**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

---&---



**Sprint 2**

**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 22 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 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**

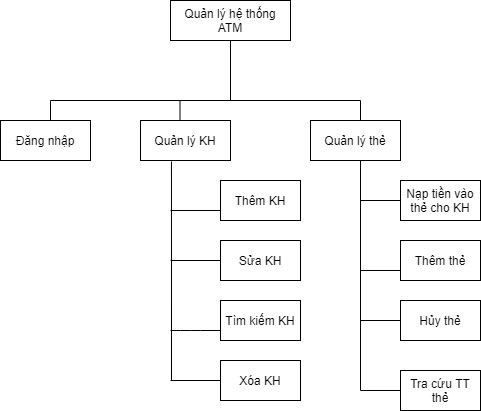
* Tính bảo mật cao: Hệ thống yêu cầu mật khẩu cho các tài khoản của người dùng. Chỉ có người quản trị hệ thống và người dùng mới nắm được tài khoản.
* Tính xác thực cao: Hệ thống hỗ trợ việc xác định tài khoản và mật khẩu đăng nhập. Mật khẩu phải có 6 chữ số.
* Khả năng bảo trì: Bảo trì hệ thống giúp 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**

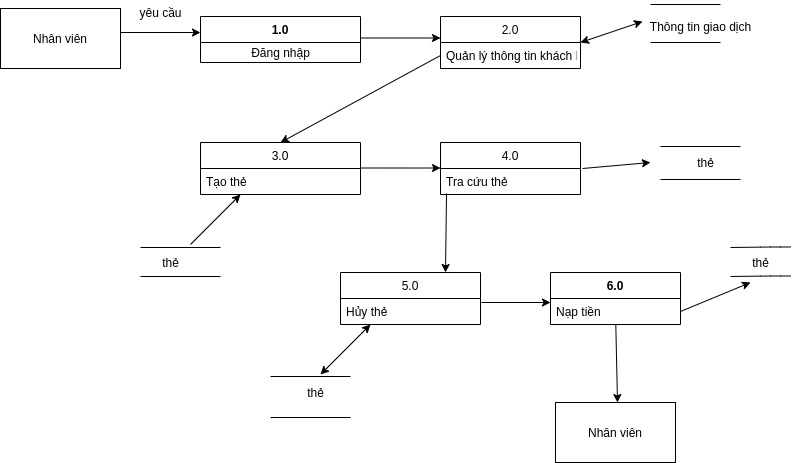
| **Các bên liên quan đến hệ thống cây ATM** | **Phương pháp thu thập yêu cầu** |
| --- | --- |
| Khách hàng của ngân hàng | Phỏng vấn, khảo sát hiện trạng, thực hiện khảo sát, bản câu hỏi thăm dò. |
| Quản lý ngân hàng | Tổ chức cuộc họp, tạo kịch bản, khảo sát hiện trạng. |
| Kiểm toán viên | Tổ chức 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**

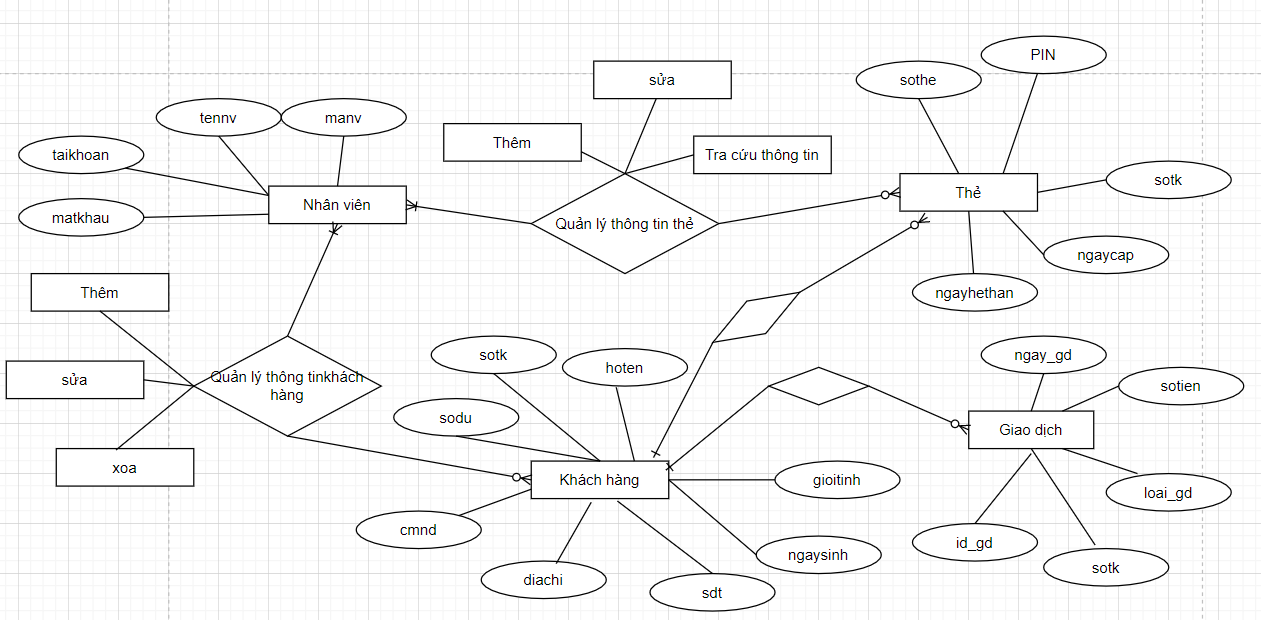
**4.1. Mô hình phân cấp chức năng (FDD)**

****

**4.2. Biểu đồ luồng dữ liệu (DFD)**

****

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

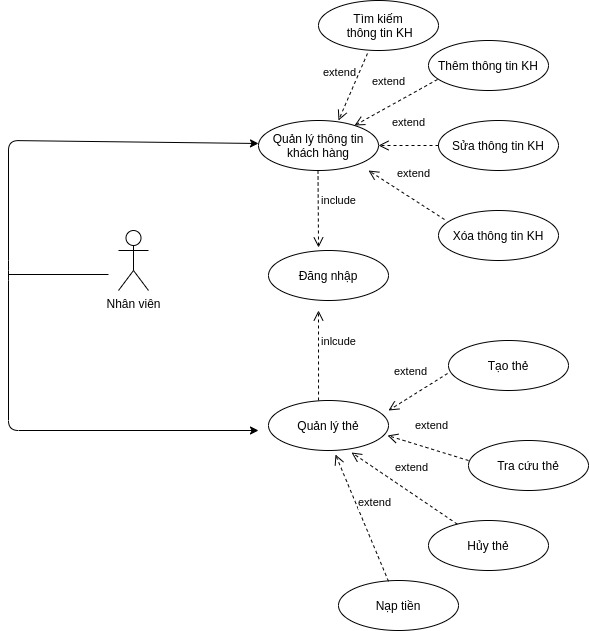
****

**5. Xác định tác nhân và Use Case**

| Tác nhân | Use case |
| --- | --- |
| Nhân viên ngân hàng | * Đăng nhập * Quản lý Khách hàng   + Thêm KH   + Sửa TT KH   + Xóa KH   + Tìm kiếm KH * Quản lý Thẻ   + Nạp tiền vào TK   + Thêm thẻ   + Hủy thẻ   + Tra cứu thông tin thẻ |

**6. Biểu đồ và mô tả Use Case**

**6.1. Biểu đồ Use Case**

****

**6.2. Mô tả Use Case**

***6.2.1. Đăng nhập***

| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Sử dụng khi nhân viên ngân hàng muốn đăng nhập vào hệ thống. |
| Điều kiện tiên quyết: | * Hồ sơ của nhân viên khách hàng đã được cập nhập vào hệ thống. * Nhân viên được cấp tên đăng nhập và mật khẩu. * Tài khoản nhân viên được phân quyền. |
| Điều kiện sau: | Đăng nhập thành công vào hệ thống và có thể sử dụng các chức năng mà hệ thống cung cấp. |
| Sự ưu tiên: | Phải có |
| Tần suất sử dụng: | Luôn luôn |
| Luồng sự kiện: | 1. Chọn chức năng đăng nhập 2. Hiển thị form đăng nhập 3. Nhân viên nhập tên đăng nhập và mật khẩu. 4. Hệ thống kiểm tra tính đúng đắn 5. Đúng thì đăng nhập thành công. Sai thì chuyển sang sự kiện thay thế. 6. UC kết thúc. |
| Trường hợp ngoại lệ: | Khi đăng nhập không thành công:   1. Hệ thống thông báo sai tên tài khoản hoặc mật khẩu. 2. Hệ thống yêu cầu nhập lại tên đăng nhập và mật khẩu. 3. Đồng ý: quay lại bước 2 của luồng sự kiện. |
| Yêu cầu đặc biệt: |  |

