|  |
| --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM |
| BÁO CÁO MÔN HỌC  **PHÁT TRIỂN PHẦN MỀM TRONG THẺ THÔNG MINH** |
| **Đề tài:**  **Xây dựng thẻ cán bộ nhân viên sử dụng Smart Card**    *Sinh viên thực hiện*:  **Lê Phi Hà – CT050215**  **Thiều Duy Quang – CT050242**  **Trần Văn Hiệp – CT050217**  **Đỗ Quang Trung – CT050251**  **Nhóm 5 – L04**  *Người hướng dẫn*:  **GV: Cao Thanh Vinh**  Khoa Công nghệ thông tin – Học viện Kỹ thuật Mật Mã  Hà Nội, 2024 |

# MỤC LỤC

[MỤC LỤC 1](#_Toc187365595)

[LỜI MỞ ĐẦU 3](#_Toc187365596)

[Chương 1 PHÂN TÍCH THIẾT KẾ 6](#_Toc187365597)

[1.1 Đặt vấn đề 6](#_Toc187365598)

[1.2 Mục tiêu đề tài 7](#_Toc187365599)

[1.3 Sơ đồ phân cấp chức năng và biểu đồ Use Case 8](#_Toc187365600)

[1.3.1 Sơ đồ phân cấp chức năng 8](#_Toc187365601)

[1.3.2 Biểu đồ hoạt động 9](#_Toc187365602)

[1.3.2.1 Khi không chứa dữ liệu hoặc khởi tạo lại dữ liệu 9](#_Toc187365603)

[1.3.2.2 Khi không chứa dữ liệu hoặc khởi tạo lại dữ liệu 10](#_Toc187365604)

[1.3.3 Biểu đồ Usecase chi tiết 11](#_Toc187365605)

[1.3.3.1 Chức năng kết nối đến thẻ 11](#_Toc187365606)

[1.3.3.2 Chức năng Đăng nhập/ Đăng xuất 12](#_Toc187365607)

[1.3.3.3 Chức năng Quản lý mã pin 13](#_Toc187365608)

[1.3.3.4 Chức năng thay đổi thông tin 17](#_Toc187365609)

[1.4 Thiết kế cơ sở dữ liệu 18](#_Toc187365610)

[Chương 2 CÀI ĐẶT VÀ THỰC NGHIỆM 19](#_Toc187365611)

[2.1 Thiết lập môi trường 19](#_Toc187365612)

[2.1.1 Phần mềm sử dụng 19](#_Toc187365613)

[2.1.1.1 Netbean IDE 8.0.2 19](#_Toc187365614)

[2.1.1.2 JCIDE 19](#_Toc187365615)

[2.1.1.3 PhpMyadmin 20](#_Toc187365616)

[2.1.1.4 SceneBuilder 20](#_Toc187365617)

[2.1.2 Ngôn ngữ sử dụng: 21](#_Toc187365618)

[2.1.2.1 Java 21](#_Toc187365619)

[2.1.2.2 JavaCard 22](#_Toc187365620)

[2.2 Thiết kế giao diện 23](#_Toc187365621)

[2.2.1 Giao diên đăng nhập 23](#_Toc187365622)

[2.2.2 Giao diện đổi mã PIN 23](#_Toc187365623)

[2.2.3 Giao diện thông tin thẻ và thay đổi thông tin 24](#_Toc187365624)

[2.2.4 Giao diện điểm danh và lịch sử điểm danh 24](#_Toc187365625)

[2.2.5 Giao diện admin mở khóa thẻ 25](#_Toc187365626)

[2.3 Các hàm mã nguồn quan trọng 25](#_Toc187365627)

[2.3.1 Mã hóa 25](#_Toc187365628)

[2.3.2 Pin Manager 26](#_Toc187365629)

[2.3.3 Xử lý ảnh 29](#_Toc187365630)

[2.3.4 Xử lý thông tin 32](#_Toc187365631)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 34](#_Toc187365632)

[1.Các kết quả đạt được 34](#_Toc187365633)

[2. Hạn chế 34](#_Toc187365634)

[3. Hướng phát triển tiếp theo 34](#_Toc187365635)

[TÀI LIỆU THAM KHẢO 35](#_Toc187365636)

# LỜI MỞ ĐẦU

Xã hội ngày càng phát triển, các thiết bị thông minh ra đời ngày càng nhiều. Đặc biệt phải kể đến là những chiếc thẻ thông minh (Smart card). Kèm theo đó là sự phát triển của công nghệ bán dẫn cho phép các nhà sản xuất chip tạo ra những con chip hay thẻ thông minh ngày càng nhỏ gọn cùng với sức mạnh tính toán cao. Tuy nhiên việc có quá nhiều nhà sản xuất chip, công việc phát triển ứng dụng cho thẻ thông minh gặp khó khăn về sự tương thích. Do đó nhu cầu về một nền tảng chung bên trong chip được đặt ra, công nghệ JavaCard được phát triển để phục vụ mục đích này. Với việc tạo ra một môi trường ảo chung trên tất cả các hệ điều hành hỗ trợ JavaCard, công nghệ này đã giúp cho việc phát triển ứng dụng chip trở nên dễ dàng giúp tiết kiệm thời gian nghiên cứu phát triển.

Nhận thấy vấn đề thực tế, ở những nơi công sở hiện nay đang rất thịnh hành hệ thống điểm danh bằng vân tay hoặc nhận diện khuôn mặt. Tuy nhiên, vấn đề chung của các khi vận hành những hệ thống này là về mặt chi phí và tính ổn định khi sử dụng. Dưới sự hướng dẫn của cô **Cao Thanh Vinh** nhóm chúng em đã chọn đề tài “Xây dựng thẻ cán bộ nhân viên sử dụng Smart Card”. Hệ thống điểm danh giờ đây sẽ sử dụng những chiếc thẻ từ và máy đọc thẻ, thông tin của người dùng vẫn sẽ được mã hoá và ẩn vào trong thẻ. Chỉ khi nào biết được mã PIN thì mới có thể xem được những thông tin đấy. Điều này giúp vẫn đảm bảo được tính bảo mật thông tin người dung của hệ thống. Do hạn chế về mặt kiến thức nên không thể tránh khỏi sai sót trong quá trình làm báo cáo, nhóm chúng em rất mong được sự giúp đỡ và góp ý của cô và các bạn.

Chúng em xin chân thành cảm ơn!

**DANH MỤC HÌNH ẢNH**

[Hình 1‑1 Biểu đồ phân cấp chức năng 5](#_Toc187364449)

[Hình 1‑2 Biểu đồ luồng hoạt động thẻ khi chưa có dữ liệu 6](#_Toc187364450)

[Hình 1‑3 Biểu đồ luồng hoạt động của thẻ đã có dữ liệu 7](#_Toc187364451)

[Hình 1‑4 Biểu đồ Usecase chi tiết chức năng kết nối thẻ 9](#_Toc187364452)

[Hình 1‑5 Biểu đồ Usecase chi tiết chức năng Đăng nhập, Đăng xuất 10](#_Toc187364453)

[Hình 1‑6 Biểu đồ Usecase chi tiết chức năng quản lý mã PIN 12](#_Toc187364454)

[Hình 1‑7 Biểu đồ Usecase chi tiết chức năng điểm danh 15](#_Toc187364455)

[Hình 1‑8 Biểu đồ Use Case chi tiết chức thay đổi thông tin 15](#_Toc187364456)

[Hình 1‑9 Bảng databse rsa\_keys 16](#_Toc187364457)

[Hình 2‑1 IDE JCIDE 17](#_Toc187364458)

[Hình 2‑2 Hệ quản trị cơ sở dữ liệu phpMyadmin 18](#_Toc187364459)

[Hình 2‑3 Ngôn ngữ Java 19](#_Toc187364460)

[Hình 2‑4 Hàm Tạo khóa AES 128-bit từ mã PIN 20](#_Toc187364461)

[Hình 2‑5 Mã hóa dữ liệu bằng AES 20](#_Toc187364462)

[Hình 2‑6 Giải mã dữ liệu bằng AES(tiếp) 21](#_Toc187364463)

[Hình 2‑7 Tạo mã PIN mới. 21](#_Toc187364464)

[Hình 2‑8 Xác minh mã PIN 21](#_Toc187364465)

[Hình 2‑9 Thay đổi mã PIN 22](#_Toc187364466)

[Hình 2‑10 Mở khóa PIN bị khóa 22](#_Toc187364467)

[Hình 2‑11 Tạo cặp khóa RSA và gửi public key 23](#_Toc187364468)

[Hình 2‑12 Xứ lý input ảnh phía Applet 24](#_Toc187364469)

[Hình 2‑13 Xứ lý output ảnh phía Applet 24](#_Toc187364470)

[Hình 2‑14 Xứ lý tải lên ảnh phía Netbeans 25](#_Toc187364471)

[Hình 2‑15 Xứ lý tải xuống ảnh phía Netbeans 26](#_Toc187364472)

[Hình 2‑16 Xứ lý input thông tin 27](#_Toc187364473)

[Hình 2‑17 Xứ lý output thông tin 28](#_Toc187364474)

[Hình 2‑18 Xứ lý thay đổi thông tin 28](#_Toc187364475)

