**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC LẠC HỒNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🙠🕮🙢🙢

A picture containing text, font, graphic design, graphics

Description automatically generated

**BÁO CÁO KẾT THÚC MÔN**

**LẬP TRÌNH C#**

***Đề tài: Chương trình quản lí ngân hàng***

Giảng viên hướng dẫn: Gv. Lê Mạnh Hùng

Nhóm sinh viên thực hiện:

1. Lê Nguyễn Thanh Tú - 121001311

2. Trịnh Đăng Quang - 121001097

3. Dương Kim Nguyên - 121001264

4. Bùi Đình Nhu - 121001407

**Đồng Nai, Tháng 11 Năm 2023**

# DANH SÁCH PHÂN CÔNG NHIỆM VỤ NHÓM

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nhiệm vụ/nội dung thực hiện** | **Ghi chú** |
| **1** | **Lê Nguyễn Thanh Tú** | * **Thiết kế logo** * **Thiết kế UI** * **Thiết kế Form: Cards, linkCard, CardDetail** |  |
| **2** | **Trịnh Đăng Quang** | * **Thiết kế form: formMain, formLogin, form ResetPassword.** * **Tìm kiếm hình ảnh và design form chữ backgroud trong tất cả các form.** * **Xây dựng database.** * **Xử lý các event cho các toolbox.** * **Thực hiện kết nối database, code xử lý các kết nối database với các form trong C#.** * **Đóng góp phần hoàn thiện file Word và Powerpoint.** |  |
| **3** | **Dương Kim Nguyên** | * **Thiết kế form: Các form con trong formMain, form newAccount.** * **Code các Event xử lý, Custom các toolbox trong một C#** * **Đóng góp phần hoàn thiện file Word và Powerpoint.** |  |
| **4** | **Bùi Đình Nhu** | * **Xây dựng database.** * **Thực hiện kết nối database, code xử lý các kết nối database với các form trong C#.** * **Đóng góp hoàn thiện file Word.** |  |
| **5** | **Nguyễn Hưng Định** | * **Lên thư viện 1 lần** | **Không tham gia làm việc nhóm** |

# PHẦN MỞ ĐẦU

## Giới thiệu về chương trình quản lí ngân hàng:

Dự án quản lý ngân hàng, không chỉ là một hệ thống phức tạp mà còn là một tác phẩm kiến trúc được xây dựng với sự tâm huyết và sáng tạo. Với mục tiêu quản lý mọi khía cạnh của giao dịch tài chính, thông tin khách hàng, và những hoạt động khác liên quan đến ngân hàng, dự án này là bước đột phá mang tính cách mạng trong lĩnh vực công nghệ ngân hàng.

Chương trình của nhóm chỉ là một sản phẩm mô phỏng một tác phẩm mô phỏng một số chức năng của các ứng dụng ngân hàng hiện đại. Với chủ đích là tìm hiểu và thiết kế UI, chức năng xử lý và cập nhật thêm, sửa, xóa từ code C# tới database, chưa thể thay thế hay cạnh tranh với những phần mềm ngân hàng hiện nay hay chí ít là so về công nghệ sử dụng trong chương trình này còn quá hạn chế.

Hệ thống được chia thành hai phần quản lý chính cho cả người quản trị và người dùng cuối. Phần quản lý admin, không chỉ cung cấp thông tin chi tiết về người dùng mà còn thống kê các giao dịch theo từng tháng một cách minh bạch và hiệu quả. Nó cũng là bảo tàng số, nơi quản trị các dịch vụ với sự linh hoạt và tiện lợi.

Trong khi đó, với vai trò người dùng sẽ có các tính năng như việc thêm mới người dùng đến việc kiểm tra số dư một cách nhanh chóng, chuyển tiền linh hoạt, mở sổ tiết kiệm đơn giản, quản lý thẻ hiệu quả, cho đến việc xem xét thông tin cá nhân, mọi điều đều trở nên dễ dàng và thuận tiện.

Cả dự án không chỉ là một công cụ quản lý ngân hàng, mà còn là thử thách đối với cả nhóm, vì từ ý tưởng đến việc thiết kế hay lập trình trên internet là khá khó tìm kiếm và đa phần là không sử dụng code c#. Vì vậy, điều đó cũng cho phép cả nhóm phải bắt tay tự tìm ý tưởng, thiết kế và cả logic trong các đoạn code xử lý. Nhưng qua đó cho cả nhóm học thêm nhiều kiến thức về việc làm việc nhóm, trao đổi thông tin tìm kiếm được, hay dạy cho nhau học về kiến thức mới. Do đó hoàn thành dự án này là một sự thành công lớn về kiến thức cũng như tư duy của mỗi người.

## Các chức năng chính của chương trình.

* Tạo tài khoản người dùng mới.
* Lấy lại mật khẩu bị quên.
* Đăng nhập vào giao diện user và giao diện admin.
* Hiển thị số dư, tên tài khoản người dùng.
* Chuyển tiền.
* Lập sổ tiết kiệm.
* Lịch sử giao dịch.
* Thêm mới thẻ, hiển thị thông tin thẻ, khóa thẻ.
* Thông tin người sử dụng, thay đổi mật khẩu.
* Đăng xuất.

## Các tài nguyên và phần mềm sử dụng trong dự án.

### C# và Visual Studio 2013

C# và WinForm có thể được sử dụng kết hợp với nhau để tạo các ứng dụng Windows có giao diện người dùng đẹp mắt và dễ sử dụng.

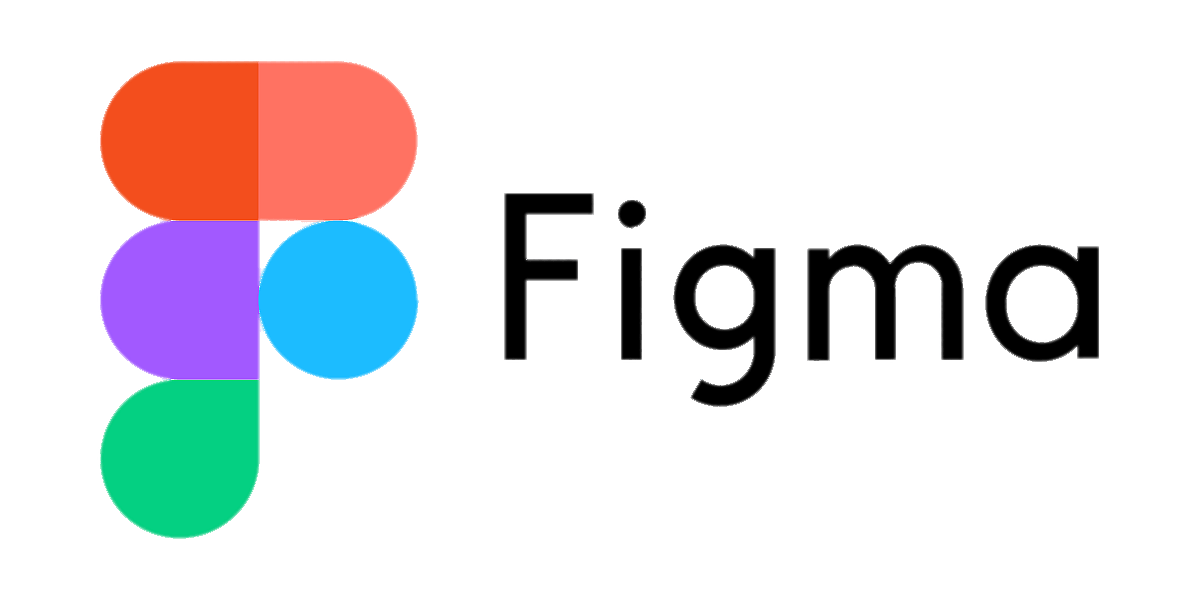
Có một số lợi ích khi sử dụng C# và WinForm trong thiết kế UI, bao gồm:

* Sức mạnh và tính linh hoạt: C# là một ngôn ngữ lập trình mạnh mẽ và linh hoạt, có thể được sử dụng để tạo các ứng dụng phức tạp với các tính năng tiên tiến. WinForm cũng là một khuôn khổ mạnh mẽ, cho phép các nhà phát triển tạo các ứng dụng với giao diện người dùng phức tạp và tùy biến.
* Hiệu suất: C# và WinForm đều được thiết kế để cung cấp hiệu suất cao. Điều này là do C# là một ngôn ngữ lập trình cấp thấp, và WinForm được tối ưu hóa cho các ứng dụng Windows.
* Khả năng mở rộng: C# và WinForm đều có thể được mở rộng để đáp ứng các nhu cầu cụ thể của ứng dụng. Điều này có thể được thực hiện bằng cách sử dụng các thành phần .NET và các thư viện của bên thứ ba.



### Figmar

Figma là một công cụ thiết kế giao diện nổi bật với khả năng hợp tác nhóm mạnh mẽ. Được sử dụng để thiết kế UI/UX, Figma cho phép tạo, chỉnh sửa và chia sẻ thiết kế một cách linh hoạt, giúp cả nhóm làm việc từ xa. Với khả năng xem trước và tương tác trực tiếp trên trình duyệt, Figma là công cụ linh hoạt, thuận tiện cho việc tạo ra các sản phẩm thiết kế mẫu form được sử dụng trong dự án.



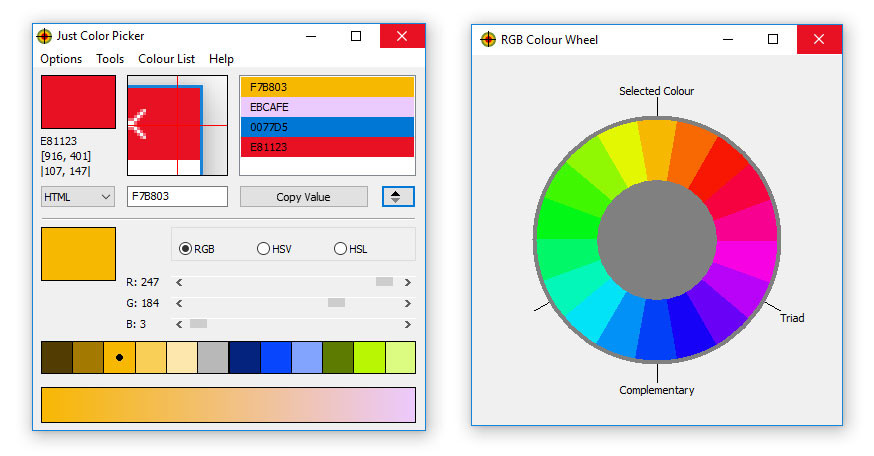
### [Free Icons, Clipart Illustrations, Photos, and Music (icons8.com)](https://icons8.com/)

Icons8.com là một nguồn tài nguyên đa dạng và đầy đủ về biểu tượng (icons) cho cộng đồng thiết kế và phát triển. Với bộ sưu tập lớn chia thành nhiều danh mục, Icons8 cung cấp hàng nghìn biểu tượng chất lượng cao có thể sử dụng miễn phí hoặc qua các gói dịch vụ trả phí. Với việc lựa chọn chương trình là quản lý ngân hàng cần phải sử dụng nhiều những icon để thiết kế dự án được đẹp hơn thì trang web này là sự lựa chọn hoàn hảo cho dự án.



### Just Color Picker

Được phát triển bởi công ty Anny, Just Color Picker là một công cụ nhỏ gọn nhưng mạnh mẽ, có khả năng chọn màu từ bất kỳ điểm nào trên màn hình và hiển thị mã màu tương ứng. Phù hợp cho dự án để có thể lấy mã màu RGB phù hợp cho việc thiết kế.



### Sql Server 2014

Microsoft SQL Server là hệ quản trị cơ sở dữ liệu mạnh mẽ của Microsoft, sử dụng ngôn ngữ truy vấn SQL. Nó cung cấp quản lý cơ sở dữ liệu toàn diện, tích hợp tốt với công nghệ Microsoft, hỗ trợ bảo mật mạnh mẽ, và có khả năng xử lý thao tác lớn. Phần mềm được sử dụng như một công cụ lưu trữ data người dùng, viết các proceduce, trigger xử lý khi cần thêm sửa xóa dữ liệu được xử lý từ code c#.



