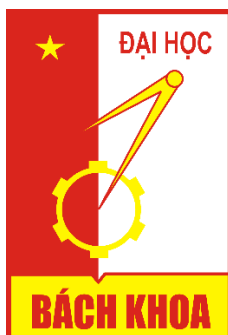


ĐẠI HỌC BÁCH KHOA HÀ NỘI
Trường Công nghệ thông tin và truyền thông

----- ∞  ∞ -----



Báo Cáo Lập Trình Hướng Đối Tượng

Đề tài:

Thiết kế hướng đối tượng

Quản lý ngân hàng

Giảng viên: **Lê Đức Hậu**

Nhóm 13.1:

Họ và tên	MSSV
Nguyễn Văn Hoàn	20225718
Trần Đức Huy	20225725
Chu Vĩnh Khang	20225863
Phan Đức Quyền	20225916
Nguyễn Khắc Tiếp	20225767

Hà Nội, năm 202

I) Mô tả khái quát bài toán.....	3
II) Cơ sở dữ liệu.....	4
<i>Hình 1: Sơ đồ thực thể</i>	<i>4</i>
III) Mô tả kiến trúc hệ thống.....	7
<i>Hình 3: Kiến trúc hệ thống</i>	<i>7</i>
IV) Mô tả UseCase Diagram và Class Diagram	8
<i>Hình 4: usecase diagram</i>	<i>8</i>
<i>Hình 5: class diagram.....</i>	<i>9</i>

Bảng phân công nhiệm vụ	
Họ và tên	Nhiệm vụ
Chu Vĩnh Khang	Tạo database và ERD, làm slide
Trần Đức Huy	Thiết kế kiến trúc hệ thống, fix bug và viết báo cáo
Nguyễn Văn Hoàn	Thiết kế và code package và class, thiết kế usecase diagram và class diagram
Nguyễn Khắc Tiếp	Thiết kế và code package và class, làm slide, viết báo cáo
Phan Đức Quyền	Làm giao diện Web, demo

Lời nói đầu

Trong thời đại công nghệ 4.0, ngành ngân hàng đã và đang chứng kiến sự chuyển đổi mạnh mẽ nhờ vào việc ứng dụng các giải pháp công nghệ thông tin hiện đại. Hệ thống ngân hàng không chỉ là nơi thực hiện các giao dịch tài chính, mà còn đóng vai trò quan trọng trong việc cung cấp các dịch vụ đa dạng, từ tiết kiệm, vay vốn, thanh toán điện tử, đến quản lý tài sản và đầu tư.

Việc thiết kế một hệ thống ngân hàng không chỉ đòi hỏi sự chính xác trong xử lý dữ liệu mà còn cần đảm bảo tính bảo mật, tốc độ xử lý và khả năng mở rộng để đáp ứng nhu cầu ngày càng cao của khách

hàng. Hơn nữa, các hệ thống này phải tuân thủ các quy định pháp luật chặt chẽ, đồng thời mang lại trải nghiệm người dùng tốt nhất.

Mục tiêu của bài toán thiết kế hệ thống ngân hàng là xây dựng một nền tảng quản lý thông tin khách hàng, giao dịch và dịch vụ một cách hiệu quả, minh bạch và an toàn. Đây không chỉ là một bài toán kỹ thuật mà còn là cơ hội để nâng cao trải nghiệm dịch vụ và góp phần vào sự phát triển bền vững của ngành tài chính.

Trong tài liệu này, chúng em sẽ trình bày chi tiết các yêu cầu, thiết kế, và giải pháp cho một hệ thống ngân hàng toàn diện, từ các chức năng cơ bản đến những tính năng nâng cao, đảm bảo đáp ứng nhu cầu thực tiễn và xu hướng hiện đại hóa trong lĩnh vực tài chính ngân hàng.