**DANH MỤC BẢNG BIỂU**

[Bảng 1.1 Đặc tả Usecase kết nối thẻ 9](#_Toc187364838)

[Bảng 1.2 Đặc tả Usecase Đăng nhập 11](#_Toc187364839)

[Bảng 1.3 Đặc tả Usecase Đăng xuất 11](#_Toc187364840)

[Bảng 1.4 Đặc tả Usecase khởi tạo mã PIN 12](#_Toc187364841)

[Bảng 1.5 Đặc tả Usecase khóa mã PIN 13](#_Toc187364842)

[Bảng 1.6 Đặc tả Usecase mở khóa mã PIN 13](#_Toc187364843)

[Bảng 1.7 Đặc tả Usecase thay đổi mã PIN 14](#_Toc187364844)

[Bảng 1.8 Đặc tả Usecase điểm danh 15](#_Toc187364845)

[Bảng 1.9 Đặc tả Usecase thay đổi thông tin 16](#_Toc187364846)

# PHÂN TÍCH THIẾT KẾ

**Thẻ thông minh (Smartcard Java)** là một loại thẻ tích hợp vi mạch (chip) với khả năng lưu trữ và xử lý dữ liệu. Nó thường được sử dụng trong các lĩnh vực yêu cầu tính bảo mật cao như thanh toán, quản lý danh tính, y tế, và giao thông. Một **Java Smartcard** là thẻ thông minh được lập trình và hoạt động dựa trên nền tảng **Java Card** – một công nghệ của Oracle cho phép chạy các ứng dụng (applet) Java trên thẻ thông minh.

## Đặt vấn đề

Hệ thống điểm danh nhân viên là một phần quan trọng trong việc quản lý nhân sự tại các doanh nghiệp, giúp theo dõi và ghi nhận thời gian làm việc của nhân viên một cách chính xác và hiệu quả. Tuy nhiên, nhiều doanh nghiệp hiện nay vẫn sử dụng phương pháp điểm danh thủ công hoặc hệ thống truyền thống không tối ưu, dẫn đến tình trạng thiếu chính xác, mất thời gian, và khó khăn trong việc quản lý dữ liệu.

Việc xây dựng và triển khai các giải pháp công nghệ hiện đại, chẳng hạn như thẻ thông minh (smartcard), có khả năng tương thích cao với các hệ thống điểm danh tự động là một yêu cầu cấp thiết. Trong bối cảnh này, việc phát triển thẻ smartcard dựa trên nền tảng Java cho hệ thống điểm danh nhân viên không chỉ giúp nâng cao hiệu quả vận hành mà còn đảm bảo tính bảo mật, độ tin cậy, và khả năng mở rộng của hệ thống.

Sử dụng Java Card, một nền tảng phổ biến trong việc phát triển các ứng dụng trên thẻ thông minh, có thể mang lại nhiều ưu điểm vượt trội như khả năng lập trình linh hoạt, tính năng mã hóa mạnh mẽ và khả năng tương thích với các tiêu chuẩn quốc tế. Tuy nhiên, việc xây dựng và triển khai hệ thống này đòi hỏi một quy trình nghiên cứu bài bản để đảm bảo rằng giải pháp đưa ra phù hợp với nhu cầu thực tiễn, có khả năng tích hợp dễ dàng với cơ sở hạ tầng hiện có và đáp ứng các yêu cầu về tính năng cũng như bảo mật.

Với lý do trên, đề tài nghiên cứu về việc xây dựng thẻ smartcard Java cho hệ thống điểm danh nhân viên mang tính ứng dụng cao, không chỉ góp phần nâng cao hiệu quả quản lý nhân sự mà còn mở ra hướng phát triển mới trong việc áp dụng công nghệ thông minh vào các quy trình quản lý nhân sự tại Việt Nam.

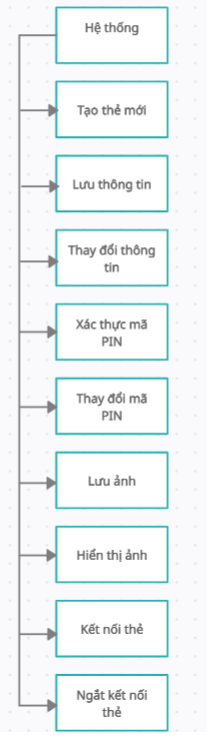
## Mục tiêu đề tài

Đề tài hướng tới việc xây dựng và triển khai giải pháp thẻ thông minh (smartcard) dựa trên nền tảng Java để ứng dụng trong hệ thống điểm danh nhân viên tự động, góp phần hiện đại hóa việc quản lý nhân sự, nâng cao hiệu quả vận hành và đảm bảo tính bảo mật, chính xác của hệ thống. Đặc biệt, thẻ smartcard sẽ được thiết kế với các chức năng chính bao gồm lưu trữ thông tin nhân viên, mã hóa và bảo mật dữ liệu, tích hợp khả năng ghi nhận thời gian điểm danh tự động, và cung cấp tiện ích tra cứu thông tin lịch sử điểm danh một cách dễ dàng và minh bạch.

Để đạt được mục tiêu trên, nghiên cứu sẽ tập trung vào phân tích các yêu cầu kỹ thuật và tính năng cần thiết của hệ thống, thiết kế và phát triển phần mềm cho thẻ smartcard trên nền tảng Java Card, đồng thời tích hợp các chức năng như điểm danh tự động, quản lý giờ làm việc, và truy xuất thông tin lịch sử điểm danh qua các thiết bị đọc thẻ. Sau khi phát triển, thẻ sẽ được kiểm thử và đánh giá hiệu năng dựa trên các tiêu chí về tốc độ xử lý, độ tin cậy và bảo mật dữ liệu. Cuối cùng, đề tài cũng sẽ đề xuất phương án triển khai hệ thống thẻ thông minh, tích hợp vào các quy trình quản lý nhân sự hiện có tại các doanh nghiệp, cùng các kế hoạch bảo trì và nâng cấp nhằm đảm bảo tính hiệu quả và khả năng mở rộng trong tương lai.

## Sơ đồ phân cấp chức năng và biểu đồ Use Case

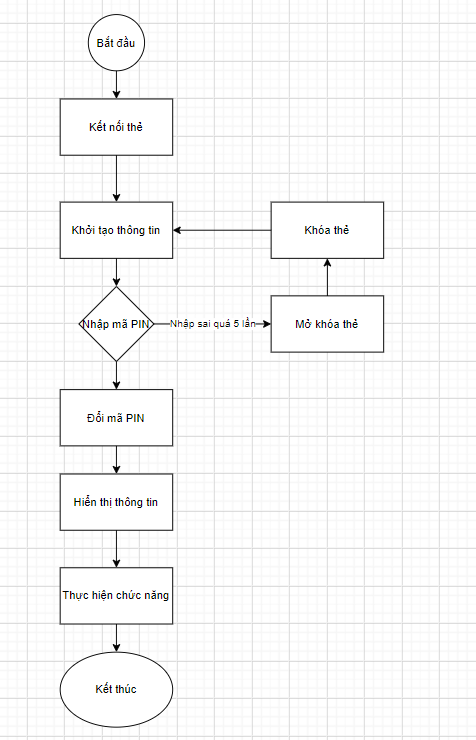
### Sơ đồ phân cấp chức năng



Hình 1‑1 Biểu đồ phân cấp chức năng

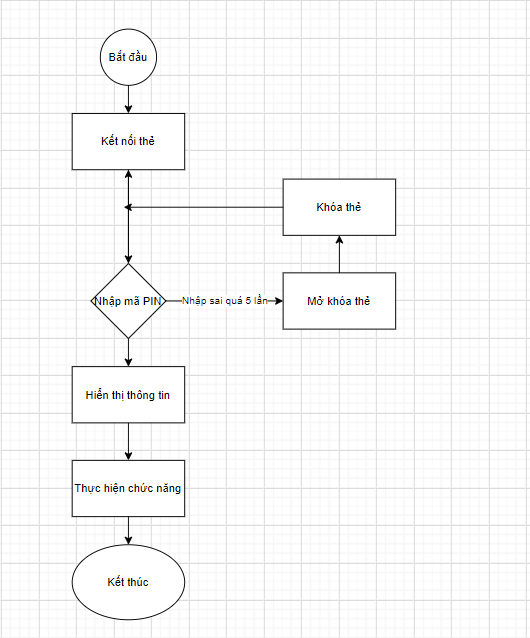
### Biểu đồ hoạt động

#### Khi không chứa dữ liệu hoặc khởi tạo lại dữ liệu



Hình 1‑2 Biểu đồ luồng hoạt động thẻ khi chưa có dữ liệu

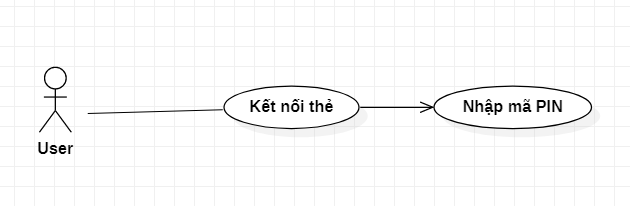
#### Khi không chứa dữ liệu hoặc khởi tạo lại dữ liệu



