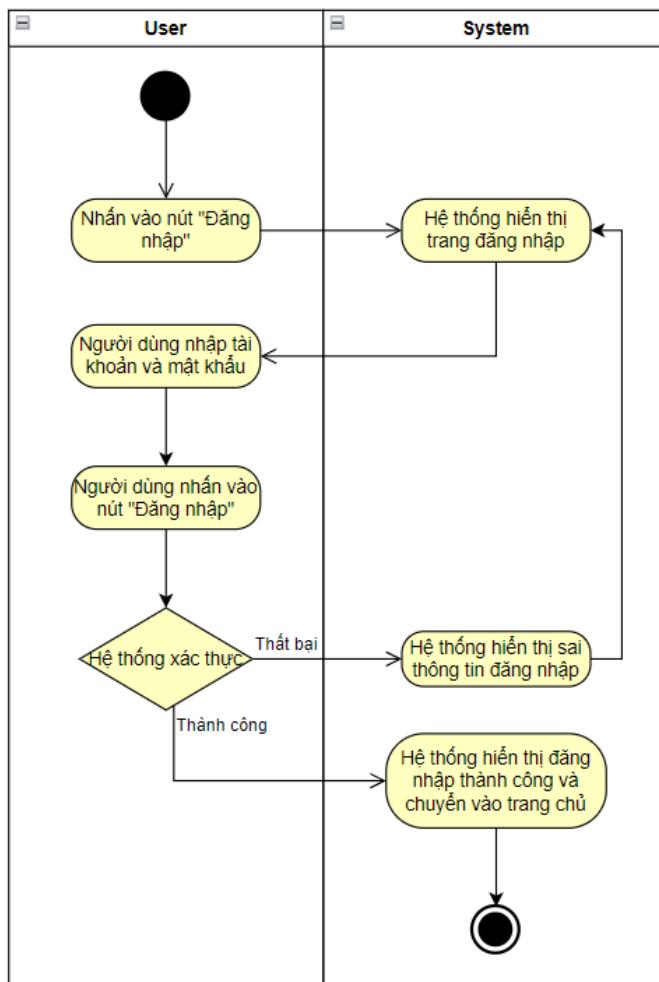
Use Case	Tạo câu trả lời
Đối tượng	Backend
Tiền điều kiện	User đã đặt câu hỏi
Hậu điều kiện	Backend tạo câu trả lời với các thông tin mà người dùng yêu cầu
Luồng chính	<ol style="list-style-type: none">1. Hệ thống nhận câu hỏi từ người dùng2. Hệ thống sử dụng các công cụ và hàm phù hợp để tạo câu trả lời cho câu hỏi3. Hệ thống trả về câu trả lời cho người dùng.
Luồng thay thế	Không
Mở rộng	Không
Ngoại lệ	Không

Use Case	Truy vấn chỉ dẫn
Đối tượng	Backend, Gemini
Tiền điều kiện	Backend thực hiện truy vấn chỉ dẫn để trả lời câu hỏi
Hậu điều kiện	Backend có chỉ dẫn phù hợp để trả lời câu hỏi
Luồng chính	<ol style="list-style-type: none">1. Hệ thống gửi truy vấn chỉ dẫn đến Gemini2. Gemini trả về chỉ dẫn trả lời câu hỏi phù hợp cho hệ thống.3. Hệ thống sử dụng chỉ dẫn để trả lời câu hỏi
Luồng thay thế	Không
Mở rộng	Không
Ngoại lệ	Không

Use Case	Truy vấn hệ cơ sở dữ liệu
Đối tượng	Backend
Tiền điều kiện	Backend thực hiện truy vấn hệ cơ sở dữ liệu để trả lời câu hỏi
Hậu điều kiện	Backend có chỉ thông tin phù hợp từ hệ cơ sở dữ liệu
Luồng chính	<ol style="list-style-type: none">1. Hệ thống gửi truy vấn dữ liệu đến hệ cơ sở dữ liệu2. Hệ cơ sở dữ liệu trả về thông tin có được từ truy vấn3. Hệ thống sử dụng thông tin để trả lời câu hỏi từ người dùng
Luồng thay thế	Không
Mở rộng	Không
Ngoại lệ	Không

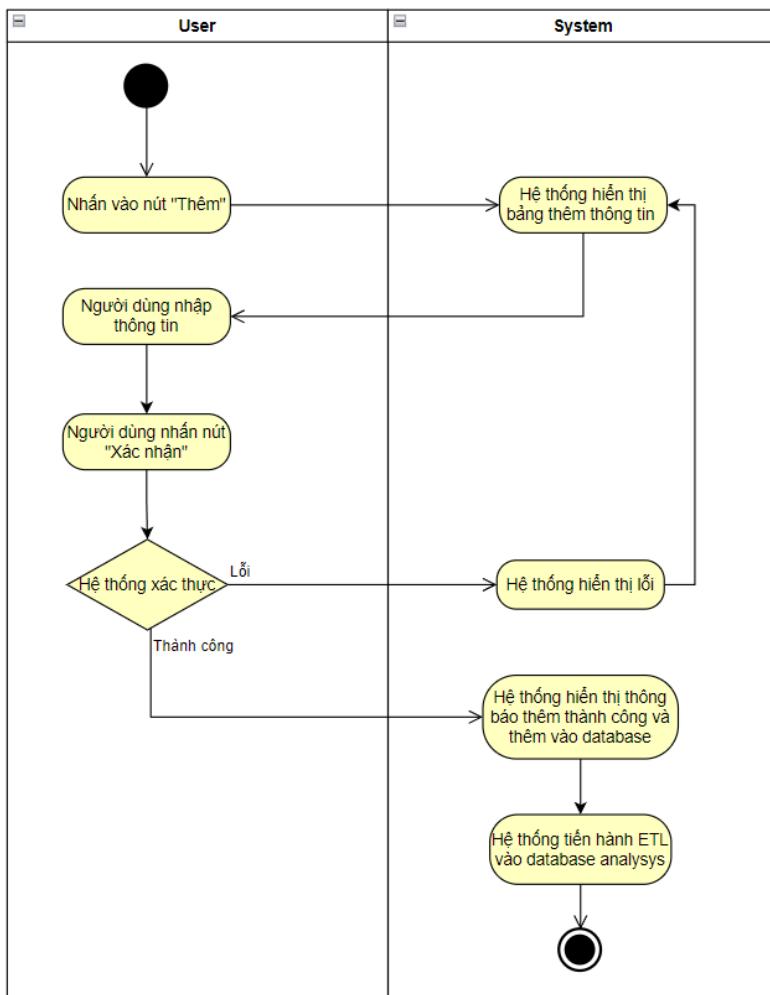
2.4.4 Activity diagram

Xem chi tiết [tại đây](#).



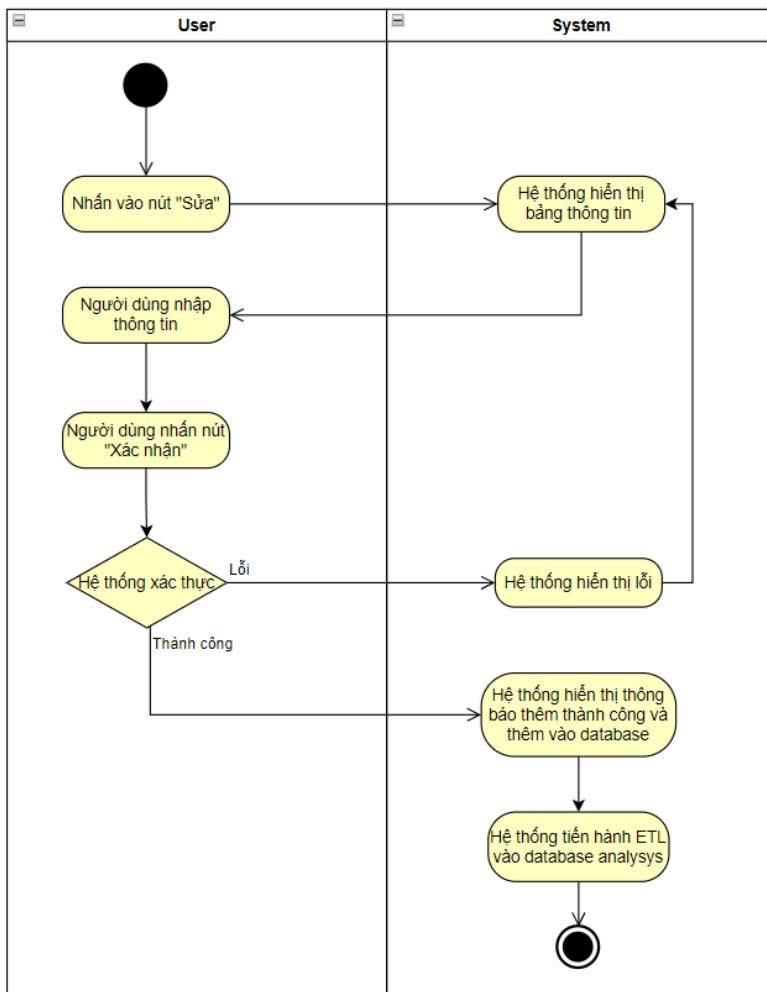
Hình 7: Activity Diagram for the login function

Khi người dùng nhấn vào nút "Đăng nhập" để tiến hành đăng nhập vào hệ thống, hệ thống sẽ chuyển hướng người dùng qua trang đăng nhập. Sau đó, người dùng tiến hành nhập thông tin đăng nhập của bản thân và sau khi hoàn tất việc nhập tài khoản và mật khẩu thì người dùng nhấn vào nút đăng nhập. Cuối cùng, hệ thống sẽ tiến hành xác thực thông tin. Nếu thông tin đăng nhập sai, hệ thống sẽ thông báo người dùng là thông tin đăng nhập sai và chuyển người dùng về trang nhập thông tin đăng nhập để người dùng tiến hành nhập thông tin lại. Nếu thông tin đăng nhập chính xác, hệ thống sẽ thông báo đăng nhập thành công và chuyển hướng người dùng vào trang chủ hệ thống sau khi đăng nhập thành công.



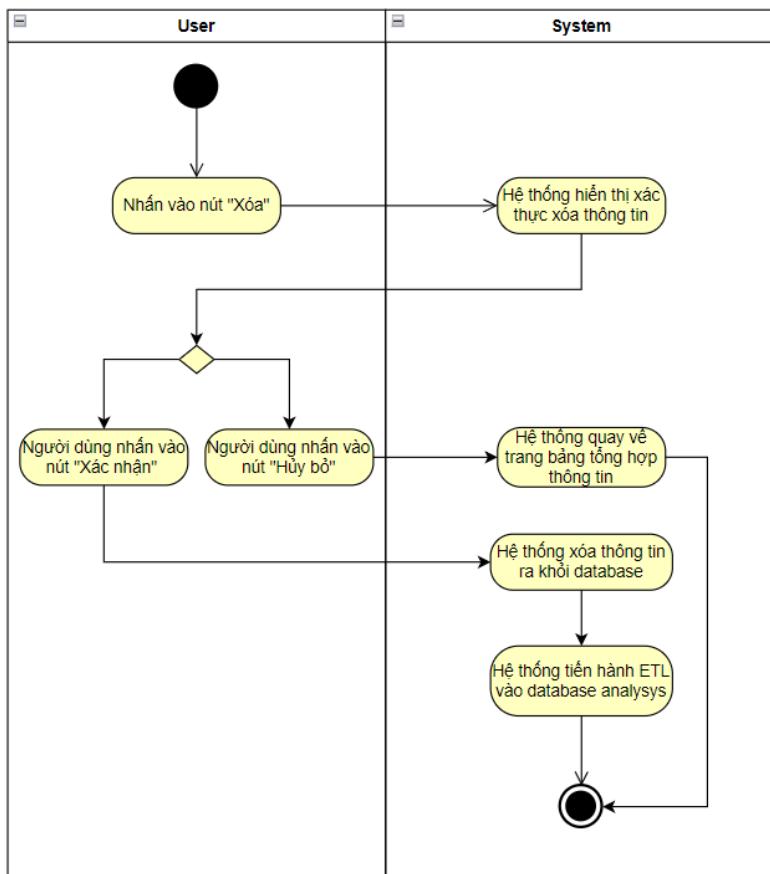
Hình 8: Activity Diagram for adding data to the system

Khi người dùng tiến hành thêm thông tin vào hệ thống. Hệ thống sẽ hiển thị trang điền thông tin để người dùng tiến hành nhập thông tin. Sau khi người dùng hoàn tất sẽ nhấn vào nút xác nhận. Sau đó, hệ thống tiến hành kiểm tra thông tin người dùng nhập vào hệ thống. Nếu thông tin không chính xác, hệ thống sẽ hiển thị lỗi cho người dùng để người dùng lại trường thông tin không chính xác. Nếu thông tin chính xác, hệ thống sẽ tiến hành thêm dữ liệu vào database. Cuối cùng hệ thống sẽ tiến hành ETL vào database analysys.



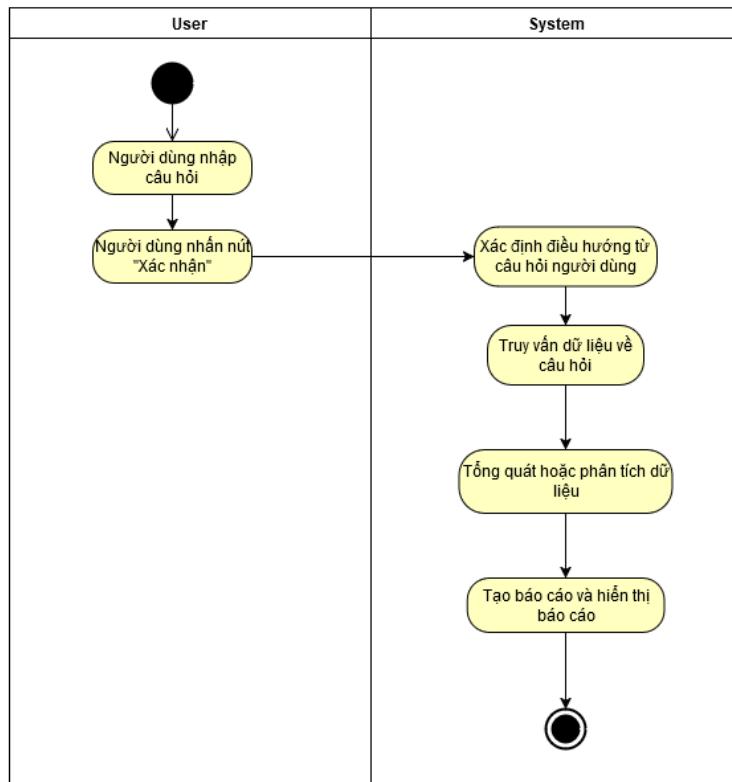
Hình 9: Activity Diagram for updating information in the system

Khi người dùng tiến hành sửa thông tin. Hệ thống sẽ hiển thị trang thông tin để người dùng tiến hành thay đổi thông tin. Sau khi người dùng hoàn tất sẽ nhấn vào nút xác nhận. Sau đó, hệ thống tiến hành kiểm tra thông tin người dùng nhập vào hệ thống. Nếu thông tin không chính xác, hệ thống sẽ hiển thị lỗi cho người dùng để người dùng nhập lại trường thông tin không chính xác. Nếu thông tin chính xác, hệ thống sẽ tiến hành lưu dữ liệu sau khi chỉnh sửa vào database. Cuối cùng hệ thống sẽ tiến hành ETL vào database analysys.



Hình 10: Activity Diagram for deleting information in the system

Khi người dùng tiến hành xóa thông tin vào hệ thống. Hệ thống sẽ hiển thị pop-up để người dùng xác nhận việc xóa thông tin. Nếu người dùng nhấn vào nút "Hủy bỏ" hệ thống sẽ trả người dùng về trang bảng tổng hợp các thông tin. Nếu người dùng nhấn vào nút "Xác nhận" hệ thống sẽ tiến hành xóa thông tin ra khỏi database. Cuối cùng hệ thống sẽ tiến hành ETL dữ liệu vào database analysys.

