**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

---&---



**Sprint 1**

**Đề tài:**

**HỆ THỐNG CÂY RÚT TIỀN ATM**

**Nhóm sinh viên thực hiện: Nhóm 6 lớp 60TH4**

**Trần Thị Châm - 1851061747**

**Mai Phương Loan - 1851061752**

**Dương Thu Hoài - 1851061240**

**Đào Thùy Dương - 1851061470**

**Ngô Thị Huệ - 1851061355**

**Trần Thị Vy - 1851061884**

**Lê Thị Duyên - 1851061481**

**Giảng viên hướng dẫn: TS. Lê Nguyễn Tuấn Thành**

Hà Nội, ngày 10 tháng 5 năm 2021

**I. TÀI LIỆU ĐẶC TẢ YÊU CẦU**

**1. Giới thiệu hệ thống**

Hệ thống rút tiền tự động (ATM).

Khách hàng có thể tương tác với hệ thống qua điều khiển bằng màn hình và nhân viên điều hành mới có quyền tắt mở hệ thống. Máy rút tiền giao tiếp với máy tính của ngân hàng bằng một liên kết giao tiếp thích hợp.

Khi bắt đầu một giao dịch khách hàng sẽ phải đưa thẻ ATM vào khe thẻ và nhập vào mã PIN - Các thông tin này sẽ được gửi trực tiếp đến ngân hàng sau đó khách hàng có thể thực hiện một hoặc nhiều giao dịch. Thẻ sẽ được giữ lại trong máy cho đến khách hàng kết thúc giao dịch thẻ sẽ được trả lại cho khách hàng.

Mọi thông tin của giao dịch sẽ được gửi trực tiếp đến ngân hàng. Trong trường hợp khách hàng nhập vào sai mã pin hệ thống sẽ yêu cầu nhập lại mã pin trước khi có thể tiến hành giao dịch.

Nếu khách hàng nhập sai quá 3 lần thì sẽ bị giữ thẻ trong máy và khách hàng sẽ phải ra ngân hàng để lấy lại thẻ. Khi đó máy sẽ hiển thị một thông báo chi tiết về các lỗi của giao dịch và sẽ yêu cầu khách hàng dừng giao dịch hiện tại hoặc thực hiện một giao dịch khác.

Khi kết thúc một phiên giao dịch khách hàng sẽ nhận được một biên lai gồm các thông tin chi tiết của giao dịch đó: thời gian, địa điểm, loại giao dịch, số dư tài khoản,...

Chỉ có nhân viên ngân hàng mới có quyền khởi động hoặc tắt máy ATM để kiểm tra bảo dưỡng định kỳ. Máy chỉ được tắt khi không có giao dịch của khách hàng.

Nhân viên có thể kiểm tra lượng tiền còn lại trong máy nếu hết thì sẽ được nạp thêm. Sau khi quá trình kiểm tra hoàn tất nhân viên sẽ khởi động lại máy.

Toàn bộ phiên giao dịch sẽ được ghi chú lại để có thể thuận lợi trong việc khắc phục sự cố khi giao dịch xảy ra lỗi do phần cứng của máy. Ghi chú này được bắt đầu từ lúc máy được khởi động cho đến lúc máy bị tắt và sẽ được gửi đến ngân hàng. Mã số thẻ và lượng rút tiền sẽ được lưu lại nhưng vì lý do bảo mật nên MÃ PIN sẽ không được lưu.

**2. Yêu cầu chức năng và yêu cầu phi chức năng**

**2.1 Yêu cầu chức năng**

Những dịch vụ mà khách hàng sẽ được cung cấp:

* Khách hàng cần nhập chính xác mã PIN mới thực hiện được các yêu cầu với cây ATM.
* Khách hàng có thể rút tiền mặt từ bất kỳ tài khoản thích hợp liên kết đến thẻ.
* Có thể đổi mã PIN để bảo mật tài khoản của mình.
* Có thể chuyển tiền đến bất kỳ tài khoản nào đã có trong hệ thống.
* Các giao dịch cần phải có sự xác minh của ngân hàng.
* Có thể xem thông tin số dư tài khoản.
* Khách hàng có thể hủy giao dịch đang diễn ra bằng cách ấn Hủy bỏ thay vì đáp ứng yêu cầu từ máy.
* Khách hàng có thể đưa tiền mặt vào cây để gửi tiền vào tài khoản.
* Nhân viên có thể thêm tiền vào cây khi hết sau khi có sự đồng ý của Quản trị viên.

**2.2 Yêu cầu phi chức năng**

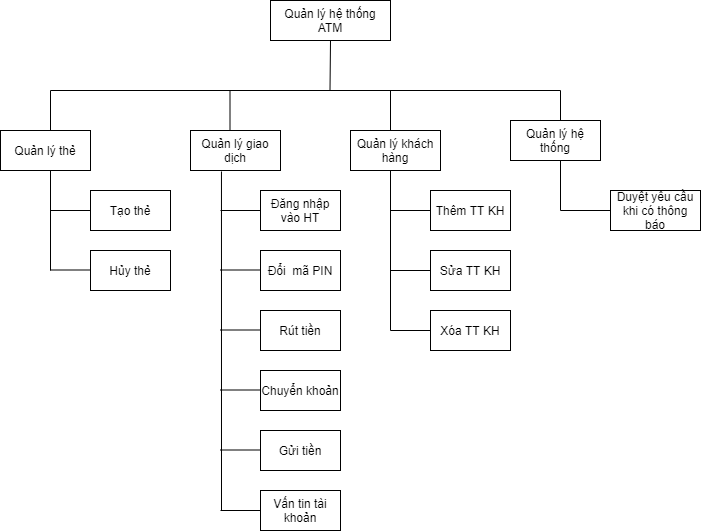
* Tính bảo mật cao: Hệ thống yêu cầu mật khẩu cho các tài khoản của người dùng. Chỉ có người quản trị hệ thống và người dùng mới nắm được tài khoản.
* Tính xác thực cao: Hệ thống hỗ trợ việc xác định tài khoản và mật khẩu đăng nhập. Mật khẩu phải có 6 chữ số.
* Khả năng bảo trì: Bảo trì hệ thống giúp vận hành tốt hơn trong quá trình hoạt động. Cho phép sửa lỗi nếu phát hiện phần mềm có sai sót. Nâng cấp thêm những chức năng mới theo nhu cầu của người dùng. Bảo trì dễ và nhanh (tối đa 30p) để đảm bảo kịp thời cho người dùng.
* Khả năng sao lưu: Có khả năng sao lưu và phục hồi dữ liệu nhanh (tối đa 10p) khi có sự cố hoặc theo định kỳ (3 tháng 1 lần)
* Khả năng mở rộng quy mô cây atm rộng rãi trên toàn quốc
* Cách thức thực hiện đảm bảo theo đúng quy định của bên đối tác về việc xử lý các giao dịch thông qua ATM.
* Tốc độ xử lý nhanh chóng (tối đa 2s) và hiệu quả (đáp ứng đầy đủ các yêu cầu)
* Thời gian hoạt động 24/24.
* Hệ thống cài đặt được trên các cây ATM
* Giao diện dùng 1 font chữ (Microsoft Sans Serif).
* Sử dụng với nhiều ngân hàng và vô hạn khách hàng.