Hình 1‑3 Biểu đồ luồng hoạt động của thẻ đã có dữ liệu

### Biểu đồ Usecase chi tiết

#### Chức năng kết nối đến thẻ



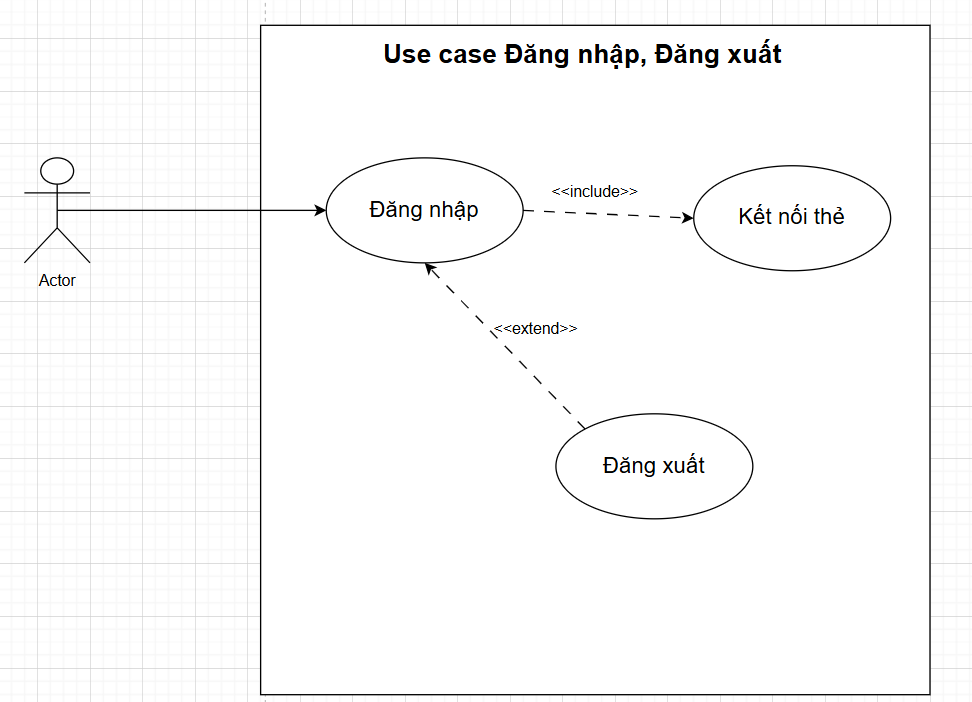
Hình 1‑4 Biểu đồ Usecase chi tiết chức năng kết nối thẻ

|  |  |
| --- | --- |
| **Use Case** | Đăng nhập |
| **Actor** | Nhân viên |
| **Brief Description** | Chức năng đăng nhập hệ thống |
| **Pre-conditions** | Kết nối thẻ |
| **Basic Flows** | Use case bắt đầu khi người dùng muốn kết nối thẻ với applet:   1. Kết nối thẻ 2. Nhập mã PIN 3. Đăng nhập |
| **Alternative Flows** | 1.1. Hiển thị thông báo “Đăng nhập thành công” |
| **Post-Conditions** |  |

Bảng 1.1 Đặc tả Usecase kết nối thẻ

#### Chức năng Đăng nhập/ Đăng xuất

a, Use case



Hình 1‑5 Biểu đồ Usecase chi tiết chức năng Đăng nhập, Đăng xuất

b, Đặc tả

|  |  |
| --- | --- |
| **Use case** | Đăng nhập |
| **Actor** | Người dùng |
| **Brief Description** | Đăng nhập vào hệ thống |
| **Pre-conditions** | Kết nối thẻ thành công |
| **Basic Flows** | Use case bắt đầu khi người dùng muốn đăng nhập với applet:   1. Người dùng nhập mã pin để kết nối thẻ: 2. Sau khi người dùng kết nối thẻ thành công thì ấn nút button để đăng nhập vào hệ thống |
| **Exception Flows** | 2.1. Nếu người dùng không kết nối thẻ thì sẽ không đăng nhập được. |
| **Post-conditions** | Nếu use case này được thực hiện thành công, người dùng sẽ đăng nhập vào hệ thống thành công |
| **Special Requirements** |  |

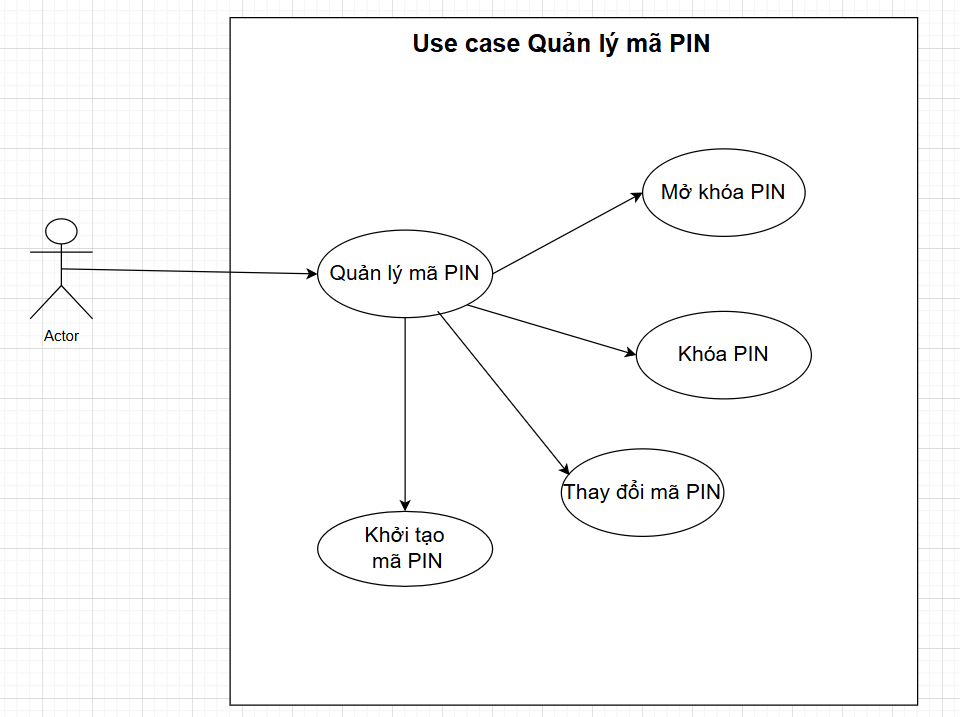
Bảng 1.2 Đặc tả Usecase Đăng nhập

|  |  |
| --- | --- |
| **Use case** | Đăng xuất |
| **Actor** | Người dùng |
| **Brief Description** | Đăng xuất ra khỏi hệ thống |
| **Pre-conditions** | Đăng nhập vào hệ thống thành công |
| **Basic Flows** | Use case bắt đầu khi người dùng muốn đăng xuất ra khỏi hệ thống:   1. Người dùng đã phải đăng nhập vào từ trước 2. Người dùng ấn nút button đăng xuất để thoát chương trình. |
| **Exception Flows** | 2.1. Nếu người dùng không đăng nhập thì sẽ không đăng nhập được. |
| **Post-conditions** | Nếu use case này được thực hiện thành công, người dùng sẽ đăng xuất ra khỏi hệ thống thành công. |
| **Special Requirements** |  |

Bảng 1.3 Đặc tả Usecase Đăng xuất

#### Chức năng Quản lý mã pin

a, Use case



Hình 1‑6 Biểu đồ Usecase chi tiết chức năng quản lý mã PIN

b, Đặc tả

|  |  |
| --- | --- |
| **Use case** | Khởi tạo mã PIN |
| **Actor** | Người dùng |
| **Brief Description** | Khởi tạo mã PIN mới cho tài khoản người dùng |
| **Pre-conditions** | Đăng nhập thẻ thành công lần đầu tiên. |
| **Basic Flows** | Use case bắt đầu khi người dùng đăng nhập thành công lần đầu tiên:   1. Người dùng đăng nhập thành công 2. Sau khi người dùng kết nối thẻ thành công thì ấn button khởi tạo thông tin. 3. Người dùng nhập trường thông tin khởi tạo mã PIN. |
| **Exception Flows** | 3.1. Nếu người dùng nhập mã PIN trùng với PIN mặc định thì sẽ trả về thông báo không được trùng mã PIN mặc định  3.2 Nếu mã PIN khởi tạo có độ dài khác từ 6-10 thì sẽ trả về thông báo yêu cầu nhập độ dài mã PIN phù hợp. |
| **Post-conditions** | Nếu use case này được thực hiện thành công, thông tin mã PIN của người dùng sẽ được lưu trên csdl. |

Bảng 1.4 Đặc tả Usecase khởi tạo mã PIN

|  |  |
| --- | --- |
| **Use case** | Khóa PIN |
| **Actor** | Người dùng |
| **Brief Description** | Khóa mã PIN của người sử dụng |
| **Pre-conditions** | Người dùng nhập sai mã PIN quá 3 lần. |
| **Basic Flows** | Use case bắt đầu khi người dùng nhập sai mã PIN quá 3 lần :   * 1. Khi chưa khởi tạo mã PIN   2. Nhập sai mã PIN mặc định quá 3 lần => khóa thẻ   2.1 Khi đã khởi tạo mã PIN  2.2 Sử dụng chức năng bất kì có liên quan đến xác thực mã PIN mà nhập sai 3 lần => khóa thẻ |
| **Exception Flows** |  |
| **Post-conditions** | Nếu use case này được thực hiện thành công, thẻ sẽ bị khóa và không thể xác thực mã PIN nữa. |
| **Special Requirements** |  |