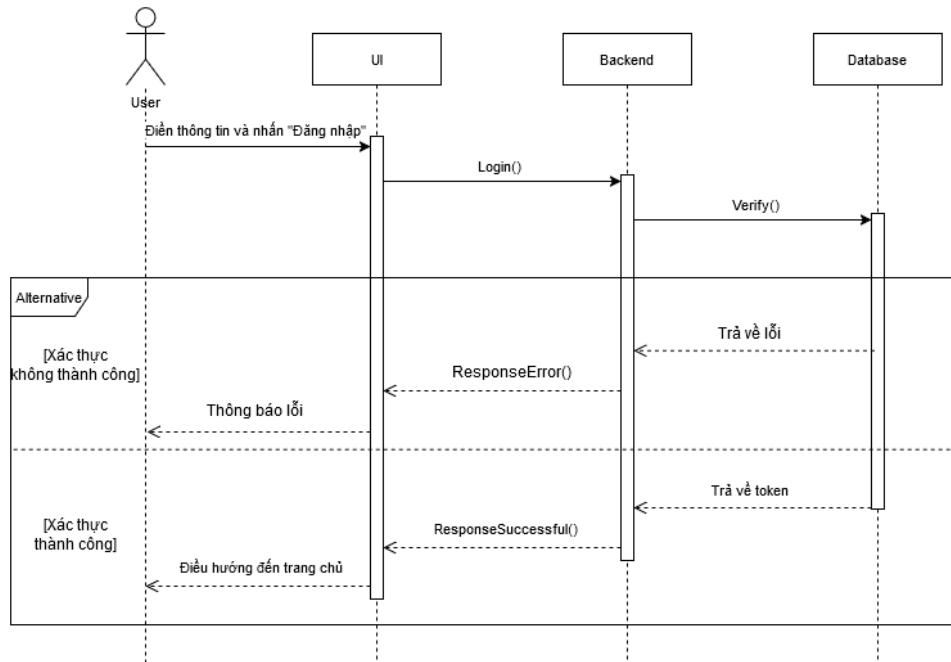


Hình 11: Activity Diagram for analysis function

Khi người dùng tiến hành truy vấn hoặc phân tích thông tin trong hệ thống. Hệ thống sẽ hiển thị trang thu thập câu hỏi từ người dùng. Sau khi người dùng điền câu hỏi và nhấn "Xác nhận", hệ thống sẽ sử dụng mô hình ngôn ngữ để xác định điều hướng từ câu hỏi của người dùng. Hệ thống sẽ tiến hành truy vấn hệ cơ sở dữ liệu về các thông tin có liên quan trong yêu cầu của người dùng và tổng quát các thông tin đó. Cuối cùng, câu trả lời dạng báo cáo sẽ được trình bày cho người dùng về câu hỏi.

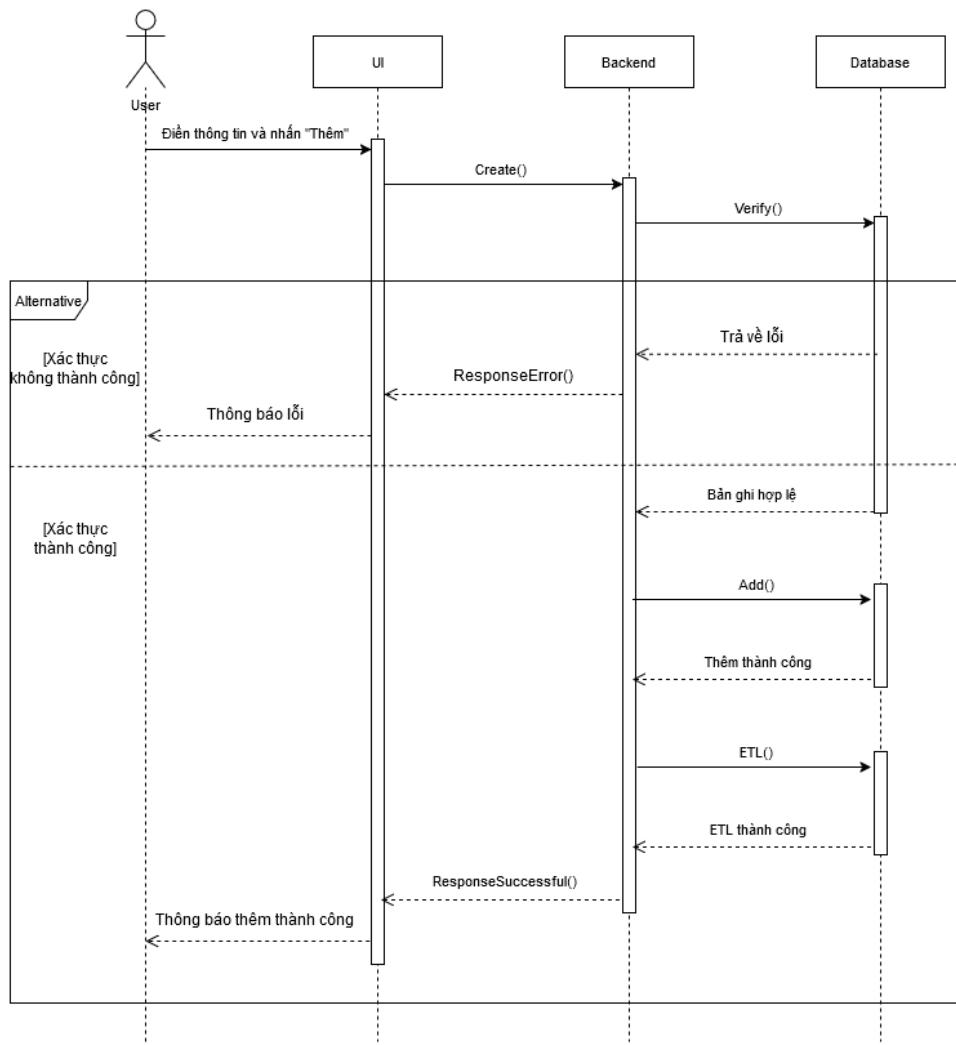
2.4.5 Sequence diagram

Xem chi tiết [tại đây](#).



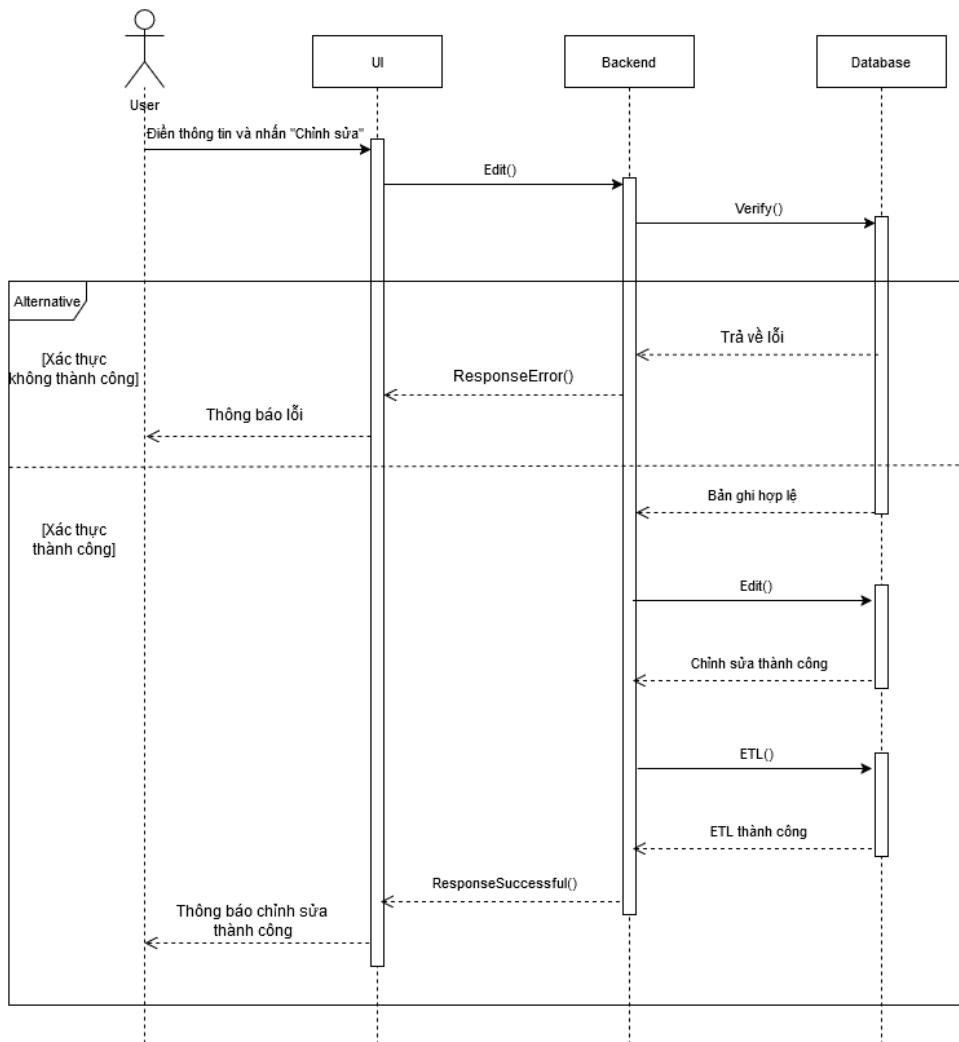
Hình 12: Sequence Diagram for logging into the system

Khi người dùng điền thông tin và nhấn "Đăng nhập", frontend sẽ gửi yêu cầu đăng nhập đến backend. Sau đó, backend sẽ tiến hành xác thực thông tin đăng nhập của người dùng. Nếu thông tin đăng nhập của người dùng chính xác, backend sẽ trả về token cho frontend và người dùng sẽ được thông báo đăng nhập thành công. Ngược lại, nếu thông tin đăng nhập của người dùng không chính xác, backend sẽ trả về lỗi và người dùng sẽ được thông báo lỗi đăng nhập.



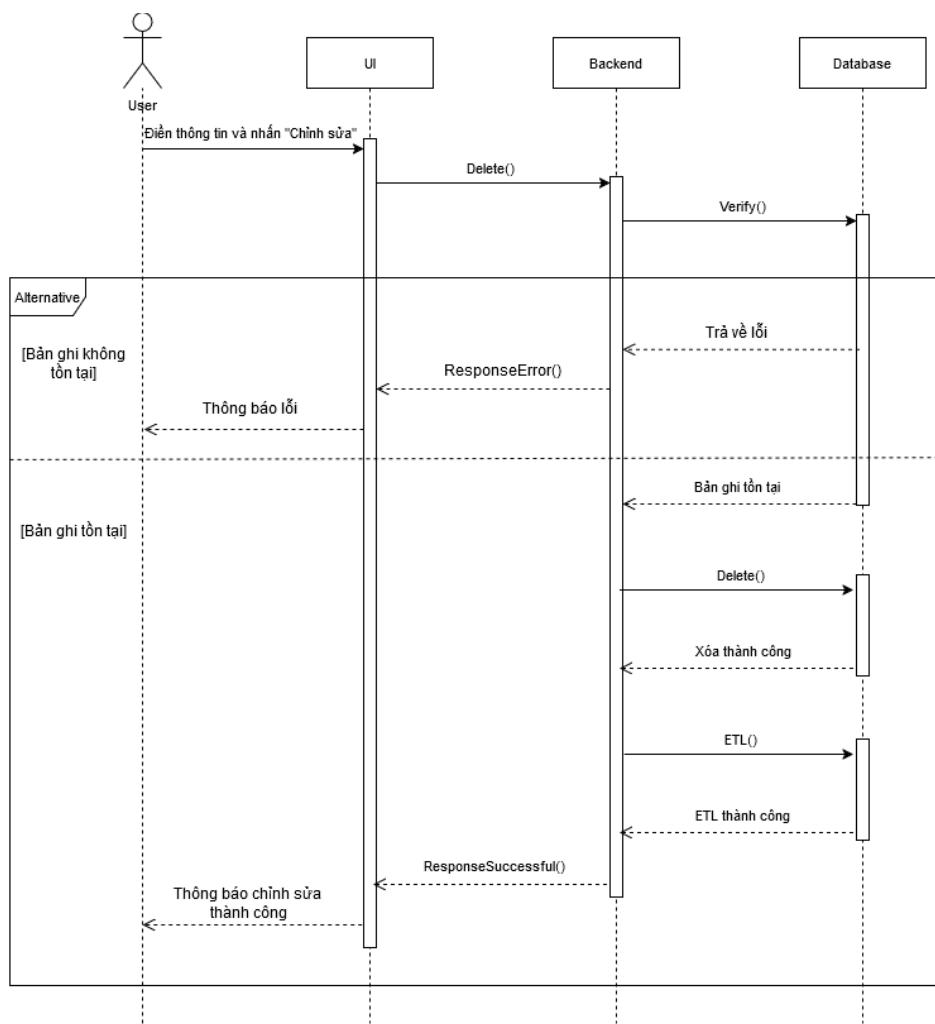
Hình 13: Sequence Diagram for adding data to the system

Khi người dùng điền thông tin của bản ghi và nhấn "Thêm", frontend sẽ gửi yêu cầu tạo bản ghi cùng thông tin của bản ghi cần được thêm đã được cung cấp bởi người dùng. Trước tiên, thông tin của bản ghi sẽ được xác thực bởi backend: các yêu cầu về số lượng, ngày tháng,... Nếu thông tin hợp lệ, backend sẽ thêm bản ghi mới vào hệ cơ sở dữ liệu và thực hiện ETL sử dụng thông tin đó. Cuối cùng, người dùng sẽ được thông báo thêm bản ghi mới thành công. Ngược lại, nếu thông tin vi phạm một trong các yêu cầu, backend sẽ trả về lỗi và người dùng sẽ được thông báo thêm bản ghi mới thất bại.



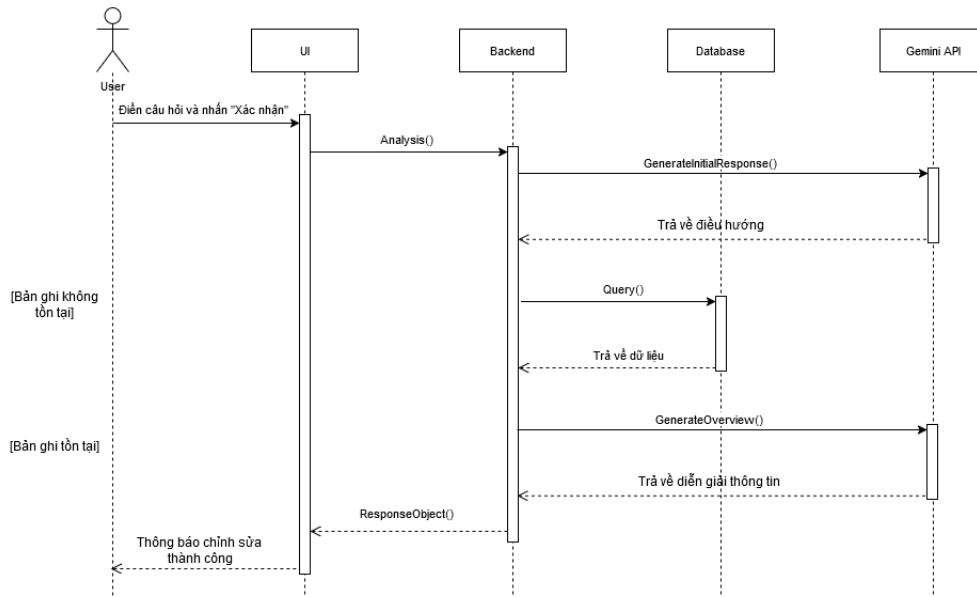
Hình 14: Sequence Diagram for updating information in the system

Khi người dùng điền thông tin của bản ghi và nhấn "Chỉnh sửa", frontend sẽ gửi yêu cầu chỉnh sửa bản ghi cụ thể cùng thông tin của bản ghi cần được chỉnh sửa đã được cung cấp bởi người dùng. Trước tiên, thông tin mới của bản ghi sẽ được xác thực bởi backend: các yêu cầu về số lượng, ngày tháng,... Nếu thông tin hợp lệ, backend sẽ chỉnh sửa bản ghi cũ trong hệ cơ sở dữ liệu và thực hiện ETL sử dụng thông tin đó. Cuối cùng, người dùng sẽ được thông báo chỉnh sửa bản ghi thành công. Ngược lại, nếu thông tin vi phạm một trong các yêu cầu, backend sẽ trả về lỗi và người dùng sẽ được thông báo chỉnh sửa bản ghi thất bại.