**3. Phương pháp thu thập yêu cầu**

|  |  |
| --- | --- |
| Các bên liên quan đến hệ thống cây ATM | Phương pháp thu thập yêu cầu |
| Khách hàng của ngân hàng | Phỏng vấn, khảo sát hiện trạng, thực hiện khảo sát, bản câu hỏi thăm dò. |
| Quản lý ngân hàng | Tổ chức cuộc họp, tạo kịch bản, khảo sát hiện trạng. |
| Kiểm toán viên | Tổ chức cuộc họp, khảo sát hiện trạng. |
| Nhân viên giao dịch | Tổ chức cuộc họp, khảo sát hiện trạng. |
| Nhân viên IT | Tổ chức cuộc họp, khảo sát hiện trạng |
| Quản trị viên cơ sở dữ liệu | Tổ chức cuộc họp, khảo sát hiện trạng |
| Nhân viên an ninh | Khảo sát hiện trạng |
| Bộ phận quảng cáo | Tổ chức cuộc họp, khảo sát hiện trạng, thực hiện khảo sát . |

**4. Đặc tả yêu cầu hệ thống**

**4.1. Mô hình phân cấp chức năng (FDD)**

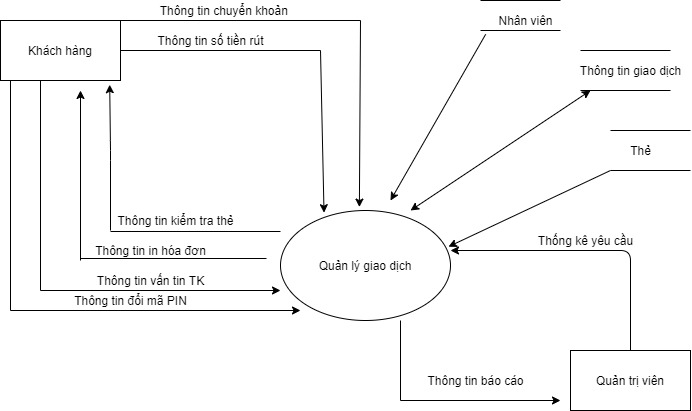


**4.2. Biểu đồ luồng dữ liệu (DFD)**

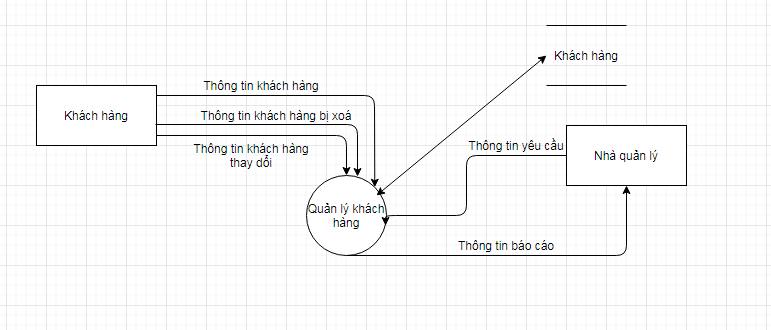
*Chức năng Quản lý thẻ*



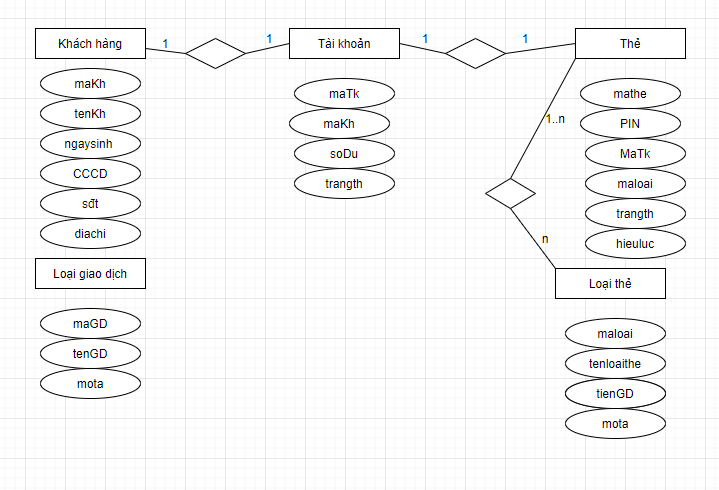
*Chức năng Quản lý giao dịch*



*Chức năng Quản lý khách hàng*



**4.3. Mô hình thực thể liên kết (ERD)**



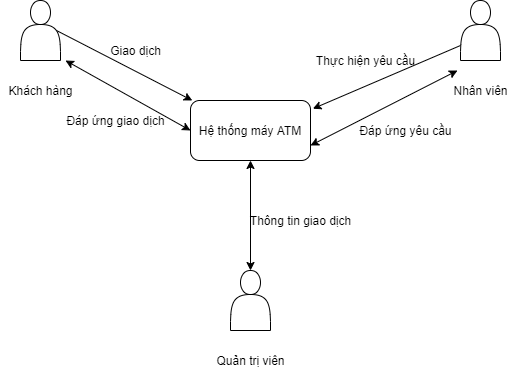
**5. Xác định tác nhân và Use Case**

Các tác nhân và các Use Case

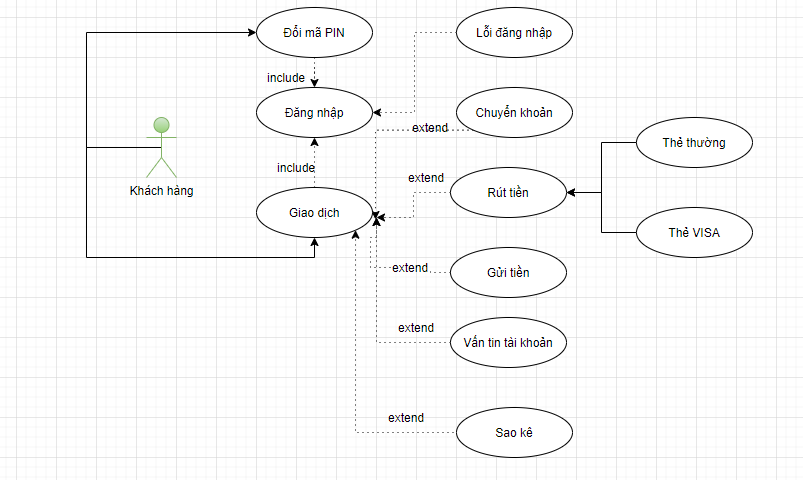
|  |  |
| --- | --- |
| Tác nhân | Use case |
| Khách hàng | * Đăng nhập * Đổi mã PIN * Sao kê tài khoản * Rút tiền * Chuyển khoản * Vấn tin tài khoản * Gửi tiền |