Bảng 1.5 Đặc tả Usecase khóa mã PIN

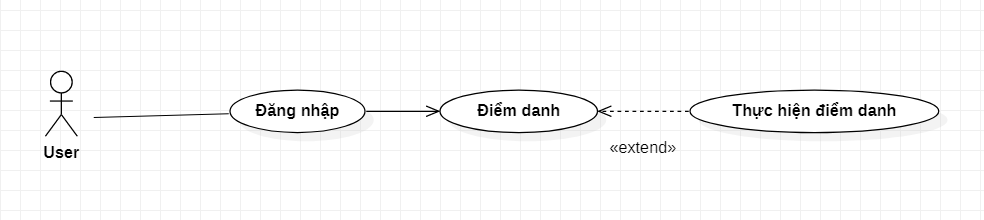
|  |  |
| --- | --- |
| **Use case** | Mở khóa PIN |
| **Actor** | Người dùng |
| **Brief Description** | Mở khóa mã PIN người dùng khi người dùng bị khóa thẻ |
| **Pre-conditions** | Mã PIN đang trong tình trạng bị khóa thẻ. |
| **Basic Flows** | Use case bắt đầu khi người dùng muốn mở khóa mã PIN:   1. Người dùng ấn nút button Mở khóa thẻ |
| **Exception Flows** |  |
| **Post-conditions** | Nếu use case này được thực hiện thành công, thẻ sẽ được mở khóa mã PIN sẽ mở khóa. |
| **Special Requirements** |  |

Bảng 1.6 Đặc tả Usecase mở khóa mã PIN

|  |  |
| --- | --- |
| **Use case** | Thay đổi mã PIN |
| **Actor** | Người dùng |
| **Brief Description** | Thay đổi mã PIN tài khoản người dùng |
| **Pre-conditions** | Mã PIN đã được khởi tạo từ trước |
| **Basic Flows** | Use case bắt đầu khi người dùng muốn thay đổi mã PIN:   1. Người dùng đăng nhập thành công 2. Sau khi người dùng kết nối thẻ thành công thì ấn button thay đổi mã PIN 3. Người dùng nhập mã PIN muốn thay đổi   3.1 Người dùng nhập cả mã PIN cũ để xác thực thông tin  3.2 Mã PIN mới được nhập 2 lần phải đảm bảo đúng độ dài và trùng khớp nhau   1. Ấn xác nhận để thay đổi mã PIN |
| **Exception Flows** | Nếu người nhập thiếu thông tin, PIN mới trùng PIN cũ hoặc nhập sai mã PIN cũ xác thực thì sẽ không cho phép thay đổi mã PIN. |
| **Post-conditions** | Nếu use case này được thực hiện thành công, thông tin mã PIN của người dùng sẽ được thay đổi. |
| **Special Requirements** |  |

Bảng 1.7 Đặc tả Usecase thay đổi mã PIN

Chức năng điểm danh

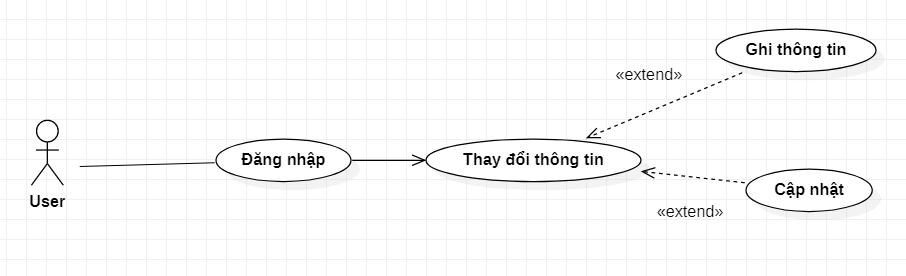


Hình 1‑7 Biểu đồ Usecase chi tiết chức năng điểm danh

|  |  |
| --- | --- |
| **Use Case** | Điểm danh |
| **Actor** | Nhân viên |
| **Brief Description** | Chức năng điểm danh |
| **Pre-conditions** | Đăng nhập thành công |
| **Basic Flows** | 1. Chọn chức năng “điểm danh” 2. Thực hiện điểm danh |
| **Alternative Flows** | 1.1. Hiển thị thông báo “Đã điểm danh thành công” |
| **Post-Conditions** |  |

Bảng 1.8 Đặc tả Usecase điểm danh

#### Chức năng thay đổi thông tin



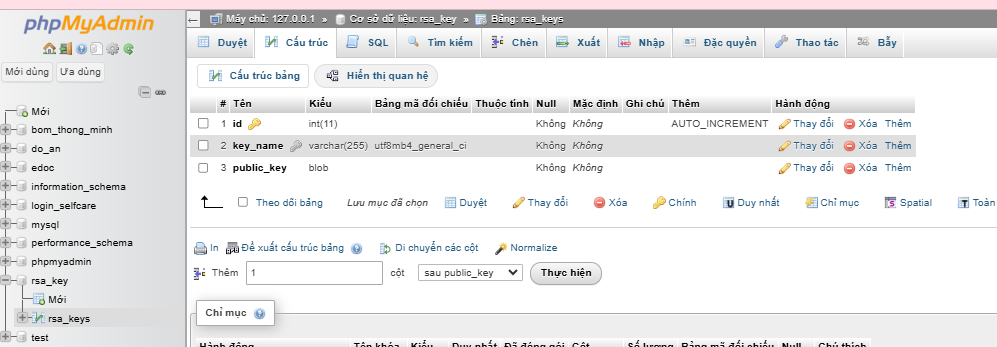
Hình 1‑8 Biểu đồ Use Case chi tiết chức thay đổi thông tin

|  |  |
| --- | --- |
| **Use Case** | Thay đổi thông tin |
| **Actor** | Nhân viên |
| **Brief Description** | Chức năng thay đổi thông tin |
| **Pre-conditions** | Đăng nhập thành công |
| **Basic Flows** | 1. Chọn thay đổi thông tin 2. Nhập thông tin 3. Cập nhật |
| **Alternative Flows** | 1.1. Hiển thị thông báo “Thay đổi thành công” |
| **Post-Conditions** |  |

Bảng 1.9 Đặc tả Usecase thay đổi thông tin

## Thiết kế cơ sở dữ liệu

Ở đề tài này chúng em nghiên cứu và xây dựng 1 bảng database liên quan đến dữ liệu quan trọng của ký dữ liệu điểm danh là bảng *rsa\_keys*



Hình 1‑9 Bảng databse rsa\_keys

# CÀI ĐẶT VÀ THỰC NGHIỆM

## Thiết lập môi trường

### Phần mềm sử dụng

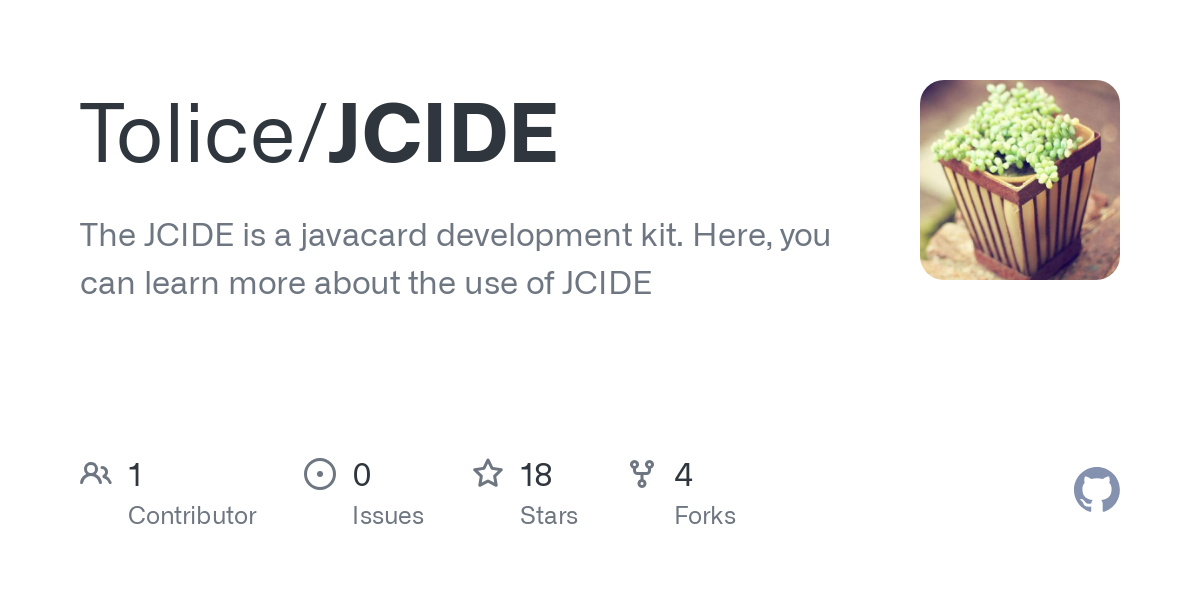
#### Netbean IDE 8.0.2

NetBeans IDE là một công cụ hỗ trợ lập trình viết mã code miễn phí được cho là tốt nhất hiện nay, được sử dụng chủ yếu cho các lập trình viên phát triển Java tuy nhiên phần mềm có dung lượng khá là nặng dành cho các máy cấu hình có RAM, CPU tương đối cao để vận hành.