***6.2.2. Quản lý khách hàng***

***6.2.2.1. Thêm thông tin khách hàng***

| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Nhân viên thêm thông tin khách hàng khi khách hàng tạo thẻ lần đầu. |
| Điều kiện tiên quyết: | * Khi có yêu cầu từ khách hàng * Đã đang nhập thành công |
| Điều kiện sau: | Thực hiện các chức năng thêm thông thành công. |
| Sự ưu tiên: | Bắt buộc |
| Tần suất sử dụng: | Thường xuyên |
| Luồng sự kiện: | 1.Nhân viên chọn chức năng thêm thông tin khách hàng.  2. Hiển thị form thông tin sau đó nhân viên sẽ thực thao tác chỉnh sửa .  3. Nhấn thêm  4. Hiển thị thông báo thành công và quay về form chính.  5. UC kết thúc |
| Luồng đi khác: |  |
| Trường hợp ngoại lệ: | 1. Hiển thị thông báo nhập sai dữ liệu và yêu cầu nhập lại.  2.Nhấp thêm thông tin.  3.Thông báo thành công và quay về form chính. |
| Yêu cầu đặc biệt: |  |

***6.2.2.2. Sửa thông tin khách hàng***

| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Nhân viên sửa thông tin khách hàng khi khách hàng |
| Điều kiện tiên quyết: | Khi khách hàng muốn thay đổi và thông tin đã có sẵn trong cơ sở dữ liệu. Đăng nhập thành công |
| Điều kiện sau: | Thực hiện các chức năng xoá thông tin thành công. |
| Sự ưu tiên: | Không bắt buộc |
| Tần suất sử dụng: | Không thường xuyên. |
| Luồng sự kiện: | 1.Nhân viên chọn chức năng sửa thông tin khách hàng.  2. Hiển thị form thông tin sau đó nhân viên sẽ thực thao tác chỉnh sửa .  3. Nhấn sửa  4. Hiển thị thông báo thành công và quay về form chính.  5. UC kết thúc |
| Luồng đi khác: | 1.Hiển thị thông báo nhập sai dữ liệu và yêu cầu nhập lại.  2.Nhấp sửa thông tin.  3.Thông báo thành công và quay về form chính. |
| Trường hợp ngoại lệ: |  |
| Yêu cầu đặc biệt: |  |

***6.2.2.3 Xóa thông tin khách hàng***

| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Nhân viên xóa thông tin khách hàng khi khách hàng đó muốn dừng sử dụng nữa. |
| Điều kiện tiên quyết: | * Đang nhập thành công * Nhận được yêu cầu từ khách hàng và thông tin khách hàng đó phải có trong cơ sở dữ liệu. * Tìm thấy khách hàng cần xóa |
| Điều kiện sau: | Thực hiện xóa thông tin của khách hàng thành công. |
| Sự ưu tiên: | Không bắt buộc |
| Tần suất sử dụng: | Không thường xuyên |
| Luồng sự kiện: | 1.Nhân viên chọn chức năng xóa thông tin khách hàng.  2. Hiển thị form thông tin của khách hàng cần xóa  3. Nhấn xóa  4. Hiển thị thông báo thành công và quay về form chính.  5. UC kết thúc. |
| Luồng đi khác: | Không muốn xóa nữa:   1. Nhấn hủy 2. Quay về form chính |
| Trường hợp ngoại lệ: | Không tìm thấy thông tin khách hàng |
| Yêu cầu đặc biệt: |  |

***6.2.2.4. Tìm kiếm khách hàng***

| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Nhân viên tìm kiếm thông tin của một khách hàng. |
| Điều kiện tiên quyết: | Khách hàng có phải có trong cơ sở dữ liệu của ngân hàng. |
| Điều kiện sau: | Hiển thị thông tin của khách hàng cần tìm kiếm |
| Sự ưu tiên: | Cần có khi sửa hoặc xóa TT khách hàng |
| Tần suất sử dụng: | Thường xuyên |
| Luồng sự kiện: | 1. Nhân viên chọn quản lý khách hàng 2. Hệ thống hiển thị form quản lý khách hàng 3. Nhập thông tin cần tìm (mã KH, tên KH, địa chỉ, …) 4. Hiển thị những khách hàng có thông tin liên quan đến dữ liệu nhập vào. 5. UC kết thúc. |
| Luồng đi khác: |  |
| Trường hợp ngoại lệ: | Không tìm thấy thông tin khách hàng. |
| Yêu cầu đặc biệt: |  |

***6.2.3. Quản lý thẻ***

***6.2.3.1. Tạo thẻ***

| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Nhân viên tạo thẻ mới cho khách hàng . |
| Điều kiện tiên quyết: | Có yêu cầu tạo thẻ |
| Điều kiện sau: | Thực hiện tạo thẻ thành công. |
| Sự ưu tiên: | Không bắt buộc |
| Tần suất sử dụng: | Thường xuyên |
| Luồng sự kiện: | 1. Nhân viên chọn chức năng tạo thẻ. 2. Hiển thị form tạo thẻ 3. Nhân viên nhập các thông tin cần thiết 4. Nhấn tạo thẻ. 5. Hiển thị thông báo thành công và quay về form chính. 6. UC kết thúc |
| Luồng đi khác: | Nhân viên nhập thiếu thông tin:   1. Hiển thị thông báo nhập thiếu 2. Quay lại bước 2 của luồng sự kiện. |
| Trường hợp ngoại lệ: | Nhân viên không tạo thẻ nữa:   1. Nhấn thoát 2. Quay trở về form đăng nhập. |
| Yêu cầu đặc biệt: |  |

***6.2.3.2. Tra cứu thông tin thẻ***

| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | * Nhân viên chọn chức năng tra cứu thẻ khi muốn thống kê thông tin,...... * Trước khi muốn hủy thẻ |
| Điều kiện tiên quyết: | Có mã thẻ hoặc tên khách hàng |
| Điều kiện sau: | Thực hiện các chức năng tra cứu thông tin thành công. |
| Sự ưu tiên: | Không bắt buộc |
| Tần suất sử dụng: | Thường xuyên. |
| Luồng sự kiện: | 1.Nhân viên chọn chức năng tra cứu thông tin thẻ.  2. Nhập mã thẻ vào ô tìm kiếm  3. Nhấn tra cứu thông tin.  4. Hiển thị form thông tin thẻ bao gồm mã thẻ, mã khách hàng, ngày lập, ngày hết hạn và loại thẻ .  5. Nhấn quay lại  6. Hiển thị form chính sau khi thoát.  7. UC kết thúc. |
| Luồng đi khác | 1.Hiển thị thông báo nhập sai dữ liệu và yêu cầu nhập lại.  2.Nhấp tra cứu thông tin  3.Thông báo thành công và hiển thị thông tin. |
| Trường hợp ngoại lệ | Không tìm thấy thẻ |
| Yêu cầu đặc biệt: |  |