**6. Các biểu đồ phân tích:**

**6.1. Biểu đồ Use Case mức khung nhìn**

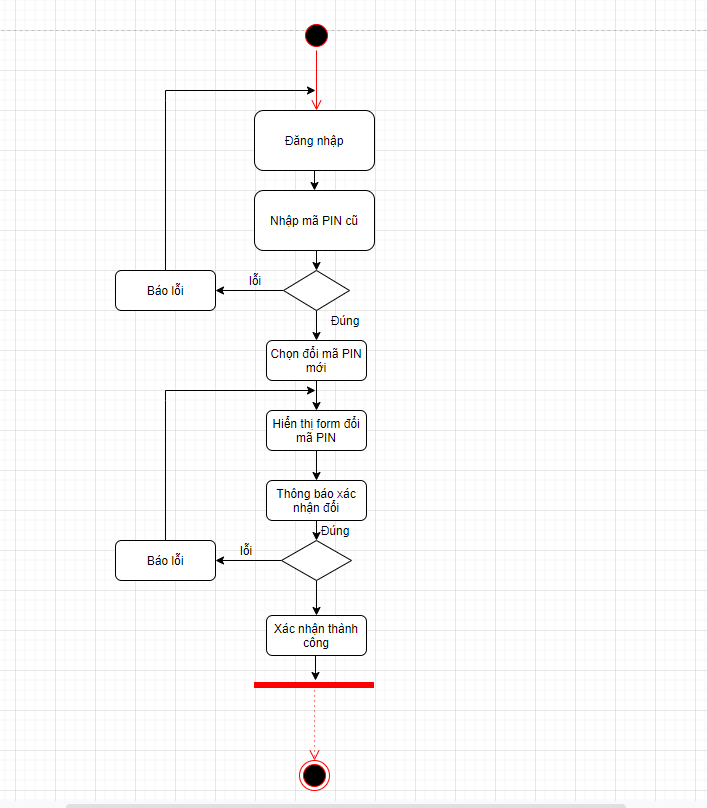
****

**6.2. Biểu đồ Use Case chính**

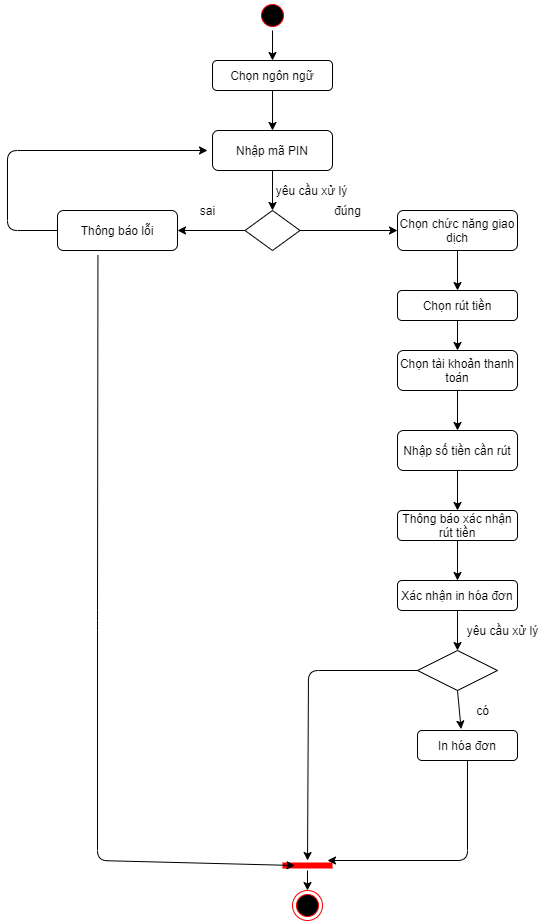
****

**6.3. Biểu đồ hoạt động cho các luồng sử dụng**

**6.3.1. Biểu đồ hoạt động Đổi mã PIN**

****

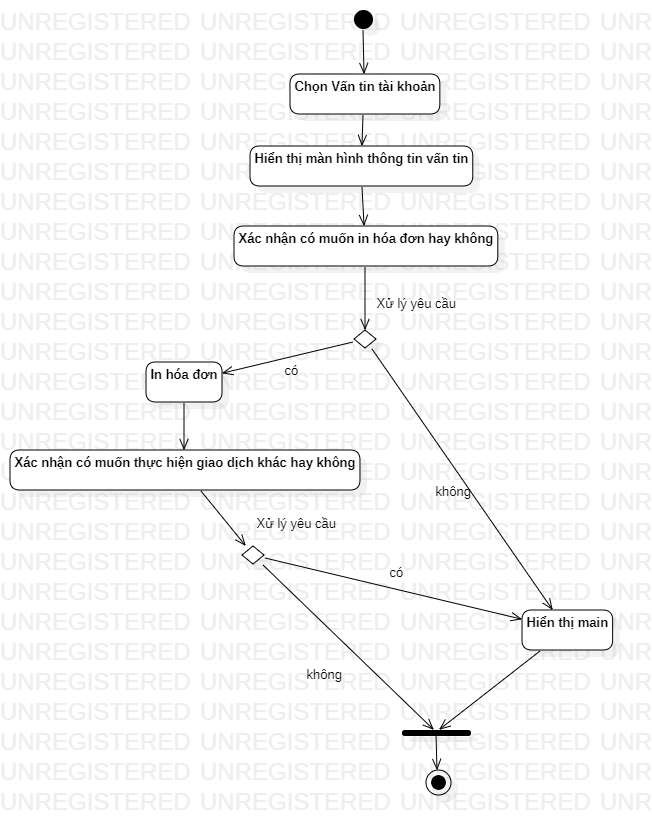
**6.3.2. Biểu đồ hoạt động Rút tiền**

****

**6.3.3. Biểu đồ hoạt động Chuyển khoản**

****

**6.3.4. Biểu đồ hoạt động Vấn tin tài khoản**

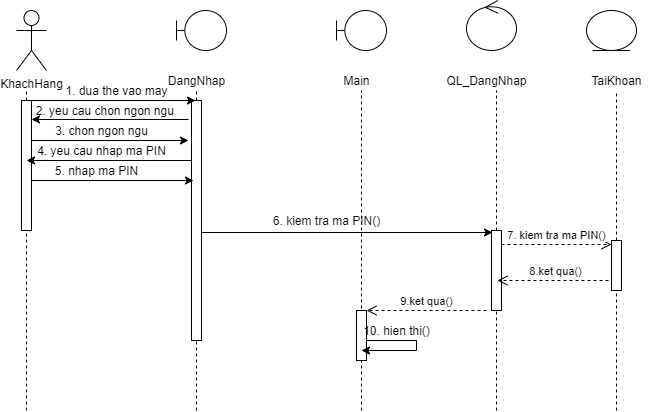
****

**6.3.5. Biểu đồ hoạt động Gửi tiền**

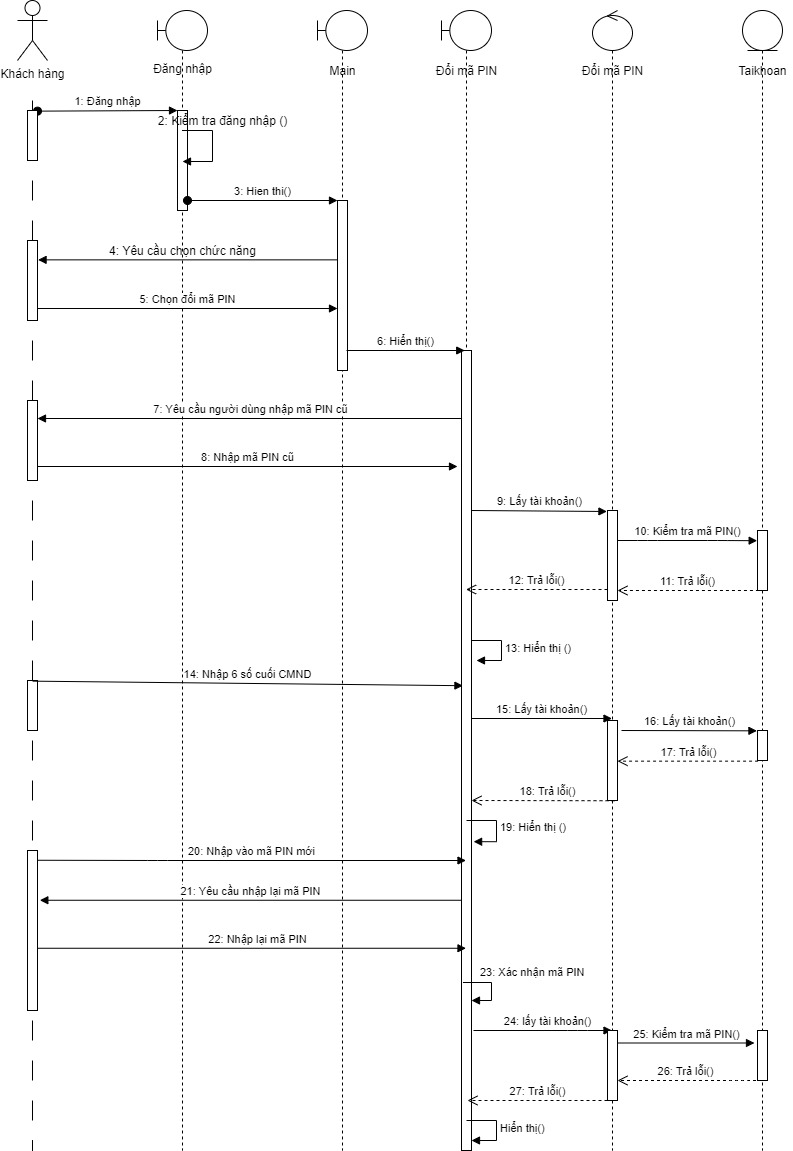
****

**6.4. Biểu đồ tuần tự**

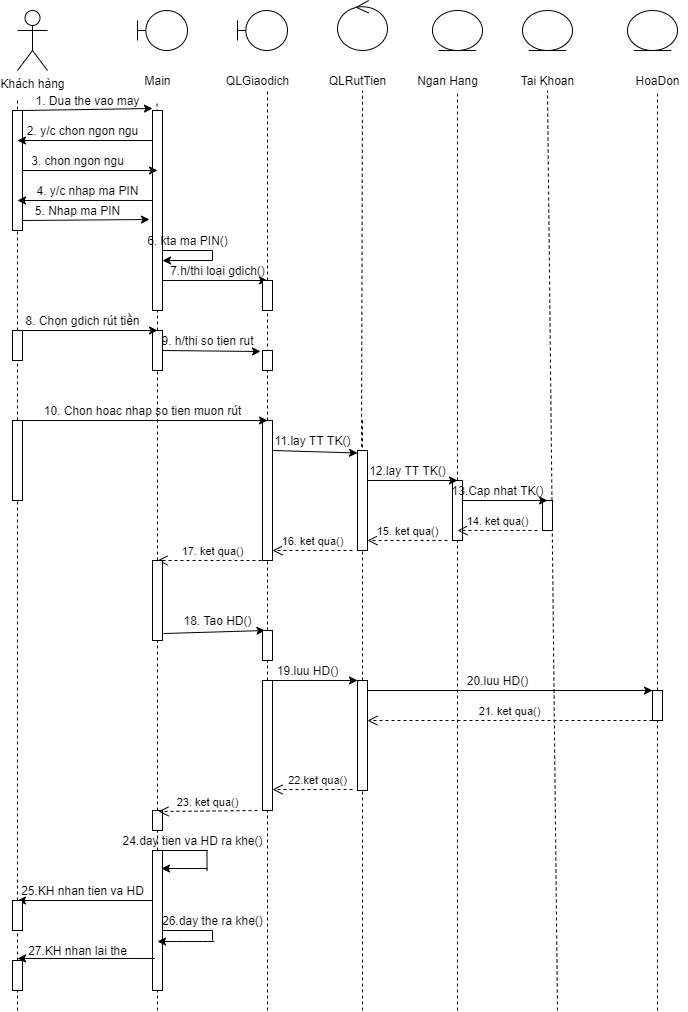
**6.4.1. Use case đăng nhập vào hệ thống**