Contents

I) Mô tả khái quát bài toán	3
II) Cơ sở dữ liệu	4
<i>Hình 1: Sơ đồ thực thể</i>	4
<i>Hình 2: ERD và mối quan hệ giữa các thực thể</i>	6
III) Mô tả kiến trúc hệ thống	7
<i>Hình 3: Kiến trúc hệ thống</i>	7
IV) Mô tả UseCase Diagram và Class Diagram	8
1. UseCase Diagram	8
<i>Hình 4: usecase diagram</i>	8
2. Class Diagram	9
<i>Hình 5: class diagram</i>	9

I) Mô tả khái quát bài toán

a) Mục tiêu:

- Xây dựng một hệ thống **ComtechBank** hỗ trợ khách hàng thực hiện các nghiệp vụ ngân hàng trực tuyến một cách dễ dàng, bảo mật và hiệu quả. Hệ thống tập trung vào việc cung cấp các dịch vụ cơ bản như quản lý tài khoản, giao dịch tài chính, vay vốn và các dịch vụ giá trị gia tăng khác.

b) Đối tượng sử dụng

- **Khách Hàng (User):** Cá nhân có tài khoản ngân hàng, sử dụng hệ thống để tra cứu thông tin, thực hiện giao dịch, hoặc sử dụng các dịch vụ khác.

- **Quản trị viên (Admin):** Quản lý, theo dõi và hỗ trợ các hoạt động của khách hàng trên hệ thống.

c) Yêu cầu nghiệp vụ

- **Quản lý tài khoản:** Tra cứu thông tin tài khoản, lịch sử giao dịch, và trạng thái thẻ ngân hàng.

- **Thực hiện giao dịch:** Chuyển tiền, rút tiền, nạp tiền điện thoại và các giao dịch mua sắm.
- **Hỗ trợ vay vốn:** Cung cấp thông tin sản phẩm vay và xử lý yêu cầu vay.
- **Sổ tiết kiệm:** gửi tiền vào sổ tiết kiệm, tính lãi hàng tháng, rút tiền từ sổ tiết kiệm
- **Dịch vụ thẻ:** Đăng ký hoặc làm mới thẻ vật lý

II) Cơ sở dữ liệu

1) Thiết kế cơ sở dữ liệu



Hình 1: Sơ đồ thực thể

Khách hàng (Mã khách hàng, Họ và tên, Ngày sinh, CCCD, SĐT, Email, Địa chỉ):

+ CCCD, SĐT, Email là unique

Tài khoản (Số tài khoản, Số dư)

Thẻ (Số thẻ, Loại thẻ, Ngày phát hành, Ngày hết hạn, Trạng thái thẻ) liên kết với thực thể Khách hàng:

+ Loại thẻ:

Thẻ ghi nợ (Debit):

+ Liên kết trực tiếp với tài khoản ngân hàng.

- + Chỉ được sử dụng khi tài khoản có đủ số dư.
- + Có thể sử dụng cho mọi hình thức giao dịch.

Thẻ tín dụng (Credit):

- + Cho phép người dùng vay tiền từ ngân hàng để chi tiêu trong hạn mức tín dụng.
- + Số dư có thể âm (trả dần hoặc toàn bộ sau kỳ hạn).
- + Chỉ có thể sử dụng cho hình thức giao dịch thanh toán dịch vụ.
 - + Trạng thái thẻ: Đang hoạt động, Đã khóa, Hết hạn

Giao dịch (Mã giao dịch, Số tiền, Thời gian, Nội dung) được thực hiện bởi thực thể Tài khoản

Sổ tiết kiệm (Mã sổ, Số tiền, Kỳ hạn, Lãi suất, Ngày đáo hạn) được sở hữu bởi thực thể Khách hàng:

- + Kỳ hạn (Đơn vị: tháng): Xác định thời gian mà khách hàng sẽ gửi tiền vào sổ tiết kiệm, sau đó có thể rút tiền ra cùng với lãi suất.
- + Lãi suất (Đơn vị: %/năm): Thể hiện tỷ lệ lãi suất áp dụng cho sổ tiết kiệm trong suốt kỳ hạn.
- + Ngày đáo hạn: Thể hiện ngày kết thúc của kỳ hạn gửi tiền.