### Github

Cuối cùng, GitHub là một nền tảng lưu trữ mã nguồn và quản lý dự án phần mềm trực tuyến. Được tạo ra để hỗ trợ việc hợp tác trong phát triển phần mềm, GitHub cung cấp không gian làm việc chia sẻ dự án, theo dõi sự thay đổi, và quản lý mã nguồn một cách hiệu quả. Hữu ích khi sử dụng cho việc chia sẻ code sau khi cập nhật một form hay một đoạn code xử lý từ xa, qua đó có thể tiếp tục công việc lập trình một cách dễ dàng.



# PHÂN TÍCH DỰ ÁN

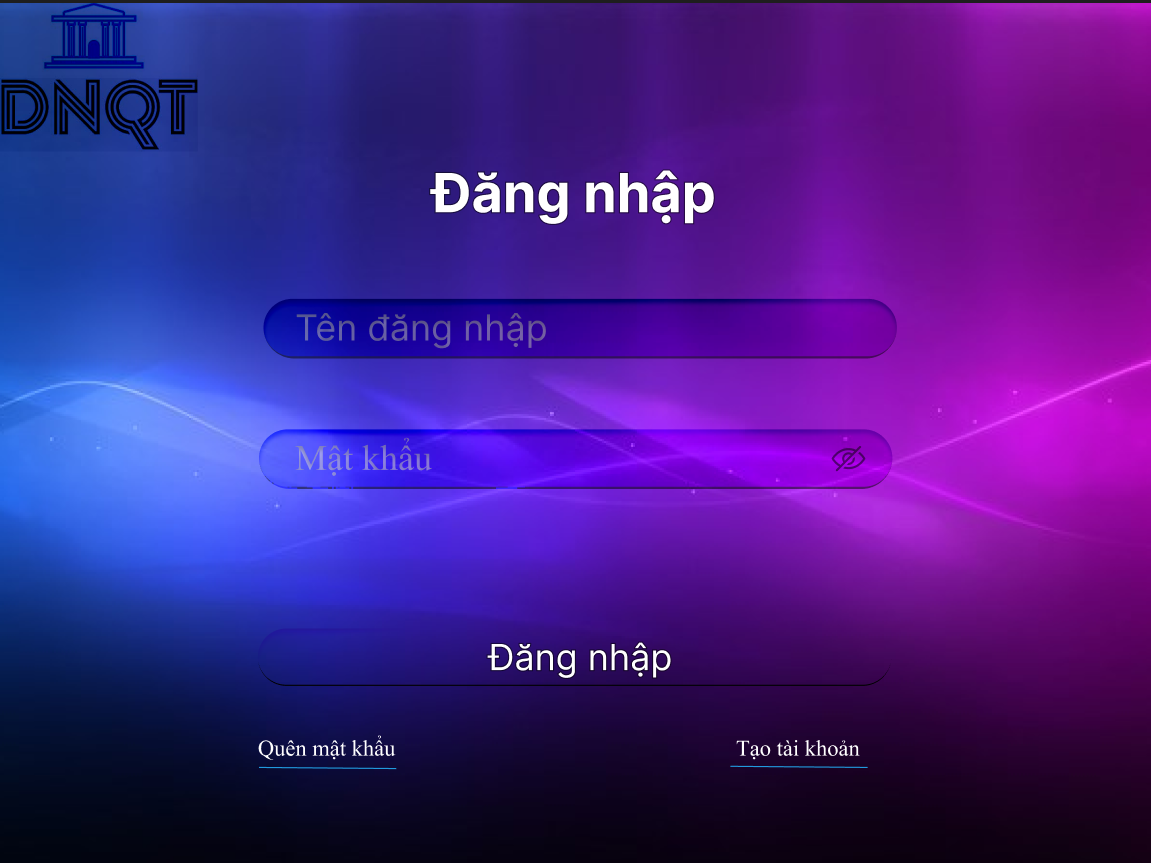
## Thiết kế giao diện:

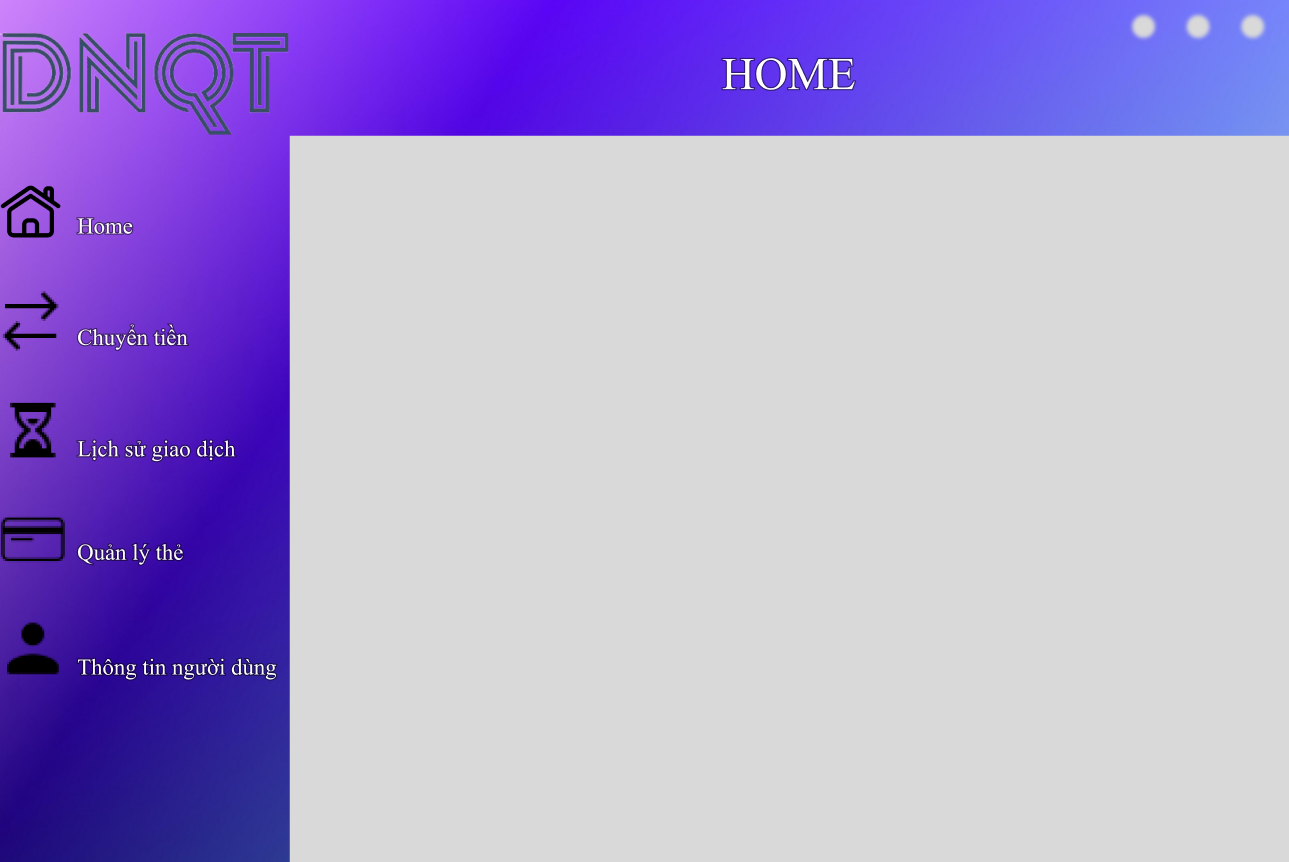
Giao diện người dùng sẽ bao gồm tất cả các yếu tố mà người dùng tương tác trực tiếp. Điều này gồm các phần tử như nút nhấn, hình ảnh, biểu mẫu, trình điều hướng và các thành phần tương tác khác.

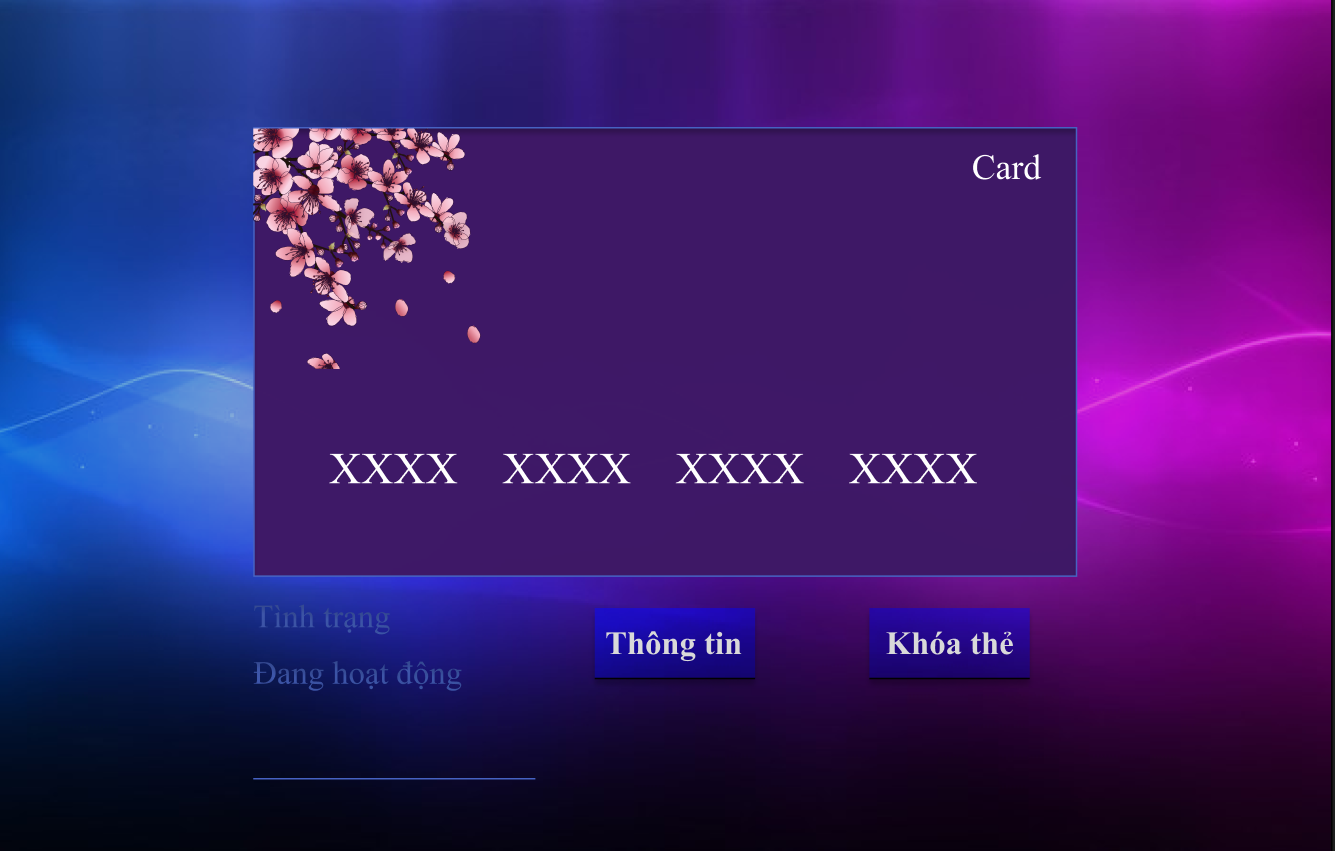
Giao diện người dùng cũng đã được thiết kế sao cho dễ sử dụng, trực quan và hướng dẫn người dùng điều hướng một cách dễ dàng hơn. Bố cục của chương trình được xác định cách các phần tử sắp xếp và tổ chức trên trang. Bố cục cần cân đối và tạo điểm nhấn cho các phần tử quan trọng, đồng thời giữ cho chương trình gọn gàng và dễ đọc.