NetBeans IDE là môi trường phát triển tích hợp và cực kỳ cần thiết cho các lập trình viên, công cụ này có thể hoạt động tốt với rất nhiều nền tảng hệ điều hành khác nhau như Linux, Windows, MacOS, ... là một mã nguồn mở cung cấp các tính năng cần thiết nhất nhằm tạo ra các ứng dụng web, thiết bị di động, desktop.

#### JCIDE

Bộ công cụ phát triển thẻ Java (JCKit) bao gồm JCIDE và pyApdutool, cung cấp một môi trường phát triển hoàn chỉnh, mạnh mẽ cho các nhà phát triển thẻ java. Nó cho phép bạn phát triển Java Card Applet một cách nhanh chóng và dễ dàng. Trang này chứa các liên kết để tải xuống Bộ phát triển thẻ Java. Hơn nữa, công cụ này hoàn toàn miễn phí.



Hình 2‑1 IDE JCIDE

#### PhpMyadmin

PhpMyadmin là một công cụ quản trị cơ sở dữ liệu mã nguồn mở, được viết bằng ngôn ngữ PHP, giúp người dùng dễ dàng quản lý các cơ sở dữ liệu MySQL hoặc MariaDB thông qua giao diện web. phpMyAdmin sử dụng ngôn ngữ SQL (Structured Query Language) để quản lý và truy vấn dữ liệu, đồng thời cung cấp một giao diện đồ họa trực quan cho phép người dùng thao tác với cơ sở dữ liệu mà không cần phải viết mã SQL trực tiếp.

Được phát triển bởi một cộng đồng mã nguồn mở, phpMyAdmin hỗ trợ nhiều tính năng hữu ích như tạo và quản lý cơ sở dữ liệu, bảng, cột, chỉ mục và thực hiện các truy vấn SQL. Công cụ này đặc biệt phù hợp với người dùng không chuyên về lập trình, giúp đơn giản hóa quá trình quản lý cơ sở dữ liệu.

PhpMyadmin có thể hoạt động trên các hệ điều hành phổ biến như Windows, macOS và Linux, và hỗ trợ nhiều ngôn ngữ giao diện khác nhau. Công cụ này là lựa chọn phổ biến trong các môi trường phát triển web, đặc biệt là với các nền tảng như WordPress, Drupal, và Joomla, giúp việc quản lý cơ sở dữ liệu trở nên thuận tiện và dễ dàng. Ngoài ra, phpMyAdmin cũng tích hợp tốt với các ngôn ngữ lập trình như PHP, Python, và Java, mang đến sự linh hoạt và hiệu quả trong việc phát triển các ứng dụng và hệ thống quản lý dữ liệu.



Hình 2‑2 Hệ quản trị cơ sở dữ liệu phpMyadmin

#### SceneBuilder

SceneBuilder là một công cụ trực quan hỗ trợ thiết kế giao diện đồ họa (GUI) cho các ứng dụng JavaFX. Công cụ này cho phép người dùng xây dựng giao diện bằng phương pháp kéo thả các thành phần giao diện trực tiếp, từ đó tự động sinh mã FXML. Nhờ đó, SceneBuilder giúp giảm thiểu thời gian viết mã thủ công, tăng hiệu quả làm việc, và hỗ trợ chỉnh sửa các thuộc tính giao diện dễ dàng. Đây là một công cụ mạnh mẽ, thân thiện với người dùng, đóng vai trò quan trọng trong việc phát triển các ứng dụng JavaFX chuyên nghiệp.

### Ngôn ngữ sử dụng:

#### Java

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class), ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995. Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.



Hình 2‑3 Ngôn ngữ Java

Java là một ngôn ngữ lập trình hướng đối tượng vì vậy nó cũng có 4 đặc điểm chung sau đây:

* Tính trừu tượng (Abstraction): là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.
* Tính đa hình (Polymorphism): cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.
* Tính kế thừa (Inheritance): điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.
* Tính đóng gói (Encapsulation): là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.

#### JavaCard

JavaCard là một phiên bản tối giản của ngôn ngữ lập trình Java, được thiết kế đặc biệt để phát triển các ứng dụng chạy trên thẻ thông minh (smart cards) và các thiết bị có tài nguyên hạn chế. JavaCard cho phép thực thi các ứng dụng Java trên các thiết bị nhỏ gọn này, tận dụng tính di động, bảo mật và tính năng hướng đối tượng của Java.

Trong JavaCard, các ứng dụng được gọi là applet, và chúng được viết bằng Java với các API chuyên dụng. Tuy nhiên, để phù hợp với môi trường hạn chế, JavaCard chỉ hỗ trợ một tập con của ngôn ngữ Java, loại bỏ các tính năng không cần thiết như đa luồng, garbage collection tự động, và thư viện tiêu chuẩn lớn

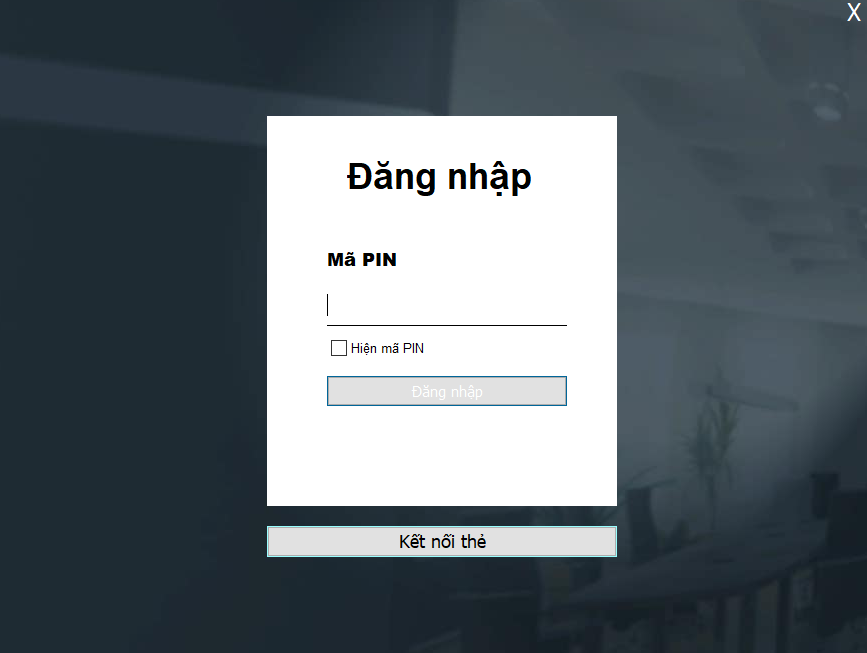


Ngôn ngữ JavaCard

JavaCard thường được sử dụng trong các lĩnh vực như phát triển thẻ SIM, thẻ tín dụng, thẻ căn cước điện tử, và các thiết bị bảo mật nhúng. Điểm mạnh của JavaCard nằm ở khả năng bảo mật cao, tính di động của mã nguồn (write once, run anywhere), và khả năng tương thích với nhiều loại thiết bị.