Hình 15: Sequence Diagram for deleting information in the system

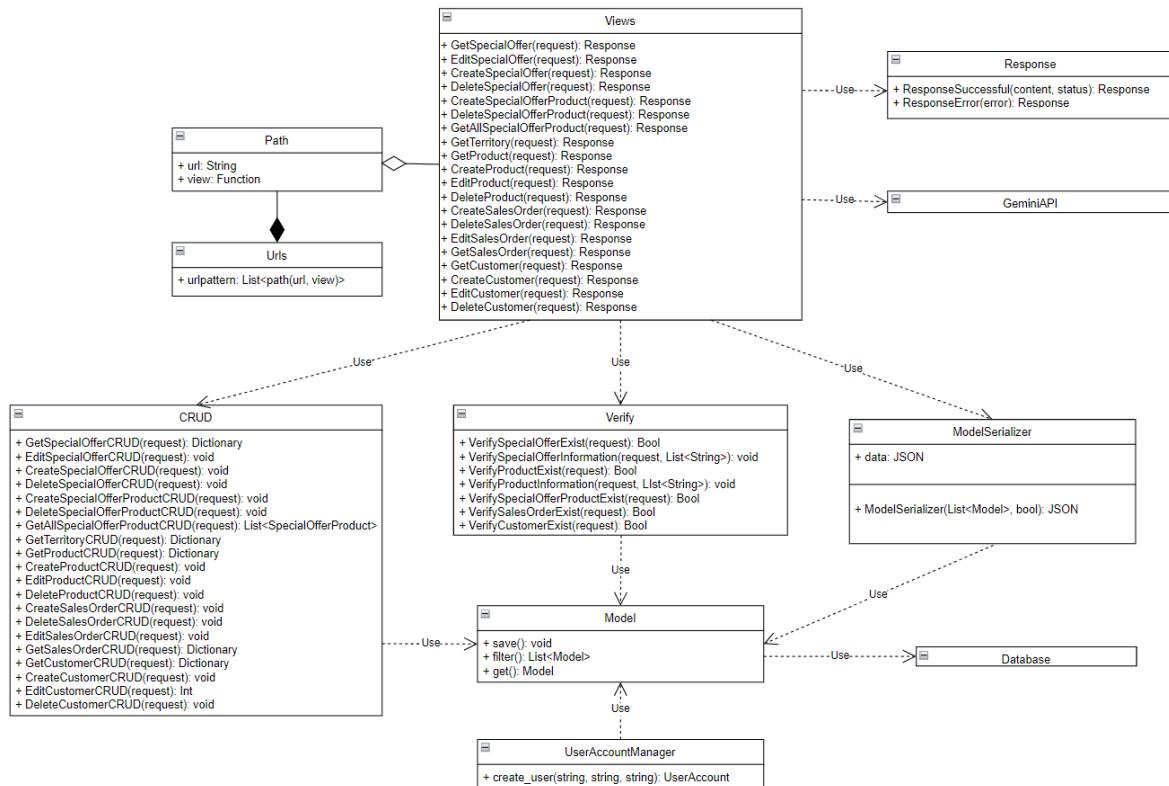
Khi người nhấn "Xóa" trên một bản ghi, frontend sẽ gửi yêu cầu xóa bản ghi cụ thể đến backend. Trước tiên, backend sẽ kiểm tra nếu bản ghi tồn tại trong hệ cơ sở dữ liệu. Nếu bản ghi đã tồn tại trong hệ cơ sở dữ liệu, bản ghi sẽ được xóa và thao tác xóa trên hệ cơ sở dữ liệu phân tích (ETL) sẽ được thực hiện. Ngược lại, nếu bản ghi không tồn tại trên hệ cơ sở dữ liệu, backend sẽ trả về lỗi và người dùng sẽ được thông báo bản ghi không tồn tại.



Hình 16: Sequence Diagram for analysis function

Khi người dùng điền câu hỏi và nhấn "Xác nhận", frontend sẽ gửi yêu cầu phân tích cùng câu hỏi người dùng đến backend. Trước tiên, backend sẽ sử dụng Gemini API để xác định yêu cầu của người dùng và xác định điều hướng. Sau đó, backend sẽ truy vấn dữ liệu dựa vào kết quả điều hướng từ mô hình ngôn ngữ. Cuối cùng, diễn giải hoặc phân tích tương ứng sẽ được thực hiện và kết quả sẽ được trả về và trình bày cho người dùng dưới dạng báo cáo.

2.4.6 Class diagram



Hình 17: Class Diagram

Xem chi tiết [tại đây](#).

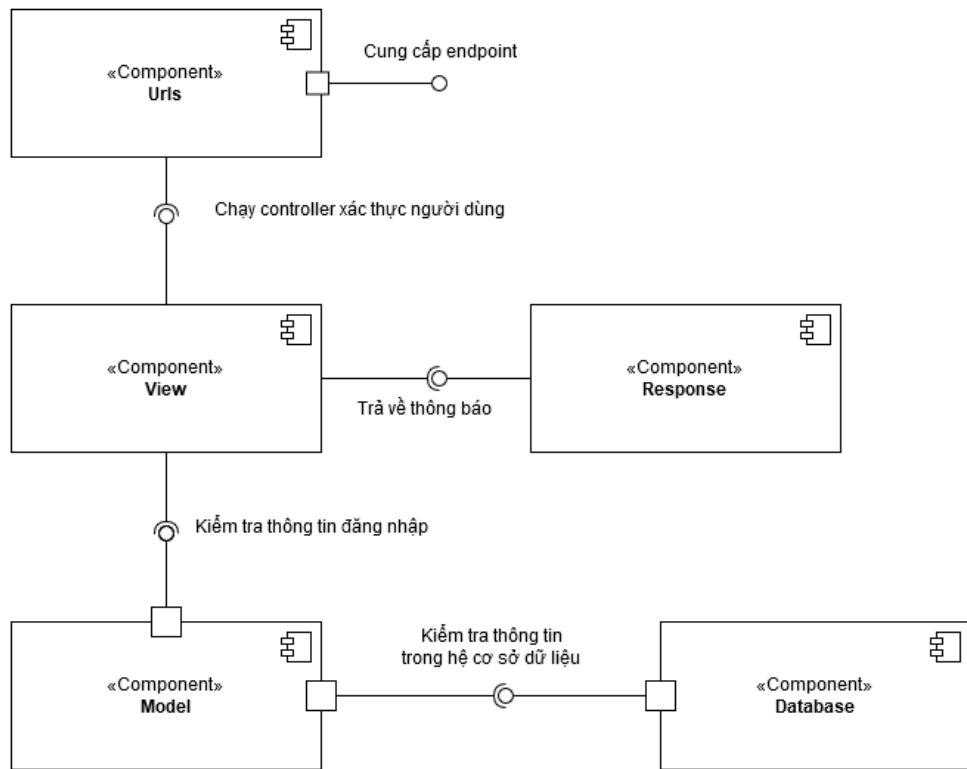
Backend được thiết kế và hiện thực sử dụng framework Django dựa trên mô hình MVC (Model, View, Controller) và Client-Server. Các thành phần quan trọng trong backend gồm có

- **Urls:** Mô đun Urls định nghĩa các đường dẫn (dưới dạng object Path) mà frontend có thể truy cập vào backend để thực hiện các thao tác. Mô đun này đóng vai trò Views của hệ thống
- **Views:** Mô đun Views định nghĩa các thao tác được thực hiện khi frontend truy cập các Urls (hay còn gọi là Views của hệ thống). Mô đun này tương tác với nhiều mô đun khác để hoàn thành quá trình thực hiện thao tác như: CRUD, Verify, ModelSerializer, GeminiAPI, Response. Mô hình này đóng vai trò Controller của hệ thống.
- **CRUD:** Mô đun chứa các hàm dùng để tương tác với hệ cơ sở dữ liệu dùng để thực hiện việc lưu dữ liệu hoặc ETL.
- **Verify:** Chứa các hàm dùng để kiểm tra yêu cầu từ frontend trước khi xử lý.
- **ModelSerializer:** Chứa các công cụ dùng để chuyển đổi dữ liệu từ model trong Django sang dạng JSON để trả về frontend.

- **GeminiAPI:** Dùng để tương tác với language model và tạo các chỉ dẫn cho hệ thống.
- **Response:** Mô đun Response được dùng để tạo response và trả về frontend.

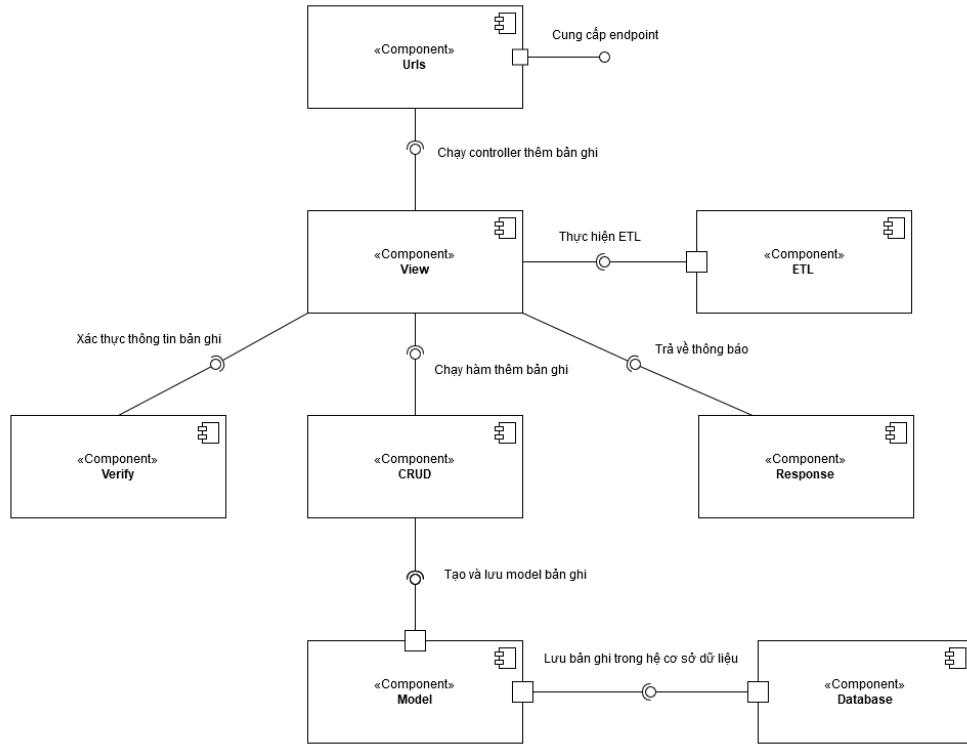
2.4.7 Component diagram

Xem chi tiết [tại đây](#).



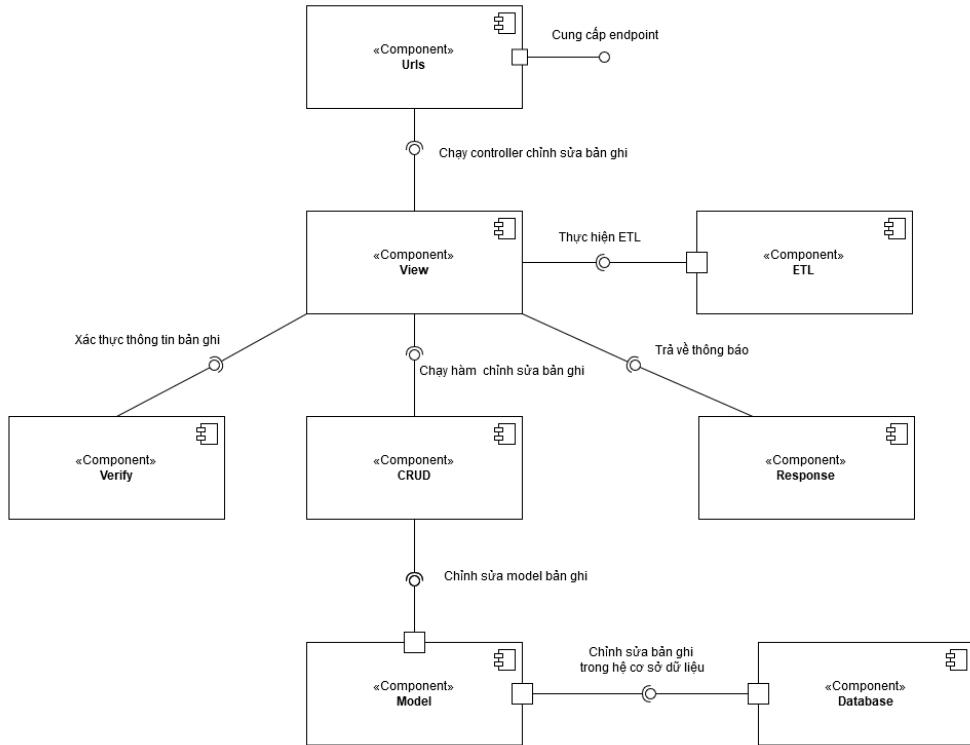
Hình 18: Component Diagram for logging into the system

Khi nhận yêu cầu đăng nhập (xác thực người dùng) từ frontend, backend sẽ tiến hành xác thực thông tin đăng nhập của người dùng trong hệ cơ sở dữ liệu. Nếu thông tin đăng nhập của người dùng chính xác, backend sẽ trả về token sử dụng hàm trong thành phần Response. Ngược lại, nếu thông tin đăng nhập của người dùng không chính xác, backend sẽ trả về lỗi.



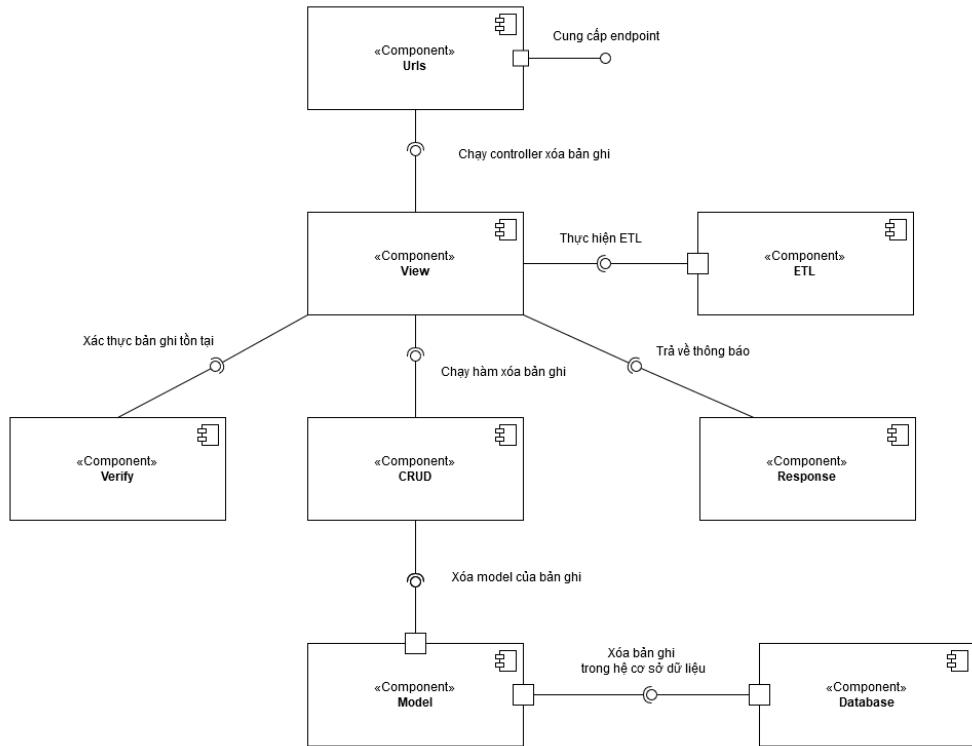
Hình 19: Component Diagram for adding data to the system

Khi nhận yêu cầu thêm bản ghi từ frontend, thông tin của bản ghi sẽ được xác thực bởi thành phần Verify. Nếu thông tin hợp lệ, hàm thêm bản ghi của thành phần CRUD sẽ được chạy để thêm bản ghi mới vào hệ cơ sở dữ liệu. Sau đó, hàm trong ETL sẽ được gọi để đồng bộ hóa hệ cơ sở dữ liệu production và analysis. Cuối cùng, thông báo thêm thành công sẽ được trả về sử dụng hàm trong thành phần Response.