**6.2.*3.3. Huỷ thẻ***

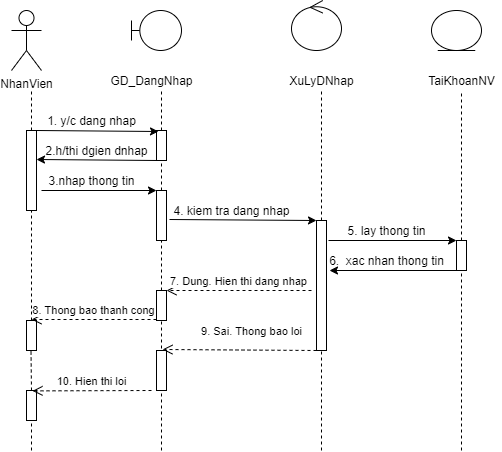
| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Nhân viên nạp tiền vào tài khoản cho khách hàng |
| Điều kiện tiên quyết: | * Thẻ hết hạn, khách hàng không muốn sử dụng thẻ nữa. * Tìm thấy thẻ cần hủy. |
| Điều kiện sau: | Thực hiện hủy thẻ thành thành công. |
| Sự ưu tiên: | Không bắt buộc |
| Tần suất sử dụng: | Không thường xuyên |
| Luồng sự kiện: | 1. Nhân viên chọn hủy thẻ 2. Hiển thị thông tin thẻ cần hủy 3. Nhấn Hủy 4. Xác nhận hủy thẻ 5. Thông báo hủy thẻ thành công và quay lại form chính 6. UC kết thúc. |
| Luồng đi khác |  |
| Trường hợp ngoại lệ | Tài khoản còn nợ tiền không thể thực hiện hủy thẻ, cần nạp tiền duy trì thẻ ban đầu sẵn có, sau đó mới thực hiện thao tác hủy thẻ được. |
| Yêu cầu đặc biệt: | Thẻ không nợ tiền |

***6.2.3.4. Nạp tiền vào thẻ***

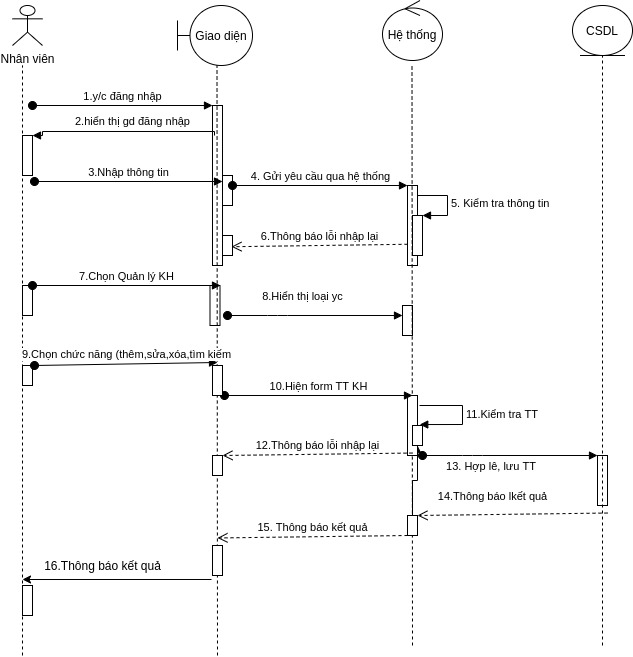
| Tác nhân: | Nhân viên ngân hàng |
| --- | --- |
| Mô tả: | Nhân viên muốn nạp tiền cho khách hàng vào thẻ |
| Điều kiện tiên quyết: | Khi khách hàng có nhu cầu nạp thẻ.. |
| Điều kiện sau: | Thực hiện nạp tiền vào thẻ cho khách hàng thành công. |
| Sự ưu tiên: | Không bắt buộc |
| Tần suất sử dụng: | Không thường xuyên |
| Luồng sự kiện: | 1. Nhân viên chọn tìm kiếm thông tin thẻ 2. Nhân viên chọn nạp tiền vào thẻ 3. Nhấn ok 4. Thông báo nạp tiền thành công và quay lại form chính. 5. UC kết thúc. |
| Luồng đi khác |  |
| Trường hợp ngoại lệ | Tài khoản đã bị khóa, không thể nạp tiền, cần thực hiện mở khóa sau đó tiếp tục giao dịch |
| Yêu cầu đặc biệt: | Tài khoản không bị khóa .(Khóa thẻ do nhấn sai mật khẩu quá 5 lần,..) |

**7. Biểu đồ trình tự**

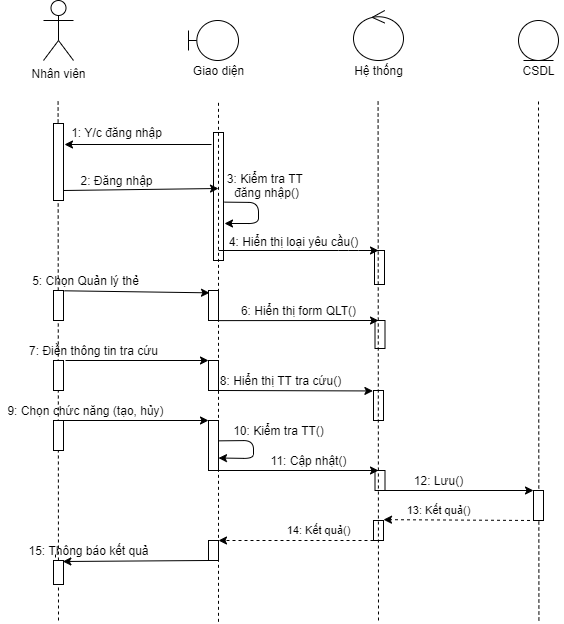
**7.1. Biểu đồ trình tự Đăng nhập**

****

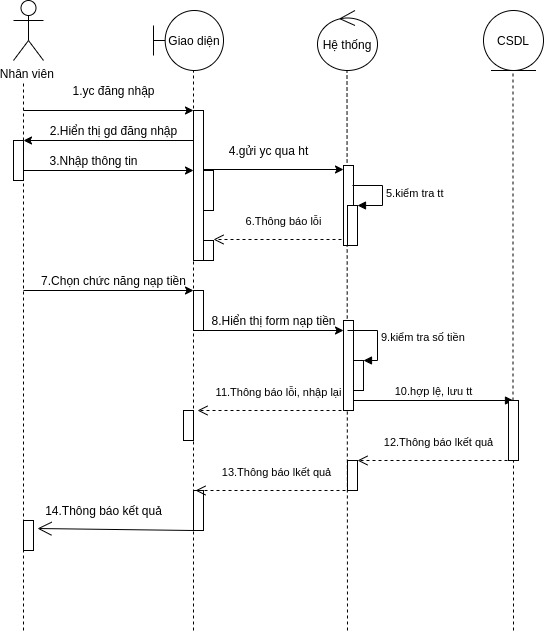
**7.2. Biểu đồ trình tự Quản lý khách hàng**

****

**7.3. Biểu đồ trình tự Tạo, Hủy thẻ**

****

**7.4. Biểu đồ trình tự Nạp tiền.**

****

**II. PHÂN TÍCH & THIẾT KẾ**

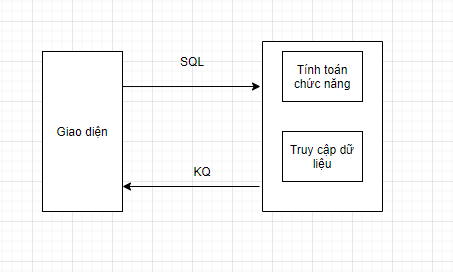
**1. Thiết kế kiến trúc**

**1.1. Kiến trúc 2 lớp**

* Thin Client

Client : là giao diện người dùng

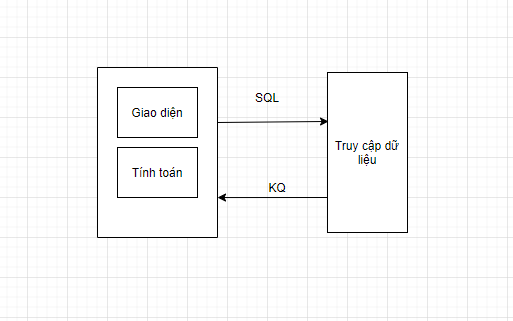
Server : bao gồm quá trình xử lý nghiệp vụ và truy cập dữ liệu



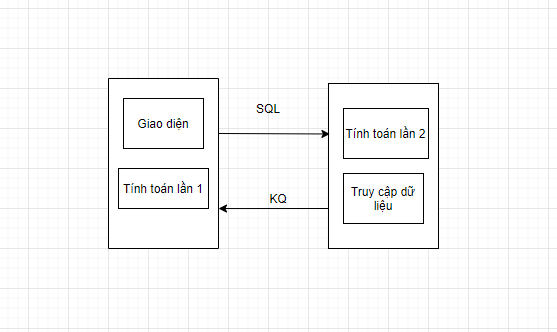
* Fat client

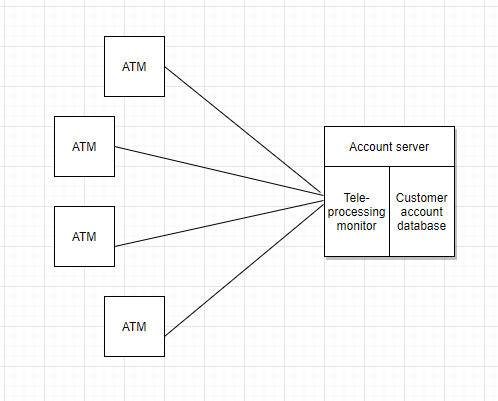
Client : là giao diện người dùng và xử lý nghiệp vụ.