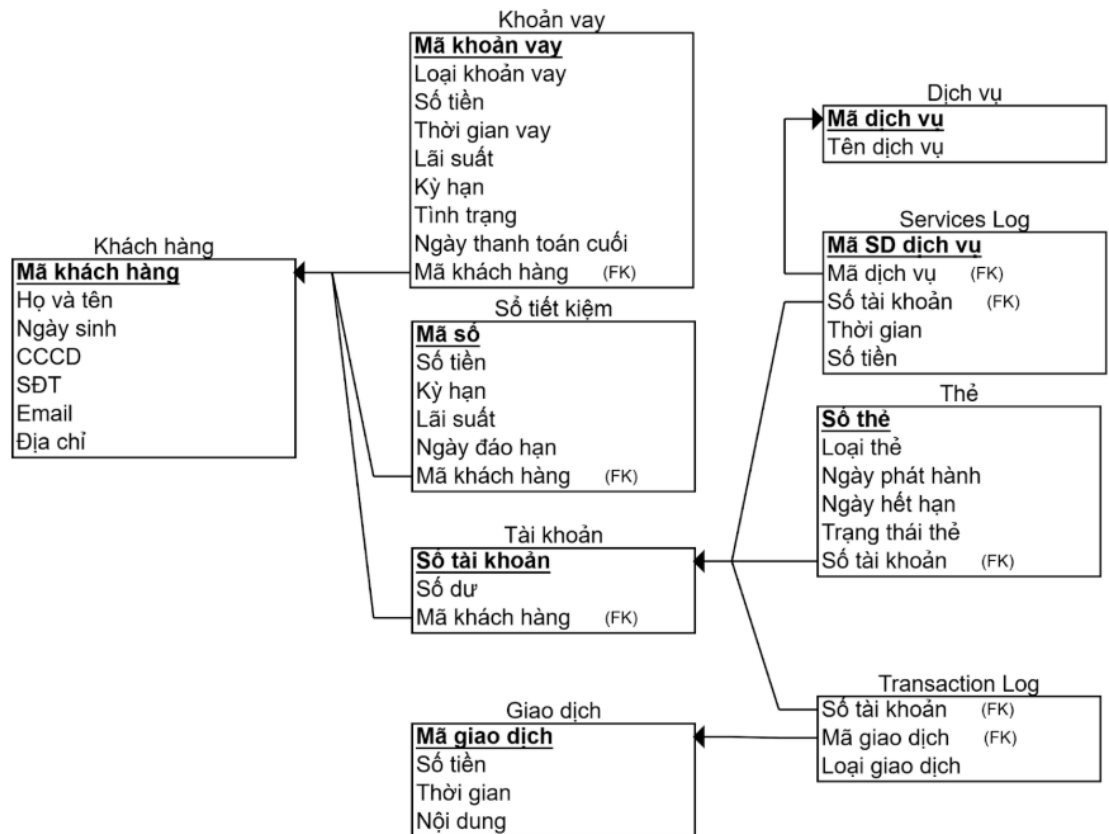
Khoản vay (Mã khoản vay, Loại khoản vay, Số tiền, Thời gian vay, Lãi suất, Kỳ hạn, Tình trạng, Ngày thanh toán cuối) được thực hiện bởi thực thể Khách hàng:

- + Thời gian vay: Ngày bắt đầu khoản vay
- + Lãi suất (Đơn vị: %/năm): Tỷ lệ lãi suất áp dụng cho khoản vay trong kỳ hạn
- + Kỳ hạn (Đơn vị: tháng): Chu kỳ thanh toán lãi và gốc
- + Tình trạng: Đang vay, Đã tắt toán, Quá hạn
- + Ngày thanh toán cuối: Thời hạn tối đa của khoản vay

Dịch vụ (Mã dịch vụ, Tên dịch vụ) được sử dụng bởi thực thể Tài khoản:

- + Tên dịch vụ: Chuyển đổi ngoại tệ, mua vé, thanh toán điện/nước, nạp thẻ,...

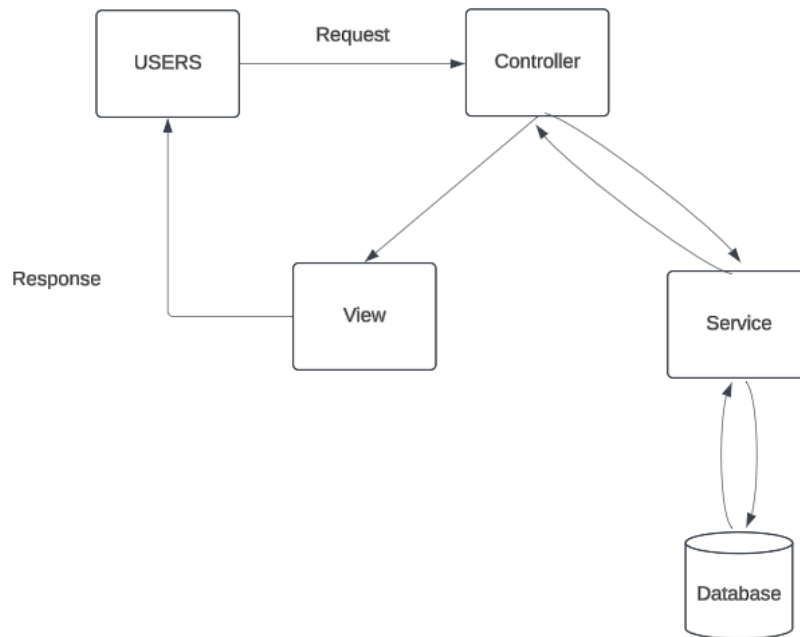
2) ERD và các quan hệ trong bảng dữ liệu