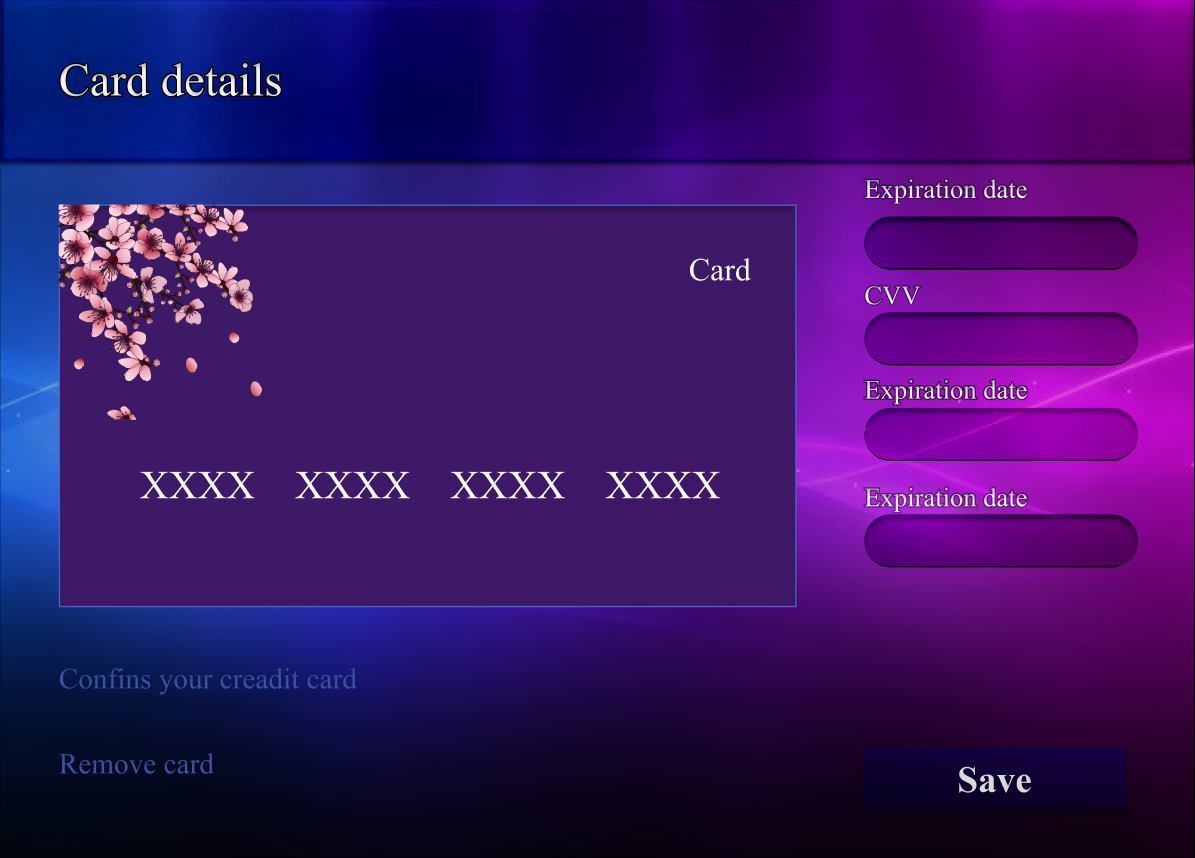
Ở phần màu sắc phải tạo ra một ảnh hưởng mạnh mẽ đến cảm xúc và thị giác của người dùng. Lựa chọn màu sắc phù hợp có thể tạo ra sự hòa hợp và tương phản, thể hiện thương hiệu và tạo nên một phong cách độc đáo cho chương trình. Màu sắc cũng có thể được sử dụng để định rõ các phần trong chương trình.

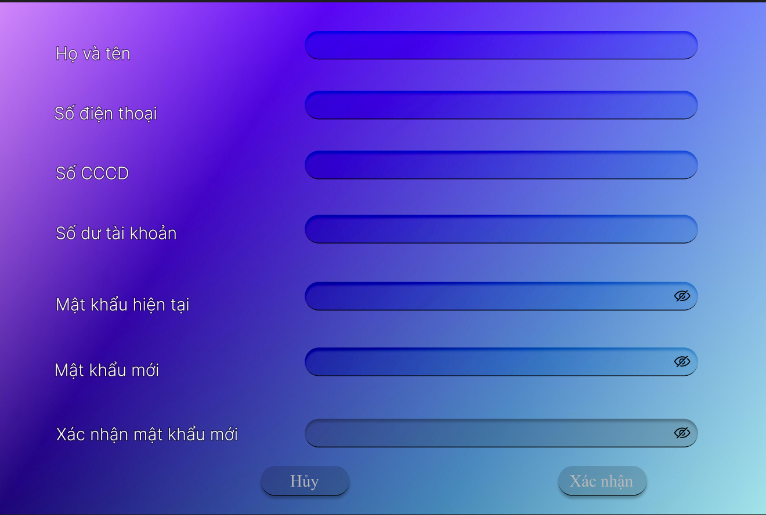
Phông chữ cũng là một yếu tố quan trọng trong thiết kế. Phông chữ nên dễ đọc và hợp lý với phong cách tổng thể. Sử dụng các loại phông chữ đơn giản để tạo sự nhất quán và giúp người dùng dễ dàng đọc và tiếp cận nội dung.



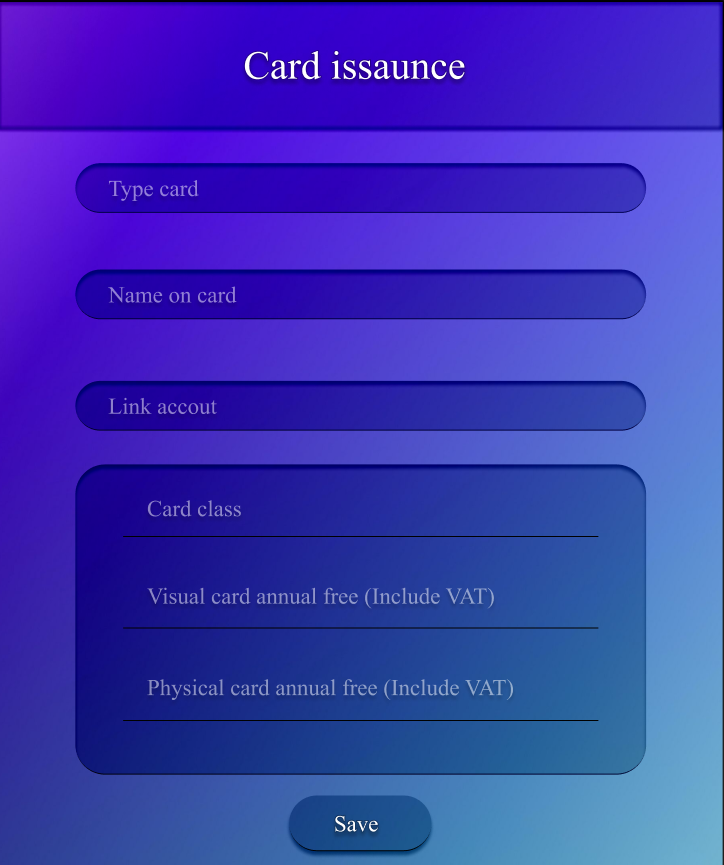












## Giới thiệu tổng quát:

### 1. Các form trong code C#.

Đây là một chương trình quản lý ngân hàng có thể giúp người dùng quản lí các tài khoản ngân hàng của họ cũng như đăng nhập dưới quyền admin để quản lí các tài khoản trong ngân hàng. Chương trình được viết bằng ngôn ngữ lập trình C# và sử dụng cơ sở dữ liệu để lưu trữ thông tin khách hàng và giao dịch.

#### A screenshot of a computer program Description automatically generatedA screenshot of a computer program Description automatically generatedCác Form chính và chi tiết:

Ban đầu, chúng em sử dụng một form Method kế thừa Form để có thể viết nhiều hàm có thể dùng kế thừa nhằm mục đích dễ tìm kiếm lỗi hơn.

Các hàm trong Method bao gồm:

HandleMouesUpActions và HandleMouseDownActions nhằm mục đích hỗ trợ ẩn mật khẩu.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generatedHandleEnter và HandleLeave hỗ trợ hover cho textbox User và textbox Password.

A screen shot of a computer

Description automatically generatedCheckNullOrWhiteSpace kiểm tra xem textbox có bị để trống hay không

A computer screen with colorful text

Description automatically generated Textbox handle empty và Textbox enter more properties nhằm giúp cho giao diện đẹp hơn, cụ thể sẽ có ví dụ khi sử dụng ở Form resetPassword.

A computer screen with text

Description automatically generated

#### Form đăng nhập:

Form đăng nhập(login) là form xuất hiện ngay khi khởi chạy chương trình, giúp người dùng đăng nhập vào tài khoản.

A screenshot of a login box

Description automatically generatedTất nhiên FormLogin kế thừa Method đã nói ở trên.

A screen shot of a computer program

Description automatically generatedTại form load, các txtTitle.Parent hay lblMessege.Parent đều được để nền là hình ảnh nền của cả form.

A computer screen shot of a program

Description automatically generatedHàm Login sẽ kiểm tra xem tài khoản mật khẩu có hợp lệ hay không để chuyển tới form main. Nếu không đúng, sẽ hiện lblMessege rằng “Incorrect username or password”.

Hàm Logout nhằm mục đích đăng xuất và trở lại form login.

A screen shot of a computer program

Description automatically generatedKhi nhấn btn đăng nhập, hàm Login sẽ được gọi.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generatedGọi hàm HandleMouesUpActions và HandleMouseDownActions khi nhấn giữ biểu tượng con mắt trong textbox password.

A computer screen with text on it

Description automatically generatedGọi hàm HandleEnter và HandleLeave khi người dùng để con trỏ chuột vào trong textbox txtUser hoặc txtPassword.

A computer screen shot of a program code

Description automatically generatedChuyển sang Form ForgetPassword hoặc NewAccount khi nhấn vào lsForgetPassword hoặc lsNewAccount.

#### A screenshot of a computer Description automatically generatedForm ResetPassword

A computer screen with white text

Description automatically generatedTương tự Form đăng nhập, khi load, các nút lblTitle và btn Home được xử lý để có nền là nền cảu form

Đoạn code này nhằm thiết kế các event khi để trỏ chuột lên các textbox họ tên, cccd, số điện thoại và tên tài khoản (gọi hàm Textbox handle empty và Textbox enter more properties từ Method). Ví dụ:

A computer screen shot of a program

Description automatically generated

A screenshot of a computer

Description automatically generatedKhi để trỏ chuột vào textbox txtCCCD, lblCCCD sẽ hiện lên và txtCCCD.text sẽ thành “”.

A screen shot of a computer program

Description automatically generatedBtnHome phía trên bên phải khi click vào sẽ quay trở lại FormLogin.

A computer screen with white text

Description automatically generatedHàm ResetPassword để gán các event Enter vào tất cả các textbox.

A computer screen shot of a program code

Description automatically generatedHàm checkLog nhằm mục đích kiểm tra tên, cccd, phone, username xem có trùng với dữ liệu của database hay không, nếu trùng sẽ chuyển sang Form newPass, không thì hiện messageBox Error.

#### A screenshot of a computer Description automatically generatedForm NewPass

A computer screen with white text

Description automatically generatedKhi Form load sẽ hiện tên user ở góc trên bên trái.

A computer screen shot of a program

Description automatically generatedHàm success được gọi khi nhấn btnConfirm, kiểm tra xem có bỏ trống 2 txt, 2 mật khẩu có trùng hay không, sau đó lưu vào trong database.

#### A screenshot of a computer Description automatically generatedFormNewAccount

A computer screen shot of a program

Description automatically generatedKhi nhấn vào btn xác nhận, chương trình sẽ kiểm tra các txt có trống hay không, tiếp đó là 2 txt mật khẩu có trùng hay không. Nếu thông tin đầy đủ, tài khoản sẽ được thêm vào database.