Server : Chỉ truy cập dữ liệu.

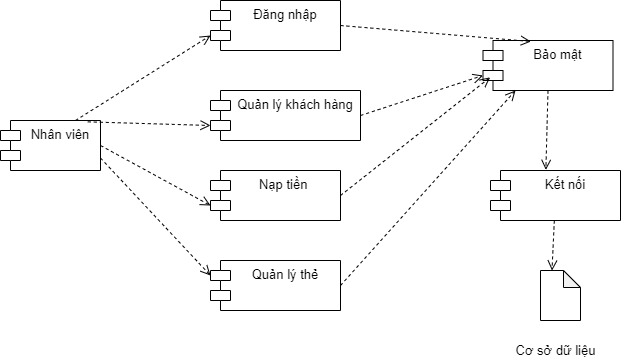


* Kết hợp

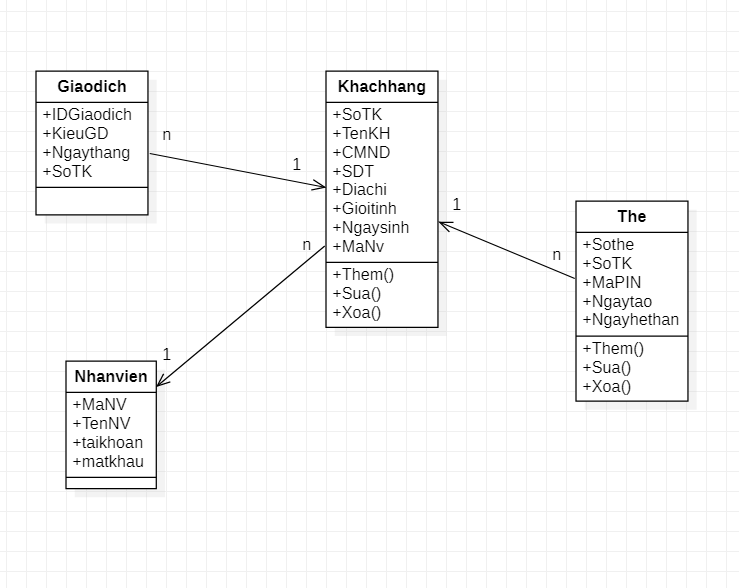




**1.2. Biểu đồ thành phần**

****

**2. Thiết kế hướng đối tượng**

****

**3.Thiết kế dữ liệu**

**3.1 Chuẩn hóa về dạng 3NF**

* Khách hàng(SoTk, Hoten, Sodu, cmnd, Diachi, Sdt, NgaySinh, GTinh)

Fo = {SoTk → Hoten, SoTk → Sodu, SoTk → cmnd, SoTk→ Diachi,

SoTk→ Ngaysinh, SoTk→ GTinh}

Lược đồ quan hệ chỉ chứa các thuộc tính nguyên tố: lược đồ ở dạng chuẩn 1NF

Lược đồ có khóa chính là SoTK, các thuộc tính khác phụ thuộc đầy đủ vào khóa chính nên ở dạng chuẩn 2NF.

Do SoTk → Hoten, SoTk → Sodu, SoTk → cmnd, SoTk→ Diachi, SoTk→Ngaysinh, SoTk→ GTinh, đã ở dạng 2NF và tất cả các thuộc tính không khóa không phụ bắc cầu vào khóa nên ở dạng chuẩn 3NF.

* Nhân viên (MaNV, TenNV, TaiKhoan,MatKhau)

F0 = { MaNV → TenNV; MaNV → TaiKhoan; MaNV → MatKhau)

Lược đồ quan hệ chỉ chứa các thuộc tính nguyên tố: lược đồ ở dạng chuẩn 1NF.

Lược đồ có khóa chính là MaNV, các thuộc tính khác phụ thuộc đầy đủ vào khóa chính nên ở dạng chuẩn 2NF.

MaNV → TenNV; MaNV → TaiKhoan; MaNV → MatKhau và MaNV là khóa chính, TenNV, TaiKhoan, MatKhau không phải là khóa nhưng phụ thuộc trực tiếp vào khóa chính nên sơ đồ ở dạng chuẩn 3NF.

* Thẻ (SoThe, SoTk, MaPin, NgayTao,NgayHetHan)

F0={SoThe→ MaPin, SoThe→ SoTk, SoThe→ NgayTao, SoThe→ NgayHetHan}

Lược đồ quan hệ chỉ chứa các thuộc tính nguyên tố : Lược đồ ở dạng chuẩn 1NF.

Lược đồ có khóa chính là SoThe , và khóa phụ là SoTk nhưng khóa phụ này không kéo theo 1 thuộc tính nào cả, nên các thuộc tính phụ thuộc đầy đủ vào SoThe → 3NF.

Xét tập Khóa Sothe,SoTk→ MaPin; SoThe,SoTk→ NgayTao; SoThe,

Sotk→ NgayHethan.

Vậy ta có SoThe,SoTk là siêu khóa chính → BCNF.

* GiaoDich(id\_gd, ngay\_gd, sotien, loai\_gd, sotk)

F0={id\_gd → ngay\_gd; id\_gd → sotien; id\_gd → loai\_gd, id\_gd → sotk}

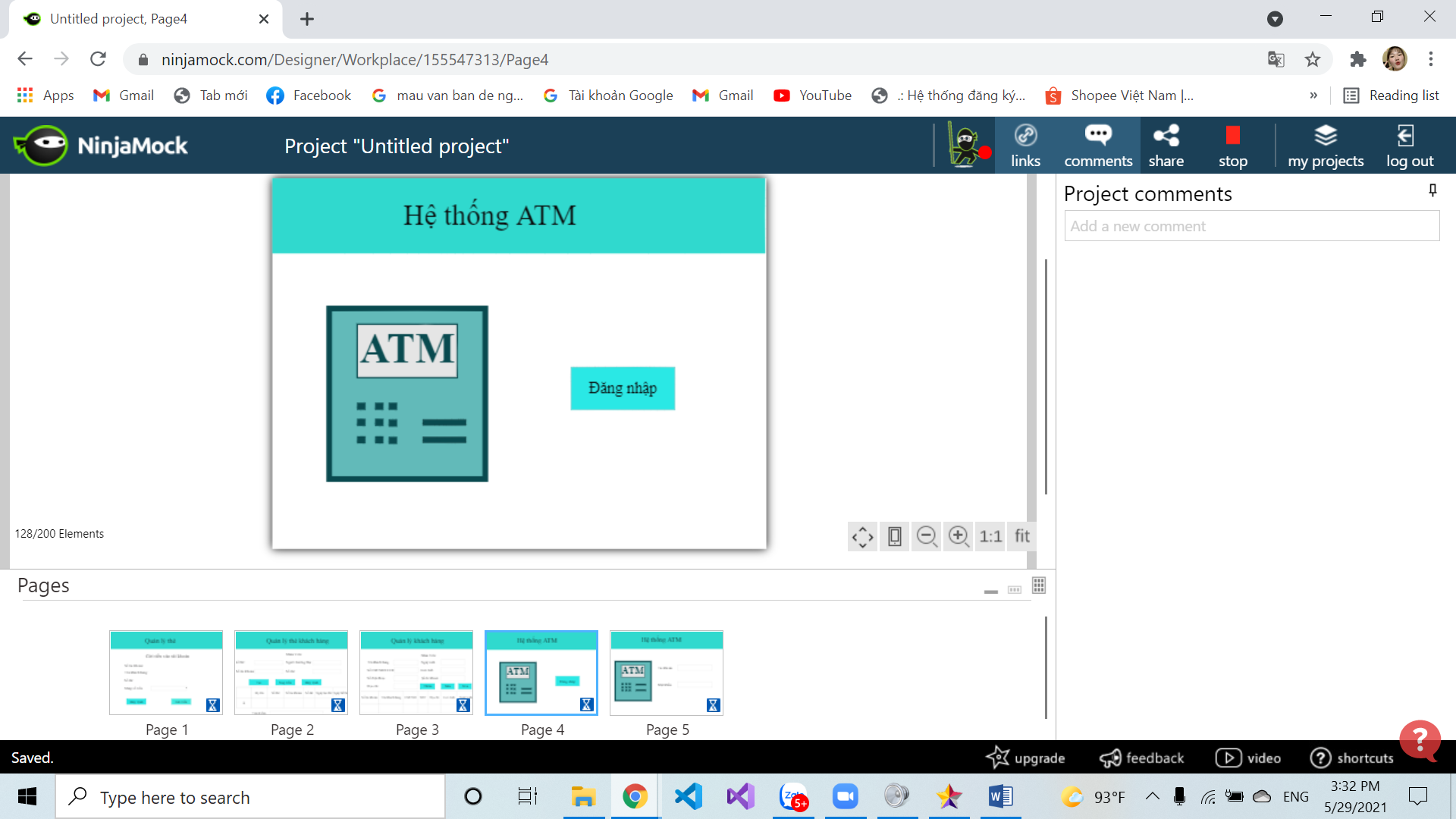
Lược đồ quan hệ chỉ chứa các thuộc tính nguyên tố nên ở dạng chuẩn 1NF.