Hình 2: ERD và mối quan hệ giữa các thực thể

- Khách hàng - Tài khoản: 1:M
+ 1 khách hàng có thể có nhiều tài khoản, 1 tài khoản chỉ thuộc 1 khách hàng
- Khách hàng - Khoản vay: 1:M
+ 1 khách hàng có thể có nhiều khoản vay, 1 khoản vay chỉ thuộc 1 khách hàng
- Khách hàng - Sổ tiết kiệm: 1:M
+ 1 khách hàng có thể có nhiều sổ tiết kiệm, 1 sổ tiết kiệm chỉ thuộc 1 khách hàng
- Tài khoản - Dịch vụ: M:M
+ 1 tài khoản có thể sử dụng nhiều dịch vụ, 1 dịch vụ có thể được sử dụng bởi nhiều tài khoản
+ Quan hệ có thuộc tính (Mã sử dụng dịch vụ, Thời gian, Số tiền)
- Tài khoản - Giao dịch: M:M
+ 1 tài khoản có thể thực hiện nhiều giao dịch, 1 giao dịch có thể liên quan đến tối đa 2 tài khoản (trong trường hợp là giao dịch Chuyển tiền)
+ Quan hệ có thuộc tính "Loại giao dịch" có thể nhận giá trị: Rút tiền, Nạp tiền, Chuyển tiền, Nhận tiền, Thanh toán.
- Tài khoản - Thẻ: 1:M
+ 1 tài khoản có thể được liên kết với nhiều thẻ, 1 thẻ chỉ có thể liên kết với 1 tài khoản
- Bảng Transaction Log thể hiện quan hệ M:M giữa Tài khoản và Giao dịch:
+ (Mã giao dịch, Loại giao dịch) là unique
- Bảng Services Log thể hiện quan hệ M:M giữa Tài khoản và Dịch vụ:
+ (Mã sử dụng dịch vụ) là unique