A black screen with white text

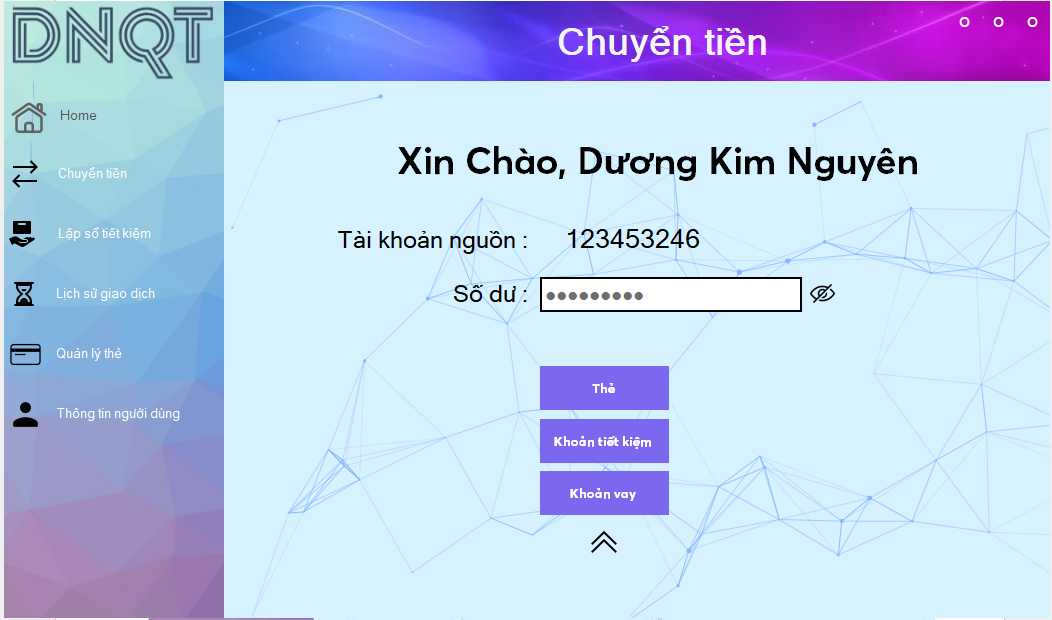
Description automatically generatedA screen shot of a computer program

Description automatically generatedForm này cũng gọi hàm HandleMouesUpActions và HandleMouseDownActions từ Method để ẩn mật khẩu cũng như gọi hàm HandleEnter và HandleLeave cùng với cài đặt nút Home.

A computer screen with text

Description automatically generatedSử dụng event leave để kiểm tra xem thông tin CCCD đã tồn tại hay chưa, tương tự với 2 txt Số điện thoại và username.

#### Form Main



A computer screen shot of a code

Description automatically generatedConstructor của FormMain.

A screen shot of a computer

Description automatically generatedEvent Load của chương trình sẽ kiểm tra xem tài khoản đăng nhập vào là thuộc quyền user hay quyền admin.

A computer screen shot of a program

Description automatically generated Điều đó sẽ dẫn đến giao diện thay đổi tùy theo quyền tài khoản mà người đăng nhập đăng nhập vào trong Form đăng nhập

A computer screen with white text

Description automatically generated

A computer screen shot of a program code

Description automatically generatedHàm OpenChildForm được gọi khi cần mở một form con nào đó.

A computer screen shot of a program

Description automatically generatedCài đặt các button khi nhấn vào sẽ mở ra form child.

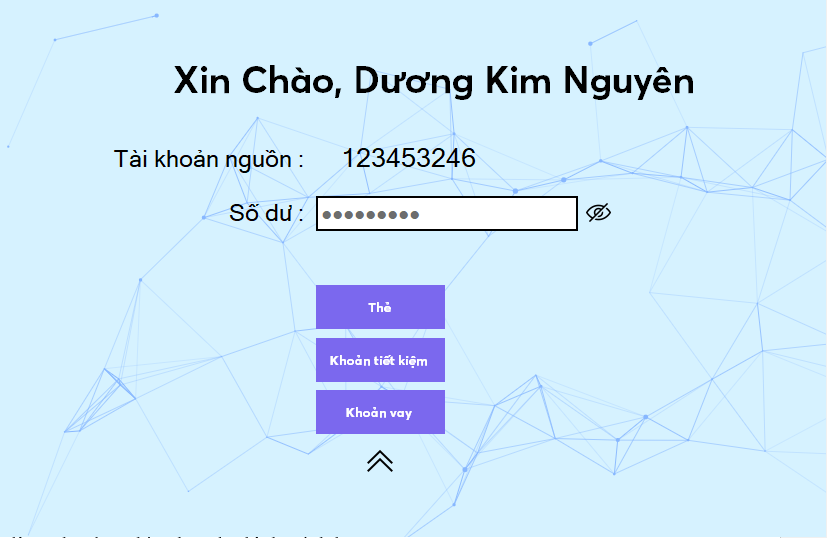
A screen shot of a computer program

Description automatically generatedHàm EnableButton được thiết kế nhằm mục đích khiến cho người dùng không thể click lại button đã click trước đó.

Custom 3 nút tắt chương trình, phóng to cửa sổ và thu nhỏ của sổ.

#### A computer screen with white and blue text Description automatically generated

#### Form Home



A computer screen shot of a program

Description automatically generated Event Load truyền dữ liệu tài khoản và số dư tài khoản từ database vào trong hai txtAccountBalance và lblNumberBank trong Form Home.

Đoạn code này nhằm mục đích hiện lên 3 btn Thẻ, Khoản tiết kiệm, và Khoản vay.

A computer screen shot of a program

Description automatically generatedĐoạn code này là để ẩn số dư tài khoản.

#### Form transferMoney

A computer screen with white text

Description automatically generatedCác giá trị khởi tạo

A screen shot of a computer program

Description automatically generatedHàm khởi tạo cũng như hàm timer tính ngày để đưa vào thông tin chuyển khoản.

A computer screen with white text

Description automatically generatedEvent Load nhằm đưa thông tin số dư và thông tin ngân hàng vào trong hai txtBank và txtAccountBalanse, cũng như ngăn việc người dùng có thể thay đổi thông tin số dư tài khoản.

Khi nhấn vào btn Confirm, chương trình sẽ kiểm tra xem txtMoney có để trống hay không sau đó thực hiện việc chuyển tiền., lưu thông tin vào trong database cũng như gọi hàm reset để reset mọi thông tin đang hiện trong textbox và đồng thời hiện messegeBox thông báo chuyển tiền thành công. Nếu txtMoney đang trống sẽ hiện messageBox Vui lòng nhập số tiền.

A computer screen shot of a program

Description automatically generatedA black rectangle with white text

Description automatically generated

Event txtBank\_OnSelectedIndexChanged được gọi nhằm mục đích tự động thêm vào nội dung chuyển tiền vào txtContent. A computer screen with white and blue text

Description automatically generated

Event này cũng cùng mục đích nhưng bên cạnh đó còn kiểm tra stk có tồn tại hoặc hợp lệ hay không.A computer screen shot of a program

Description automatically generated

Event txtMoney\_Leave nhằm kiểm tra xem số tiền chuyển có lớn hơn 1000vnđ hay không, nếu không sẽ xuất hiện MessageBox. A computer screen shot of a program code

Description automatically generated

Event txtContent\_Leave kiểm tra nội dung chuyển khoản. A computer screen with white text

Description automatically generated

#### Form Saving

#### A screenshot of a login form Description automatically generated

A computer screen with white text

Description automatically generatedCác giá trị khởi tạo:

A computer screen shot of a program code

Description automatically generatedHàm khởi tạo cùng hàm timer để lấy thông tin ngày tháng.

Hàm UpdateValue gọi Hàm tinhLaiSuat nhằm tính giá trị số tiền gửi và số tiền thu được sau khi đăng kí sổ tiết kiệm.

A computer screen shot of a code

Description automatically generated

A black screen with white text

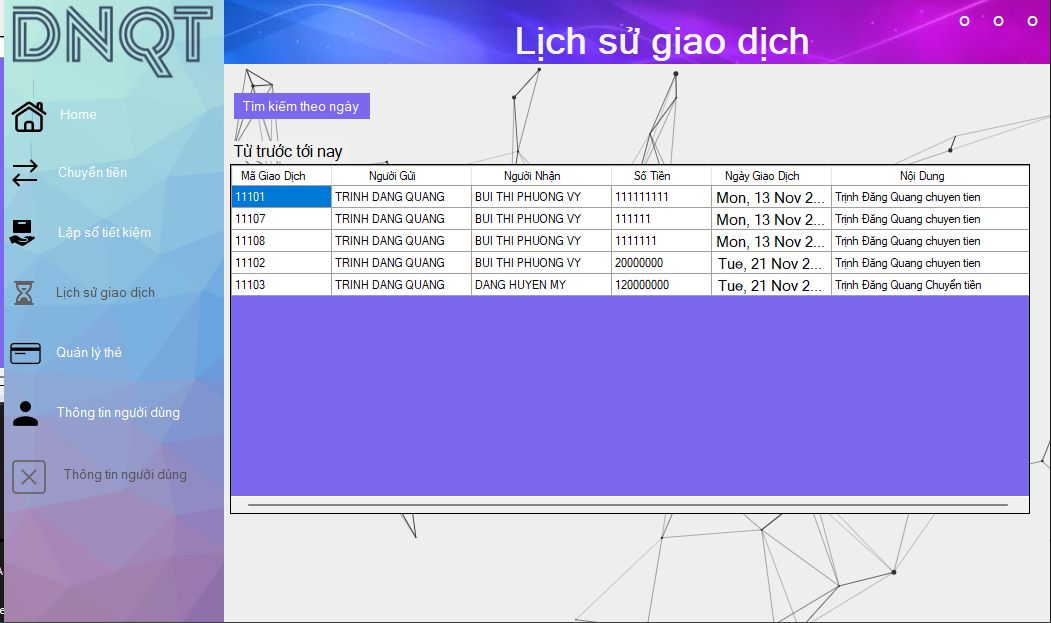
Description automatically generatedA screen shot of a computer program

Description automatically generatedEvent txtSendingTerm\_OnSelectedIndexChanged thực hiện gán laiSuat và dateEnd mỗi lần giá trị được chọn thay đổi cùng với đó là gọi hàm UpdateValue để thay đổi giá trị tiền nhận được liên tục. Và hàm UpdateValue cũng được gọi mỗi khi txtMoney bị thay đổi nội dung.

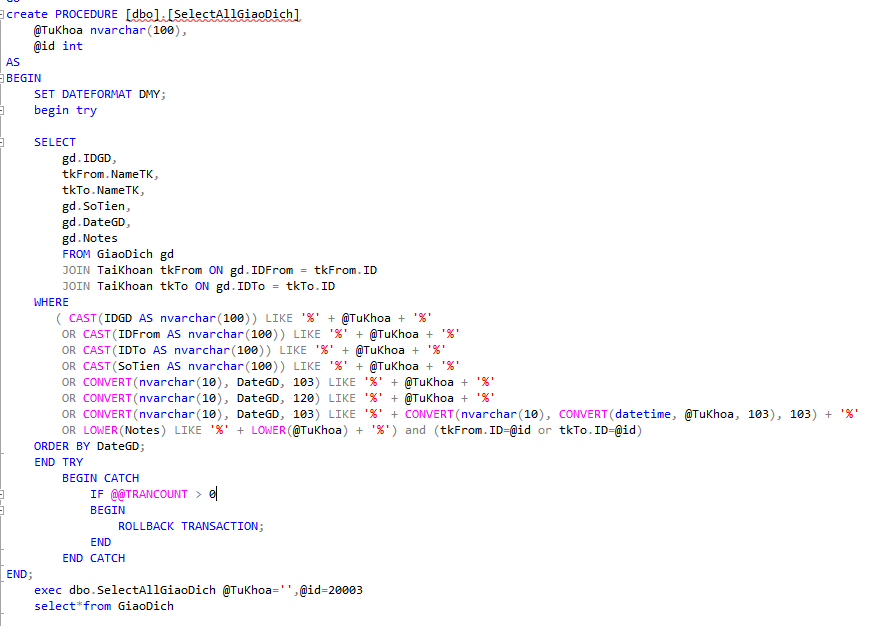
A computer screen with text

Description automatically generatedKhi nhấn btn Confirm, tài khoản sẽ tự động bị trừ đi số tiền đã nộp vào sổ tiết kiệm, cùng với đó là thành công trong việc đăng kí dịch vụ lập sổ tiết kiệm, mọi thông tin sẽ được đưa vào trong database.