****

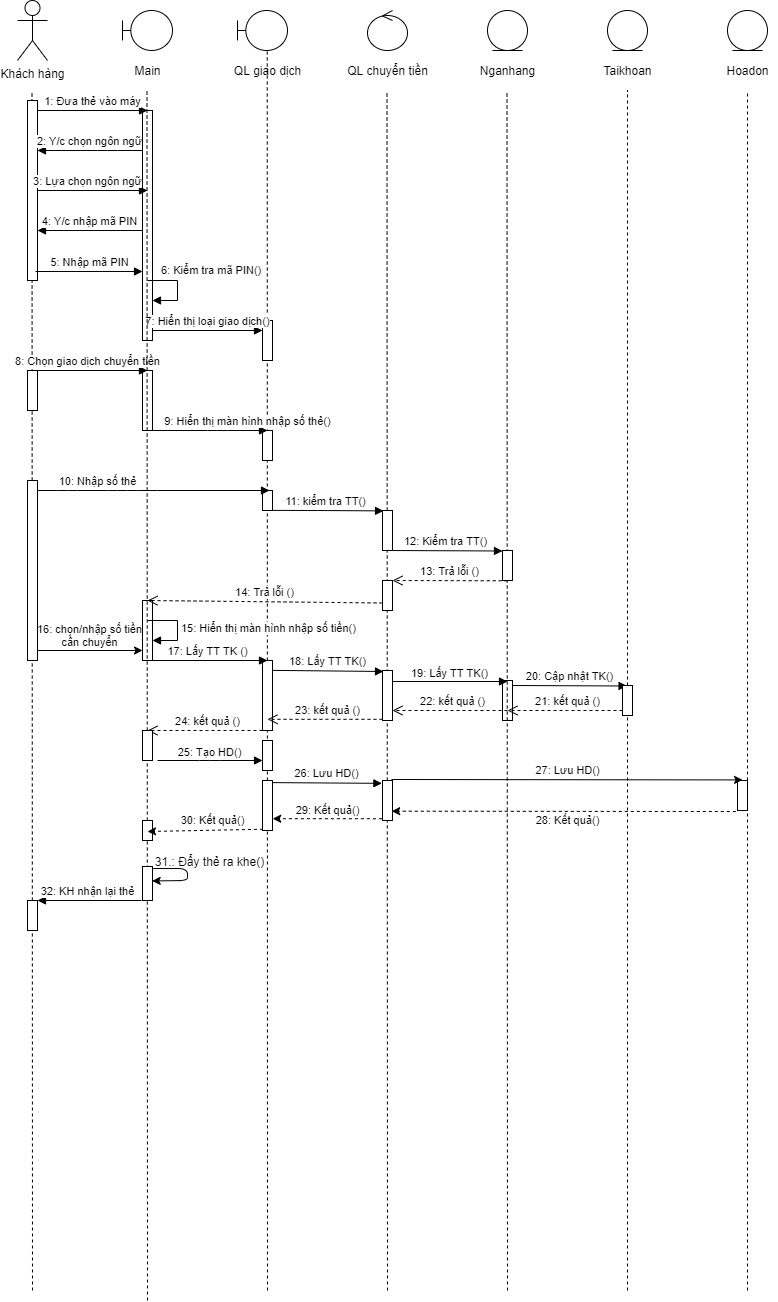
**6.4.2. Use case đổi mã PIN**

****

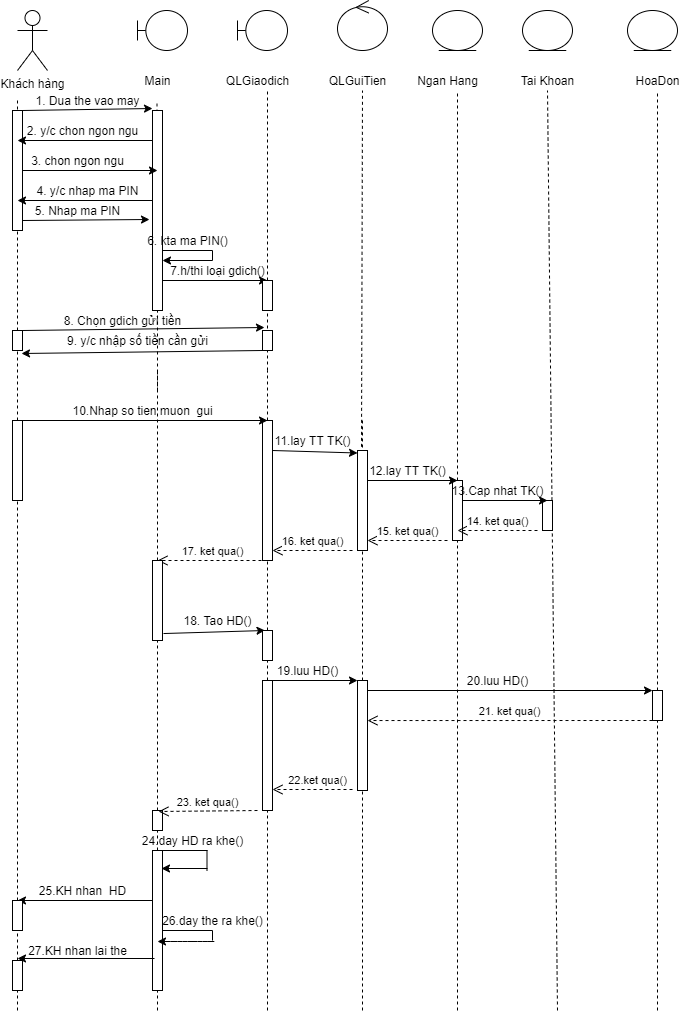
**6.4.3. Use case rút tiền**

****

**6.4.4. Use case chuyển khoản**

****

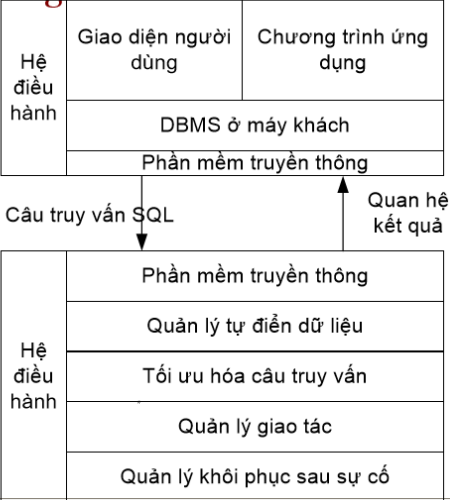
**6.4.5 Use case Gửi tiền**

****

**II. PHÂN TÍCH & THIẾT KẾ**

**1. Thiết kế kiến trúc(các loại kiến trúc và server thông dụng)**

**1.1. Kiến trúc 3 lớp**

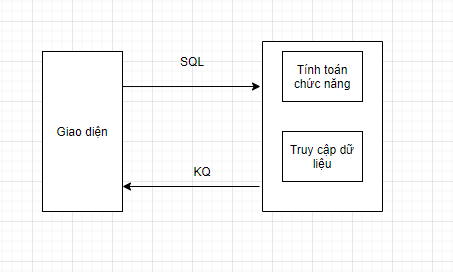


**1.2. Kiến trúc 2 lớp**

* Thin Client

Client : là giao diện người dùng

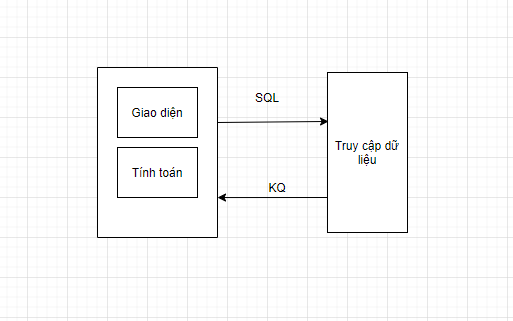
Server : bao gồm quá trình xử lý nghiệp vụ và truy cập dữ liệu