III) Mô tả kiến trúc hệ thống



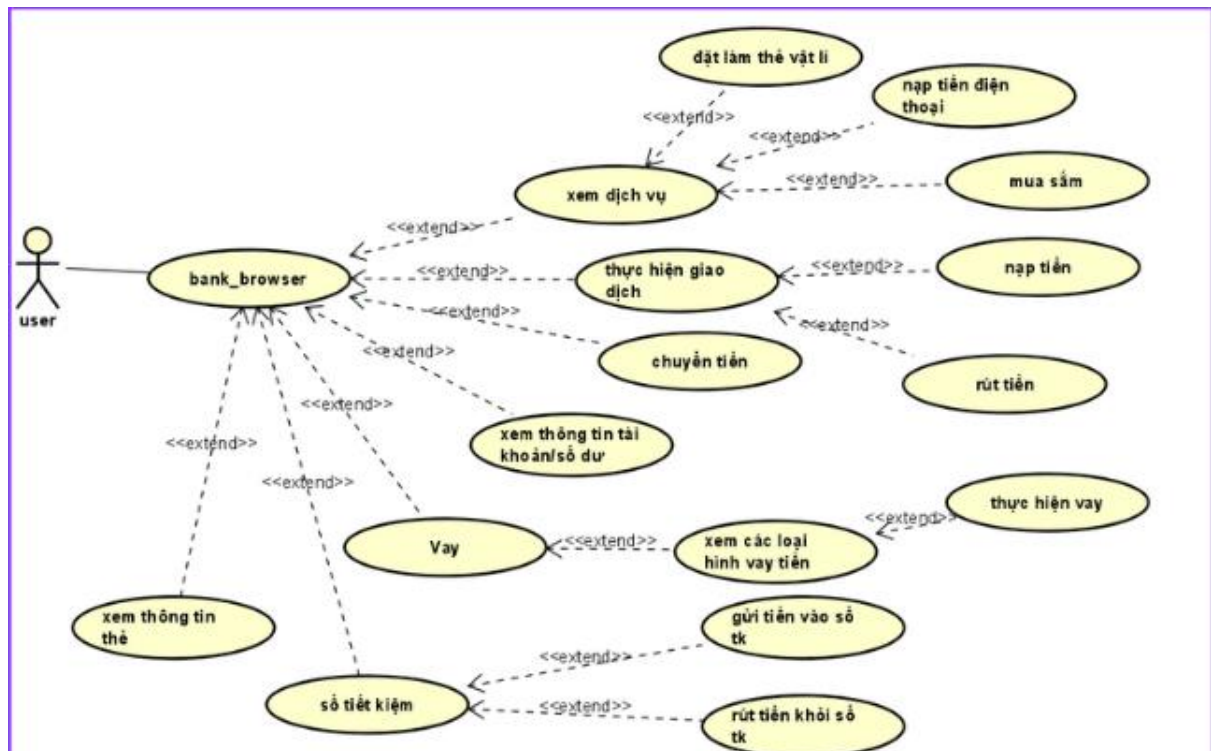
Hình 3: Kiến trúc hệ thống

- **Users (Người dùng):**
Đây là thành phần bên ngoài tương tác với hệ thống, gửi yêu cầu (requests) đến hệ thống và nhận lại phản hồi (responses).
- **Controller (Bộ điều khiển):**
Là thành phần chịu trách nhiệm nhận yêu cầu từ người dùng và xử lý chúng.
Controller sẽ phân tích yêu cầu, giao tiếp với các thành phần khác trong hệ thống (như Service hoặc View) để thực hiện logic nghiệp vụ.
- **Service (Dịch vụ):**
Đây là lớp trung gian chứa logic nghiệp vụ chính của hệ thống.
Service sẽ xử lý dữ liệu từ Controller và giao tiếp với Database (Cơ sở dữ liệu) để truy vấn hoặc lưu trữ dữ liệu.
- **Database (Cơ sở dữ liệu):**
Thành phần lưu trữ dữ liệu của hệ thống.
Nó phục vụ các yêu cầu từ Service để cung cấp hoặc cập nhật dữ liệu.
- **View (Giao diện):**
View chịu trách nhiệm hiển thị thông tin cho người dùng.
Sau khi nhận kết quả từ Controller, nó sẽ định dạng dữ liệu và gửi phản hồi về cho người dùng.
- **Luồng dữ liệu:**
 - + Người dùng gửi Request đến Controller.

- + Controller xử lý yêu cầu, giao tiếp với Service để thực hiện logic nghiệp vụ.
- + Service có thể truy vấn hoặc cập nhật dữ liệu trong Database.
- + Sau khi xử lý, kết quả được gửi từ Service về Controller.
- + Controller chuyển kết quả đến View để định dạng và gửi phản hồi cuối cùng cho người dùng.