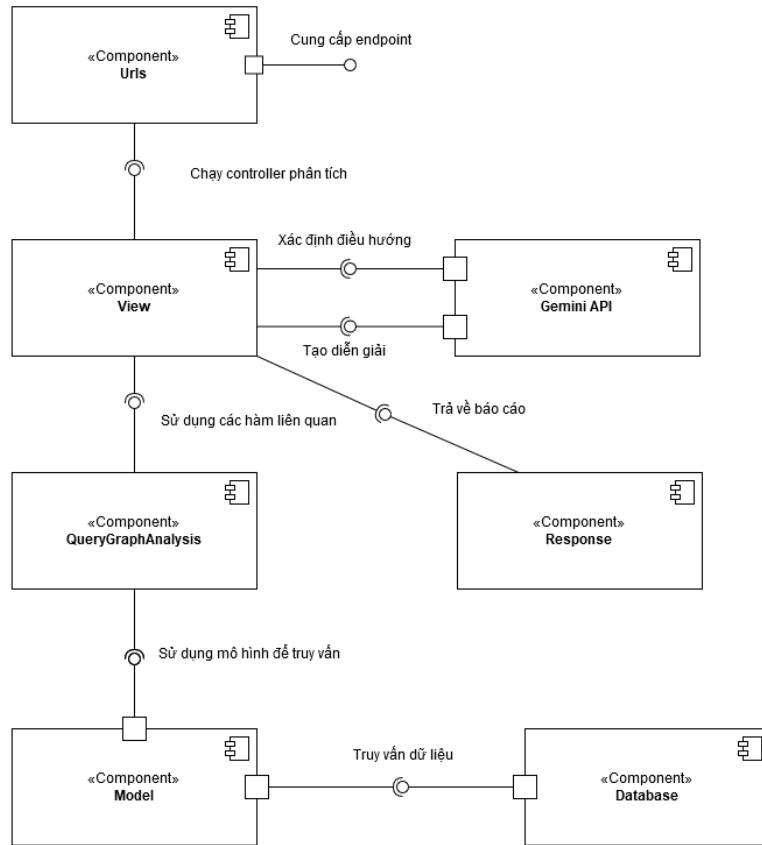
Hình 20: Component Diagram for updating information in the system

Khi nhận yêu cầu chỉnh sửa bản ghi từ frontend, thông tin của bản ghi sẽ được xác thực bởi thành phần Verify. Nếu thông tin hợp lệ, hàm thêm bản ghi của thành phần CRUD sẽ được chạy để chỉnh sửa bản ghi trong hệ cơ sở dữ liệu. Sau đó, hàm trong ETL sẽ được gọi để đồng bộ hóa hệ cơ sở dữ liệu production và analysis. Cuối cùng, thông báo chỉnh sửa thành công sẽ được trả về sử dụng hàm trong thành phần Response.



Hình 21: Component Diagram for deleting information in the system

Khi nhận yêu cầu xóa bản ghi từ frontend, thành phần Verify sẽ kiểm tra nếu bản ghi tồn tại trong hệ cơ sở dữ liệu. Nếu bản ghi tồn tại, hàm xóa bản ghi của thành phần CRUD sẽ được chạy để xóa bản ghi trong hệ cơ sở dữ liệu. Sau đó, hàm trong ETL sẽ được gọi để đồng bộ hóa hệ cơ sở dữ liệu production và analysis. Cuối cùng, thông báo xóa thành công sẽ được trả về sử dụng hàm trong thành phần Response.



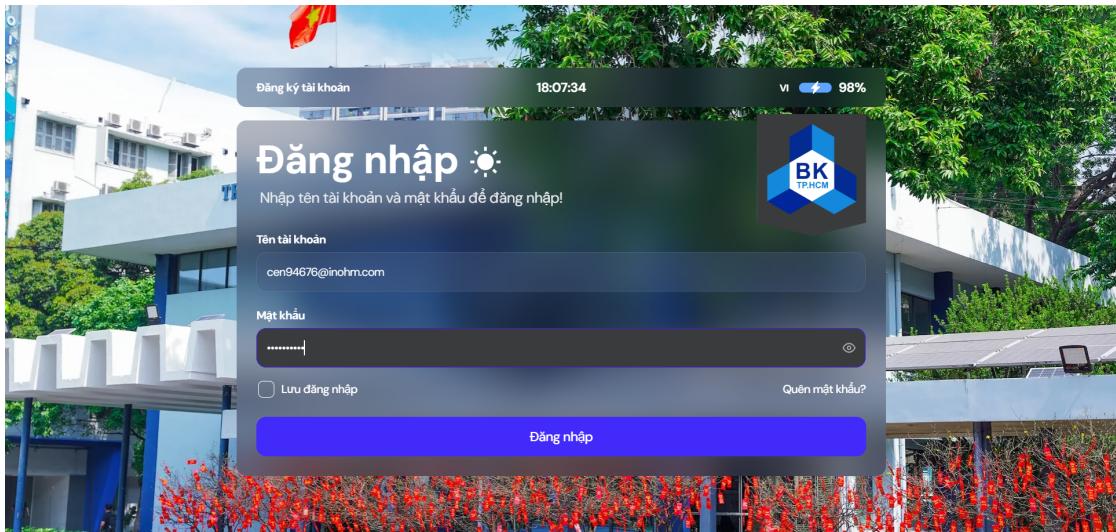
Hình 22: Component Diagram for analysis function

Khi nhận yêu cầu truy vấn và phân tích dữ liệu, trước tiên, controller phân tích sẽ sử dụng Gemini API để xác định yêu cầu của người dùng và xác định điều hướng. Sau đó, các hàm Query của thành phần QueryGraphAnalysis sẽ được sử dụng để truy vấn dữ liệu. Controller phân tích sẽ tiếp tục thực hiện các bước trong điều hướng và tạo báo cáo. Cuối cùng, báo cáo được trả về.

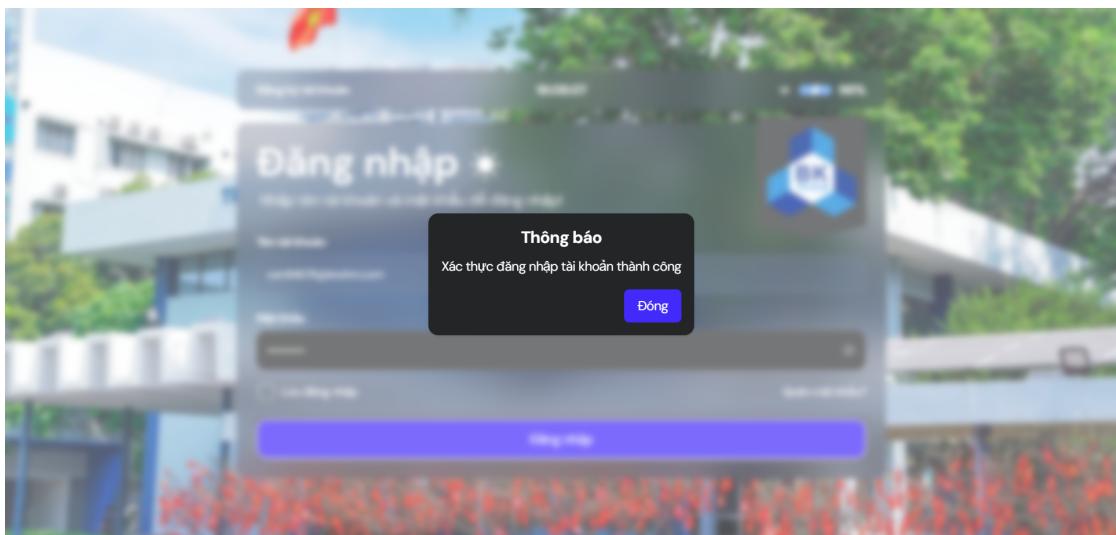


2.4.8 Demo

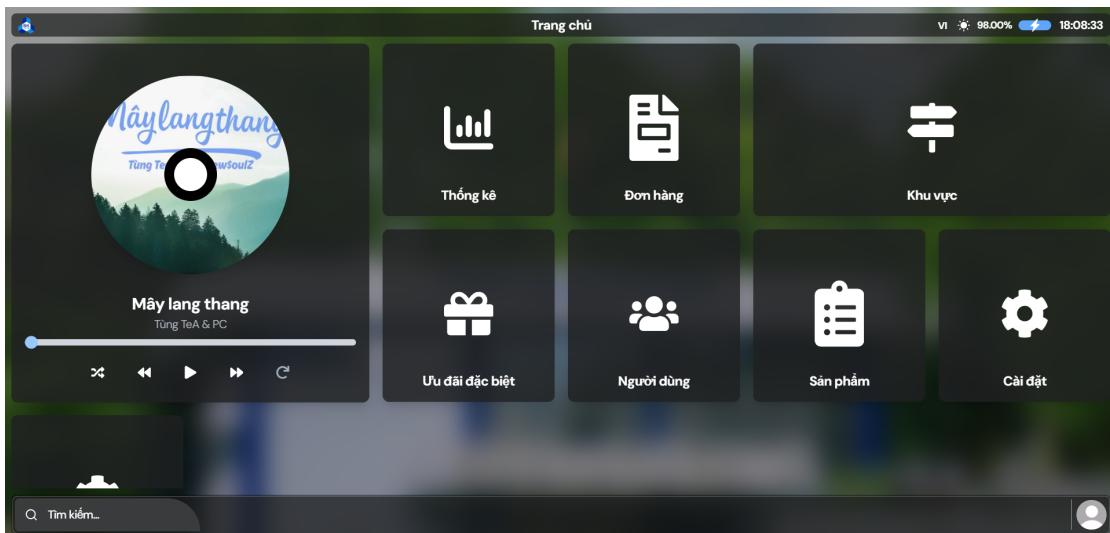
2.4.8.a Đăng nhập vào hệ thống



Hình 23: Form đăng nhập của ứng dụng



Hình 24: Thông báo đăng nhập thành công



Hình 25: Diều hướng đến trang chủ sau khi đăng nhập

2.4.8.b Các tính năng CRUD

	Tên sản phẩm	Nhà sản xuất	Giá sản xuất	Giá bán	Kích thước
<input type="checkbox"/>	Mountain Bike Socks, L	dásdadsd	3.4000	9.5000	XXL
<input type="checkbox"/>	Gyatt	Rizz	140.0000	156.6000	1234
<input type="checkbox"/>	Mountain Bike Socks, L	dásdadsd	3.4000	9.5000	sadasdadasa
<input type="checkbox"/>	Test product	test manufacturer	10.0000	10.0000	12
<input type="checkbox"/>	Test editing ast	test manufacturer	14.5000	10.0000	12
<input type="checkbox"/>	Road-750 Black, 52	Không có thông tin	343.6500	539.9900	XXL
<input type="checkbox"/>	Road-750 Black, 48	Publisher	343.6500	539.9900	XXL
<input type="checkbox"/>	Road-750 Black, 44	Không có thông tin	343.6500	539.9900	XXL
<input type="checkbox"/>	Hi. Bottom Bracket	Không có thông tin	539.400	121.4900	XXL
<input type="checkbox"/>	Mi. Bottom Bracket	Không có thông tin	44.9500	101.2400	XXL

Hình 26: Trang xem danh sách sản phẩm của ứng dụng

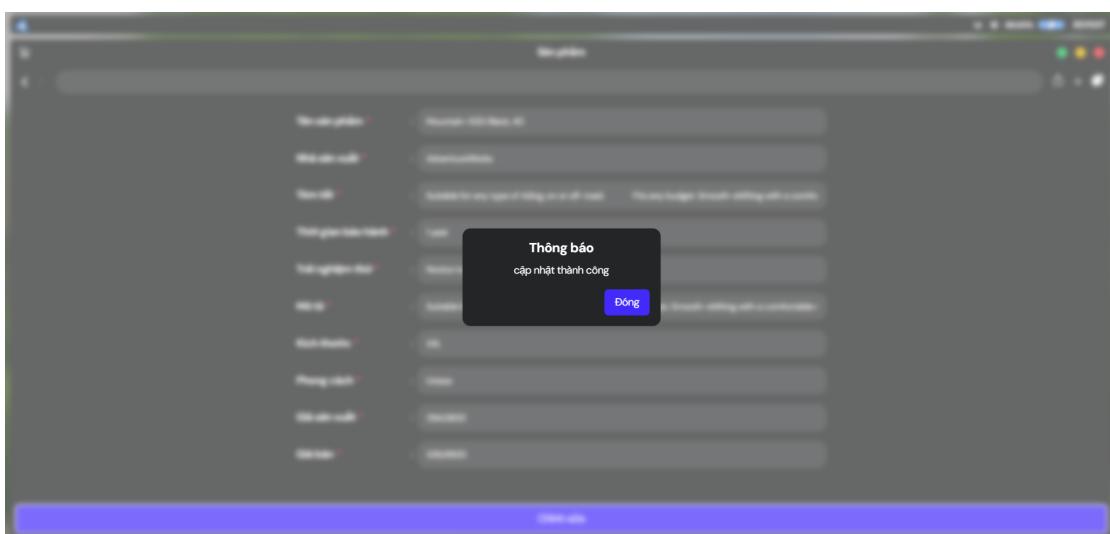


The screenshot shows a product editing form titled "Sản phẩm". The form contains the following fields:

Tên sản phẩm *	: Mountain-500 Black, 40
Nhà sản xuất *	: AdventureWorks
Tóm tắt *	: Suitable for any type of riding, on or off-road. Fits any budget. Smooth-shifting with a comfortable
Thời gian bảo hành *	: 1 year
Trải nghiệm thử *	: Novice to Intermediate riders
Mô tả *	: Suitable for any type of riding, on or off-road. Fits any budget. Smooth-shifting with a comfortable r
Kích thước *	: XXL
Phong cách *	: Unisex
Giá sản xuất *	: 2945800
Giá bán *	: 5399900

At the bottom right of the form is a blue button labeled "Chỉnh sửa".

Hình 27: Chính sửa thông tin sản phẩm



Hình 28: Chính sửa thông tin sản phẩm thành công

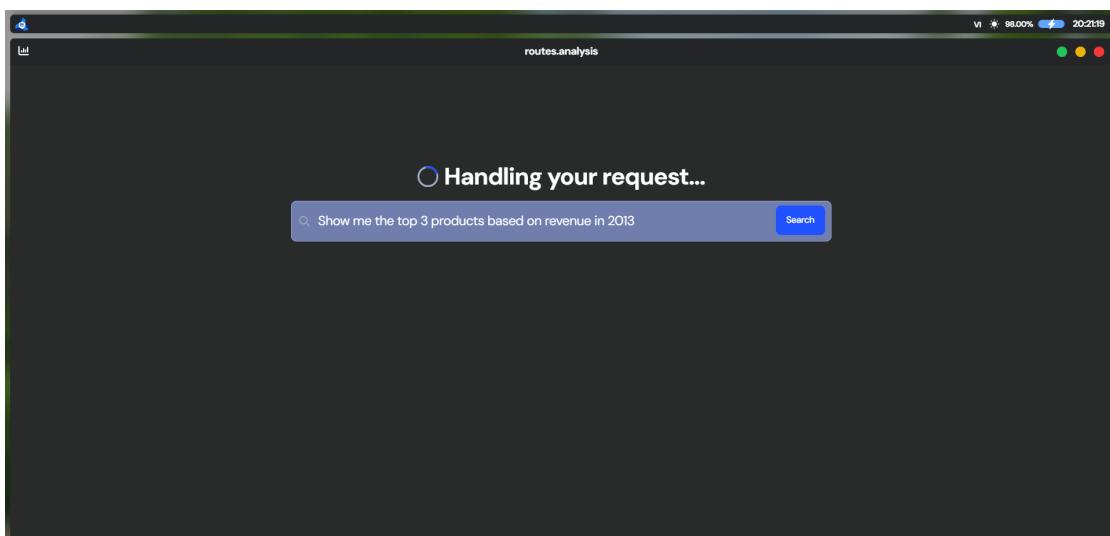


2.4.8.c Dashboard



Hình 29: Trang dashboard thống kê của ứng dụng

2.4.8.d Tính năng phân tích



Hình 30: Người dùng nhập câu hỏi



The screenshot shows a dark-themed application window titled "routes.analysis". At the top, it says "Report: 'Show me the top 3 products based on revenue in 2013'". Below this is a "QUERY" section containing the following SQL code:

```
SELECT p.Name, SUM(sod.LineTotal) AS Revenue FROM analysis_productdim p JOIN analysis_salesorderdetailfact sod ON p.Id = sod.Product_id JOIN analysis_salesorderheaderfact soh ON sod.SalesOrder_Id = soh.Id WHERE EXTRACT(YEAR FROM soh.OrderDate) = 2013 GROUP BY p.Name ORDER BY Revenue DESC LIMIT 3;
```

Below the query is a "QUERY RESULT" section displaying a table with three rows:

Name	revenue
Mountain-200 Black, 38	2212974.7827
Mountain-200 Black, 42	1932388.2907
Mountain-200 Silver, 38	1815673.0932

At the bottom, there is an explanation of the query steps and information about joined tables.