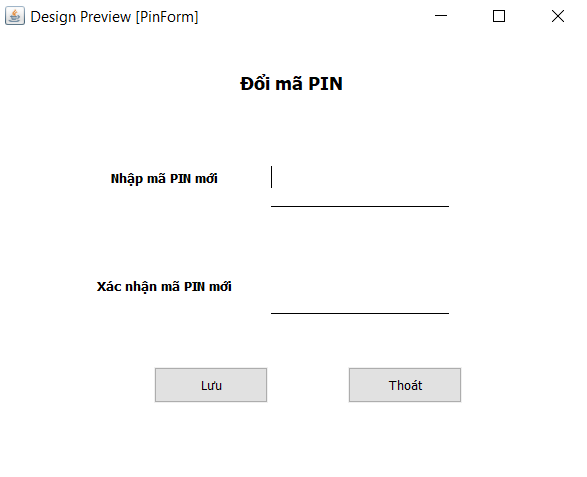
## Thiết kế giao diện

### Giao diên đăng nhập



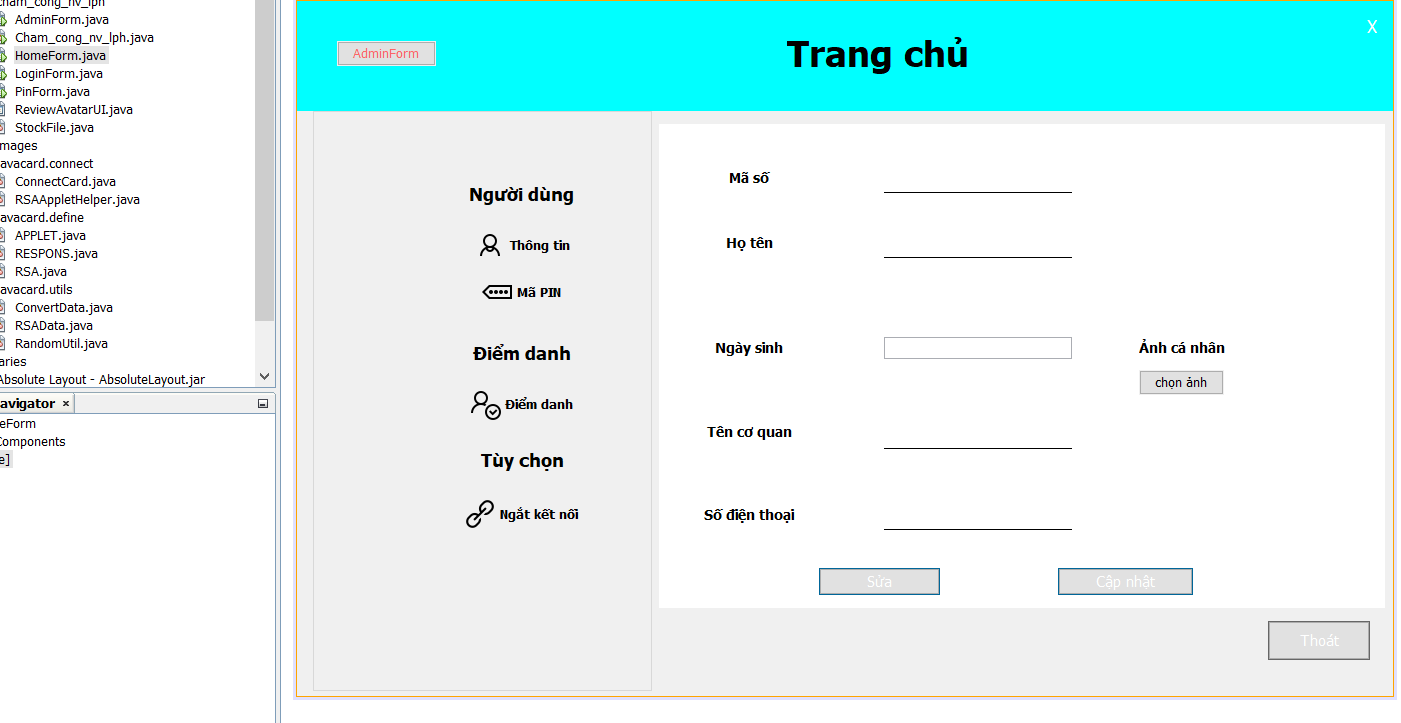
Hình ảnh giao diện đăng nhập

### Giao diện đổi mã PIN



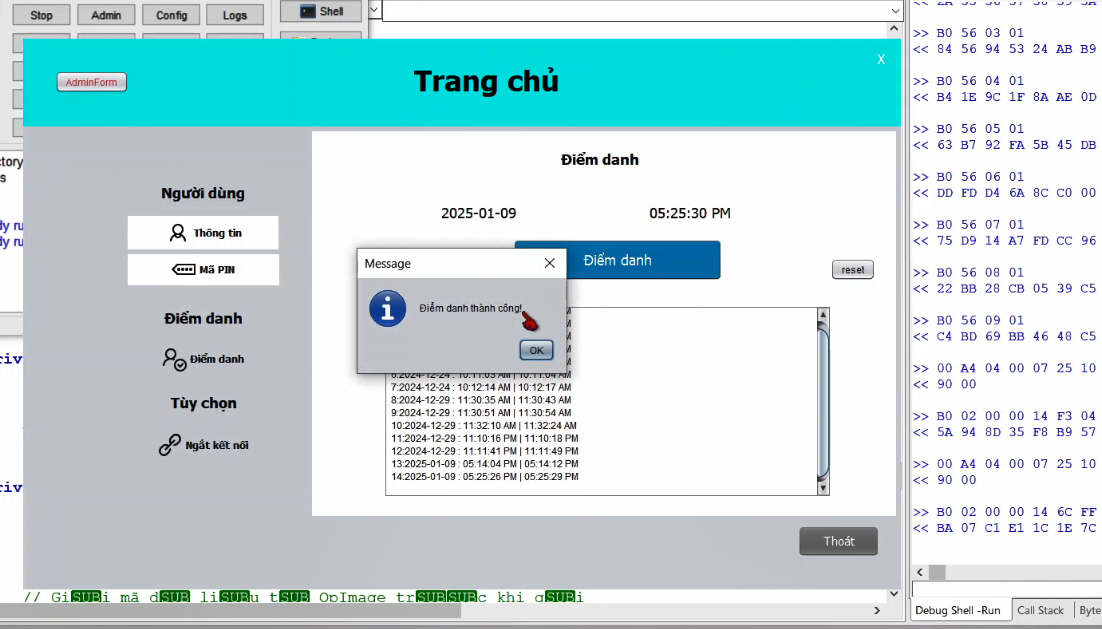
Giao diện đổi mã PIN

### Giao diện thông tin thẻ và thay đổi thông tin



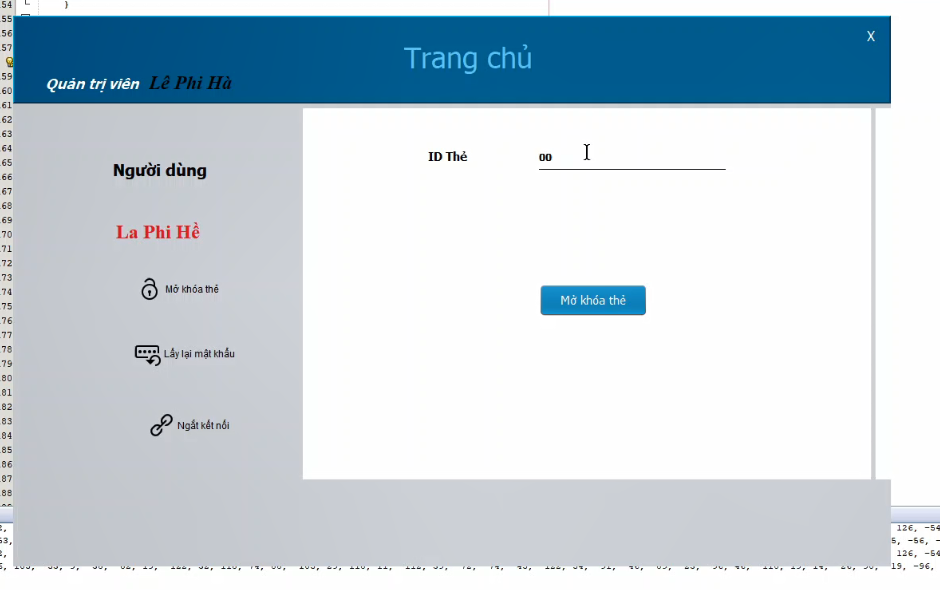
Giao diện thông tin thẻ và thay đổi thông tin