IV) Mô tả UseCase Diagram và Class Diagram

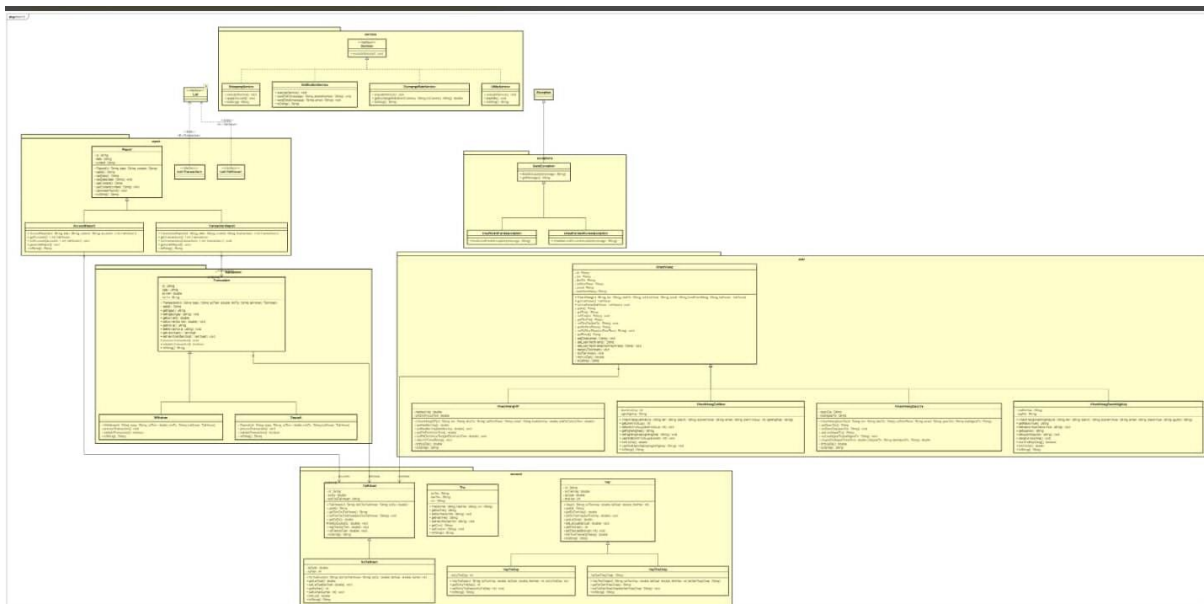
1. UseCase Diagram



Hình 4: usecase diagram

- User
 - Đây là tác nhân chính, đại diện cho người dùng thực hiện các thao tác trên ứng dụng ngân hàng trực tuyến. Người dùng có thể truy cập và thực hiện nhiều chức năng khác nhau, từ xem thông tin đến thực hiện các giao dịch tài chính.
- Use Case
 - **bank_browser (Trình duyệt ngân hàng):**
 - + Là trường hợp sử dụng chính, đại diện cho việc truy cập vào các chức năng của ứng dụng ngân hàng trực tuyến.
 - + Các chức năng con được mở rộng bằng các mối quan hệ <<extend>>.
 - **xem dịch vụ (Xem các dịch vụ):**
 - + Cho phép người dùng duyệt qua các dịch vụ mà ngân hàng cung cấp như đặt làm thẻ vật lý, nạp tiền điện thoại, mua sắm.
 - + Các chức năng này được mở rộng qua các mối quan hệ:
 - <<extend>> đặt làm thẻ vật lý
 - <<extend>> nạp tiền điện thoại
 - <<extend>> mua sắm
 - **thực hiện giao dịch:** Người dùng thực hiện các thao tác tài chính, bao gồm:

- + chuyển tiền: Gửi tiền từ tài khoản này sang tài khoản khác.
 - + nạp tiền: Nạp tiền vào tài khoản.
 - + rút tiền: Rút tiền từ tài khoản.
 - xem thông tin tài khoản/số dư:
 - + Người dùng có thể kiểm tra số dư và thông tin tài khoản cá nhân.
 - **vay (Khoản vay):** Cung cấp thông tin về các khoản vay, bao gồm:
 - + xem các loại hình vay tiền: Tìm hiểu các gói vay mà ngân hàng cung cấp.
 - + thực hiện vay: Thực hiện một khoản vay trực tuyến.
 - **sổ tiết kiệm:** Cho phép người dùng quản lý sổ tiết kiệm, bao gồm:
 - + gửi tiền vào sổ tk: Thêm tiền vào sổ tiết kiệm.
 - + rút tiền khỏi sổ tk: Rút tiền từ sổ tiết kiệm.
 - **xem thông tin thẻ:**
 - + Xem thông tin chi tiết về các loại thẻ mà người dùng sở hữu.
 - c. Mỗi quan hệ trong sơ đồ
 - **Quan hệ <<extend>>:**
 - + Dùng để biểu diễn các chức năng mở rộng của một trường hợp sử dụng chính. Ví dụ: thực hiện giao dịch được mở rộng với các chức năng như chuyển tiền, rút tiền, nạp tiền. xem dịch vụ được mở rộng với các chức năng như đặt làm thẻ vật lý, nạp tiền điện thoại, mua sắm.
 - **Hướng tương tác:**
 - + Dòng chấm gạch thể hiện sự tương tác giữa người dùng và từng chức năng cụ thể trong hệ thống
2. Class Diagram



Hình 5: class diagram

- a. Tổng quan
 - Class Diagram minh họa các lớp (Classes), các mối quan hệ (Relationships) giữa chúng, và cách các lớp tương tác với nhau. Nó gồm:
 - + Các lớp chính với tên, thuộc tính và phương thức.
 - + Mối quan hệ giữa các lớp: Quan hệ kế thừa, liên kết, phụ thuộc...
 - + Phạm vi hệ thống tập trung vào việc mô hình hóa các thực thể chính như tài khoản, người dùng, giao dịch, và các dịch vụ liên quan.
- b. Các lớp chính
 - **Lớp Account (Tài khoản):**

Thuộc tính:

- + accountId: ID duy nhất cho tài khoản.
- + accountType: Loại tài khoản (ví dụ: tiết kiệm, vãng lai).
- + balance: Số dư trong tài khoản.