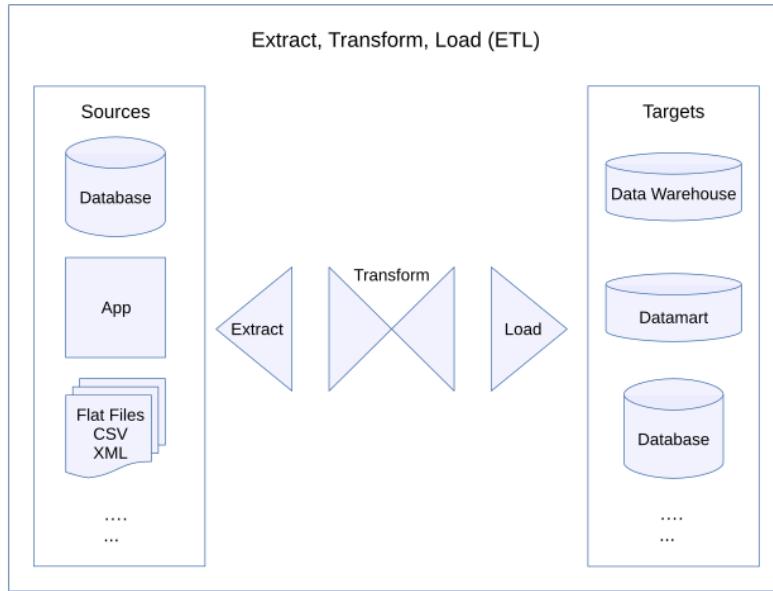
Hình 31: Câu trả lời nhận được từ hệ thống

2.5 Trích xuất, chuyển đổi, tải (ETL)

2.5.1 Giới thiệu

ETL (Extract, Transform, Load) là một quy trình quan trọng trong lĩnh vực quản lý và xử lý dữ liệu, đặc biệt trong xây dựng kho dữ liệu (Data Warehouse). Đây là một tập hợp các bước xử lý để chuyển đổi dữ liệu từ các nguồn khác nhau thành một dạng dễ sử dụng hơn trong các hệ thống phân tích và báo cáo. Các bước chính bao gồm:

- Extract (Trích xuất dữ liệu):** Thu thập dữ liệu từ nhiều nguồn khác nhau như cơ sở dữ liệu, file log, API, hoặc các dịch vụ bên ngoài. Dữ liệu thu thập thường không đồng nhất về định dạng và cấu trúc.
- Transform (Chuyển đổi dữ liệu):** Dữ liệu trích xuất được làm sạch, chuyển đổi và định dạng lại theo yêu cầu của hệ thống đích. Quá trình này bao gồm chuẩn hóa, tổng hợp, lọc bỏ dữ liệu không hợp lệ và xử lý các mối quan hệ giữa các bảng dữ liệu.
- Load (Tải dữ liệu):** Dữ liệu đã được chuyển đổi sẽ được tải vào kho dữ liệu hoặc một hệ thống phân tích để sử dụng. Quá trình này có thể được thực hiện định kỳ hoặc liên tục tùy thuộc vào nhu cầu.



Hình 32: ETL Diagram

2.5.2 Các kỹ thuật ETL từ cơ bản đến nâng cao

A. Kỹ thuật cơ bản:

- **Phân loại dữ liệu (Data Profiling):** Trước khi thực hiện bất kỳ công việc ETL nào, việc hiểu rõ về dữ liệu bạn đang làm việc là rất quan trọng. Phân loại dữ liệu bao gồm việc kiểm tra chất lượng dữ liệu, khám phá cấu trúc dữ liệu và tìm hiểu mối liên hệ giữa các bảng, cột và giá trị.
- **Làm sạch dữ liệu (Data Cleaning):** Trong quy trình chuyển đổi, dữ liệu thường được làm sạch để loại bỏ giá trị bị lỗi, bị thiếu hoặc không hợp lệ. Ví dụ, một dữ liệu thống kê mô tả về tuổi của khách hàng có thể bị làm sạch bằng cách thay thế các giá trị tuổi không hợp lệ (như "-1" hoặc "999") bằng tuổi trung bình của toàn bộ dữ liệu.
- **Định dạng dữ liệu (Data Formatting):** Dữ liệu thường được định dạng lại để phù hợp với yêu cầu của hệ thống mục tiêu. Ví dụ, một công ty có thể muốn định dạng lại ngày tháng từ dạng "Ngày-Tháng-Năm" thành "Năm-Tháng-Ngày" để phù hợp với cơ sở dữ liệu mục tiêu.

B. Kỹ thuật nâng cao:

- **Tự động hóa ETL (ETL Automation):** Trong một số trường hợp, quy trình ETL cần phải được thực hiện định kỳ, ví dụ hàng ngày hoặc hàng tuần. Trong trường hợp này, việc tự động hóa quá trình ETL là rất quan trọng, giúp tiết kiệm thời gian và công sức.
- **Xử lý dữ liệu lớn (Big Data Processing):** Với sự bùng nổ dữ liệu, việc xử lý dữ liệu lớn là một nhu cầu không thể thiếu trong quy trình ETL. Điều này đòi hỏi sự áp dụng của



các công nghệ như Hadoop hoặc Spark, cũng như việc sử dụng các kỹ thuật phân tách dữ liệu (data partitioning) và song song hóa (parallelism).

- **ETL Real-Time:** Trong một số ngữ cảnh, dữ liệu cần được xử lý ngay lập tức sau khi nó được tạo ra. Đây được gọi là xử lý dữ liệu theo thời gian thực (Real-Time ETL). Điều này đòi hỏi sự áp dụng của các công nghệ như Apache Kafka, Amazon Kinesis, hoặc Google Cloud Pub/Sub, những công nghệ cho phép xử lý dữ liệu streaming.
- **Sử dụng AI và Machine Learning:** Một số công cụ ETL hiện đại có thể sử dụng AI và Machine Learning để cải thiện chất lượng dữ liệu, phân loại dữ liệu, và tự động hóa quy trình ETL. Ví dụ, AI có thể được sử dụng để nhận dạng và loại bỏ các giá trị bất thường trong dữ liệu, còn Machine Learning có thể được sử dụng để dự đoán giá trị bị thiếu dựa trên các giá trị khác trong dữ liệu.

2.5.3 Intial Load ETL

Toàn bộ quá trình Initial Load ETL trong đồ án được làm trong 2 giai đoạn chính:

- Từ dữ liệu gốc sang hệ cơ sở dữ liệu production
- Từ dữ liệu production sang hệ cơ sở dữ liệu analysis

Quy trình này được thực hiện nhằm trích xuất, chuyển đổi, và tích hợp dữ liệu từ một file sao lưu cơ sở dữ liệu (.bak) vào PostgreSQL, đồng thời chuẩn bị dữ liệu để sử dụng cho các mục đích phân tích. Quy trình đảm bảo dữ liệu được làm sạch, tổ chức hợp lý và sẵn sàng phục vụ phân tích hiệu quả. Trong các giai đoạn trên, cần chọn lọc, xử lý các bảng (tables), các đoạn truy vấn đã viết sẵn (views) và loại dữ liệu sao cho phù hợp với model.

2.5.3.1 ETL từ file dữ liệu sang hệ cơ sở dữ liệu production

1. Trích xuất dữ liệu từ file .bak

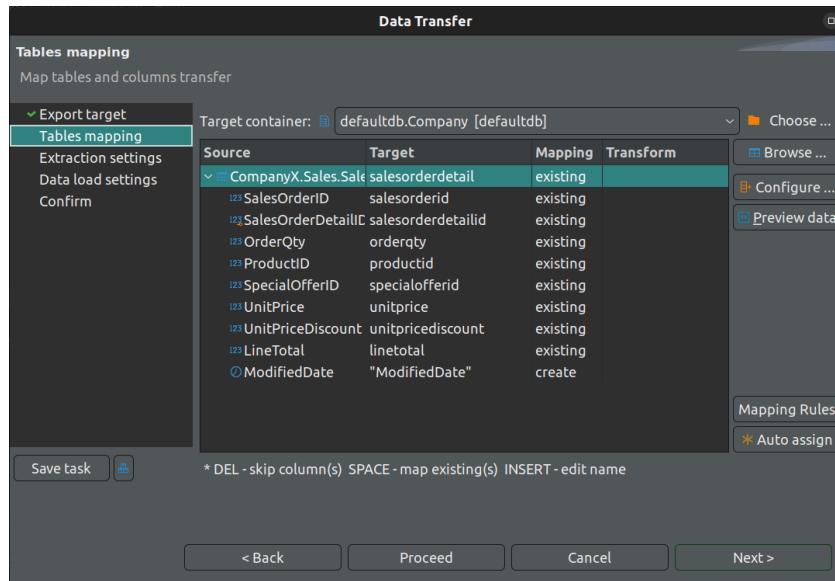
- File .bak là một bản sao lưu chứa toàn bộ thông tin của một công ty.
- Dữ liệu trong file .bak được phục hồi từ hệ thống gốc (SQL Server) và chuyển đổi sang định dạng tương thích với PostgreSQL.
- Quá trình này đảm bảo dữ liệu gốc được bảo toàn và không bị thay đổi trong giai đoạn đầu tiên.

```
RESTORE DATABASE CompanyX
FROM DISK = '/var/opt/mssql/backups/CompanyX.bak'
WITH REPLACE, MOVE 'AdventureWorks2022' TO '/var/opt/mssql/data/CompanyTemp.mdf',
MOVE 'AdventureWorks2022_log' TO '/var/opt/mssql/data/CompanyTemp.ldf';
```

Hình 33: Restore using SQL script

2. Chuyển đổi dữ liệu vào schema Company

- Sau khi trích xuất, dữ liệu được tải vào schema Company trong cơ sở dữ liệu PostgreSQL.
- Schema này đóng vai trò là nơi lưu trữ dữ liệu gốc trong Postgre, đảm bảo rằng mọi thao tác xử lý sau đó không ảnh hưởng trực tiếp đến dữ liệu trong production.



Hình 34: Migration from MySQL to the Company schema in PostgreSQL

- Sử dụng tính năng Export có trong dbeaver-ce để xuất dữ liệu
- Các bước xử lý dữ liệu ban đầu được thực hiện ở đây, bao gồm xóa đi các cột không cần thiết, mà không vi phạm các constraint, index ở trong dữ liệu gốc

	salesorderid	salesorderdetailid	orderqty	productid	specialofferid	unitprice	unitpricediscount	linetotal
1	43,659	1	1	776	1	2,024,99	0	2,024,99
2	43,659	2	3	777	1	2,024,99	0	6,074,98
3	43,659	3	1	778	1	2,024,99	0	2,024,99
4	43,659	4	1	771	1	2,039,99	0	2,039,99
5	43,659	5	1	772	1	2,039,99	0	2,039,99
6	43,659	6	2	773	1	2,039,99	0	4,079,99
7	43,659	7	1	774	1	2,039,99	0	2,039,99
8	43,659	8	3	714	1	28,84	0	86,5212
9	43,659	9	1	716	1	28,84	0	28,8404
10	43,659	10	6	709	1	5,7	0	34,2
11	43,659	11	2	712	1	5,19	0	10,373
12	43,659	12	4	711	1	20,19	0	80,746
13	43,660	13	1	762	1	419,46	0	419,4589
14	43,660	14	1	758	1	874,79	0	874,794
15	43,661	15	1	745	1	809,76	0	809,76
16	43,661	16	1	743	1	714,7	0	714,7043
17	43,661	17	2	747	1	714,7	0	1,429,4086
18	43,661	18	4	712	1	5,19	0	20,746
19	43,661	19	4	715	1	28,84	0	115,3616
20	43,661	20	2	742	1	722,59	0	1,445,1898
21	43,661	21	3	775	1	2,024,99	0	6,074,98
22	43,661	22	2	778	1	2,024,99	0	4,049,988
23	43,661	23	2	711	1	20,19	0	40,373
24	43,661	24	2	741	1	818,7	0	1,637,4
25	43,661	25	4	776	1	2,024,99	0	8,099,976
26	43,661	26	2	773	1	2,039,99	0	4,079,988
27	43,661	27	2	716	1	28,84	0	57,6808
28	43,661	28	2	777	1	2,024,99	0	4,049,988
29	43,661	29	5	708	1	20,19	0	100,9325
30	43,662	30	3	764	1	419,46	0	1,258,3767
31	43,662	31	5	770	1	419,46	0	2,097,2945

Hình 35: The saleorderdetail table data when migrated to the Company schema



3. Xử lý dữ liệu gốc

Một ví dụ xử lý kiểu dữ liệu, cho phù hợp với model đã xây dựng:

```
SELECT "AnnualSales", "AnnualRevenue"
FROM store
WHERE "AnnualSales" IS NOT NULL OR "AnnualRevenue" IS NOT NULL;

UPDATE store
SET
    "AnnualSales" = CAST(REPLACE(REPLACE("AnnualSales)::text, '$', ''), ',', ',')
AS numeric(20, 4)),
    "AnnualRevenue" = CAST(REPLACE(REPLACE("AnnualRevenue)::text, '$', ''), ',', ',')
AS numeric(20, 4))
;
SELECT "AnnualSales", "AnnualRevenue"
FROM store
```

Một ví dụ gom nhóm dữ liệu từ nhiều bảng, tối ưu hiệu suất và không gian:

```
CREATE TABLE "Company".product1 AS(
SELECT
    p."ProductID",
    p."Name",
    p."Description",
    pm."Summary",
    pm."Manufacturer",
    pm."WarrantyPeriod",
    pm."Wheel",
    pm."Saddle",
    pm."Pedal",
    pm."Crankset",
    pm."Style",
    pm."RiderExperience"
FROM "Company".productanddescription p
LEFT JOIN "Company".productmodelcatalogdescription pm
ON p."ProductModel" = pm."Name"
where p."CultureID" = 'en'
);
```

4. Chuyển đổi dữ liệu vào schema chính (public)

- Di chuyển dữ liệu từ schema Company sang schema public bằng python
- Thực hiện các thao tác làm sạch như:
 - Loại bỏ các hàng dữ liệu không hợp lệ hoặc chứa giá trị lỗi.
 - Chuẩn hóa các giá trị trong cột để đảm bảo tính nhất quán.



- Giữ lại các cột cần thiết, xóa các cột không liên quan để giảm thiểu dung lượng lưu trữ và tăng tốc độ truy vấn.

Ví dụ một đoạn import từ schema Company vào public, trong đó đảm bảo kết nối với database, đồng thời chọn đúng Model, cài đặt các primary key, foreign key tương ứng. **Thiết lập kết nối với database và chọn đúng bảng cần truy xuất:**

```
connection.rollback()
cur = connection.cursor()
cur.execute("""
SELECT * FROM "Company".salesorderdetail
ORDER BY "salesorderdetailid" ASC
""")
# Result
rows = cur.fetchall()
cur.close()
```

Và tạo bảng bằng model, lấy các foreign key tương ứng, truyền đúng số lượng tham số:

```
for row in tqdm(rows):
try:
    try:
        product = Product.objects.get(id=row[3])
    except Product.DoesNotExist:
        product = None

    try:
        specialOffer = SpecialOffer.objects.get(id=row[4])
    except SpecialOffer.DoesNotExist:
        specialOffer = None

    try:
        saleOrder = SalesOrderHeader.objects.get(id=row[0])
    except SalesOrderHeader.DoesNotExist:
        saleOrder = None

    salesOrderDetail = SalesOrderDetail(id = row[1],
                                         SalesOrder = saleOrder,
                                         Product = product,
                                         SpecialOffer = specialOffer,
                                         OrderQty = row[2],
                                         UnitPrice = row[5],
                                         UnitPriceDiscount = row[6],
                                         LineTotal = row[7],
                                         CarrierTrackingNumber = row[8],
```