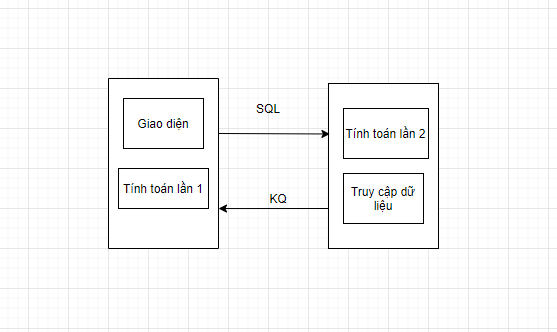
* Fat client

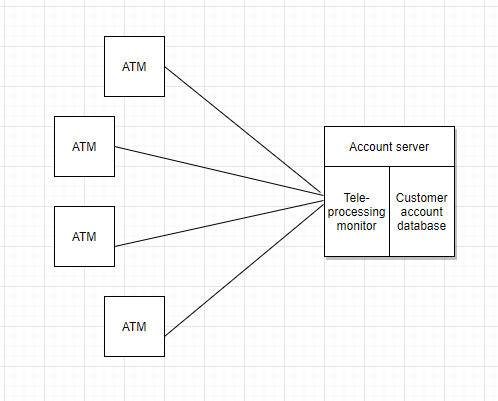
Client : là giao diện người dùng và xử lý nghiệp vụ.

Server : Chỉ truy cập dữ liệu.



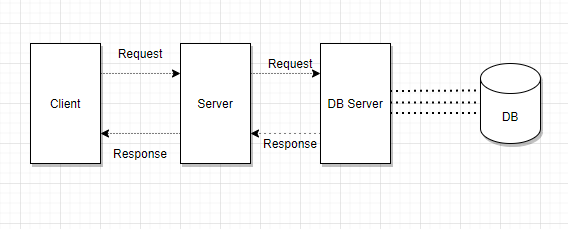
* Kết hợp

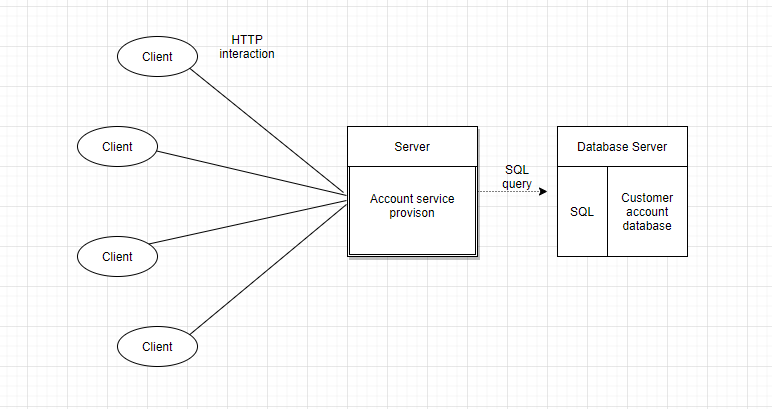




**1.3 Kiến trúc 3 lớp**

* Client: Gửi yêu cầu, dữ liệu và nhận kết quả từ lớp ứng dụng.
* Server ứng dụng: thực hiện các xử lý dữ liệu được gửi từ các client.
* Server cơ sở dữ liệu: lưu trữ dữ liệu cho lớp ứng dụng, cung cấp các dịch vụ cho các máy tính và chương trình trong lớp ứng dụng. Thường sử dụng các hệ quản trị cơ sở dữ liệu để thực hiện các dịch vụ.





**1.4. Thiết kế các chức năng trên tầng kiến trúc**

* Phân chia công việc giữa client server để tối ưu hóa xử lý.
* Tận dụng server, phân chia công việc giữa các thành phần.

Ví dụ:

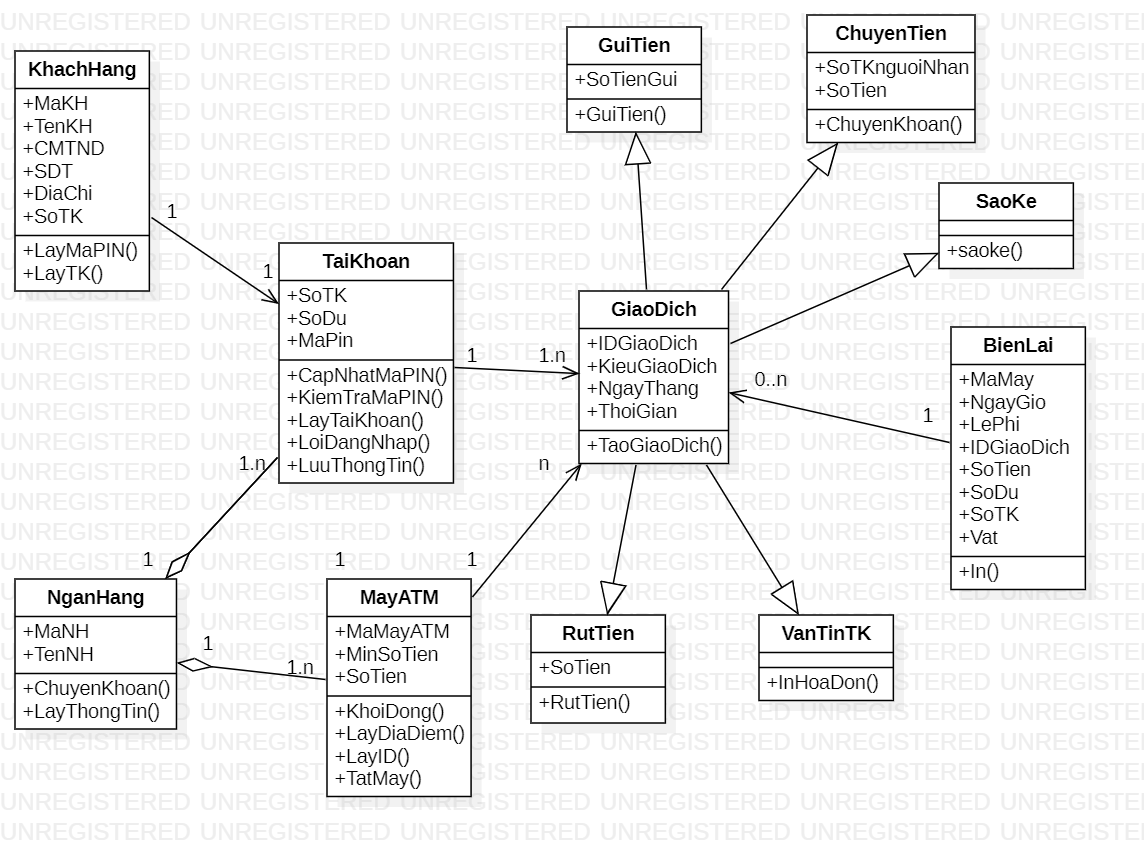
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Xử lý** | **Mô tả** | **Nơi thực hiện** |
| 1 | Hiển thị màn hình nhập mã PIN | Màn hình MH001 | Client |
| 2 | Kiểm tra đầy đủ thông tin | Kiểm tra người dùng có nhập đầy đủ mã PIN không | Client |
| 3 | Kiểm tra mã PIN | Kiểm tra mã PIN của người dùng có tồn tại trong CSDL hay không | Server |
| 4 | Hiển thị màn hình chính | Hiển thị màn hình sau khi đăng nhập thành công | Client |
| 5 | Hiển thị màn hình đăng nhập lại và thông báo lỗi | Màn hình 001 | Client |
| 6 | Hiển thị màn hình lựa chọn giao dịch | Màn hình 002 | Client |
| 7 | Hiển thị màn hình rút tiền | Màn hình 003 | Client |
| 8 | Nhập số tiền cần rút | Kiểm tra người dùng có nhập số tiền cần rút nhỏ hơn số tiền trong tài khoản và lớn hơn 0 không | Client |
| 9 | Kiểm tra số tiền rút | Kiểm tra số tiền người dùng cần rút có đúng điều kiện trong CSDL không | Server |
| 10 | Hiển thị màn hình hủy giao dịch | Màn hình 002 | Client |
| 11 | Kiểm tra giao dịch | Sau khi nhấn giao dịch rút tiền hệ thống sẽ kiểm tra số dư | Server |
| 12 | Hiển thị màn hình giao dịch thành công và hỏi có muốn giao dịch tiếp không | Màn hình 004 | Client |
| 13 | Muốn giao dịch tiếp | Màn hình 002 | Client |
| 14 | Hiển thị màn hình nhập lại số tiền và thông báo lỗi | Màn hình 005 | Client |