Phương thức:

- + deposit(amount): Gửi tiền vào tài khoản.
- + withdraw(amount): Rút tiền từ tài khoản.
- + checkBalance(): Kiểm tra số dư tài khoản.

- **Lớp User (Người dùng):**

Thuộc tính:

- + userId: ID duy nhất cho người dùng.
- + name: Tên người dùng
- + email: Địa chỉ email.
- + phoneNumber: Số điện thoại liên lạc.

Phương thức:

- + register(): Đăng ký tài khoản.
- + login(): Đăng nhập vào hệ thống.
- + logout(): Đăng xuất khỏi hệ thống.

- **Lớp Transaction (Giao dịch):**

Thuộc tính:

- + transactionId: ID duy nhất cho giao dịch.
- + transactionType: Loại giao dịch (nạp tiền, rút tiền, chuyển khoản).
- + amount: Số tiền giao dịch.
- + date: Ngày thực hiện giao dịch.

Phương thức:

- + executeTransaction(): Thực hiện giao dịch.
- + rollbackTransaction(): Hoàn tác giao dịch (nếu xảy ra lỗi).

- **Lớp Loan (Khoản vay):**

Thuộc tính:

- + loanId: ID của khoản vay.
- + loanAmount: Số tiền vay.
- + interestRate: Lãi suất vay.
- + loanTerm: Thời hạn vay.

Phương thức:

- + applyForLoan(): Đăng ký khoản vay.
- + repayLoan(amount): Trả tiền vay.

- **Lớp SavingAccount (Tài khoản tiết kiệm):** Kế thừa từ lớp Account.

Thuộc tính:

- + interestRate: Lãi suất tiết kiệm.
- + maturityDate: Ngày đáo hạn.

Phương thức:

- + calculateInterest(): Tính lãi cho tài khoản tiết kiệm.

- **Lớp CheckingAccount (Tài khoản thanh toán):** Kế thừa từ lớp Account.

Phương thức:

- + writeCheck(): Viết séc để thanh toán.

c. Các mối quan hệ

- **Kế thừa (Inheritance):**

- + SavingAccount và CheckingAccount kế thừa từ lớp Account: Hai lớp này được chuyên biệt hóa từ lớp Account để cung cấp các tính năng đặc thù cho từng loại tài khoản.

+ PersonalLoan và HomeLoan kế thừa từ lớp Loan: Các lớp này biểu diễn các loại hình vay chuyên biệt như vay cá nhân hoặc vay mua nhà.

- **Liên kết (Association):**

+ User - Account: Mỗi quan hệ 1-nhiều: Một người dùng (đại diện bởi User) có thể sở hữu nhiều tài khoản (đại diện bởi Account).

Tác dụng: Tác nhân User liên kết với tài khoản để thực hiện các thao tác như giao dịch hoặc kiểm tra số dư.

+ Account - Transaction: Mỗi quan hệ 1-nhiều: Mỗi tài khoản có thể tham gia nhiều giao dịch.

Tác dụng: Lớp Transaction thực hiện các giao dịch như nạp tiền, rút tiền hoặc chuyển tiền giữa các tài khoản.

+ Loan - Account: Mỗi quan hệ 1-1: Mỗi khoản vay thường liên quan đến một tài khoản để quản lý việc thanh toán.

Tác dụng: Lớp Loan phụ thuộc vào Account để kiểm tra tình trạng số dư và khả năng vay

- **Phụ thuộc (Dependency):**

+ Transaction phụ thuộc vào Account: Transaction sử dụng Account để thao tác với số dư, bao gồm các hành động như nạp/rút/chuyển tiền.

+ Loan phụ thuộc vào Account: Loan tham chiếu Account như một phần để đảm bảo khả năng thanh toán khoản vay.

+ User phụ thuộc vào Transaction: User gọi các giao dịch, vì vậy phụ thuộc vào Transaction để thực thi các tác vụ.