Schemas	Tables	Grid	Text							
		#	id	OrderQty	UnitPrice	UnitPriceDiscount	LineTotal	Product_id	SalesOrder_id	SpecialOffer_id
public	account_employee	1	1	1	2,024.99	0	2,024.994	776	43,659	1
	account_employee_groups	2	2	3	2,024.99	0	6,074.982	777	43,659	1
	account_employee_user_permission	3	3	1	2,024.99	0	2,024.994	778	43,659	1
	analysis_customerdim	4	4	1	2,039.99	0	2,039.994	771	43,659	1
	analysis_employeeadmin	5	5	1	2,039.99	0	2,039.994	772	43,659	1
	analysis_productdim	6	6	2	2,039.99	0	4,079.988	773	43,659	1
	analysis_salesorderdetailfact	7	7	1	2,039.99	0	2,039.994	774	43,659	1
	analysis_salesorderheaderfact	8	8	3	28.84	0	86.5212	[NULL]	43,659	1
	analysis_specialofferdim	9	9	1	28.84	0	28.8404	[NULL]	43,659	1
	auth_group	10	10	6	5.7	0	34.2	709	43,659	1
	auth_group_permissions	11	11	2	5.19	0	10.373	[NULL]	43,659	1
	auth_permission	12	12	4	20.19	0	80.746	[NULL]	43,659	1
	django_admin_log	13	13	1	419.46	0	419.4589	762	43,660	1
	django_content_type	14	14	1	874.79	0	874.794	758	43,660	1
	django_migrations	15	15	1	809.76	0	809.76	745	43,661	1
	django_session	16	16	1	714.7	0	714.7043	743	43,661	1
	sales_customer	17	17	2	714.7	0	1,429.4086	747	43,661	1
	sales_customerindividual	18	18	4	5.19	0	20.746	[NULL]	43,661	1
	sales_customerstore	19	19	4	28.84	0	115.3616	[NULL]	43,661	1
	sales_product	20	20	2	722.59	0	1,445.1898	742	43,661	1
	sales_salesorderdetail	21	21	3	2,024.99	0	6,074.982	775	43,661	1
	sales_salesorderheader	22	22	2	2,024.99	0	4,049.988	778	43,661	1
	sales_specialoffer	23	23	2	20.19	0	40.373	[NULL]	43,661	1
	sales_specialofferproduct	24	24	2	818.7	0	1,637.4	741	43,661	1
	sales_specialofferproduct	25	25	4	2,024.99	0	8,099.976	776	43,661	1
	sales_specialofferproduct	26	26	2	2,039.99	0	4,079.988	773	43,661	1
	sales_salesorderdetail	27	27	2	28.84	0	57.6808	[NULL]	43,661	1
	sales_salesorderheader	28	28	2	2,024.99	0	4,049.988	777	43,661	1
	sales_specialoffer	29	29	5	20.19	0	100.9325	708	43,661	1
	sales_specialofferproduct	30	30	3	419.46	0	1,258.3767	764	43,662	1
	sales_specialofferproduct	31	31	5	419.46	0	2,097.2945	770	43,662	1

Hình 36: The saleorderdetail table data when migrated to the public schema

)

```

salesOrderDetail.save()
except Exception as e:
    print(e)
    pass

```

5. Những vấn đề và giải pháp

- Xử lý lỗi dữ liệu:

- Một số hàng chứa dữ liệu không hợp lệ (ví dụ: giá trị null ở các cột bắt buộc hoặc dữ liệu sai định dạng).
- Giải pháp: Loại bỏ các hàng này và ghi nhận vào log để theo dõi.

```
/home/dn/PycharmProjects/DA_HTTT_BE/.venv/lib/python3.10/site-packages/django/db/models/fields/_init_.py:1665: Run
timewarning: DateTimeField SpecialOffer.EndDate received a naive datetime (2014-11-30 00:00:00) while time zone supp
ort is active.
```

Hình 37: A datetime format error

- Khối lượng dữ liệu lớn:

- Do file .bak chứa toàn bộ thông tin của công ty, quá trình xử lý yêu cầu tối ưu hóa hiệu năng.
- Giải pháp: Sử dụng staging schema (Company) làm vùng đệm để tách biệt các bước xử lý.

- Cột dữ liệu không cần thiết:

- Nhiều cột trong dữ liệu nguồn không phục vụ mục đích phân tích.



- Giải pháp: Loại bỏ các cột này trong bước chuyển đổi bằng cách sử dụng câu lệnh SQL và script Python.

2.5.3.2 ETL từ hệ cơ sở dữ liệu production sang hệ cơ sở dữ liệu analysis

Quy trình ETL từ **production** (hoặc **public**) sang phân tích (**analysis**) được thiết kế nhằm chuẩn bị dữ liệu tối ưu cho các bài toán phân tích và báo cáo. Đây là bước quan trọng trong việc chuyển đổi dữ liệu từ dạng "hoạt động" (operational) sang dạng "phân tích" (analytical).

Phương pháp trích xuất:

- Sử dụng các truy vấn SQL và python script để chọn lọc dữ liệu phù hợp với mục đích phân tích.
- Lọc bỏ các bản ghi không cần thiết dựa trên các điều kiện cụ thể (ví dụ: dữ liệu cũ, lỗi, hoặc không liên quan).

Tương tự như phần trước, tuy nhiên do lần này các dữ liệu production đều nằm cùng trong Postgre và cùng một schema, việc chuyển dữ liệu không còn phức tạp, nhưng vẫn đòi hỏi việc chính xác vì đây là dữ liệu quan trọng được dùng để phân tích và làm BI.

```
lst = SalesOrderDetail.objects.all()
for item in tqdm(lst):

    try:
        salesOrder = SalesOrderHeaderFact.objects.get(id = item.SalesOrder.id)
    except:
        salesOrder = None

    try:
        product = ProductDim.objects.get(id = item.Product.id)
    except:
        product = None

    try:
        specialOffer = SpecialOfferDim.objects.get(id = item.SpecialOffer.id)
    except:
        specialOffer = None

    instance = SalesOrderDetailFact(
            id=item.id,
            OrderQty=item.OrderQty,
            LineTotal=item.LineTotal,
            Product=product,
            SpecialOffer=specialOffer,
            SalesOrder=salesOrder
```

```
)
# Save instance
instance.save()
```

	id	OrderQty	LineTotal	Product_id	SpecialOffer_id	SalesOrder_id
1	1	1	2,024.994	776	1	43,659
2	2	3	6,074.982	777	1	43,659
3	3	1	2,024.994	778	1	43,659
4	4	1	2,039.994	771	1	43,659
5	5	1	2,039.994	772	1	43,659
6	6	2	4,079.988	773	1	43,659
7	7	1	2,039.994	774	1	43,659
8	8	3	86.5212	[NULL]	1	43,659
9	9	1	28.8404	[NULL]	1	43,659
10	10	6	34.2	709	1	43,659
11	11	2	10.373	[NULL]	1	43,659
12	12	4	80.746	[NULL]	1	43,659
13	13	1	419.4589	762	1	43,660
14	14	1	874.794	758	1	43,660
15	15	1	809.76	745	1	43,661
16	16	1	714.7043	743	1	43,661
17	17	2	1,429.4086	747	1	43,661
18	18	4	20.746	[NULL]	1	43,661
19	19	4	115.3616	[NULL]	1	43,661
20	20	2	1,445.1898	742	1	43,661
21	21	3	6,074.982	775	1	43,661
22	22	2	4,049.988	778	1	43,661
23	23	2	40.373	[NULL]	1	43,661
24	24	2	1,637.4	741	1	43,661
25	25	4	8,099.976	776	1	43,661
26	26	2	4,079.988	773	1	43,661
27	27	2	57.6808	[NULL]	1	43,661
28	28	2	4,049.988	777	1	43,661
29	29	5	100.9325	708	1	43,661
30	30	3	1,258.3767	764	1	43,662
31	31	5	2,097.2945	770	1	43,662

Hình 38: The saleorderdetail table data when migrated to the analysis table

2.5.4 Streaming ETL

Để hệ cơ sở dữ liệu phân tích có thể cập nhật các dữ liệu mới nhất từ hệ cơ sở dữ liệu production, quy trình ETL được áp dụng như sau:

- Với mỗi thao tác trên hệ cơ sở dữ liệu production, thao tác đó sẽ được thực hiện trên hệ cơ sở dữ liệu analysis. Để đảm bảo không có sự khác nhau giữa hai hệ cơ sở dữ liệu, thao tác phải được thực hiện thành công trước khi được thực hiện trên hệ cơ sở dữ liệu analysis.
- Các bản ghi (record) trong các bảng trong hệ cơ sở dữ liệu analysis phải có cùng khóa chính với các bản ghi trong hệ cơ sở dữ liệu để đảm bảo tính mạch lạc trong hai hệ cơ sở dữ liệu.
- Dữ liệu được lưu trong hệ cơ sở dữ liệu production sẽ được chọn lọc các trường nhất định hoặc được biến đổi trước khi được lưu vào hệ cơ sở dữ liệu analysis.

Ví dụ luồng ETL mẫu cho thao tác tạo record mới trong bảng SpecialOffer (production database):

```
def CreateSpecialOfferCRUD(request):
    # Converting request.body to dictionary type
    dict = request.body.decode("UTF-8")
    specialOfferInfo = json.loads(dict)

    # Get special offer info from dictionary
    Description = specialOfferInfo['Description']
    DiscountPct = specialOfferInfo['DiscountPct']
    Type = specialOfferInfo['Type']
    StartDate = specialOfferInfo['StartDate']
    EndDate = specialOfferInfo['EndDate']
    MinQty = specialOfferInfo['MinQty']
    MaxQty = specialOfferInfo['MaxQty']

    # Create new special offer object
    specialOffer = SpecialOffer(
        Description=Description,
        DiscountPct=DiscountPct,
        Type=Type,
        StartDate=StartDate,
        EndDate=EndDate,
        MinQty=MinQty,
        MaxQty=MaxQty)

    # Save new special offer object
    specialOffer.save()

    # ETL
    CreateSpecialOfferDimETL(specialOffer)
```

Hình 39: The process of creating a record in the production database

Khi backend nhận yêu cầu tạo record mới trong bảng SpecialOffer, tất cả các thông tin chi tiết của bản ghi sẽ được trích từ yêu cầu và lưu trong hệ cơ sở dữ liệu production. Sau khi bản ghi đã được lưu trong hệ cơ sở dữ liệu production thành công, thao tác ETL sẽ được thực hiện để lưu thông tin mới đó trong hệ cơ sở dữ liệu analysis.

```
def CreateSpecialOfferDimETL(object):
    # Create new object
    specialOfferDim = SpecialOfferDim(
        id=object.id,
        Description=object.Description,
        DiscountPct=object.DiscountPct)

    # Save object
    specialOfferDim.save()
```

Hình 40: ETL process in the analysis database

Khi bản ghi tương ứng được tạo trong hệ cơ sở dữ liệu phân tích, các trường quan trọng sẽ được chọn lọc. Cụ thể, trong trường hợp này, chỉ 3 trường được chọn khi lưu trong hệ cơ sở dữ liệu analysis: id, Description và DiscountPct.



Dưới đây là thao tác thêm record vào bảng SpecialOffer và kết quả:

Tạo ưu đãi

Description *	: Test ETL SpecialOffer
Discount Percentage *	: 0.3
Type *	: Test discount
Start Date *	: 2024-05-01 00:00:00.000
End Date *	: 2024-05-05 00:00:00.000
Min Quantity *	: 10
Max Quantity *	: 15
<button>Tạo</button> <button>Hủy</button>	

Hình 41: Fill in information when creating a new SpecialOffer

Query Query History

```
1 SELECT * FROM public.sales_specialoffer
2 ORDER BY id DESC LIMIT 100
```

Data Output Messages Notifications

			Type	StartDate	EndDate	MinQty	MaxQty
1	35	Test ETL SpecialOffer	Test discount	2024-05-01 00:00:00+00	2024-05-05 00:00:00+00	10	15

Hình 42: A new record created in the SpecialOffer table (production database)

Query Query History

```
1 SELECT * FROM public.analysis_specialofferdim
2 ORDER BY id DESC LIMIT 100
```

Data Output Messages Notifications

		DiscountPct	Description
1	35	0.3000	Test ETL SpecialOffer

Hình 43: A new record created in the SpecialOfferDim table (analysis database)



2.6 Business Intelligence Dashboard

2.6.1 Giới thiệu

Dashboard là công cụ hoặc giao diện hiển thị giúp cung cấp một cái nhìn tổng quan về các chỉ số liên quan đến một mục tiêu hoặc một quá trình kinh doanh nào đó. Dashboard sử dụng số liệu, các thống kê và biểu đồ để biểu diễn thông tin một cách tổng quan và dễ hiểu. Hiện nay, dashboard có nhiều loại như: business dashboard (bảng phân tích kinh doanh), KPI dashboard (bảng theo dõi KPI), performance dashboard (bảng hiển thị hiệu suất), financial dashboard (bảng theo dõi tài chính),... Mỗi loại Dashboard sẽ có những thành phần khác nhau tùy mục đích doanh nghiệp sử dụng, nhưng phần lớn, chúng đều có cấu trúc và chức năng tương tự:

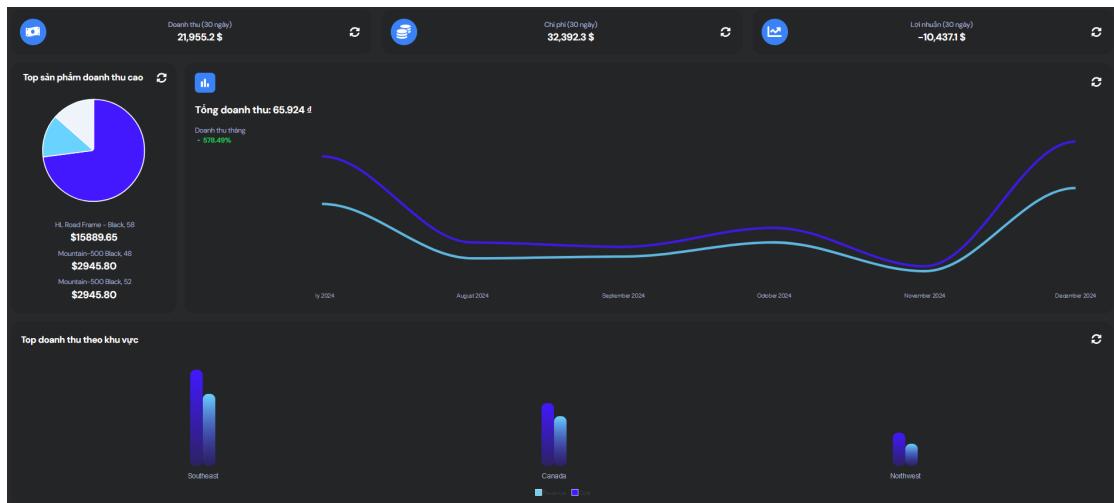
- Biểu đồ:** Thông tin được biểu diễn trực quan. Nhiều loại biểu đồ được sử dụng như: biểu đồ cột, biểu đồ tròn hoặc biểu đồ đường,... Các đồ thị này biểu diễn thông tin phức tạp một cách ngắn gọn, giúp người dùng hiểu được dữ liệu nhanh chóng.
- Điểm dữ liệu và thước đo:** Hầu hết các dashboard theo dõi các điểm dữ liệu hoặc KPI. Theo dõi các chi tiết này không chỉ giúp nhân viên biết tình hình hoặc hiệu suất hiện tại của quá trình kinh doanh mà còn khuyến khích các ý tưởng và hoạt động để gia tăng chỉ số.
- Biểu tượng:** Các biểu tượng được sử dụng rất nhiều trong dashboard để biểu diễn một ý nghĩa nào đó. Ví dụ biểu tượng mũi tên chỉ lên màu xanh lá có thể được dùng để hiển thị sự gia tăng tích cực của một chỉ số nào đó. Ngược lại, các chữ số màu đỏ có thể được dùng để diễn thị một xu hướng tiêu cực nào đó trong dữ liệu.
- Báo cáo:** Báo cáo hoặc bản tóm tắt có thể giúp hiển thị tập trung vào các thông tin cụ thể nào đó trong quá trình kinh doanh giúp người dùng xác định các xu hướng và thông tin cần thiết mà không tốn thời gian.
- Lịch và các thông báo:** Dashboard là một trong những nơi cung cấp người dùng các thông tin quan trọng nhất nên lịch và các thông báo có thể giúp người dùng theo dõi và nhắc nhở về các mục tiêu và việc cần phải làm.

2.6.2 Các nội dung hiển thị

Qua tìm hiểu và xem xét hệ cơ sở dữ liệu và các yêu cầu về sales, nhóm đã xác định các thông tin quan trọng cần được hiển thị trên sales dashboard trong hệ thống như sau:

- Lợi nhuận, doanh thu và chi phí:** Giúp người dùng nắm được cái nhìn tổng quan của quá trình kinh doanh của doanh nghiệp trong tháng vừa qua. Ngoài ra, thông tin tổng quan về lợi nhuận, doanh thu và chi phí trong 6 tháng tính đến thời điểm hiện tại cũng cần được hiển thị.
- Sản phẩm bán chạy:** Thông tin về 3 sản phẩm được bán chạy nhất trong 30 ngày vừa qua và các thông số cụ thể của từng sản phẩm (lợi nhuận, doanh thu và chi phí).
- Khu vực kinh doanh:** Thông tin về các khu vực kinh doanh tiêu biểu trong khoảng thời gian gần đây, dựa vào doanh thu của khu vực và số lượng hàng hóa được phân phối đến khu vực đó.