### Giao diện điểm danh và lịch sử điểm danh



Giao diện điểm danh và lịch sử điểm danh

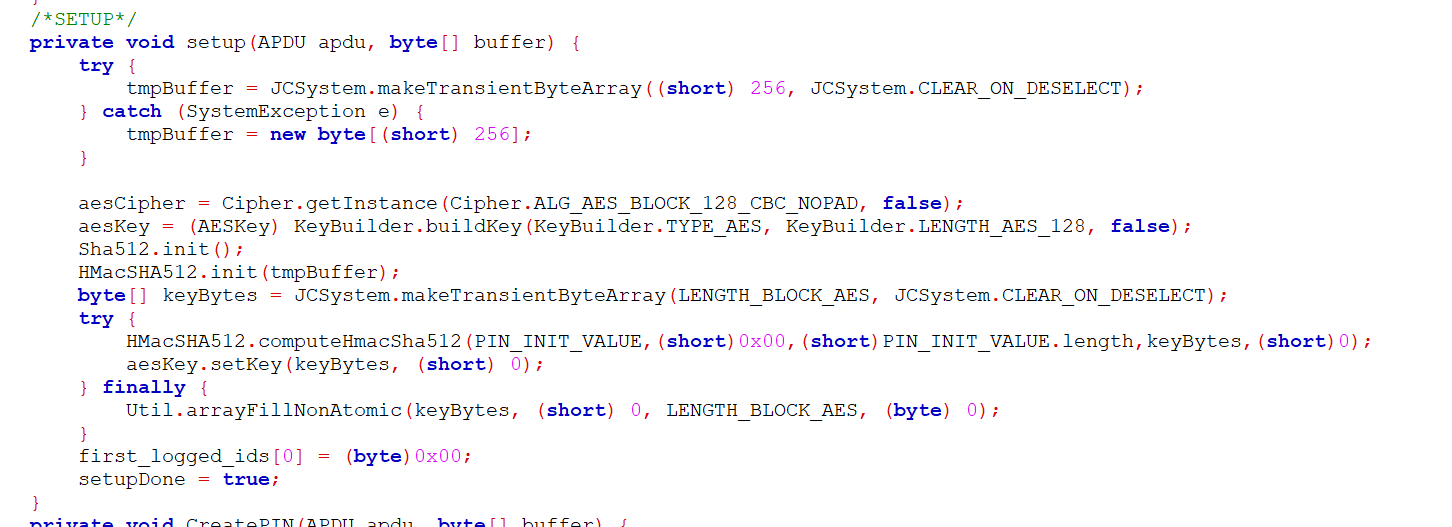
### Giao diện admin mở khóa thẻ



Giao diện mở khóa thẻ

## Các hàm mã nguồn quan trọng

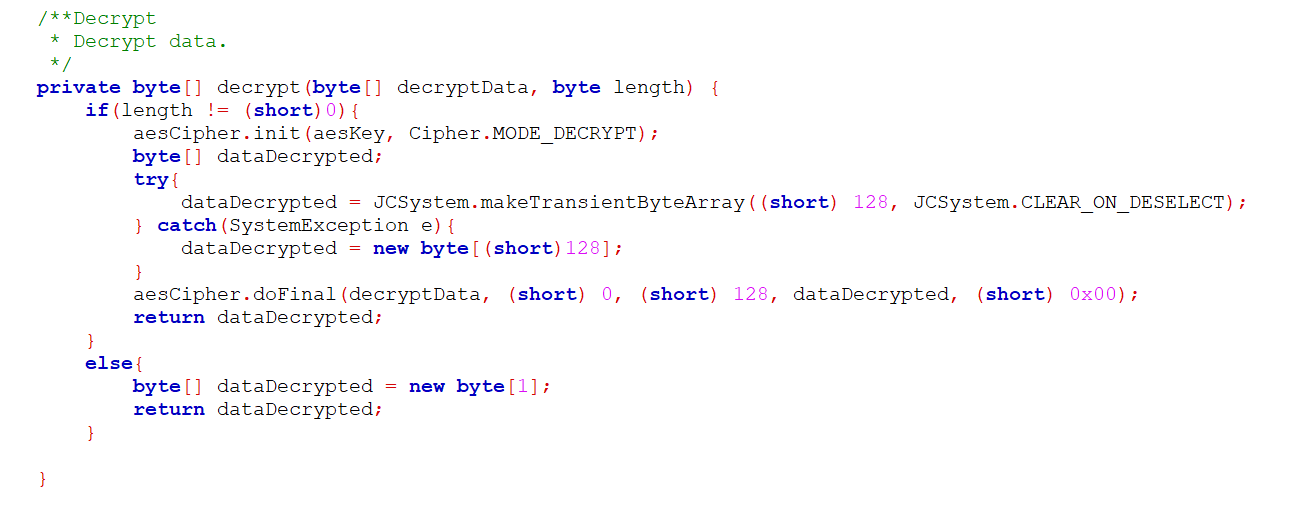
### Mã hóa



Hình 2‑4 Hàm Tạo khóa AES 128-bit từ mã PIN

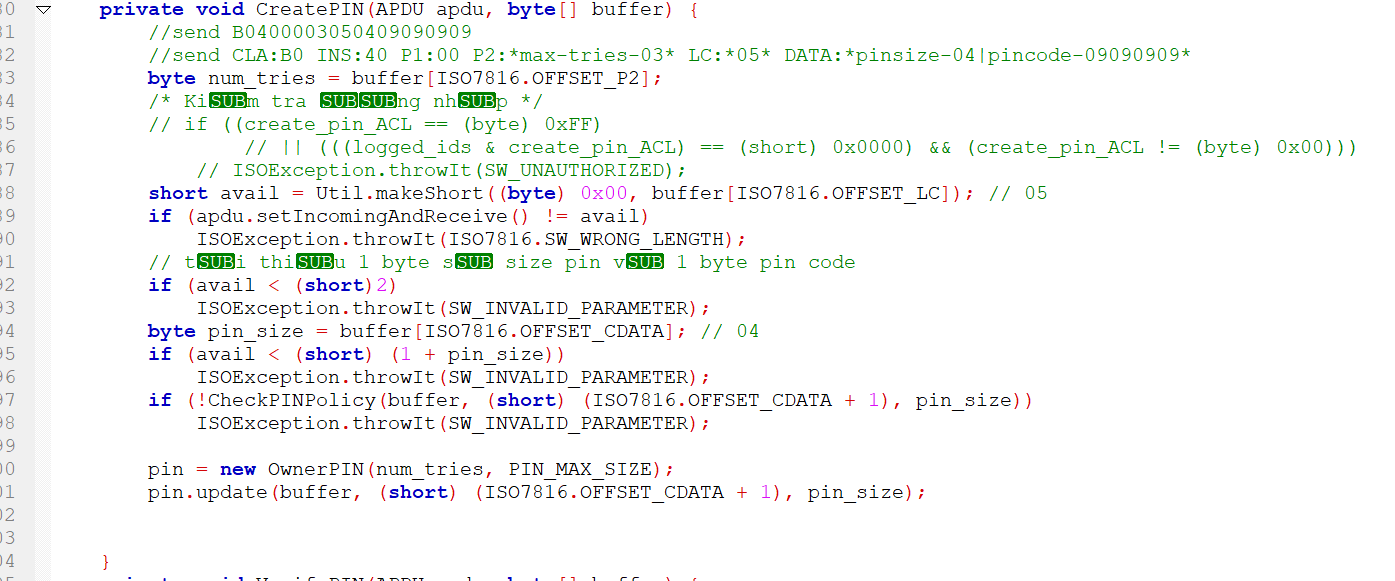


Hình 2‑5 Mã hóa dữ liệu bằng AES

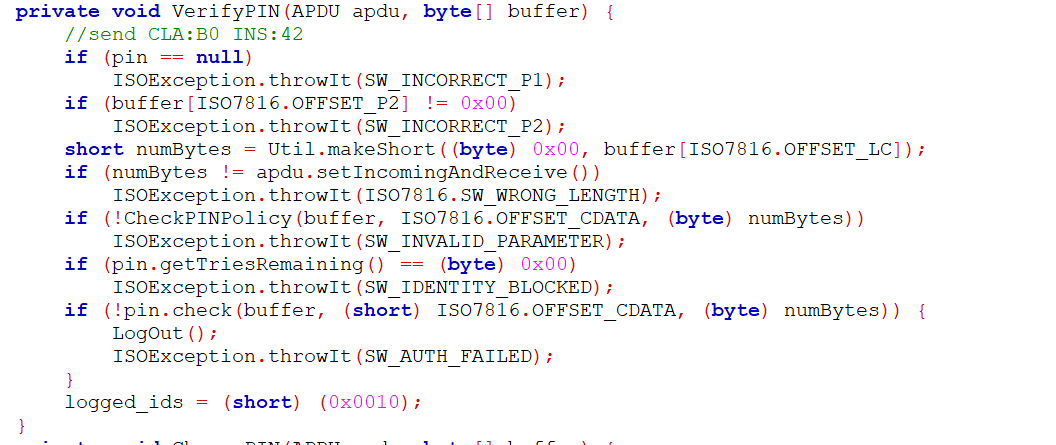


Hình 2‑6 Giải mã dữ liệu bằng AES(tiếp)

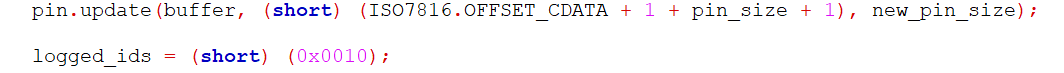
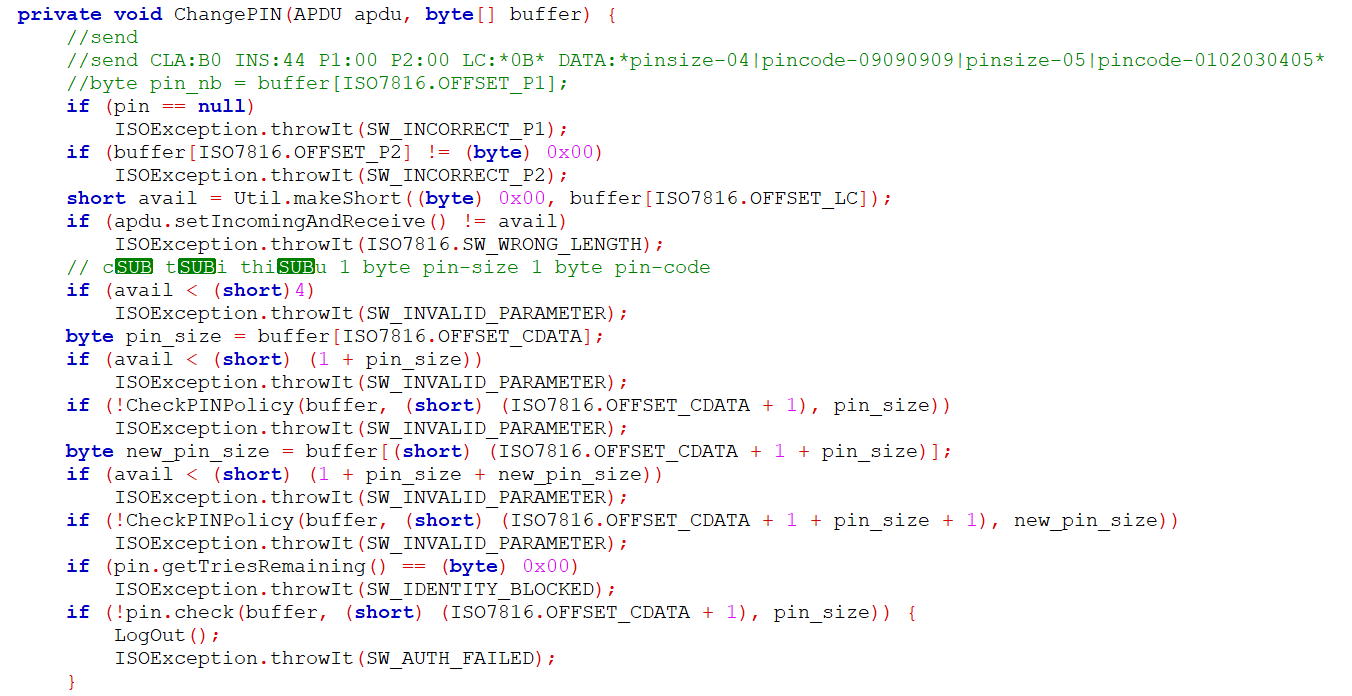
### Pin Manager