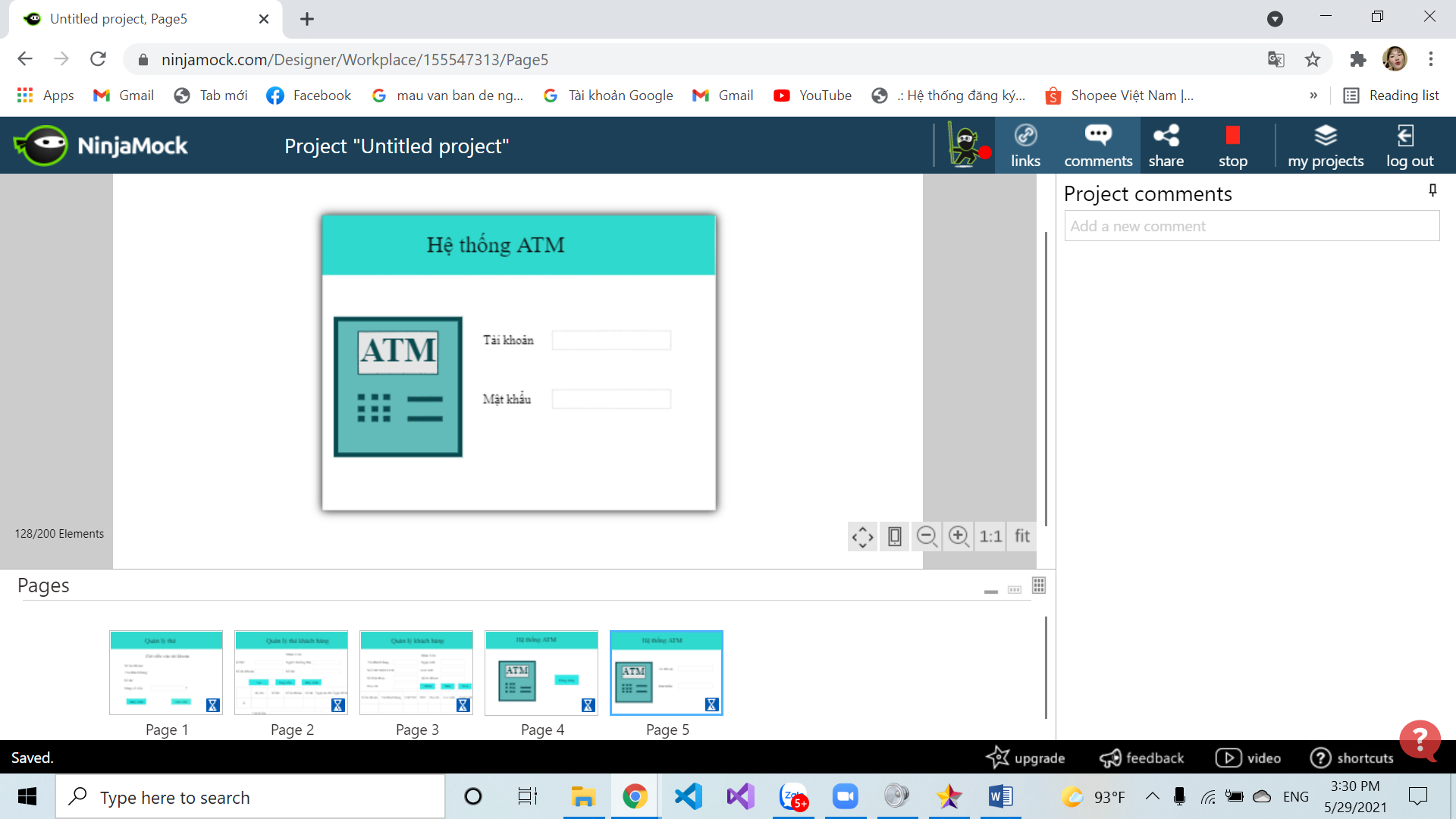
Lược đồ có khóa chính là id\_gd và khóa phụ là sotk nhưng khóa phụ này không kéo theo 1 thuộc tính nào cả nên các thuộc tính phụ thuộc đầy đủ vào khóa chính nên sẽ ở dạng chuẩn 3NF.

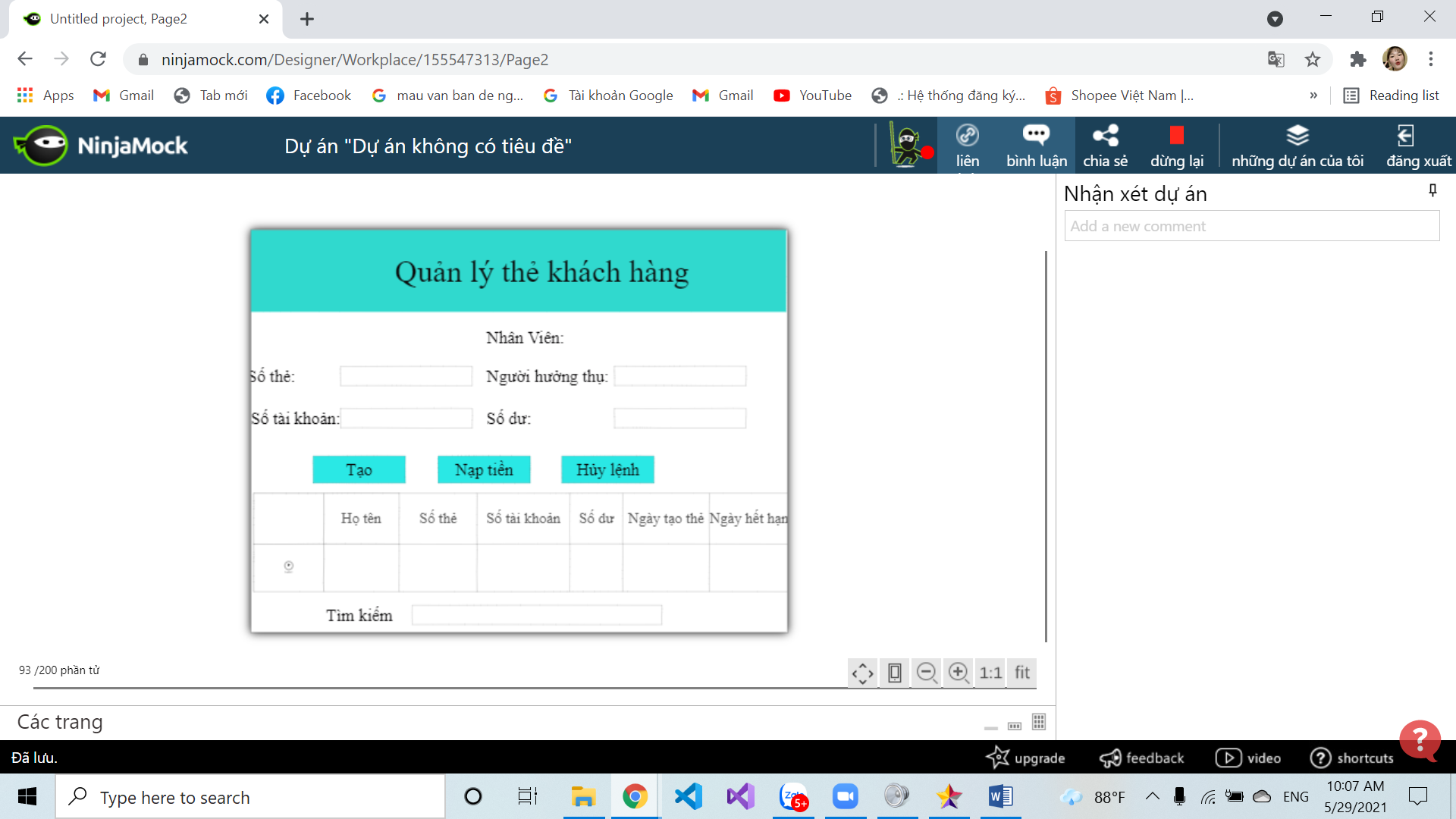
Xét tập khóa F1={ id\_gd,sotk → sotien; id\_gd,sotk → loai\_gd}