2.6.3 Thiết kế dashboard



Hình 44: Application Dashboard

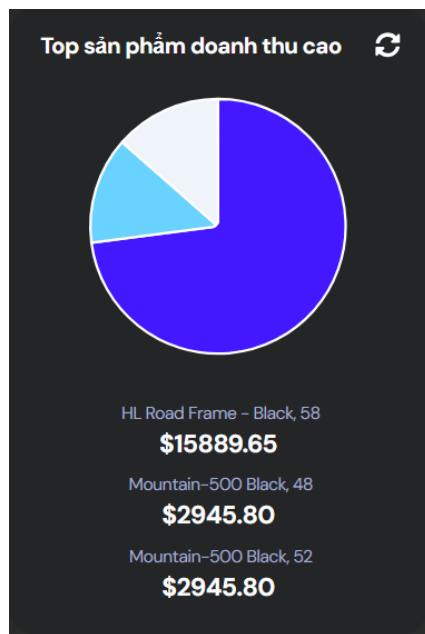
Về tổng quan, dashboard của ứng dụng sẽ có 4 thành phần. Đó là:

- **Doanh thu, chi phí và lợi nhuận trong 30 ngày vừa qua:** Thành phần này đóng vai trò như thông số KPI của công ty trong thời gian gần. Thành phần được thiết kế đơn giản và dễ quan sát (đặt trên đầu trang).



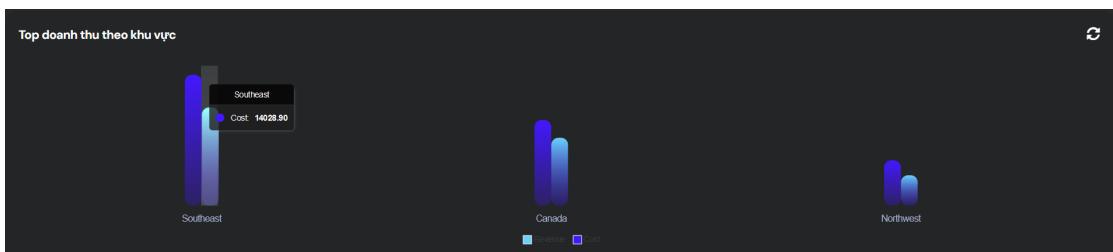
Hình 45: Doanh thu, chi phí và lợi nhuận trong 30 ngày vừa qua

- **Sản phẩm bán chạy trong 30 ngày vừa qua:** Thành phần này giúp người dùng theo dõi và kịp thời nắm bắt xu hướng thị trường trong thời gian gần đây để tiếp thị và quảng bá với khách hàng. Thành phần này bao gồm thông tin về 3 sản phẩm bán chạy nhất với biểu đồ tròn giúp người dùng có thể dễ dàng so sánh xu hướng sản phẩm.



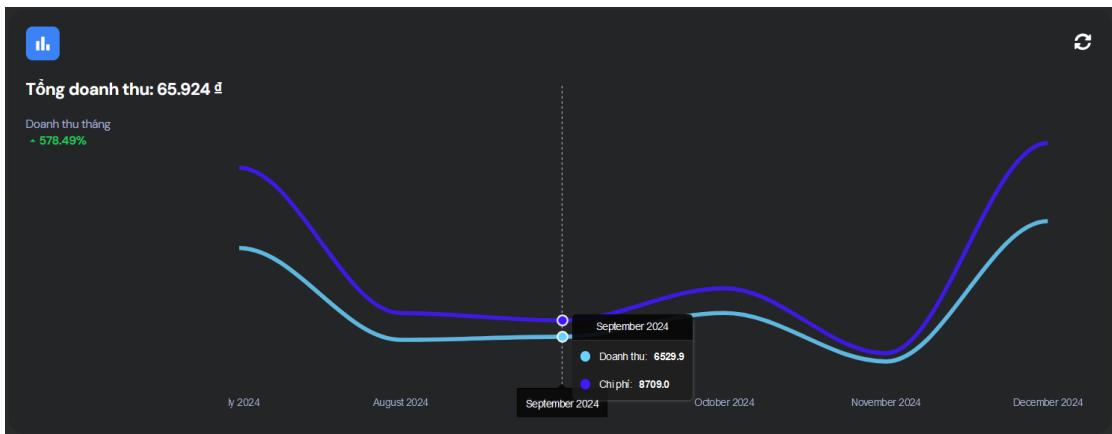
Hình 46: Sản phẩm bán chạy trong 30 ngày vừa qua

- **Vùng thị trường thịnh hành nhất trong 30 ngày vừa qua:** Thành phần này giúp người dùng theo dõi 3 vùng có nhu cầu cao nhất trong 30 ngày vừa qua để người dùng có thể lập kế hoạch và thực hiện việc tiếp thị một cách hiệu quả. Thành phần này bao gồm



Hình 47: Vùng thị trường thịnh hành nhất trong 30 ngày vừa qua

- **Thông số doanh thu và chi phí tổng quát trong 6 tháng vừa qua:** Thành phần này giúp người dùng có thể theo dõi hiệu suất của công ty trong vòng 6 tháng vừa qua, nhận biết các xu hướng xảy ra trong thời gian dài và đưa ra những giải pháp, phương án sales phù hợp. Thông tin được hiển thị dưới dạng biểu đồ đường giúp biểu diễn sự tiến triển của doanh thu và chi phí của công ty.



Hình 48: Thông số doanh thu và chi phí tổng quát trong 6 tháng vừa qua

2.7 Phân tích dữ liệu

2.7.1 Giới thiệu

Qua tìm hiểu và thảo luận, nhóm đã xác định các vấn đề và nhu cầu sau trong một hệ thống thông tin:

- Nhu cầu truy vấn và phân tích dữ liệu:** Trong một hệ thống thông tin, luôn luôn tồn tại nhu cầu truy vấn và phân tích dữ liệu. Tuy nhiên, để thực hiện được các thao tác đó, người dùng cần sử dụng các công cụ truy vấn nhưdbeaver, pgadmin4,... và viết các câu lệnhsqlđể có thể truy vấn các thông tin cần thiết. Ngoài ra, để phân tích dữ liệu người dùng cần phải biết cách sử dụng các công cụ hoặc cách sử dụng mã Python. Trong nhiều trường hợp, người dùng bình thường có thể không có đủ khả năng để thực hiện các thao tác trên.
- Tính hạn chế của tính năng phân tích thông dụng:** Vấn đề vừa nêu trên có thể được giải quyết bằng cách thêm một tính năng phân tích một vấn đề cụ thể vào hệ thống thông tin. Tuy nhiên, tính năng đó chỉ có thể thử sử dụng một số tài nguyên hoặc thông tin cụ thể. Ví dụ, một tính năng dự đoán doanh thu của tổ chức trong 3 tháng tiếp theo tùy theo vùng chỉ sử dụng dữ liệu trong các bảng TerritoryDim (vùng kinh doanh) và SalesOrderHeaderFact (đơn hàng).

Để có thể giải quyết các hạn chế và nhu cầu trên, nhóm đã quyết định sử dụng các tính năng của AI như: prompt engineering, chat completion và function calling của nền tảng Google AI Studio để hiện thực tính năng phân tích của hệ thống:

- Chat completion:** Là một loại mô hình ngôn ngữ có thể tạo ra câu trả lời sau khi người dùng cung cấp yêu cầu. Câu trả lời có thể là tổng hợp của đoạn văn bản, đoạn mã hoặc giải thích,...
- Prompt engineering:** Là hoạt động thiết kế yêu cầu để mô hình ngôn ngữ có thể đưa ra câu trả lời phù hợp với ngữ cảnh của người dùng. Mô hình ngôn ngữ có thể được cung cấp các chỉ dẫn hoặc thông tin để có thể tạo ra câu trả lời đúng nhất.

- **Function calling:** Một tính năng của nhiều mô hình ngôn ngữ giúp chúng tương tác với các công cụ hoặc API được cung cấp bởi nhà phát triển giúp nâng cao khả năng phản hồi của mô hình ngôn ngữ. Nhờ vào tính năng này, mô hình ngôn ngữ có thể thực hiện các hành động để trợ giúp cho câu trả lời hoặc thực hiện các hành động như câu trả lời cho người dùng.

Các tính năng trên sẽ được sử dụng để tạo các câu lệnh SQL để truy vấn dữ liệu, điều hướng thích hợp với nhu cầu của người dùng, tạo báo cáo cho câu hỏi của người dùng và tóm tắt để người dùng có thể hiểu nội dung và tùy chỉnh câu hỏi hoặc gợi ý cho thích hợp.

2.7.2 Thiết kế tính năng phân tích dữ liệu

2.7.2.a Thiết kế câu trả lời

Câu trả lời được thiết kế theo báo cáo (report). Báo cáo hoặc câu trả lời sẽ bao gồm 3 thành phần chính:

- **text (đoạn văn bản):** Thành phần này bao gồm đoạn văn bản được tạo bởi mô hình ngôn ngữ Gemini. Đoạn văn bản có thể là một câu trả lời cho một câu hỏi, một giải thích cho câu truy vấn (query) hoặc một mô tả, diễn giải cho một biểu đồ được vẽ,...

```
The analysis_salesorderdetailfact table has the following fields:  
  
id (primary key)  
OrderQty (int)  
LineTotal (float)  
Product_id (foreign key)  
SpecialOffer_id (foreign key)  
SalesOrder_id (foreign key)
```

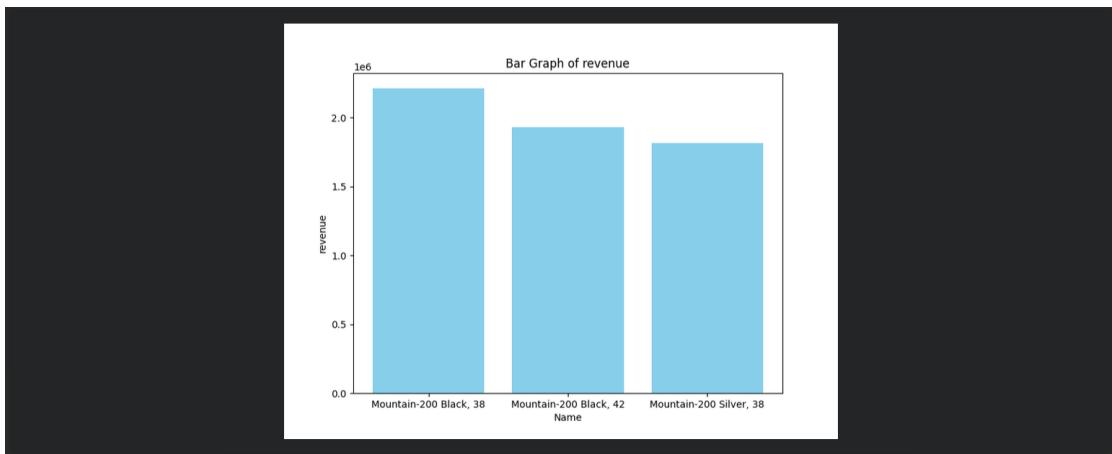
Hình 49: Đoạn văn bản trả lời cho câu hỏi về các trường của bảng analysis_salesorderdetailfact

- **queryResult (kết quả truy vấn):** Thành phần này gồm có câu truy vấn tạo bởi mô hình ngôn ngữ để chạy trên hệ cơ sở dữ liệu. Câu truy vấn được tạo tùy vào yêu cầu của người dùng. Ngoài ra, thành phần này còn có kết quả của câu truy vấn đó dưới dạng bảng.

QUERY:	
<pre>SELECT p."Name", SUM(sod."LineTotal") AS Revenue FROM analysis_salesorderdetailfact sod JOIN analysis_productdim p ON sod."Product_Id" = p."id" JOIN analysis_salesorderheaderfact soh ON sod."SalesOrder_Id" = soh."Id" WHERE EXTRACT(YEAR FROM soh."OrderDate") = 2013 GROUP BY p."Name" ORDER BY Revenue DESC LIMIT 3;</pre>	
QUERY RESULT	
Name	revenue
Mountain-200 Black, 38	2212974.7827
Mountain-200 Black, 42	1932388.2907
Mountain-200 Silver, 38	1815673.0932

Hình 50: Kết quả truy vấn cho câu hỏi về các sản phẩm bán chạy

- **graphResult (biểu đồ):** Thành phần này gồm có biểu đồ minh họa cho dữ liệu được truy vấn.



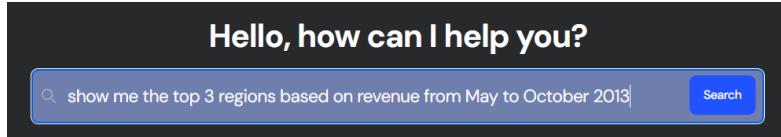
Hình 51: Kết quả biểu đồ cho câu hỏi về các sản phẩm bán chạy

Sau khi nhận được câu trả lời của người dùng, hệ thống sẽ tạo báo cáo sử dụng các thành phần trên để biểu diễn câu trả lời hoặc các thông tin quan trọng phù hợp với yêu cầu của người dùng. Một báo cáo, câu trả lời có thể có trình tự thành phần như sau: kết quả truy vấn - trình bày thông tin mà người dùng yêu cầu, văn bản - giải thích câu truy vấn, biểu đồ - trình bày kết quả truy vấn dưới dạng biểu đồ, văn bản - diễn giải biểu đồ, xác định các xu hướng trong biểu đồ.

2.7.2.b Quy trình phân tích và truy vấn dữ liệu

Hệ thống sử dụng kết hợp các API Gemini của Google AI Studio và các công cụ, thư viện trong Python để tạo ra câu trả lời cho người dùng. Về tổng quát, quy trình phân tích của hệ thống sẽ có những bước sau:

- Người dùng sẽ nhập câu hỏi theo ý muốn. Câu hỏi của người dùng sẽ được gửi đến backend để xử lý.



Hình 52: Người dùng nhập và gửi câu hỏi

- Backend sẽ nhận câu hỏi của người dùng và tính năng chat completion được sử dụng để xác định yêu cầu của người dùng. Tính năng chat completion sẽ được cung cấp chỉ dẫn (instruction) các thông tin về lược đồ hệ cơ sở dữ liệu (database schema) để có thể tạo các lệnh truy vấn phù hợp.

```
You are named Faust. You are an assistant for a sales organization. In the Postgres database, in schema "public", there are these tables as a datamart.  
Those tables follow the structure following:  
  
<table><analysis_salesorderheaderfact>: Contain information about sale orders  
  <field>id (primary key)</field>  
  <field>OrderDate (date): The date that the order is ordered.</field>  
  <field>SubTotal (float): The total cost of the items of this order.</field>  
  <field>TaxAmt (float): Tax cost of this order.</field>  
  <field>Freight (float): The cost of shipping this order.</field>  
  <field>TotalDue (float): The overall total cost of this order. TotalDue = Subtotal + TaxAmt + Freight.</field>  
  <field>Employee_id (foreign key): Foreign key to analysis_employeedim table. This field describe the employee who created the order.</field>  
  <field>Customer_id (foreign key): Foreign key to analysis_customerdim table. This field describe the customer that placed the order.</field>  
</table>  
  
<table><analysis_salesorderdetailfact>: Contain information about the types of products in a sale order  
  <field>id (primary key)</field>  
  <field>OrderQty (int): The number of the products of this type in the order.</field>  
  <field>LineTotal (float): The total cost of this product type in the order. LineTotal = OrderQty x ListPrice x DiscountPct.</field>  
  <field>Product_id (foreign key): Foreign key to analysis_productdim table.</field>  
  <field>SpecialOffer_id (foreign key): Foreign key to analysis_specialofferdim table. This field describe the special offer that was applied to this type  
  <field>SalesOrder_id (foreign_key): Foreign key to analysis_salesorderheaderfact table.</field>  
</table>
```

Hình 53: Một phần chỉ dẫn cho mô hình ngôn ngữ về lược đồ hệ cơ sở dữ liệu

Ngoài ra, tại đây, tính năng function calling cũng được sử dụng để định nghĩa hai hàm: QueryPostgresDatamart - dùng để truy vấn dữ liệu và PredictFromQueryData - dùng để dự đoán dữ liệu trong tương lai. Mô hình sẽ lựa chọn câu trả lời hoặc hàm thích hợp với yêu cầu của người dùng và điền các tham số thích hợp để hệ thống gọi hàm.



```
def QueryPostgresDatamart(query: str):
    """Create a Postgres query as parameter to query the information asked by the user in the Postgres datamart.
    The function will search the postgres database using the query and return the user the table result.
    After that, the function will draw a graph based on the graphType parameter.
    For bar graph and pie graph, the first column of the result in the query MUST be categories or labels (the x axis).
    For line graph, the first column (usually a date column) is the x axis.

    Args:
        query: Create a query that will be queried in the Postgres datamart. The query must be on a single line.
        Only generate the syntactically correct and bare query, without any newline or other special characters.
        You can rename the columns of the query result to match the content.

    Returns:
        The result of the query (and maybe a graph)
    """
    return []
```