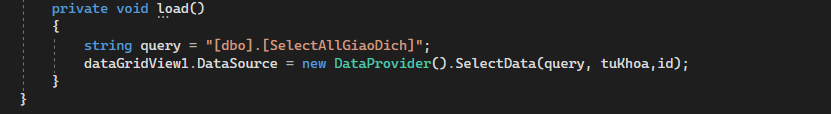
#### FormHistory

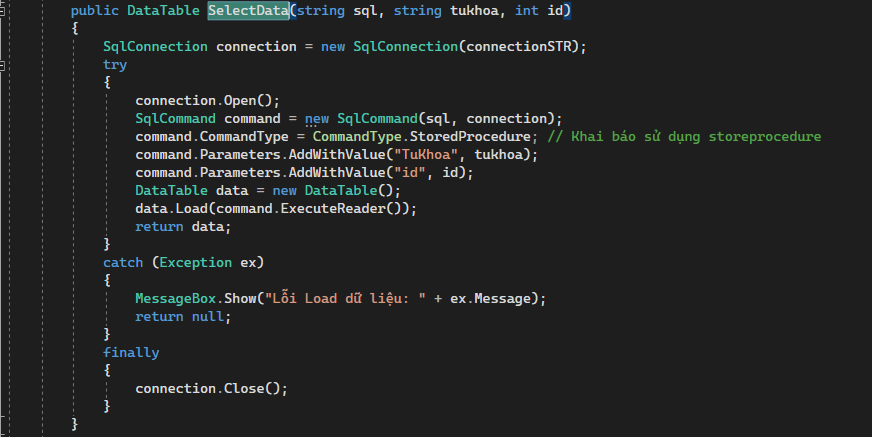


Procedure select data trong bảng Giao dịch kết hợp với tìm kiếm

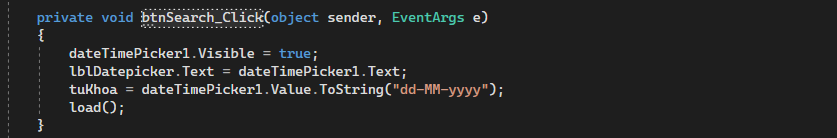


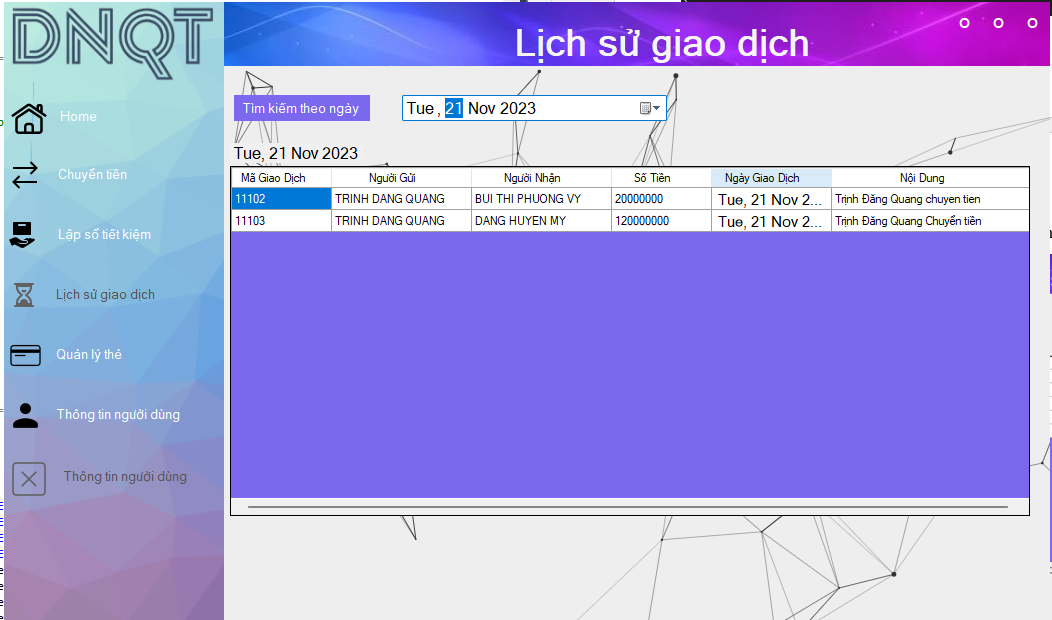
Hàm load dùng để lấy dữ liệu trong code c# để hiển thị trên form C#



Hàm kết nối c# với database

Sự kiện xử lý tìm kiếm gắn cho event click của btnSearch và valueChanged dateTimePicker1



Form History sau khi lọc kết quả tìm kiếm

#### FormCardManegement

A screenshot of a card

Description automatically generated Một card.cs đã được tạo ra để định hình hình dáng của tấm thẻ ngân hàng public partial class Card: UserControlA screenshot of a computer

Description automatically generated.

A computer screen shot of code

Description automatically generatedCác giá trị và hàm khởi tạo

A computer screen with white text

Description automatically generatedEvent CardLoad sắp xếp thứ tự các thành phần trong Card.

A computer screen with white text and black background

Description automatically generatedClick vào nút khóa thẻ để có thể khóa thẻ

A screen shot of a computer code

Description automatically generatedClick vào nút thông tin để xem được thông tin chi tiết của thẻ

A screenshot of a card

Description automatically generatedDẫn tới Form card\_detail

A screen shot of a computer program

Description automatically generatedA computer screen shot of a code

Description automatically generatedThiết kế đơn giản hiệu quả với mục đích kiểm tra thông tin là chính.

A computer screen with white text

Description automatically generatedQuay trở lại với FormCardManegement, các giá trị và hàm khởi tạo:

A computer screen with white text

Description automatically generatedNgay ở event FormCardManegement\_Load đã thực hiện việc tạo card mới và truyền thông tin người dùng vào các giá trị của card từ database.

A computer screen with many colorful text

Description automatically generated with medium confidenceNhấn qua lại giữa 2 btn để xem thêm các thẻ ngân hàng khả dụng.

Nếu chưa có bất kì thẻ ngân hàng nào, thực hiện tạo thẻ mới bằng cách nhấn vào btn Tạo thẻ.

A screenshot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generatedĐiều này dẫn đến form Link\_card.

A screen shot of a computer code

Description automatically generatedCác giá trị khởi tạo

A computer screen shot of a code

Description automatically generatedHàm khởi tạo lấy dữ liệu thông tin user và điền sẵn vào txtName, linkedAcc.

A screen shot of a computer program

Description automatically generatedEvent load bổ sung thông tin số dư và cho dateStart có giá trị ngày hiện tại.

Sau khi nhấn btn Save, hệ thống sẽ random một mã ngân hàng ngẫu nhiên, cho giá trị dateEnd là 5 năm sau đó. Dữ liệu được đưa về hệ thống, tạo một card mới và Show lên panel parent.

A computer screen shot of a program

Description automatically generated

#### A screenshot of a computer Description automatically generated FormInformation

Các giá trị, hàm khởi tạo cùng với event FormInformation\_Load nhằm truyền thông tin vào các txt tên, sđt, cccd, số dư tài khoản và không thể bị thay đổi

A computer screen shot of a program

Description automatically generated

A screen shot of a computer

Description automatically generatedHàm select để lấy thông tin từ database và truyền vào giá trị các biến.

Nhấn btn Đổi mật khẩu để hiện ra các txt mới phục vụ cho việc đổi mật khẩu. A black screen with white text

Description automatically generated

A screen shot of a computer

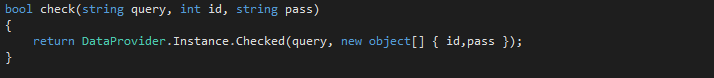
Description automatically generated Hàm VisualChange giúp các label thay đổi mật khẩu cùng các txt và 2 btn thay đổi dựa trên value.

Giao diện khi nhấn btn Đổi mật khẩu: A screenshot of a computer

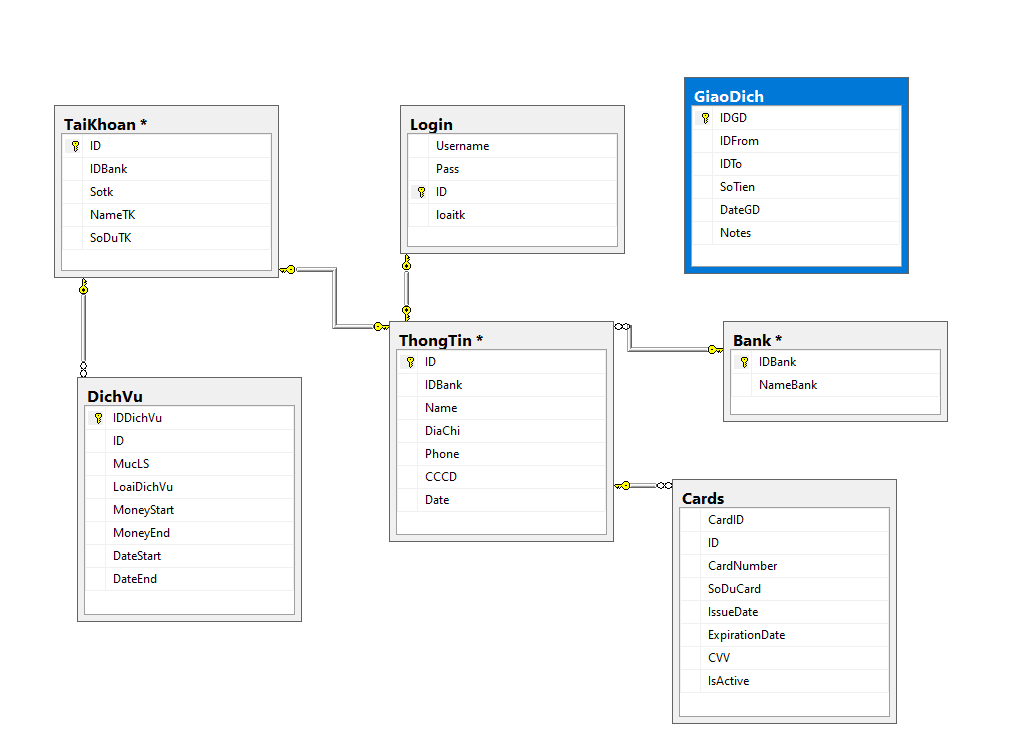
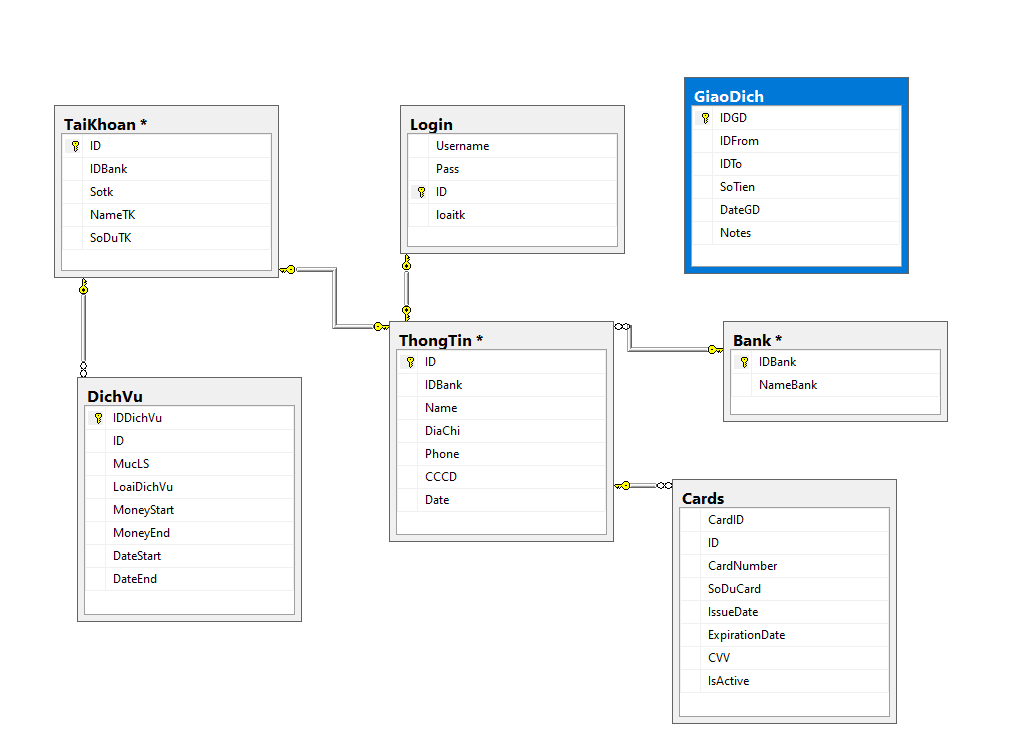
Description automatically generated

A computer screen shot of a program

Description automatically generatedKhi nhấn btn Confirm, cần phải nhập đúng mật khẩu hiện tại (lấy thông tin từ database để so sánh) nếu không sẽ phải nhập lại từ đầu, sau đó chương trình kiểm tra xem txtPassword và txtPassword2 có trùng nhau hay không, và kiểm tra luôn xem cả 2 có trùng mật khẩu cũ hay không. Sau tất cả các bước, mật khẩu sẽ được thay đổi và thực hiện đóng lại 3 ô thay đổi mật khẩu bằng cách gọi hàm VisualChange.