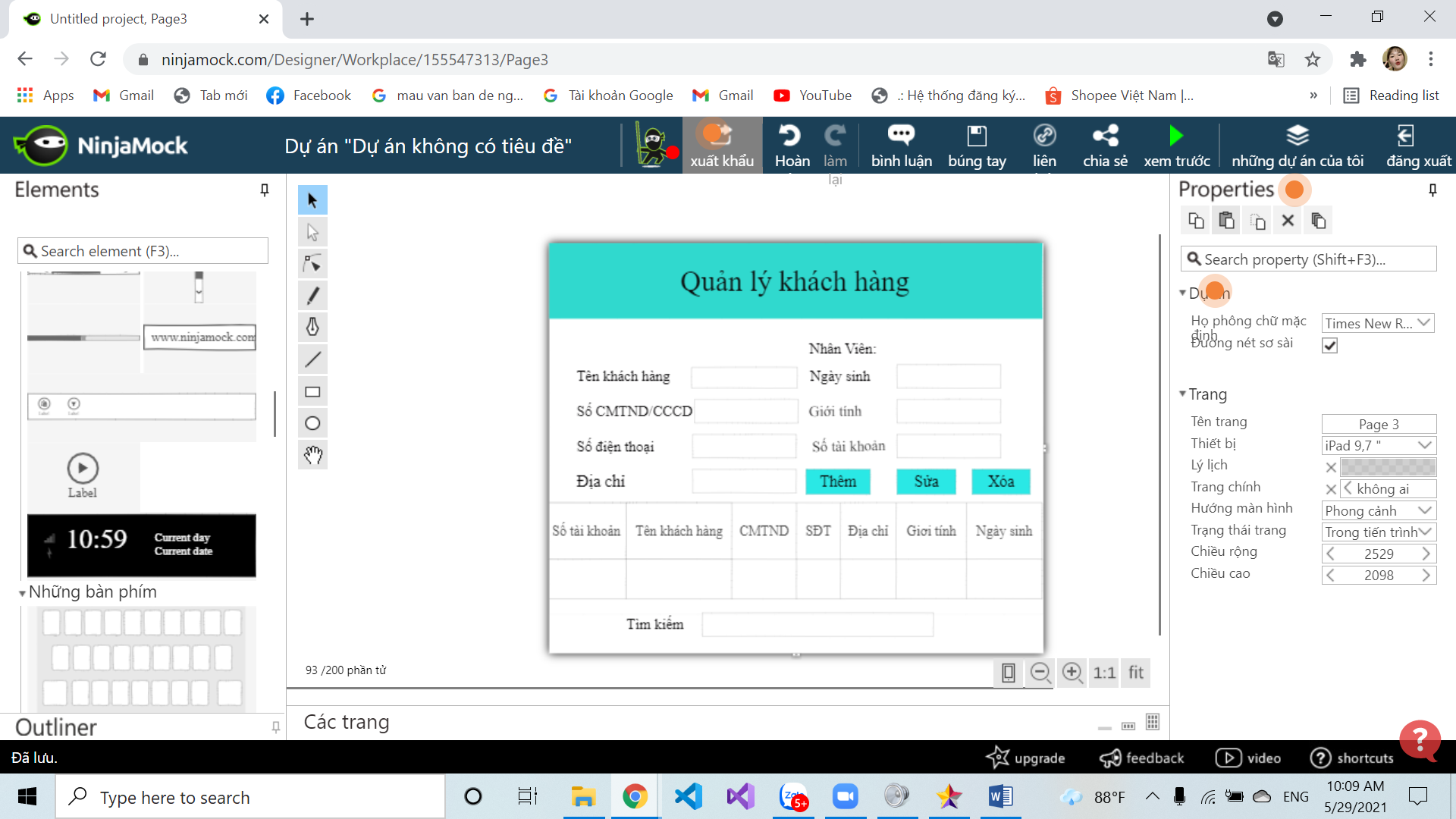
Ta có id\_gd,sotk là một siêu khóa (vì id\_gd, sotk là khóa chính) nên ở dạng BCNF.