**1.5. Biểu đồ thành phần**

****

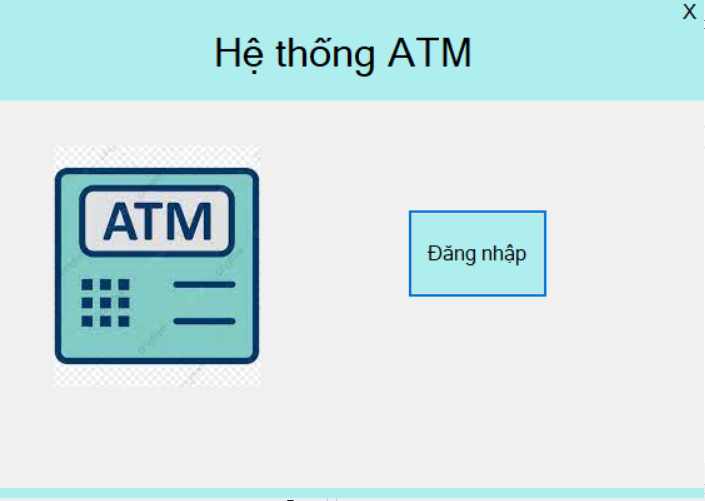
**2. Thiết kế hướng đối tượng**

*Biểu đồ lớp*

****

**3. Thiết kế giao diện**

*a, Giao diện Form1*

****

|  |  |
| --- | --- |
| Điều khiển | Text |
| Panel | Chứa lab hệ thống ATM |
| Label | Hệ thống ATM |
| Button | Đăng nhập |
| Picturebox | ATM |

*b, Giao diện flogin*

# 

|  |  |
| --- | --- |
| Điều khiển | Text |
| Label | Hệ thống ATM |
| Số tài khoản |
| X |
| Mã PIN |
| Textbox |  |
| Button | Đăng nhập |
| Panel |  |
| Picturebox |  |

*c, Giao diện fhome*



|  |  |
| --- | --- |
| Điều khiển | Text |
| Label | Hệ thống ATM |
| Lựa chọn giao dịch |
| Số tài khoản |
| X |
| Button | Gửi tiền |
| Đổi mã PIN |
| Chuyển tiền |
| Rút tiền |
| Sao kê |
| Truy vấn |
| Thoát |
| Panel |  |

*d, Giao diện fguitien*



|  |  |
| --- | --- |
| Điều khiển | Text |
| Text Box |  |
| Button | Hủy |
| Gửi |
| Label | Gửi tiền |
| Số tiền |
| X |
| Hệ thống ATM |
| Panel |  |

*e, Giao diện fchuyentien*



|  |  |
| --- | --- |
| Điều khiển | Text |
| Text Box |  |
| Số dư |
| Button | Kiểm tra |
| Hủy |
| Label | Chuyển tiền |
| Nhập số tài khoản |
| X |
| Hệ thống ATM |
| Panel |  |

*f, Giao diện fchuyen*



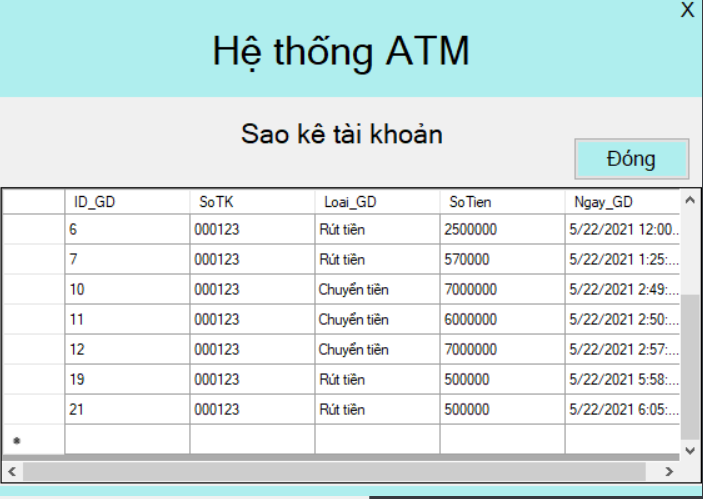
|  |  |
| --- | --- |
| Điều khiển | Text |
| Label | Hệ thống ATM |
| Chuyển tiền |
| Nhập số tiền chuyển |
| Số dư |
| TextBox | Nhập tiền |
| Button | Hủy |
| Chuyển |

*g, Giao diện frut*



|  |  |
| --- | --- |
| Điều khiển | Text |
| Label | Hệ thống ATM |
| Rút tiền |
| Số dư |
| Button | 200 000 VNĐ |
| 2 000 000 VNĐ |
| 500 000 VNĐ |
| 1 000 000 VNĐ |
| 3 000 000 VNĐ |
| Số khác |
| Hủy |

*h, Giao diện fsaoke*



|  |  |
| --- | --- |
| Điều khiển | Text |
| Label | Hệ thống ATM |
| Sao kê tài khoản |
| Button | Đóng |
| DataGridView | Bảng sao kê |

i, Giao diện fbalance



|  |  |
| --- | --- |
| Điều khiển | Text |
| Label | Hệ thống ATM |
| Kiểm tra số dư |
| Số tài khoản |
| Số dư |
| Button | Đóng |

# 

# **Phần III. TÀI LIỆU KIỂM THỬ**

# **1. Kiểm thử hộp đen**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hình thức** | **Mã kiểm thử** | **Miêu tả** | **Tiến trình kiểm thử** | | **Kết quả các tiến trình kiểm thử** | **Trạng thái** |
| **Các bước thực hiện** | **Kết quả từng bước** |
| Rút Tiền | 01 | 0<Số tiền rút < số dư  số tiền rút<số tiền ATM | Nhập 1 số tiền lớn hơn 0 và nhỏ hơn số dư | True | Yêu cầu được thực hiện. Trả số tiền khách hàng cần rút. | Pass |
| So sánh số tiền rút với số dư | True |
| So sánh số tiền rút với số tiền trong ATM | True |
| 02 | 0<Số tiền rút < số dư  số tiền rút<số tiền ATM | Nhập 1 số tiền lớn hơn 0 và nhỏ hơn số dư | True | Thông báo lỗi | Failed |
| So sánh số tiền rút với số dư | True |
| So sánh số tiền rút với số tiền trong ATM | False |
| 03 | Số tiền>0, >số dư | Nhập 1 số tiền >0, >số dư | True | Thông báo lỗi | Failed |
| So sánh với số dư | False |
| 04 | Số tiền = 0 | Nhập 1 số tiền <0 | False | Thông báo lỗi | Failed |
| Chuyển tiền | 05 | Nhập đúng mã thẻ, 0 < Số tiền < số dư | Nhập mã thẻ cần chuyển | True | Thông báo thành công | Pass |
| Nhập số tiền | True |
| Kiểm tra số dư | True |
| 06 | Nhập đúng mã thẻ, số tiền > 0, > số dư | Nhập mã thẻ cần chuyển | True | Thông báo lỗi | Failed |
| Nhập số tiền | True |
| Kiểm tra số dư | False |
| 07 | Nhập đúng mã thẻ, số tiền = 0 | Nhập mã thẻ cần chuyển | True | Thông báo lỗi | Failed |
| Nhập số tiền | True |
| 08 | Nhập sai mã thẻ | Nhập mã thẻ | False | Thông báo lỗi | Failed |
| Đổi mật khẩu | 09 | Nhập đúng mật khẩu cũ, mới | Nhập mật khẩu cũ và mới | True | Thông báo thành công | Pass |
| 10 | Nhập đúng mật khẩu cũ, sai mật khẩu mới | Nhập mật khẩu cũ và mới | False | Thông báo lỗi | Failed |
| 11 | Nhập sai mật khẩu cũ | Nhập mật khẩu cũ | False | Thông báo lỗi | Failed |