### 2. Database

1. Database diagrams
2. Chi tiết các table

* Table [dbo].[Bank]

*SET QUOTED\_IDENTIFIER ON*

*GO*

*CREATE TABLE [dbo].[Bank](*

*[IDBank] [int] NOT NULL,*

*[NameBank] [nvarchar](50) NOT NULL,*

*PRIMARY KEY CLUSTERED*

*(*

*[IDBank] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

Trong bảng Bank gồm các column là [IDBank], [NameBank] với chức năng lưu trữ data danh sách các ngân hàng.

* Table [dbo].[GiaoDich]

*CREATE TABLE [dbo].[GiaoDich] (*

*[IDGD] [int] NOT NULL,*

*[IDFrom] [int] NOT NULL,*

*[IDTo] [int] NOT NULL,*

*[SoTien] [int] NOT NULL,*

*[DateGD] [smalldatetime] NOT NULL,*

*[Notes] [nvarchar](200) NULL,*

*PRIMARY KEY CLUSTERED*

*(*

*[IDGD] ASC*

*) WITH (*

*PAD\_INDEX = OFF,*

*STATISTICS\_NORECOMPUTE = OFF,*

*IGNORE\_DUP\_KEY = OFF,*

*ALLOW\_ROW\_LOCKS = ON,*

*ALLOW\_PAGE\_LOCKS = ON*

*) ON [PRIMARY]*

*) ON [PRIMARY];*

Bảng GiaoDich với chức năng lưu trữ data các giao dịch từ các tài khoản trong ngân hàng.

* Table [dbo]. [Cards]

*CREATE TABLE [dbo].[Cards] (*

*CardID INT,*

*[ID] INT NOT NULL,*

*CardNumber VARCHAR(19) NOT NULL,*

*[SoDuCard] INT NOT NULL,*

*IssueDate SMALLDATETIME NOT NULL,*

*ExpirationDate SMALLDATETIME NOT NULL,*

*CVV NVARCHAR(3) NOT NULL,*

*IsActive BIT NOT NULL DEFAULT 1,*

*CONSTRAINT UQ\_CardID\_ID UNIQUE (CardID, ID)*

*);*

Bảng Cards với chức năng lưu trữ data các thẻ đã được thêm vào từ các tài khoản nguồn.

* Table [dbo].[ DichVu]

*CREATE TABLE [dbo].[GiaoDich] (*

*[IDGD] [int] NOT NULL,*

*[IDFrom] [int] NOT NULL,*

*[IDTo] [int] NOT NULL,*

*[SoTien] [int] NOT NULL,*

*[DateGD] [smalldatetime] NOT NULL,*

*[Notes] [nvarchar](200) NULL,*

*PRIMARY KEY CLUSTERED*

*(*

*[IDGD] ASC*

*) WITH (*

*PAD\_INDEX = OFF,*

*STATISTICS\_NORECOMPUTE = OFF,*

*IGNORE\_DUP\_KEY = OFF,*

*ALLOW\_ROW\_LOCKS = ON,*

*ALLOW\_PAGE\_LOCKS = ON*

*) ON [PRIMARY]*

*) ON [PRIMARY];*

Chức năng: Lưu trữ thông tin về các dịch vụ ngân hàng, bao gồm mức lãi suất ([MucLS]), loại dịch vụ ([LoaiDichVu]), khoảng tiền áp dụng dịch vụ ([MoneyStart] và [MoneyEnd]), ngày bắt đầu ([DateStart]) và ngày kết thúc ([DateEnd]).

* Table [dbo]. [Login]

*CREATE TABLE [dbo].[Login](*

*[Username] [char](50) NOT NULL,*

*[Pass] [char](50) NOT NULL,*

*[ID] [int] NOT NULL,*

*[loaitk] [int] NOT NULL*

*) ON [PRIMARY]*

Chức năng: Lưu trữ thông tin đăng nhập, bao gồm tên đăng nhập ([Username]), mật khẩu ([Pass]), mã người dùng ([ID]), và loại tài khoản ([loaitk]).

* TABLE [dbo].[TaiKhoan]

*CREATE TABLE [dbo].[TaiKhoan](*

*[ID] [int] NOT NULL,*

*[IDBank] [int] Not NUll,*

*[Sotk] [int] NOT NULL,*

*[NameTK] [nvarchar](50) NOT NULL,*

*[SoDuTK] [int] NOT NULL,*

*PRIMARY KEY CLUSTERED*

*(*

*[ID] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

Chức năng: Lưu trữ thông tin về các tài khoản ngân hàng, bao gồm mã tài khoản ([ID]), mã ngân hàng ([IDBank]), số tài khoản ([Sotk]), tên tài khoản ([NameTK]), và số dư tài khoản ([SoDuTK]).

* TABLE [dbo].[ThongTin]

*CREATE TABLE [dbo].[ThongTin](*

*[ID] [int] NOT NULL,*

*[IDBank] [int] NOT NULL,*

*[Name] [nvarchar](50) NOT NULL,*

*[DiaChi] [nvarchar](100) NULL,*

*[Phone] [char](10) NOT NULL,*

*[CCCD] [char](12) NOT NULL,*

*[Date] [smalldatetime] NOT NULL,*

*PRIMARY KEY CLUSTERED*

*(*

*[ID] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

*) ON [PRIMARY]*

Chức năng: Lưu trữ thông tin cá nhân của người dùng, bao gồm mã người dùng ([ID]), mã ngân hàng ([IDBank]), tên người dùng ([Name]), địa chỉ ([DiaChi]), số điện thoại ([Phone]), số chứng minh nhân dân ([CCCD]), và ngày sinh ([Date]).

#### Dữ liệu các bảng.

Dữ liệu trong các bảng được thêm vào để test các tính năng và một số được thêm vào từ proceduce thông qua dữ liệu được lấy từ code c#.

*INSERT [dbo].[Bank] ([IDBank], [NameBank]) VALUES (1210001, N'DNQT Bank')*

*INSERT [dbo].[Bank] ([IDBank], [NameBank]) VALUES (1210002, N'DQ Bank')*

*INSERT [dbo].[Bank] ([IDBank], [NameBank]) VALUES (1210003, N'VietNam Bank ')*

*INSERT [dbo].[Bank] ([IDBank], [NameBank]) VALUES (1210004, N'MC Bank')*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate],*

*[ExpirationDate], [CVV], [IsActive]) VALUES (1, 20001, N'1234 5678 9012 3456', 1000, CAST(N'2023-01-01 00:00:00' AS SmallDateTime), CAST(N'2025-12-31 00:00:00' AS SmallDateTime), N'561', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (2, 20001, N'9876 5432 1098 7654', 1500, CAST(N'2023-02-01 00:00:00' AS SmallDateTime), CAST(N'2026-01-31 00:00:00' AS SmallDateTime), N'371', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (1, 20002, N'1111 2222 3333 4444', 0, CAST(N'2023-03-01 00:00:00' AS SmallDateTime), CAST(N'2026-02-28 00:00:00' AS SmallDateTime), N'393', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (2, 20002, N'5555 6666 7777 8888', 0, CAST(N'2023-04-01 00:00:00' AS SmallDateTime), CAST(N'2026-03-31 00:00:00' AS SmallDateTime), N'666', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (1, 20003, N'3333 9999 5555 3333', 0, CAST(N'2023-03-01 00:00:00' AS SmallDateTime), CAST(N'2026-02-28 00:00:00' AS SmallDateTime), N'422', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (2, 20003, N'3333 3333 1111 1111', 0, CAST(N'2023-04-01 00:00:00' AS SmallDateTime), CAST(N'2026-03-31 00:00:00' AS SmallDateTime), N'538', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (3, 20003, N'1234 1234 1234 4444', 0, CAST(N'2023-03-01 00:00:00' AS SmallDateTime), CAST(N'2026-02-28 00:00:00' AS SmallDateTime), N'934', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (4, 20003, N'4321 4321 4321 8888', 0, CAST(N'2023-04-01 00:00:00' AS SmallDateTime), CAST(N'2026-03-31 00:00:00' AS SmallDateTime), N'812', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (3, 20002, N'3233 9199 6655 1133', 0, CAST(N'2023-03-01 00:00:00' AS SmallDateTime), CAST(N'2026-02-28 00:00:00' AS SmallDateTime), N'696', 1)*

*INSERT [dbo].[Cards] ([CardID], [ID], [CardNumber], [SoDuCard], [IssueDate], [ExpirationDate], [CVV], [IsActive]) VALUES (4, 20002, N'6783 3453 2211 1331', 0, CAST(N'2023-04-01 00:00:00' AS SmallDateTime), CAST(N'2026-03-31 00:00:00' AS SmallDateTime), N'747', 1)*

*SET IDENTITY\_INSERT [dbo].[DichVu] ON*

*INSERT [dbo].[DichVu] ([IDDichVu], [ID], [MucLS], [LoaiDichVu], [MoneyStart], [MoneyEnd], [DateStart], [DateEnd]) VALUES (11, 20003, 0.07, N'Sổ tiết kiệm', 50000000, 53500000, CAST(N'2023-11-21 00:00:00' AS SmallDateTime), CAST(N'2026-11-21 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[DichVu] ([IDDichVu], [ID], [MucLS], [LoaiDichVu], [MoneyStart], [MoneyEnd], [DateStart], [DateEnd]) VALUES (12, 20003, 0.07, N'Sổ tiết kiệm', 20000000, 21400000, CAST(N'2023-11-21 00:00:00' AS SmallDateTime), CAST(N'2024-11-21 00:00:00' AS SmallDateTime))*

*SET IDENTITY\_INSERT [dbo].[DichVu] OFF*

*INSERT [dbo].[GiaoDich] ([IDGD], [IDFrom], [IDTo], [SoTien], [DateGD], [Notes]) VALUES (11101, 20003, 20002, 111111111, CAST(N'2023-11-13 00:00:00' AS SmallDateTime), N'Trịnh Đăng Quang chuyen tien')*

*INSERT [dbo].[GiaoDich] ([IDGD], [IDFrom], [IDTo], [SoTien], [DateGD], [Notes]) VALUES (11102, 20003, 20002, 20000000, CAST(N'2023-11-21 00:00:00' AS SmallDateTime), N'Trịnh Đăng Quang chuyen tien')*

*INSERT [dbo].[GiaoDich] ([IDGD], [IDFrom], [IDTo], [SoTien], [DateGD], [Notes]) VALUES (11103, 20003, 20009, 120000000, CAST(N'2023-11-21 00:00:00' AS SmallDateTime), N'Trịnh Đăng Quang Chuyển tiền')*