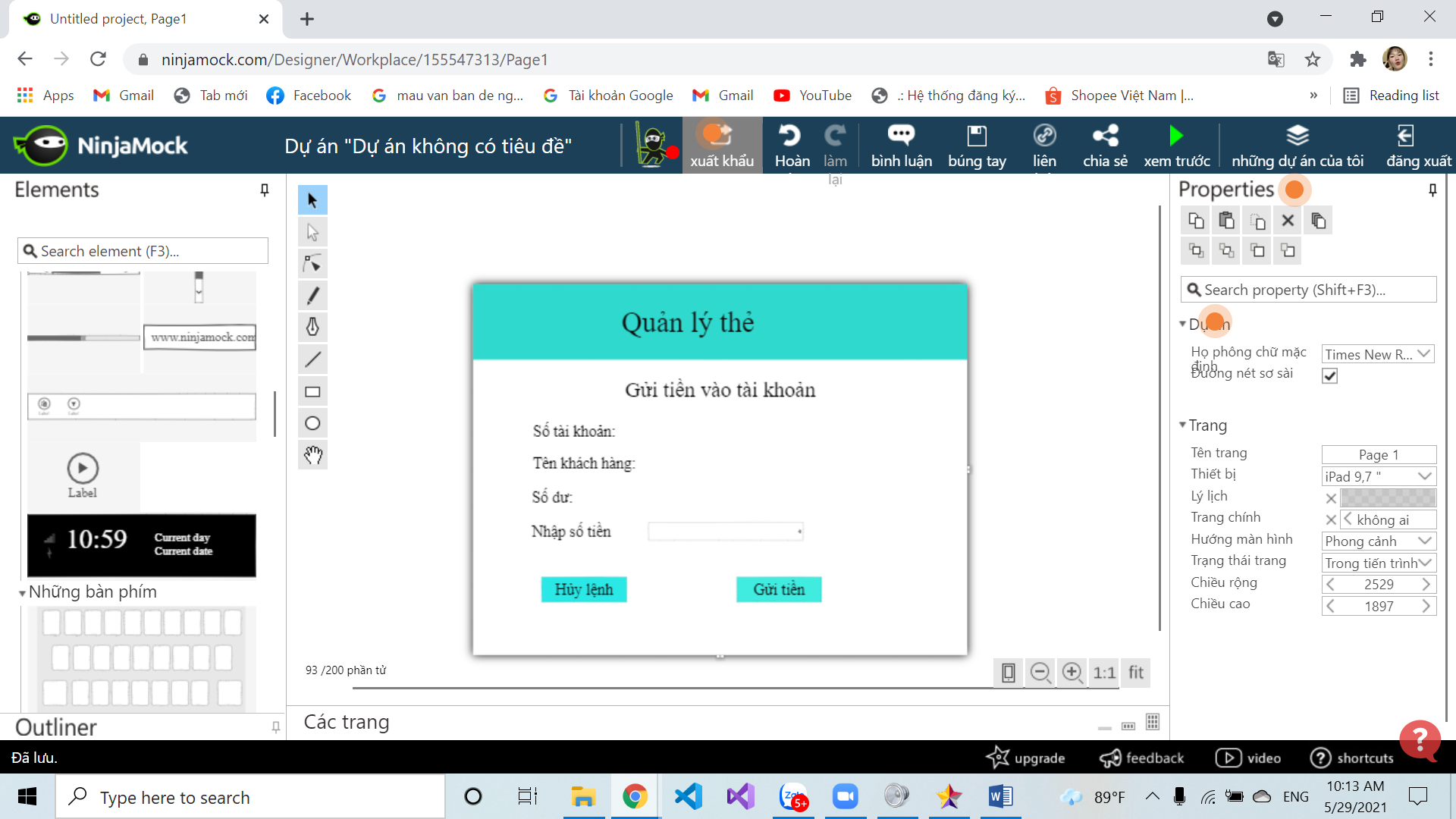
**4. Thiết kế giao diện**

****

****

****

****

****

**III. CODE**

**IV. TÀI LIỆU KIỂM THỬ**

1. **Kiểm thử hộp đen**

(Bắt buộc phải có dữ liệu cụ thể)

| **Hình thức** | **Mã kiểm thử** | **Miêu tả** | **Tiến trình kiểm thử** | | **Kết quả các tiến trình kiểm thử** | **Kết quả thực tế** | **Trạng thái** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Các bước thực hiện** | **Kết quả từng bước** |
| Đăng nhập | 01 | Nhập đúng tài khoản và mật khẩu nhân viên | Nhập tài khoản nhân viên  (nhân viên nào) | True | Yêu cầu được thực hiện. Vào được hệ thống | Đăng nhập thành công | Pass |
| Nhập mật khẩu nhân viên | True |
| Nhập đúng tài khoản nhưng sai mật khẩu | Nhập tài khoản nhân viên | True | Thông báo lỗi | Thông báo lỗi | Pass |
| Nhập mật khẩu nhân viên | False |
| Nhập đúng mật khẩu nhưng sai tài khoản | Nhập tài khoản  nhân viên | False | Thông báo lỗi | Thông báo lỗi | Pass |
| Nhập mật khẩu nhân viên | True |
| Tạo thẻ | 02 | Nhập đầy đủ thông tin yêu cầu | Nhân viên nhập thông tin cần thiết để tạo thẻ | True | Tạo thẻ thành công | Tạo thẻ thành công | c |
| Nhập thiếu thông tin thẻ | Nhân viên nhập thiếu thông tin cần thiết để tạo thẻ | False | Thông báo lỗi | Thông báo lỗi và mời nhập lại |
| Nạp tiền | 03 | Nhập số tiền >0 | Nhập số tiền cần nạp | True | Thông báo thành công | Nạp thành công | Pass |
| Nhập số tiền <0 hoặc không nhập | Nhập số tiền cần nạp | False | Thông báo lỗi | Thông báo lỗi mời nhập lại |
| Xóa thẻ | 04 | Tìm thấy thẻ cần xóa | Nhân viên tìm được thẻ cần xóa và xóa | True | Xóa thẻ thành công | Thẻ đã được xóa | Pass |
| Không tìm thấy thẻ | Nhân viên không tìm được thẻ cần xóa | False | Không hiển thị thông tin thẻ được tìm | Không hiển thị thông tin thẻ được tìm |
| Thêm khách hàng | 05 | Nhập đầy đủ thông tin về khách hàng và chưa tồn tại số CCCD, số tài khoản trong CSDL | Nhập đầy đủ tất cả thông tin cần thiết của khách hàng | True | Thêm thành công | Thêm thành công và lưu vào CSDL | Pass |
| Nhập đầy đủ thông tin về khách hàng và số CCCD, số tài khoản đã tồn tại trong CSDL | Nhập đầy đủ tất cả thông tin cần thiết của khách hàng | False | Thông báo lỗi | Yêu cầu nhập lại thông tin |
| Nhập thiếu thông tin của khách hàng | Nhập thiếu một trong những thông tin của khách hàng | False | Thông báo lỗi | Yêu cầu nhập lại thông tin |
| Tìm kiếm thông tin | 06 | Nhập đúng thông tin cần tìm kiếm | Nhập thông tin cần tìm vào ô tìm kiếm | True | Tìm kiếm thành công | Hiển thị thông tin cần tìm | Pass |
| Nhập sai thông tin | Nhân viên nhập thông tin sai vào ô tìm kiếm | False | Thông báo lỗi | Yêu cầu nhập lại thông tin |
| Sửa thông tin KH | 07 | Tìm thấy KH và nhập đúng đầy đủ thông tin cần sửa | Nhân viên tìm thấy và nhập đúng đầy đủ thông tin cần sửa | True | Sửa thành công | Sửa thành công và lưu vào CSDL | Pass |
| Tìm thấy thông tin KH và nhập thiếu thông tin cần sửa | Nhân viên tìm thấy và nhập thiếu thông tin cần sửa | False | Thông báo lỗi | Thông báo lỗi |
| Không tìm thấy thông tin KH | Nhân viên không tìm thấy | False | Không hiển thị thông tin KH được tìm | Không hiển thị thông tin KH được tìm |
| Xóa khách hàng | 08 | Tìm thấy khách hàng cần xóa | Nhân viên tìm được KH cần xóa | True | Thông báo thành công | Khách hàng đã được xóa | Pass |
| Không tìm thấy khách hàng | Nhân viên không tìm được khách hàng cần xóa | False | Không hiển thị thông tin khách hàng được tìm | Không hiển thị thông tin KH được tìm |

1. **Kiểm thử hộp trắng**

**2.1. Thêm khách hàng**

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

{

(1) if (txbDiachi.Text == "" || txbSDT.Text == "" || txbTenKH.Text == "" || txbSTK.Text == "" || txbSoCMT.Text == "" || comboGioitinh.Text == "")

{

(2) MessageBox.Show("Chưa nhập đủ thông tin khách hàng!");

return;

}

(3) else

{

(4) try

{

(5) conn.Open();

(5) string query = "Insert into khachhang values('"+txbSTK.Text+"',N'"+txbTenKH.Text+"','"+txbSoCMT.Text+"','"+txbSDT.Text+"',N'"+txbDiachi.Text+"','0',N'"+comboGioitinh.Text+"','"+dtpkNgaysinh.Text+"')";

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

(5) cmd.ExecuteNonQuery();

(5) MessageBox.Show("Thêm thành công!");

conn.Close();

populate();

(5) txbDiachi.Clear();

(5) txbSDT.Clear();

(5) txbSoCMT.Clear();

(5) txbTenKH.Clear();

(5) txbSTK.Clear();

}

(6) catch (Exception Ex)

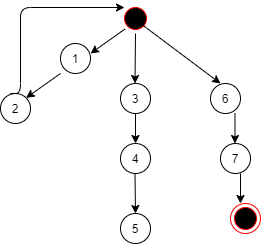
{

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

}

}

}



* Số lập trình độc lập (độ phức tạp lặp) = 3
* Có 3 đường thi hành tuyến tính độc lập cơ bản là:

1. 1→ 2
2. 3→ 4→ 5
3. 6→ 7

* Thiết kế testcase
* Test case cho đường 1:

+) Kết quả kỳ vọng: Hiển thị ra thông điệp “Chưa nhập đủ thông tin khách hàng”

* Test Case cho đường 2:

+)Kết quả kỳ vọng: Thêm khách hàng thành công

* Testcase cho đường 3:

+)Kết quả kỳ vọng: Hiển thị thông tin về lỗi đang gặp phải.