**2. Kiểm thử hộp trắng**

Rút tiền

(1) private void button1\_Click(object sender, EventArgs e)

{

(2) **if** (txbsotien.Text == "")

{

(3) MessageBox.Show("Mời bạn nhập số tiền");

}

(4) else if (Convert.ToInt32(txbsotien.Text) <= 0)

{

(5) MessageBox.Show("Số tiền nhập vào không đúng!");

}

(6) else if (Convert.ToInt32(txbsotien.Text) % 10000 != 0)

{

(7) MessageBox.Show("Số tiền rút phải chia hết cho 10000");

}

(8) else if(Convert.ToInt32(txbsotien.Text) > sodu)

{

(9) MessageBox.Show("Số tiền trong tài khoản của bạn không đủ!");

}

(10) **else**

{

(11) try

{

(121) newsodu = sodu - Convert.ToInt32(txbsotien.Text);

(121) conn.Open();

(121) string query = "update Taikhoan set sodu ='" + newsodu + "' where SoTK ='" + acc + "'";

(121) SqlCommand cmd = new SqlCommand(query, conn);

cmd.ExecuteNonQuery();

(121) DialogResult result = MessageBox.Show("Bạn có muốn in hóa đơn không? ", "", MessageBoxButtons.YesNo);

switch (result)

{

(12) case DialogResult.No:

(13) DialogResult rs = MessageBox.Show("Bạn có muốn thực hiện giao dịch khác không?", "", MessageBoxButtons.YesNo);

switch (rs)

{

(14) case DialogResult.No:

(15) Form1 form1 = new Form1();

(15) form1.Show();

(15) this.Hide();

break;

(16) case DialogResult.Yes:

(17) fhome f = new fhome();

(17) f.Show();

(17) this.Hide();

break;

}

break;

(18) case DialogResult.Yes:

(19) DialogResult r = MessageBox.Show("Mời bạn nhận hóa đơn. Bạn có muốn thực hiện giao dịch khác không?", "", MessageBoxButtons.YesNo);

switch (r)

{

(20) case DialogResult.No:

(21) Form1 form1 = new Form1();

(21) form1.Show();

(21) this.Hide();

break;

(22) case DialogResult.Yes:

(23) fhome f = new fhome();

(23) f.Show();

(23) this.Hide();

break;

}

break;

}

conn.Close();

}

(24) catch (Exception Ex)

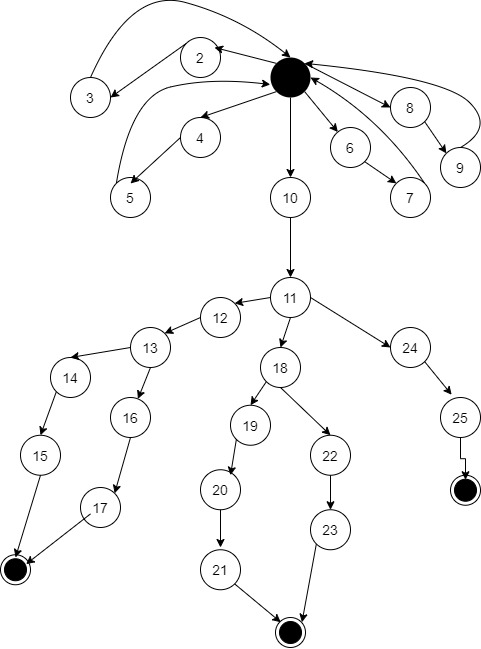
{

(25) MessageBox.Show(Ex.Message);

}

}

}



Số lập trình độc lập (độ phức tạp lặp) = 9

Có 9 đường thi hành tuyến tính độc lập cơ bản là :

* 1. 2→3
* 2. 4→5
* 3. 6→7
* 4. 8→9
* 5. 10→11→12→13→14→15
* 6. 10→11→12→13→16→17
* 7. 10→11→18→19→20→21
* 8. 10→11→18→22→23
* 9. 10→11→24→25

Thiết kế test case:

* Test case cho đường 1:

+)Kết quả kỳ vọng: In ra thông điệp:*“Mời bạn nhập số tiền”*

* Test case cho đường 2:

sotien<=0

+)Kết quả kỳ vọng: In ra thông điệp

“Số tiền nhập vào không đúng”

* Test case cho đường 3:

txbsotien.Text % 10000 != 0+)Kết quả kỳ vọng: in ra thông điệp:

“Số tiền rút phải chia hết cho 10000”

* Test case cho đường 4:

txbsotien.Text > sodu

+)Kết quả kỳ vọng: in ra thông điệp:

“Số tiền trong tài khoản của bạn không đủ!”

* Test case cho đường 5:

1. newsodu = sodu - Convert.ToInt32(txbsotien.Text);

+)Kết quả kỳ vọng:

"update Taikhoan set sodu ='" + newsodu + "' where SoTK ='" + acc + "'";

2. DialogResult result = MessageBox.Show("Bạn có muốn in hóa đơn không? ", "", MessageBoxButtons.YesNo);

+) DialogResult.No:

Kết quả kỳ vọng: form1.Show();

+)DialogResult.Yes:

Kết quả kỳ vọng: fhome.Show();

* Test case cho đường 9:

newsodu = sodu - Convert.ToInt32(txbsotien.Text);

+)Kết quả kỳ vọng:

"update Taikhoan set sodu ='" + newsodu + "' where SoTK ='" + acc + "'";

+)Kết quả kỳ vọng: in ra thông điệp:

“Ex.Message” // thông tin chi tiết về lỗi đang gặp phải

Chuyển tiền

private void btnchuyen\_Click(object sender, EventArgs e)

{

(2) if (txbtien.Text == "")

{

(3) MessageBox.Show("Mời bạn nhập số tiền");

}

(4) else if (Convert.ToInt32(txbtien.Text) <= 0)

{

(5) MessageBox.Show("Số tiền nhập vào không đúng!");

}

(6) else if (Convert.ToInt32(txbtien.Text) > sodu)

{

(7) MessageBox.Show("Số tiền trong tài khoản của bạn không đủ!");

}

(8) else

{

(9) try

{

(10) newsodu = sodu - Convert.ToInt32(txbtien.Text);

(10) conn.Open();

(10) string query = "update Taikhoan set sodu ='" + newsodu + "' where SoTK ='" + acc + "'";

(10) SqlCommand cmd = new SqlCommand(query, conn);

(10) cmd.ExecuteNonQuery();

(11) string qr = "update Taikhoan set sodu = sodu + '" + Convert.ToInt32(txbtien.Text) + "' where SoTK ='" + stk + "'";

(11) SqlCommand cd = new SqlCommand(qr, conn);

(11) cd.ExecuteNonQuery();

(11) DialogResult result = MessageBox.Show("Bạn có muốn in hóa đơn không? ", "", MessageBoxButtons.YesNo);

switch (result)

{

(12) case DialogResult.No:

(13) DialogResult rs = MessageBox.Show("Bạn có muốn thực hiện giao dịch khác không?", "", MessageBoxButtons.YesNo);

switch (rs)

{

(14) case DialogResult.No:

(15) Form1 form1 = new Form1();

(15) form1.Show();

(15) this.Hide();

break;

(16) case DialogResult.Yes:

(17) fhome f = new fhome();

(17) f.Show();

(17) this.Hide();

break;

}

break;

(18)case DialogResult.Yes:

(19) DialogResult r = MessageBox.Show("Mời bạn nhận hóa đơn. Bạn có muốn thực hiện giao dịch khác không?", "", MessageBoxButtons.YesNo);

switch (r)

{

(20) case DialogResult.No:

(21) Form1 fo = new Form1();

(21) fo.Show();

(21) this.Hide();

break;

(22)case DialogResult.Yes:

(23) fhome fh = new fhome();

(23) fh.Show();

(23) this.Hide();

break;

}

break;

}

conn.Close();

}

(24) catch (Exception Ex)