*INSERT [dbo].[GiaoDich] ([IDGD], [IDFrom], [IDTo], [SoTien], [DateGD], [Notes]) VALUES (11107, 20003, 20002, 111111, CAST(N'2023-11-13 00:00:00' AS SmallDateTime), N'Trịnh Đăng Quang chuyen tien')*

*INSERT [dbo].[GiaoDich] ([IDGD], [IDFrom], [IDTo], [SoTien], [DateGD], [Notes]) VALUES (11108, 20003, 20002, 1111111, CAST(N'2023-11-13 00:00:00' AS SmallDateTime), N'Trịnh Đăng Quang chuyen tien')*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'quang ', N'123 ', 10001, 0)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'trinhdangquang ', N'123 ', 20001, 1)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'buithiphuongvy ', N'123 ', 20002, 1)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'1 ', N'1 ', 20003, 1)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'2 ', N'2 ', 20006, 1)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'nhudinhbui ', N'123 ', 20007, 1)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'tule ', N'123 ', 20008, 1)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'my ', N'1 ', 20009, 1)*

*INSERT [dbo].[Login] ([Username], [Pass], [ID], [loaitk]) VALUES (N'raucon ', N'1 ', 20011, 1)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20001, 1210001, 123453246, N'TRINH DANG QUANG', 5000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20002, 1210001, 7983246, N'BUI THI PHUONG VY', 1000000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20003, 1210003, 999654111, N'TRINH DANG QUANG', 55000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20004, 1210002, 988888812, N'TRINH DANG QUANG', 40000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20005, 1210004, 988555821, N'BUI DINH NHU', 20000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20006, 1210001, 123453246, N'DUONG KIM NGUYEN', 5000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20007, 1210001, 343873212, N'BUI DINH NHU', 1000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20008, 1210001, 123453246, N'LE NGUYEN THANH TU', 500000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20009, 1210001, 653183187, N'DANG HUYEN MY', 175000000)*

*INSERT [dbo].[TaiKhoan] ([ID], [IDBank], [Sotk], [NameTK], [SoDuTK]) VALUES (20011, 1210001, 653183186, N'MESSI', 0)*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (10001, 1210001, N'Trịnh Đăng Quang', NULL, N'0843123654', N'040203013420', CAST(N'2003-12-26 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20001, 1210001, N'Trịnh Đăng Quang', NULL, N'0843345868', N'040203010919', CAST(N'2003-12-26 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20002, 1210001, N'Bùi Thị Phương Vy', N'Dĩ An, Bình Dương', N'0986318750', N'030101562343', CAST(N'2002-11-18 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20003, 1210003, N'Trịnh Đăng Quang', NULL, N'0843345869', N'040203010916', CAST(N'2003-12-26 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20004, 1210002, N'Trịnh Đăng Quang', NULL, N'0843341869', N'040444410916', CAST(N'2003-12-26 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20005, 1210004, N'Bùi Đình Nhu', NULL, N'0982762877', N'040203567843', CAST(N'2003-02-06 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20006, 1210001, N'Dương Kim Nguyên', NULL, N'0943543651', N'040203343321', CAST(N'2003-11-08 00:00:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20007, 1210001, N'Bùi Đình Nhu', NULL, N'0978546432', N'1234567321 ', CAST(N'2023-11-21 10:52:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20008, 1210001, N'Lê Nguyễn Thanh Tú', NULL, N'0364345765', N'75433212343 ', CAST(N'2023-11-21 10:52:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20009, 1210001, N'Đặng Huyền My', NULL, N'0987654321', N'1123452324 ', CAST(N'2023-11-21 10:50:00' AS SmallDateTime))*

*INSERT [dbo].[ThongTin] ([ID], [IDBank], [Name], [DiaChi], [Phone], [CCCD], [Date]) VALUES (20011, 1210001, N'messi', NULL, N'0999999999', N'444444444 ', CAST(N'2023-11-21 11:10:00' AS SmallDateTime))*

#### Khóa ngoại(foreign key constraints)

Thêm các ràng buộc và chỉ mục (index), cũng như việc thiết lập các ràng buộc khóa ngoại (foreign key constraints) giữa các bảng.

*ALTER TABLE [dbo].[Cards] ADD CONSTRAINT [UQ\_CardID\_ID] UNIQUE NONCLUSTERED*

*(*

*[CardID] ASC,*

*[ID] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

Đoạn mã này thêm một ràng buộc duy nhất (UNIQUE CONSTRAINT) cho bảng Cards trên cột CardID và ID, đảm bảo rằng mỗi cặp giá trị trong hai cột này là duy nhất. Các tùy chọn khác được cấu hình, chẳng hạn như PAD\_INDEX (tắt chế độ padding), STATISTICS\_NORECOMPUTE, SORT\_IN\_TEMPDB, IGNORE\_DUP\_KEY, ONLINE, ALLOW\_ROW\_LOCKS, và ALLOW\_PAGE\_LOCKS.

*GO*

*SET ANSI\_PADDING ON*

*GO*

*/\*\*\*\*\*\* Object: Index [UQ\_\_ThongTin\_\_5C7E359EEBF33560] Script Date: Tue, 21 Nov 2023 05:00:21 PM \*\*\*\*\*\*/*

*ALTER TABLE [dbo].[ThongTin] ADD UNIQUE NONCLUSTERED*

*(*

*[Phone] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

Tương tự, đoạn mã này thêm một chỉ mục duy nhất cho cột Phone trong bảng ThongTin.

*GO*

*SET ANSI\_PADDING ON*

*GO*

*/\*\*\*\*\*\* Object: Index [UQ\_\_ThongTin\_\_A955A0AA3FF80D51] Script Date: Tue, 21 Nov 2023 05:00:21 PM \*\*\*\*\*\*/*

*ALTER TABLE [dbo].[ThongTin] ADD UNIQUE NONCLUSTERED*

*(*

*[CCCD] ASC*

*)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]*

Tương tự như trên, đoạn mã này thêm một chỉ mục duy nhất cho cột CCCD trong bảng ThongTin.

*GO*

*SET ANSI\_PADDING ON*

*GO*

*ALTER TABLE [dbo].[Cards] WITH CHECK ADD FOREIGN KEY([ID])*

*REFERENCES [dbo].[ThongTin] ([ID])*

*GO*

*ALTER TABLE [dbo].[DichVu] WITH CHECK ADD FOREIGN KEY([ID])*

*REFERENCES [dbo].[TaiKhoan] ([ID])*

*GO*

*ALTER TABLE [dbo].[Login] WITH CHECK ADD FOREIGN KEY([ID])*

*REFERENCES [dbo].[ThongTin] ([ID])*

*GO*

*ALTER TABLE [dbo].[TaiKhoan] WITH CHECK ADD FOREIGN KEY([ID])*

*REFERENCES [dbo].[ThongTin] ([ID])*

*GO*

*ALTER TABLE [dbo].[ThongTin] WITH CHECK ADD FOREIGN KEY([IDBank])*

*REFERENCES [dbo].[Bank] ([IDBank])*

*GO*

Đoạn mã này thêm ràng buộc khóa ngoại cho năm bảng khác nhau ([Cards], [DichVu], [Login], [TaiKhoan], [ThongTin]) trỏ đến bảng [ThongTin]. Ràng buộc này đảm bảo rằng giá trị trong cột ID của các bảng khác đều phải là khóa ngoại trỏ đến giá trị ID trong bảng [ThongTin]. WITH CHECK đảm bảo rằng dữ liệu đã tồn tại phải tuân thủ ràng buộc.

#### Các Proceduce và trigger

##### Proceduce [dbo].[ADD\_ACCOUNT]

create proc [dbo].[ADD\_ACCOUNT]

@name nvarchar(50),

@cccd char(12),

@phone char(10),

@user varchar(150),

@pass varchar(50),

@id int,

@date smalldatetime,

@loaitk int

,@bank int

,@stk int

,@sodutk int

,@Nametk nvarchar(50)

AS

begin

set dateformat DMY

begin try

BEGIN TRANSACTION;

-- Kiểm tra xem tài khoản có tồn tại chưa

IF NOT EXISTS (SELECT 1 FROM ThongTin tt join Login lg on tt.ID=lg.ID where tt.CCCD=@cccd or tt.Phone=@phone or lg.Username=@user

)

BEGIN

-- Thêm tài khoản mới

INSERT INTO ThongTin (Name, ID, Phone, CCCD,Date,IDBank) VALUES (@name, @id, @phone, @cccd,@date,@bank);

INSERT INTO Login (Username, Pass,ID,loaitk) VALUES (@user, @pass,@id,@loaitk);

insert into TaiKhoan(ID,IDBank,Sotk,SoDuTK,NameTK)values(@id,@bank,@stk,@sodutk,@Nametk)

COMMIT TRANSACTION;

END

ELSE

BEGIN

ROLLBACK TRANSACTION;

END

end try

BEGIN CATCH

IF @@TRANCOUNT >0

BEGIN

ROLLBACK TRANSACTION;

END

END CATCH

END;

Stored procedure này được thiết kế để thêm một tài khoản mới vào cơ sở dữ liệu, đồng thời đảm bảo rằng không có trùng lặp về CCCD, số điện thoại hoặc tên đăng nhập. Nếu có lỗi xảy ra trong quá trình thực hiện, nó sẽ quay lại trạng thái trước khi thực hiện bất kỳ thay đổi nào.

##### Proceduce [dbo].[CHECK\_ACCOUNT\_EXISTENCE]

CREATE PROCEDURE [dbo].[CHECK\_ACCOUNT\_EXISTENCE]

@Sotk INT,

@nameBank NVARCHAR(50)

AS

BEGIN

SELECT tk.NameTK

FROM dbo.TaiKhoan tk

JOIN dbo.Bank b ON tk.IDBank = b.IDBank

WHERE tk.Sotk = @Sotk AND b.NameBank = @nameBank

END

Stored procedure này có nhiệm vụ chủ yếu là kiểm tra sự tồn tại của tài khoản trong bảng TaiKhoan với số tài khoản (Sotk) và tên ngân hàng (nameBank) được chỉ định. Nếu có bản ghi nào thỏa mãn điều kiện, stored procedure sẽ trả về cột NameTK của bản ghi đó.

##### 5.3 Proceduce [dbo].[CHECK\_ACCOUNT\_EXISTENCE]

CREATE PROCEDURE [dbo].[CHECK\_FORGOT]

@name NVARCHAR(50),

@cccd CHAR(12),

@phone CHAR(10),

@userName NVARCHAR(100)

AS

BEGIN

SELECT \*

FROM dbo.Login

INNER JOIN dbo.ThongTin ON Login.ID = ThongTin.ID

WHERE (Username = @name OR Username = @userName) AND CCCD = @cccd AND Phone = @phone

END

Stored procedure này được thiết kế để kiểm tra thông tin khi người dùng quên mật khẩu. Nó kiểm tra xem có bản ghi nào trong bảng Login và ThongTin mà có Username giống với @name hoặc @userName, CCCD giống với @cccd, và Phone giống với @phone hay không. Nếu có, nó sẽ trả về tất cả các cột của các bản ghi thỏa mãn điều kiện.

##### 5.4 Proceduce [dbo].[CHECK\_LOGIN] CREATE PROC [dbo].[CHECK\_LOGIN]

@userName nvarchar(100),

@passWord nvarchar(100)

AS

BEGIN

SELECT \* FROM dbo.Login, dbo.ThongTin

WHERE Username =@userName AND Pass = @passWord

END

Stored procedure này được thiết kế để kiểm tra thông tin đăng nhập của người dùng. Nếu tìm thấy bản ghi nào trong bảng Login mà có Username giống @userName và Pass giống @passWord, nó sẽ trả về tất cả các cột từ cả hai bảng cho bản ghi đó.

##### 5.5 Proceduce [dbo].[GET\_NAME]

create proc [dbo].[GET\_NAME]

@userName nvarchar(100)

as

begin

select \* from dbo.Login inner join dbo.ThongTin on Login.ID = ThongTin.ID

where Username = @userName

end

Stored procedure này được thiết kế để lấy thông tin (tất cả các cột) từ bảng Login và ThongTin cho người dùng có tên đăng nhập (Username) bằng với giá trị được truyền vào tham số @userName.