**2.2. Sửa khách hàng**

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

{

(2) if (txbSTK.Text != stk)

{

(3) MessageBox.Show("Bạn không được phép sửa số tài khoản!");

}

(4) else

{

(5) try

{

(6) conn.Open();

(6) string query1 = "Update khachhang set hoten =N'" + txbTenKH.Text + "', soCMT= '" + txbSoCMT.Text + "', diachi = N'" + txbDiachi.Text + "', soDT = '" + txbSDT.Text + "', gioitinh = N'" + comboGioitinh.Text + "' where sotk = '" + txbSTK.Text+"' ";

(6) SqlCommand cmd1 = new SqlCommand(query1, conn);

(6) cmd1.ExecuteNonQuery();

(6) MessageBox.Show("Sửa thành công!");

(6) conn.Close();

(6) populate();

(6) txbDiachi.Clear();

(6) txbSDT.Clear();

(6) txbSoCMT.Clear();

(6) txbTenKH.Clear();

(6) txbSTK.Clear();

}

(7) catch (Exception Ex)

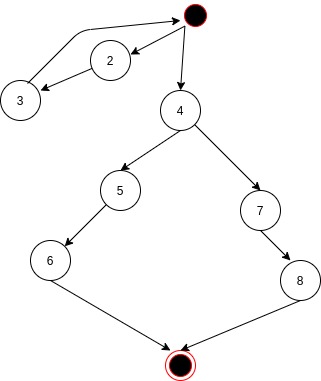
{

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

}

}

}



Số lập trình độc lập (độ phức tạp lặp) = 3

Có 2 đường thi hành tuyến tính độc lập cơ bản là:

* 2 --> 3
* 4 --> 5 --> 6
* 4 --> 7 --> 8

Thiết kế test case:

* Test case cho đường 1 :

+) Kết quả kỳ vọng: in ra thông báo "Bạn không được phép sửa số tài khoản".

* Test case cho đường 2:

+) Kết quả kỳ vọng: in ra thông báo " Sửa thành công!".

* Test case cho đường 3:

+)Kết quả kỳ vọng: in ra thông điệp:

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

**2.3. Xóa khách hàng**

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

{

(1) DialogResult result = MessageBox.Show("Bạn có chắc muốn xóa khách hàng này?", "", MessageBoxButtons.YesNo);

switch (result)

{

(2) case DialogResult.No:

break;

(3) case DialogResult.Yes:

conn.Open();

(4) string query2 = "Delete from the where sotk = '"+txbSTK.Text+ "'";

SqlCommand cmd2 = new SqlCommand(query2, conn);

cmd2.ExecuteNonQuery();

string query4 = "Delete from giaodich where sotk = '" + txbSTK.Text + "'";

(4) SqlCommand cmd4 = new SqlCommand(query4, conn);

cmd4.ExecuteNonQuery();

(4) string query3 = "delete from khachhang where sotk = '" + txbSTK.Text + "'";

SqlCommand cmd3 = new SqlCommand(query3, conn);

cmd3.ExecuteNonQuery();

conn.Close();

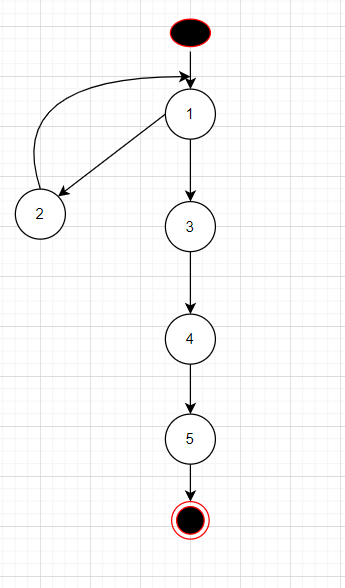
(5) MessageBox.Show("Xóa thành công!");

populate();

break;

}

}



* Số lập trình độc lập (độ phức tạp lặp) = 2
* Có 2 đường thi hành tuyến tính độc lập cơ bản là :
  + 1→ 2
  + 1→ 3→ (4,5,6)→ 7
* Thiết kế test case:
* Test case cho đường 1:

+)Kết quả kỳ vọng: Sau khi nhấn xóa, in ra thông báo:*“Bạn có muốn xóa khách hàng không?”.*

*+)* Nhấn hủy xóa và quay lại form quản lý khách hàng.

* Test case cho đường 2:

+)Kết quả kỳ vọng: Sau khi nhấn xóa, in ra thông báo:*“Bạn có muốn xóa khách hàng không?”.*

*+)* Nhấn đồng ý, in ra thông báo : “*Xóa thành công”*

+) Thực hiện xóa thông tin từ bảng các bảng thẻ, khách hàng, giao dịch với thông tin đã chọn..

**2.4. Tìm kiếm khách hàng**

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

{

(1) string k = txbTimkiem.Text;

(2) string query5 = "Select sotk, Hoten, soCMT, soDT, Diachi, gioitinh, ngaysinh, sodu from khachhang where SoTK like '%" + k + "%' or Diachi like N'%" + k + "%' or Hoten like N'%" + k + "%' or soCMT like '%" + k + "%' or SoDT like '%" + k + "%'";

(3) SqlDataAdapter adapter1 = new SqlDataAdapter(query5, conn);

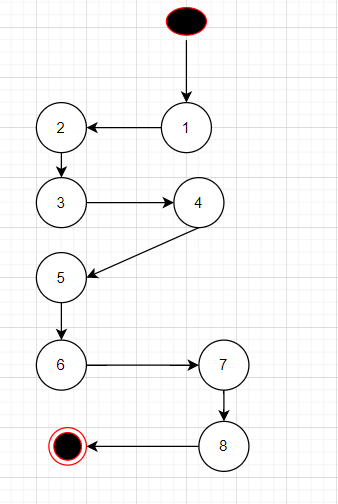
(4) SqlCommandBuilder builder = new SqlCommandBuilder(adapter1);

(5) var dss = new DataSet();

(6) adapter1.Fill(dss);

(7) dtgvKH.DataSource = dss.Tables[0];

}



* Số lập trình độc lập = 1
* Có 1 đường thi hành tuyến tính độc lập cơ bản là :
  + 1→ 2→ 3→ 4→ 5→ 6→ 7
* Thiết kế test case:

Test case cho đường 1:

+) Kết quả kỳ vọng: Sau khi nhấn tìm kiếm thì hiển thị ra thông tin liên quan đến từ khóa cần tìm kiếm.

**2.5. Nạp tiền vào tài khoản**

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

{

(1) if (txbSotien.Text == "" || Convert.ToInt32(txbSotien.Text) < 0)

{

(2) MessageBox.Show("Nhập số tiền");

}

(3) else

{

(4) sodu = soducu + Convert.ToInt32(txbSotien.Text);

(5) try

{

(6) conn.Open();

(6) string query = "update khachhang set sodu ='" + sodu + "' where SoTK ='" + sotk + "'";

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

(6) cmd.ExecuteNonQuery();

(6) MessageBox.Show("Nạp thành công!");

(6) home home = new home();

(6) home.Show();

(6) this.Hide();

(6) conn.Close();

}

(7) catch(Exception Ex)

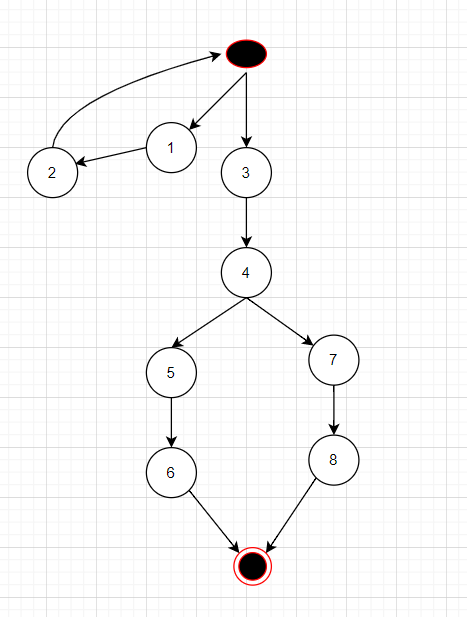
{

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

}

}

}



* Số lập trình độc lập = 2
* Có 2 đường thi hành tuyến tính độc lập cơ bản là:
  + 1 -> 2
  + 3 -> 4 -> 5 -> (6,7,8)
* Thiết kế test case:

- Test case cho đường 1:

+) Kết quả kỳ vọng: Quay lại form “Nhập số tiền” nếu nhập vào số tiền nạp <0

- Test case cho đường 2:

+) Kết quả kỳ vọng: Cập nhập lại số dư: sodu = soducu + sotiennap sau khi nạp thêm số tiền >0.

+) In ra thông báo: “Nạp thành công”.

+) Quay trở lại form quản lý ban đầu.