Hình 54: Prototype hàm QueryPostgresDatamart được cung cấp cho mô hình

```
def PredictFromQueryData(query: str, step: int):
    """Create a Postgres query as parameter to query the data needed in order to predict user's desired future trends.
    The result of the query must be the history data leading up. This function will predict future trends based on the data queried
    on the datamart using the query you created

    Args:
        query: Create a query that will be queried in the Postgres datamart. The query must be on a single line.
        Only generate the syntactically correct and bare query, without any newline or other special characters.
        The result table of the query must have the last column as the column used as data for the algorithm.
        step: How many steps into the future that the algorithm should predict

    Returns:
        Prediction result for the user's prompt
    """
    return []
```

Hình 55: Prototype hàm PredictFromQueryData được cung cấp cho mô hình

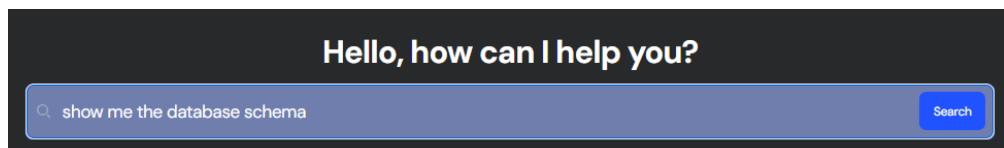
- Sau khi nhận được kết quả điều hướng từ mô hình. Hệ thống sẽ thực hiện các thao tác sau:
 - **Nếu mô hình trả về văn bản:** Nếu mô hình ngôn ngữ chỉ trả về văn bản, hệ thống sẽ tạo câu trả lời bao gồm chỉ một thành phần văn bản.
 - **Nếu mô hình trả về gọi hàm QueryPostgresDatamart:** Nếu mô hình ngôn ngữ trả về gọi hàm QueryPostgresDatamart, hệ thống sẽ thực hiện truy vấn dữ liệu sử dụng tham số query được truyền từ mô hình. Kết quả của truy vấn sẽ được thể hiện bằng thành phần queryResult. Sau đó, mô hình ngôn ngữ sẽ được sử dụng để giải thích câu truy vấn được tạo để người dùng có thể điều chỉnh yêu cầu để có kết quả như mong muốn. Phần giải thích truy vấn sẽ được trình bày bằng thành phần text. Cuối cùng, biểu đồ sẽ được vẽ và diễn giải biểu đồ sẽ được tạo bởi mô hình ngôn ngữ. Hai thành phần này sẽ được thể hiện bằng graphResult và text. Báo cáo cho trường hợp này sẽ bao gồm: queryResult - kết quả truy vấn, text - giải thích câu truy vấn, graphResult - hình vẽ biểu đồ và text - diễn giải biểu đồ.
 - **Nếu mô hình trả về gọi hàm PredictFromQueryData:** Nếu mô hình ngôn ngữ trả về gọi hàm PredictFromQueryData, hệ thống sẽ thực hiện truy vấn dữ liệu sử dụng tham số query được truyền từ mô hình. Kết quả của truy vấn sẽ được thể hiện bằng thành phần queryResult. Mô hình ngôn ngữ sẽ được sử dụng để giải thích câu

truy vấn được tạo để người dùng có thể điều chỉnh yêu cầu để có kết quả như mong muốn. Phần giải thích truy vấn sẽ được trình bày bằng thành phần text. Sau đó, hệ thống sẽ sử dụng kết quả truy vấn để phân tích và dự đoán tùy vào yêu cầu của người dùng. Kết quả sau khi phân tích sẽ được vẽ thành biểu đồ và biểu diễn bằng thành phần graphResult. Cuối cùng, diễn giải cho biểu đồ sẽ được tạo bởi mô hình và thể hiện bằng thành phần text. Báo cáo cho trường hợp này sẽ bao gồm: queryResult - kết quả truy vấn, text - giải thích câu truy vấn, graphResult - biểu đồ biểu thị kết quả phân tích và text - diễn giải biểu đồ.

2.7.3 Demo

2.7.3.a Chat completion

Ta sẽ kiểm tra tính năng chat completion qua câu hỏi "Show me the database schema" trong ứng dụng.



Hình 56: Câu hỏi được gửi bởi người dùng

Câu trả lời nhận được từ backend:

The screenshot shows a terminal window with a dark background and white text. The title bar says "Report: 'show me the database schema'". The content starts with a description of the database schema, mentioning seven tables: analysis_salesorderheaderfact, analysis_salesorderdetailfact, analysis_productdim, analysis_specialofferdim, analysis_employeedim, analysis_customerdim, and analysis_territorydim. It then provides detailed descriptions for each table, such as the fields in analysis_salesorderheaderfact and the calculations for LineTotal in analysis_salesorderdetailfact. The tables are linked through foreign keys, with analysis_salesorderheaderfact linking to analysis_employeedim and analysis_customerdim, and analysis_salesorderdetailfact linking to analysis_salesorderheaderfact, analysis_productdim, and analysis_specialofferdim. The schema is described as a star schema centered around analysis_salesorderheaderfact and analysis_salesorderdetailfact.

Hình 57: Câu trả lời nhận được từ hệ thống



2.7.3.b Truy vấn dữ liệu

Ta sẽ kiểm tra tính năng truy vấn dữ liệu qua câu hỏi "Show me the top 5 products based on revenue in 2014" trong ứng dụng.



Hình 58: Câu hỏi được gửi bởi người dùng

Câu trả lời nhận được từ backend:

Report: "show me the top 5 products based on revenue in 2014"

QUERY:
SELECT p.Name, SUM(sod.LineTotal) AS Revenue FROM analysis_productdim p JOIN analysis_salesorderdetailfact sod ON p.Id = sod.Product_Id JOIN analysis_salesorderheaderfact soh ON sod.SalesOrder_Id = soh.Id WHERE EXTRACT(YEAR FROM soh.OrderDate) = 2014 GROUP BY p.Name ORDER BY Revenue DESC LIMIT 5;

QUERY RESULT

Name	revenue
Mountain-200 Black, 38	1003904.897
Mountain-200 Black, 42	847338.7659
Mountain-200 Silver, 38	838223.523
Mountain-200 Silver, 46	74390.3615
Mountain-200 Black, 46	724539.359

This query identifies the top 5 products with the highest revenue in 2014.
Here's a breakdown:
Joins: It joins three tables (analysis_productdim, analysis_salesorderdetailfact, analysis_salesorderheaderfact) to link products to sales order details and then to the sales order header.
Filtering: It filters sales orders to include only those from 2014.
Aggregation: It groups the results by product name and calculates the sum of LineTotal for each product, representing the revenue for that product.
Ordering and Limiting: It orders the results by revenue in descending order and limits the output to the top 5 products.
In essence, the query provides a ranking of the best-selling products in terms of revenue generated during 2014.

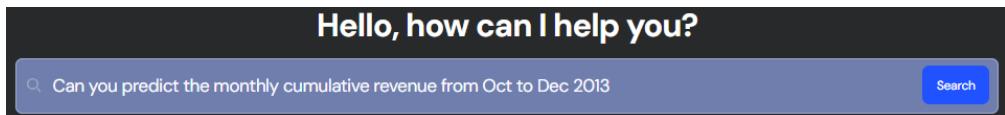
Bar Graph of revenue

The provided data represents the revenue generated from sales of different variations of a product, likely a bicycle model named "Mountain-200" across various colors (Black, Silver) and sizes (38, 42, 46).
Trends:
The data shows a clear trend of higher revenue for the Mountain-200 Black, 38 model compared to other variations. Revenue gradually decreases as we consider other size and color combinations. This suggests that size 38 in Black is the most popular variation.
Examples and Explanations:
Mountain-200 Black, 38: This variation generated the highest revenue (\$1003.904.82). This could be due to several factors: higher demand for this specific size and color combination, effective marketing campaigns targeted at this demographic, or potentially a favorable price point.
Mountain-200 Black, 42 & 46, and Mountain-200 Silver, 38 & 46: These variations all show a significantly lower revenue compared to the Mountain-200 Black, 38. This may indicate that either size 42 and 46 are less popular sizes, the silver color is less preferred, or a combination of these factors. The decrease in revenue is probably due to less demand for these less popular variations.
In summary, the data suggests that product variations greatly influence revenue. The "Mountain-200 Black, 38" variation is a clear top performer, and understanding why it is most popular could be beneficial for future marketing and inventory management decisions. Further analysis could be carried out to determine the reasons behind the popularity of the best performing variation and the underperformance of others.

Hình 59: Câu trả lời nhận được từ hệ thống

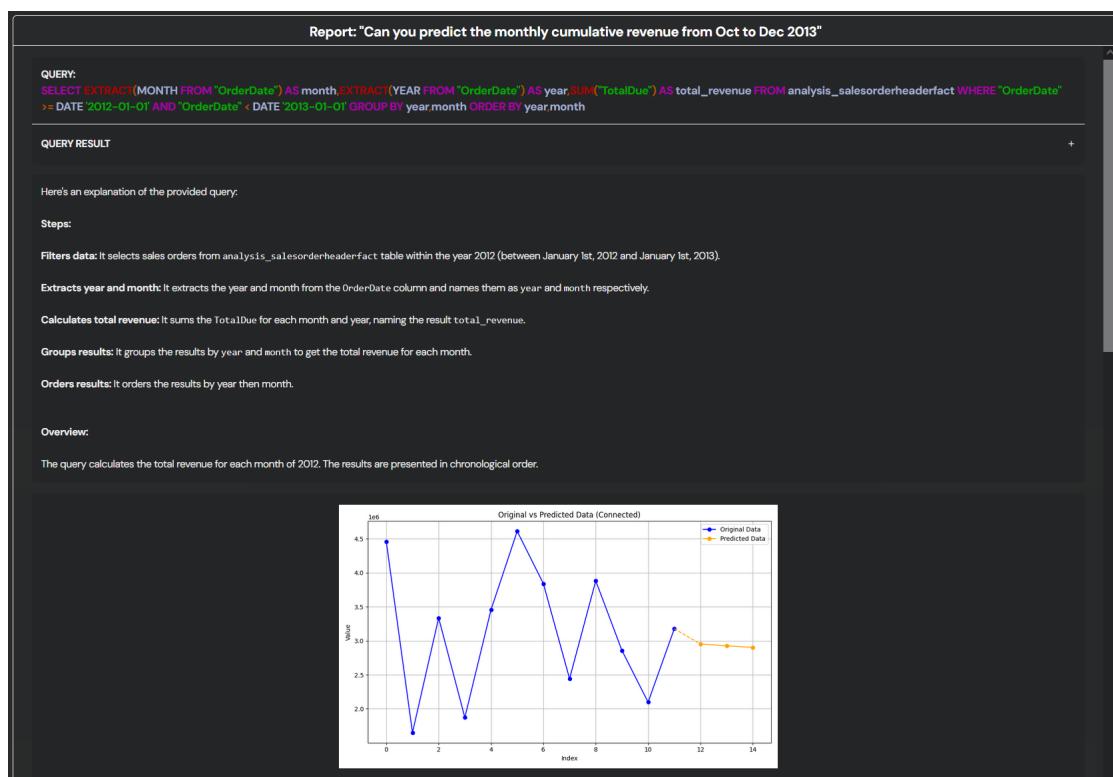
2.7.3.c Phân tích dữ liệu

Ta sẽ kiểm tra tính năng phân tích dữ liệu qua câu hỏi "Show me the top 5 products based on revenue in 2014" trong ứng dụng.



Hình 60: Câu hỏi được gửi bởi người dùng

Câu trả lời nhận được từ backend:



Hình 61: Câu trả lời nhận được từ hệ thống

2.7.4 Các hạn chế và cách khắc phục

Hiện thực tính năng phân tích kết hợp với mô hình ngôn ngữ có các lợi thế như: tiện lợi, tự động hóa và vận dụng tiềm năng của hệ cơ sở dữ liệu phân tích,... Tuy nhiên, việc sử dụng mô hình ngôn ngữ mang lại các hạn chế như:

- Câu trả lời không phù hợp:** Trong một số trường hợp, điều hướng hoặc câu trả lời đưa



ra bởi mô hình ngôn ngữ có thể không như ý muốn của người dùng; không gọi hàm, gọi hàm sai hoặc diễn giải không đúng trọng tâm.

- **Câu trả lời không nhất quán:** Các câu trả lời của mô hình ngôn ngữ mang tính chất ngẫu nhiên. Cùng một câu hỏi có thể mang lại các câu trả lời khác nhau.

Report: "show me the top 3 regions based on revenue from May to October 2013"

```
SELECT T1.Name, SUM(T3.TotalDue) AS revenue ;
FROM analysis_territorydim AS T1
JOIN analysis_customerdim AS T2 ON T1.id = T2.Territory_id;
JOIN analysis_salesorderheaderfact AS T3 ON T2.id = T3.Customer_id;
WHERE T3.OrderDate BETWEEN '2013-05-01' AND '2013-10-31';
GROUP BY T1.Name;
ORDER BY revenue DESC;
LIMIT 3;
```

Hình 62: Điều hướng sai bởi mô hình ngôn ngữ

Các hạn chế trên có thể được khắc phục bằng các phương pháp như sau:

- **Thêm các hướng dẫn vào mô tả (prompt engineering):** Các chỉ dẫn chi tiết hoặc các ví dụ có thể được thêm vào các mô tả để mô hình có thể xử lý chính xác trong các trường hợp tương tự.

```
IMPORTANT ADDITIONAL NOTES THAT YOU SHOULD PAY ATTENTION TO WHEN GIVING A RESPONSE:
+ When you are asked to give a query to calculate a cumulative total, you do not need to use GROUP BY on "OrderDate".
Instead, you should aggregate the totals by day in a subquery first and then apply the cumulative sum.
For example: When you are asked to cumulative total of sales order for each day in 2012. This is the query:
WITH daily_totals AS (
    SELECT OrderDate, SUM(TotalDue) AS daily_total
    FROM analysis_salesorderheaderfact
    WHERE EXTRACT(YEAR FROM OrderDate) = 2012
    GROUP BY OrderDate
)
SELECT OrderDate,
       SUM(daily_total) OVER (ORDER BY OrderDate) AS CumulativeTotalDue
FROM daily_totals
ORDER BY OrderDate;

+ Always try to find a function to satisfy user prompt. If you are not calling a function, DO NOT reply with answer in SQL or Python code.
```

Hình 63: Các chỉ dẫn được thêm để khắc phục điều hướng sai bởi mô hình

- **Nâng cấp mô hình:** Mô hình được sử dụng trong hệ thống là Gemini-1.5-flash, một mô hình tương đối nhanh và hiệu quả. Tuy nhiên, các câu trả lời của mô hình vẫn chưa mang lại độ chính xác cao so với các mô hình ngôn ngữ mới hiện nay.



References

- [1] Sales là gì? Mô tả công việc và tố chất cần có của nhân viên sales. Truy cập từ: <https://www.pace.edu.vn/tin-kho-tri-thuc/sales-la-gi/>
- [2] The Key Benefits of Business Intelligence in Marketing. Truy cập từ: <https://addepto.com/blog/business-intelligence-in-marketing-benefits-for-the-industry/>
- [3] Data Mart là gì? Tìm hiểu tổng quan về Data mart. Truy cập từ: <https://viblo.asia/p/data-mart-la-gi-tim-hieu-tong-quan-ve-data-mart-GyZJZn0kJjm>
- [4] Designing the Star Schema in Data Warehousing. Truy cập từ: <https://www.geeksforgeeks.org/designing-the-star-schema-in-data-warehousing/>
- [5] Extract, transform, load. Truy cập từ: https://en.wikipedia.org/wiki/Extract,_transform,_load
- [6] Next.js Documentation. Truy cập từ: <https://nextjs.org/docs>
- [7] Django documentation. Truy cập từ: <https://docs.djangoproject.com/en/5.1/>
- [8] What Is a Data Dashboard? The Ultimate 101 Guide. Truy cập từ: <https://www.netsuite.com/portal/resource/articles/human-resources/dashboard.shtml>
- [9] Design and build a great dashboard. Truy cập từ: <https://www.geckoboard.com/best-practice/dashboard-design/>
- [10] Gemini Developer API . Truy cập từ: <https://ai.google.dev/gemini-api/docs>