##### 5.6 Proceduce [dbo].[INSERT\_GIAODICH]

CREATE PROCEDURE [dbo].[INSERT\_GIAODICH]

@IDGD INT,

@IDTo INT,

@IDFrom INT,

@SoTien INT,

@DateGD smalldatetime,

@Notes NVARCHAR(200)

AS

BEGIN

INSERT INTO [dbo].[GiaoDich] ([IDGD], [IDTo], [IDFrom], [SoTien], [DateGD], [Notes])

VALUES (@IDGD, @IDTo, @IDFrom, @SoTien, @DateGD, @Notes)

END

Stored procedure này được thiết kế để thêm một giao dịch mới vào bảng GiaoDich với thông tin được chỉ định qua các tham số đầu vào.

##### 5.7 Proceduce [dbo].[InsertCard]

CREATE PROCEDURE [dbo].[InsertCard]

@ID INT,

@CardNumber VARCHAR(19),

@SoDuCard INT,

@IssueDate SMALLDATETIME,

@ExpirationDate SMALLDATETIME

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO Cards (ID, CardNumber, SoDuCard, IssueDate, ExpirationDate)

VALUES (@ID, @CardNumber, @SoDuCard, @IssueDate, @ExpirationDate);

END;

Stored procedure này được thiết kế để thêm một thẻ mới vào bảng Cards với thông tin được chỉ định qua các tham số đầu vào. Việc tắt thông báo số lượng bản ghi ảnh hưởng có thể hữu ích trong trường hợp stored procedure chỉ thực hiện một thao tác thay đổi dữ liệu và không trả về kết quả dữ liệu.

##### 5.8 Proceduce [dbo].[InsertDichVu]

CREATE PROCEDURE [dbo].[InsertDichVu]

@ID INT,

@MucLS Float,

@LoaiDichVu NVARCHAR(50),

@MoneyStart INT,

@MoneyEnd INT,

@DateStart SMALLDATETIME,

@DateEnd SMALLDATETIME

AS

BEGIN

INSERT INTO [dbo].[DichVu] ([ID], [MucLS], [LoaiDichVu], [MoneyStart], [MoneyEnd], [DateStart], [DateEnd])

VALUES (@ID, @MucLS, @LoaiDichVu, @MoneyStart, @MoneyEnd, @DateStart, @DateEnd);

END;

Stored procedure này được thiết kế để thêm một dịch vụ mới vào bảng DichVu với thông tin được chỉ định qua các tham số đầu vào. Điều này có thể là phần của hệ thống quản lý các dịch vụ, trong đó mỗi dịch vụ được định nghĩa bởi các thuộc tính như mức lãi suất (MucLS), loại dịch vụ (LoaiDichVu), giới hạn số tiền (MoneyStart và MoneyEnd), và khoảng thời gian có hiệu lực (DateStart và DateEnd).

##### 5.9 Proceduce [dbo]. [SelectAllGiaoDich]

create PROCEDURE [dbo].[SelectAllGiaoDich]

@TuKhoa nvarchar(100),

@id int

AS

BEGIN

SET DATEFORMAT DMY;

begin try

SELECT

gd.IDGD,

tkFrom.NameTK,

tkTo.NameTK,

gd.SoTien,

gd.DateGD,

gd.Notes

FROM GiaoDich gd

JOIN TaiKhoan tkFrom ON gd.IDFrom = tkFrom.ID

JOIN TaiKhoan tkTo ON gd.IDTo = tkTo.ID

WHERE

( CAST(IDGD AS nvarchar(100)) LIKE '%' + @TuKhoa + '%'

OR CAST(IDFrom AS nvarchar(100)) LIKE '%' + @TuKhoa + '%'

OR CAST(IDTo AS nvarchar(100)) LIKE '%' + @TuKhoa + '%'

OR CAST(SoTien AS nvarchar(100)) LIKE '%' + @TuKhoa + '%'

OR CONVERT(nvarchar(10), DateGD, 103) LIKE '%' + @TuKhoa + '%'

OR CONVERT(nvarchar(10), DateGD, 120) LIKE '%' + @TuKhoa + '%'

OR CONVERT(nvarchar(10), DateGD, 103) LIKE '%' + CONVERT(nvarchar(10), CONVERT(datetime, @TuKhoa, 103), 103) + '%'

OR LOWER(Notes) LIKE '%' + LOWER(@TuKhoa) + '%') and (tkFrom.ID=@id or tkTo.ID=@id)

ORDER BY DateGD;

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

END CATCH

END;

Stored procedure [dbo].[SelectAllGiaoDich] được thiết kế để lấy thông tin về các giao dịch từ bảng GiaoDich. Nó nhận vào hai tham số: @TuKhoa (từ khóa tìm kiếm) và @id (ID của tài khoản liên quan).

Lọc và Tìm Kiếm:

Sử dụng các điều kiện LIKE để tìm kiếm theo nhiều trường dữ liệu như IDGD, IDFrom, IDTo, SoTien, DateGD, và Notes.

Có kiểm tra ngày tháng theo nhiều định dạng khác nhau.

Chỉ lấy các giao dịch liên quan đến tài khoản có ID là @id trong cột tkFrom hoặc tkTo.

Sắp Xếp Kết Quả:

Sắp xếp kết quả theo cột DateGD.

Xử Lý Ngoại Lệ:

Trong trường hợp có lỗi, kiểm tra nếu có giao dịch đang chạy, thực hiện rollback để quay trở lại trạng thái trước khi bắt đầu giao dịch.

Tóm lại, stored procedure này hỗ trợ việc lấy thông tin chi tiết về các giao dịch dựa trên từ khóa tìm kiếm và ID của tài khoản liên quan, đồng thời xử lý ngoại lệ khi cần thiết.

##### 5.9 TRIGGER Cards\_Insert

CREATE TRIGGER Cards\_Insert

ON dbo.Cards

INSTEAD OF INSERT

AS

BEGIN

SET NOCOUNT ON;

IF NOT EXISTS (

SELECT 1

FROM dbo.Cards c

JOIN inserted i ON c.CardNumber = i.CardNumber

)

BEGIN

DECLARE @MaxCardID INT;

SELECT @MaxCardID = MAX(CardID)

FROM dbo.Cards

WHERE ID IN (SELECT ID FROM inserted);

IF @MaxCardID IS NULL

SET @MaxCardID = 0;

IF EXISTS (SELECT 1 FROM inserted i WHERE NOT EXISTS (SELECT 1 FROM dbo.Cards c WHERE c.CardNumber = i.CardNumber))

BEGIN

INSERT INTO dbo.Cards (CardID, ID, CardNumber, SoDuCard, IssueDate, ExpirationDate, CVV, IsActive)

SELECT

ROW\_NUMBER() OVER (PARTITION BY i.ID ORDER BY i.CardNumber) + @MaxCardID,

i.ID,

i.CardNumber,

i.SoDuCard,

i.IssueDate,

i.ExpirationDate,

CAST((RAND() \* (999 - 100 + 1) + 100) AS INT),

i.IsActive

FROM inserted i;

END

END

END;

Trigger này được sử dụng để xử lý việc thêm bản ghi vào bảng Cards trong trường hợp có sự trùng lặp về số thẻ. Nó đặt ra các điều kiện để đảm bảo tính duy nhất của số thẻ và tự động tạo CardID mới nếu cần thiết.

# TỔNG KẾT

## I. Mục tiêu dự án

Mục tiêu của dự án là mô phỏng một hệ thống quản lý ngân hàng hiệu quả, có đầy đủ tính năng của một hệ thống quản lý ngân hàng, đáp ứng nhu cầu người dùng và đảm bảo tính bảo mật.

## II. Chức năng chính

Chương trình cung cấp các chức năng quản lý tài khoản, giao dịch, quản lý thẻ, mở sổ tiết kiệm online và tra cứu lịch sử giao dịch.

## III. Công nghệ sử dụng

Sử dụng cơ sở dữ liệu SQL Server, framework ASP.NET, ngôn ngữ lập trình C#, và các công nghệ khác để đảm bảo ổn định và linh hoạt.

## IV. Thách thức và giải quyết

Đối mặt với thách thức về việc đưa ra ý tưởng, thiết kế và kết nối với cơ sở dữ liệu, nhóm đã mở nhiều cuộc họp nhỏ cả online và offline để thống nhất hướng đi gỡ rối từng bước của tiến độ công việc.

## V. Vấn đề tồn động

Chưa thể triển khai một số ý tưởng đưa ra ban đầu như là thiết kế hệ thống quản lý của quản trị viên, một số lỗi nhỏ chưa kịp sửa kiến trải nghiệm người dùng không được như ý, cũng như chưa xử lý được lỗi "Managed Debugging Assistant 'CallbackOnCollectedDelegate'" xuất hiện khi có một hàm callback (hàm gọi lại) được chuyển giao cho một đối tượng delegate đã bị thu gom bởi bộ thu gom rác (garbage collector) và một số lỗi khác chưa xác định được.

## VI. Hướng phát triển

Đưa ra bản cập nhật để có thể sửa những vấn đề được đưa ra ở trên và đưa ra kế hoạch mở rộng tính năng, cải thiện giao diện người dùng, và tích hợp các dịch vụ tài chính mới để duy trì tính độc đáo và cạnh tranh.

# Mục lục

[DANH SÁCH PHÂN CÔNG NHIỆM VỤ NHÓM 2](#_Toc151514239)

[PHẦN MỞ ĐẦU 3](#_Toc151514240)

[I. Giới thiệu về chương trình quản lí ngân hàng: 3](#_Toc151514241)

[II. Các chức năng chính của chương trình. 3](#_Toc151514242)

[III. Các tài nguyên và phần mềm sử dụng trong dự án. 4](#_Toc151514243)

[1. C# và Visual Studio 2013 4](#_Toc151514244)

[2. Figmar 4](#_Toc151514245)

[3. Free Icons, Clipart Illustrations, Photos, and Music (icons8.com) 5](#_Toc151514246)

[4. Just Color Picker 5](#_Toc151514247)

[5. Sql Server 2014 6](#_Toc151514248)

[6. Github 7](#_Toc151514249)

[PHÂN TÍCH DỰ ÁN 8](#_Toc151514250)

[I. Thiết kế giao diện: 8](#_Toc151514251)

[II. Giới thiệu tổng quát: 12](#_Toc151514252)

[1. Các form trong code C#. 12](#_Toc151514253)

[1.1 Các Form chính và chi tiết: 12](#_Toc151514254)

[1.2 Form đăng nhập: 14](#_Toc151514255)

[1.3 Form ResetPassword 17](#_Toc151514256)

[1.4 Form NewPass 20](#_Toc151514257)

[1.5 FormNewAccount 21](#_Toc151514258)

[1.6 Form Main 23](#_Toc151514259)

[1.7 Form Home 26](#_Toc151514261)

[1.8 Form transferMoney 28](#_Toc151514262)

[1.9 Form Saving 31](#_Toc151514263)

[1.10 FormHistory 34](#_Toc151514265)

[1.11 FormCardManegement 37](#_Toc151514266)

[1.12 FormInformation 42](#_Toc151514267)

[2. Database 46](#_Toc151514268)

[1. Database diagrams 46](#_Toc151514269)

[2. Chi tiết các table 46](#_Toc151514270)

[3. Dữ liệu các bảng. 50](#_Toc151514271)

[4. Khóa ngoại(foreign key constraints) 53](#_Toc151514272)

[5. Các Proceduce và trigger 54](#_Toc151514273)

[TỔNG KẾT 60](#_Toc151514274)

[I. Mục tiêu dự án 60](#_Toc151514275)

[II. Chức năng chính 60](#_Toc151514276)

[III. Công nghệ sử dụng 60](#_Toc151514277)

[IV. Thách thức và giải quyết 60](#_Toc151514278)

[V. Vấn đề tồn động 60](#_Toc151514279)

[VI. Hướng phát triển 60](#_Toc151514280)