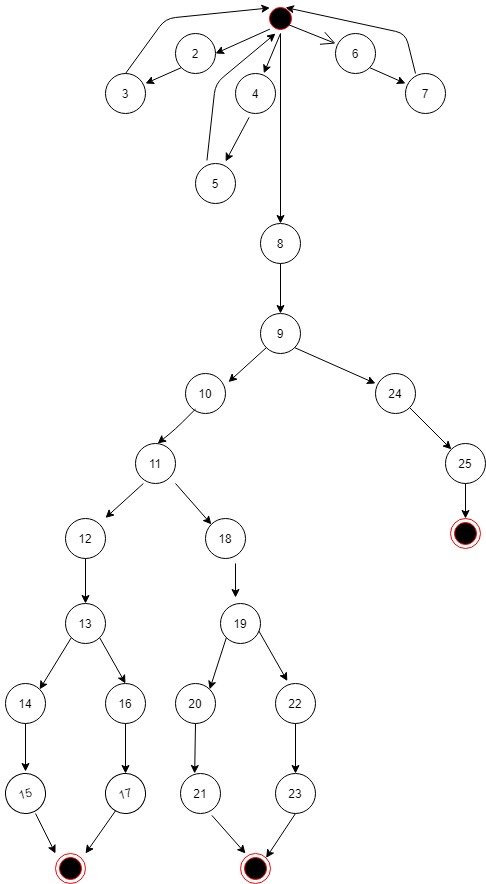
{

(25) MessageBox.Show(Ex.Message);

}

}

}

****

Số lập trình độc lập (độ phức tạp lặp) = 8

Có 8 đường thi hành tuyến tính độc lập cơ bản là :

* 1. 2→3
* 2. 4→5
* 3. 6→7
* 4. 8→ 9 → 10 → 11 → 12→ 13 → 14 → 15
* 5. 8→9→10→11→12→13→ 16 →17
* 6. 8→ 9→10→11→18→ 19 → 20 → 21
* 7. 8→ 9→10→11→18→ 19 → 22 → 23
* 8. 8 → 9 → 24 → 25

Thiết kế test case:

* Test case cho đường 1:

+)Kết quả kỳ vọng: In ra thông điệp:*“Mời bạn nhập số tiền”*

* Test case cho đường 2:

sotien<=0

+)Kết quả kỳ vọng: In ra thông điệp

“Số tiền nhập vào không đúng”

* Test case cho đường 3 :

txbsotien.Text > sodu

+)Kết quả kỳ vọng: in ra thông điệp:

“Số tiền trong tài khoản của bạn không đủ!”

* Test case cho đường 4:

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi chuyển

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘không.’

+)Hiện ra dòng thông báo : “Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘không’ thì quay lại form1 - form login.

* Test case cho đường 5:

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi chuyển

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘không.’

+)Hiện ra dòng thông báo : “Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘có’ thì quay lại fhome - form giao dịch.

* Test case cho đường 6:

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi chuyển

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘có’

+)Hiện ra dòng thông báo : “Mời bạn nhận hóa đơn. Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘không’ thì quay lại form1 - form login.

* Test case cho đường 7:

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi chuyển

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘có.’

+)Hiện ra dòng thông báo : “Mời bạn nhận hóa đơn. Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘có’ thì quay lại fhome - form giao dịch.

* Test case cho đường 8:

newsodu = sodu - Convert.ToInt32(txbsotien.Text);

+)Kết quả kỳ vọng:

"update Taikhoan set sodu ='" + newsodu + "' where SoTK ='" + acc + "'";

+)Kết quả kỳ vọng: in ra thông điệp:

“Ex.Message” // thông tin chi tiết về lỗi đang gặp phải

Gửi tiền

private void button1\_Click(object sender, EventArgs e)

{

(2) if(txbTien.Text == "" || Convert.ToInt32(txbTien.Text)<0) (3)

{

(4) MessageBox.Show("Nhập số tiền gửi");

}

(5) else

{

(6) newsodu = oldsodu + Convert.ToInt32(txbTien.Text);

(7) try

{

(8) conn.Open();

(8) string query = "update Taikhoan set sodu ='" + newsodu + "' where SoTK ='" + acc + "'";

(8) SqlCommand cmd = new SqlCommand(query,conn);

(8) cmd.ExecuteNonQuery();

(8) DialogResult result = MessageBox.Show("Bạn có muốn in hóa đơn không? ", "", MessageBoxButtons.YesNo);

switch (result)

{

(9) case DialogResult.No:

(10) DialogResult rs = MessageBox.Show("Bạn có muốn thực hiện giao dịch khác không?", "", MessageBoxButtons.YesNo);

switch (rs)

{

(11) case DialogResult.No:

(12) Form1 form1 = new Form1();

(12) form1.Show();

(12) this.Hide();

break;

(13) case DialogResult.Yes:

(14) fhome f = new fhome();

(14) f.Show();

(14) this.Hide();

break;

}

break;

(15) case DialogResult.Yes:

(16) DialogResult r = MessageBox.Show("Mời bạn nhận hóa đơn. Bạn có muốn thực hiện giao dịch khác không?", "", MessageBoxButtons.YesNo);

switch (r)

{

(17) case DialogResult.No:

(18) Form1 form1 = new Form1();

(18) form1.Show();

(18) this.Hide();

break;

(19) case DialogResult.Yes:

(20) fhome f = new fhome();

(20) f.Show();

(20) this.Hide();

break;

}

break;

}

conn.Close();

addgiaodich();

(21)}catch(Exception Ex)

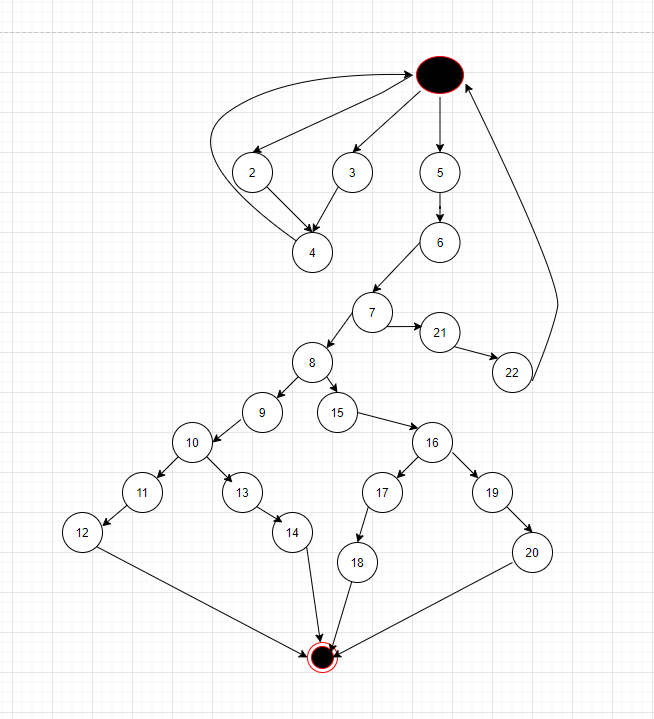
{

(22) MessageBox.Show(Ex.Message);

}

}

}



Số lập trình độc lập (độ phức tạp lặp) = 7

Có 6 đường thi hành tuyến tính độc lập cơ bản là :

* 1. 2→4
* 2. 3→ 4
* 3. 5→6→7→8→9→10→11→ 12
* 4. 5→6→7→8→9→10→13→ 14
* 5. 5→6→7→8→15→17→18
* 6. 5→6→7→8→15→19→20
* 7. 5→6→7→21→22

Thiết kế test case:

* Test case cho đường 1,2:

+)Kết quả kỳ vọng: In ra thông điệp:*“Nhập số tiền gửi”*

* Test case cho đường 3:

Đã nhập vào số tiền gửi.

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi chuyển

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘không.’

+)Hiện ra dòng thông báo : “Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘không’ thì quay lại form1 - form login.

* Test case cho đường 4:

Đã nhập vào số tiền gửi.

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi gửi.

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘không.’

+)Hiện ra dòng thông báo : “Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘có’ thì quay lại fhome - form giao dịch sau khi đăng nhập.

* Test case cho đường 5:

Đã nhập vào số tiền gửi.

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi chuyển

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘có.’

+)Hiện ra dòng thông báo : “Mời bạn nhận hóa đơn. Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘không’ thì quay lại form1 - form login.

* Test case cho đường 6:

Đã nhập vào số tiền gửi.

+)Kết quả kỳ vọng: trừ tiền trong tài khoản sau khi chuyển.

+) Hiện ra dòng thông báo : “Bạn có muốn in hóa đơn không?”, nếu chọn ‘có.’

+)Hiện ra dòng thông báo : “Mời bạn nhận hóa đơn. Bạn có muốn thực hiện giao dịch khác không?”, nếu chọn ‘có’ thì quay lại fhome - form giao dịch sau khi login..

* Test case cho đường 7:

+)Kết quả kỳ vọng: bắt lỗi khi nhập sai số tiền thì in ra thông báo lỗi.