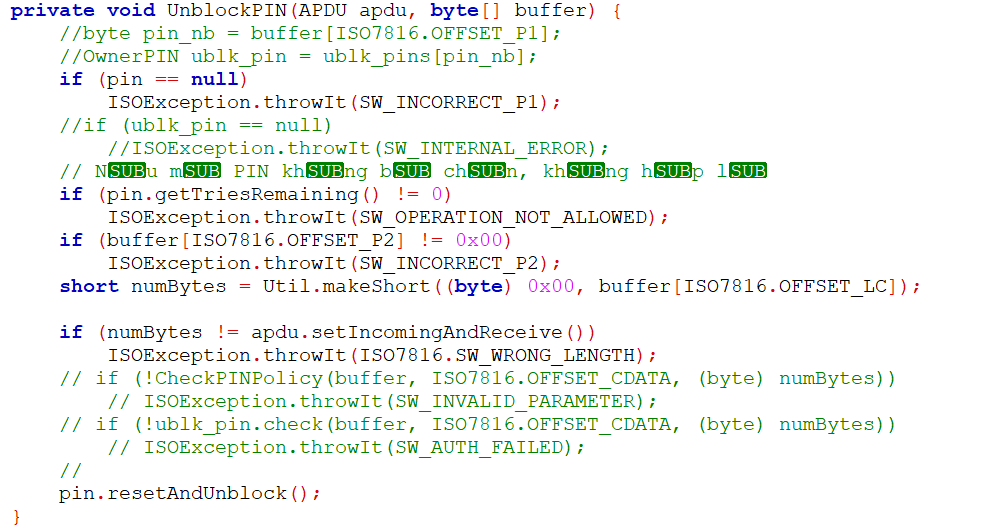
Hình 2‑7 Tạo mã PIN mới.



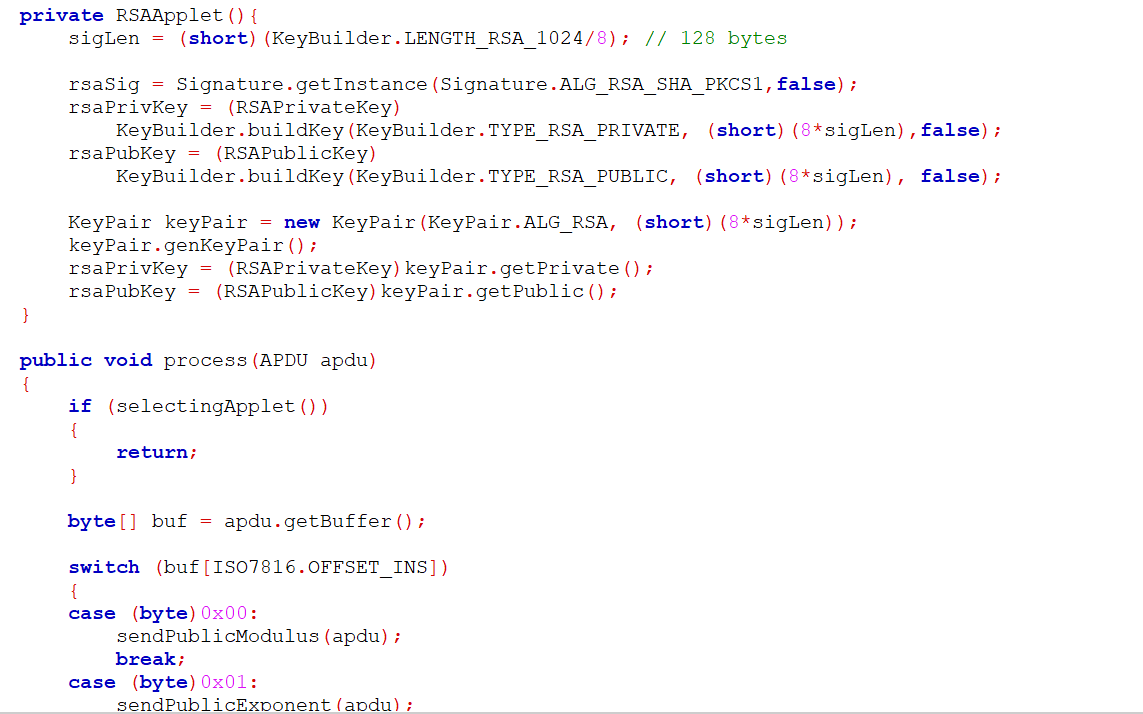
Hình 2‑8 Xác minh mã PIN

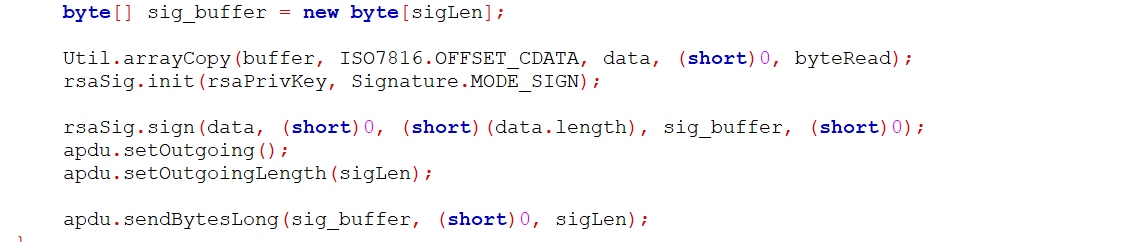


Hình 2‑9 Thay đổi mã PIN



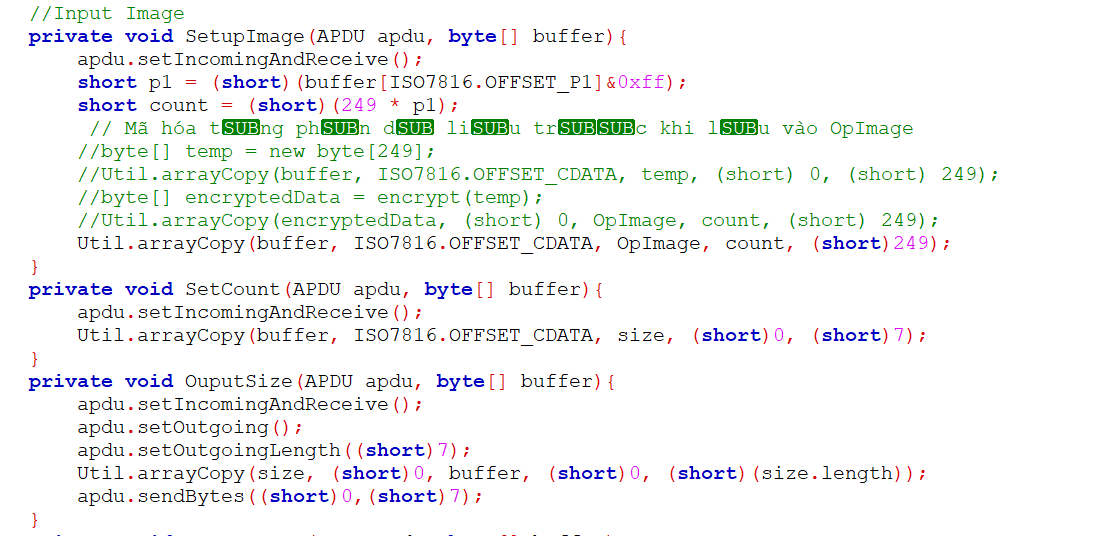
Hình 2‑10 Mở khóa PIN bị khóa



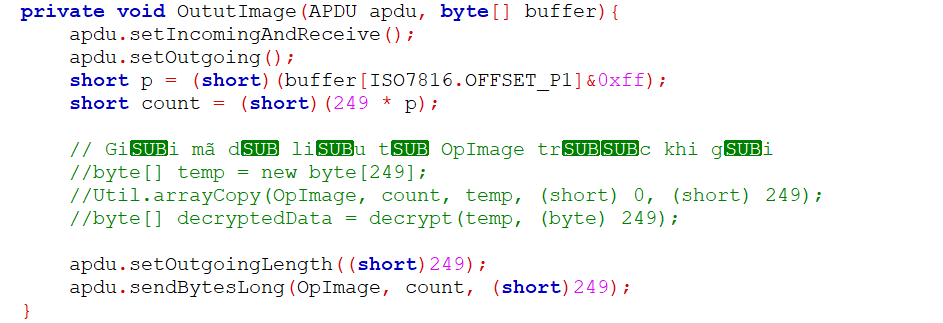
 

Hình 2‑11 Tạo cặp khóa RSA và gửi public key

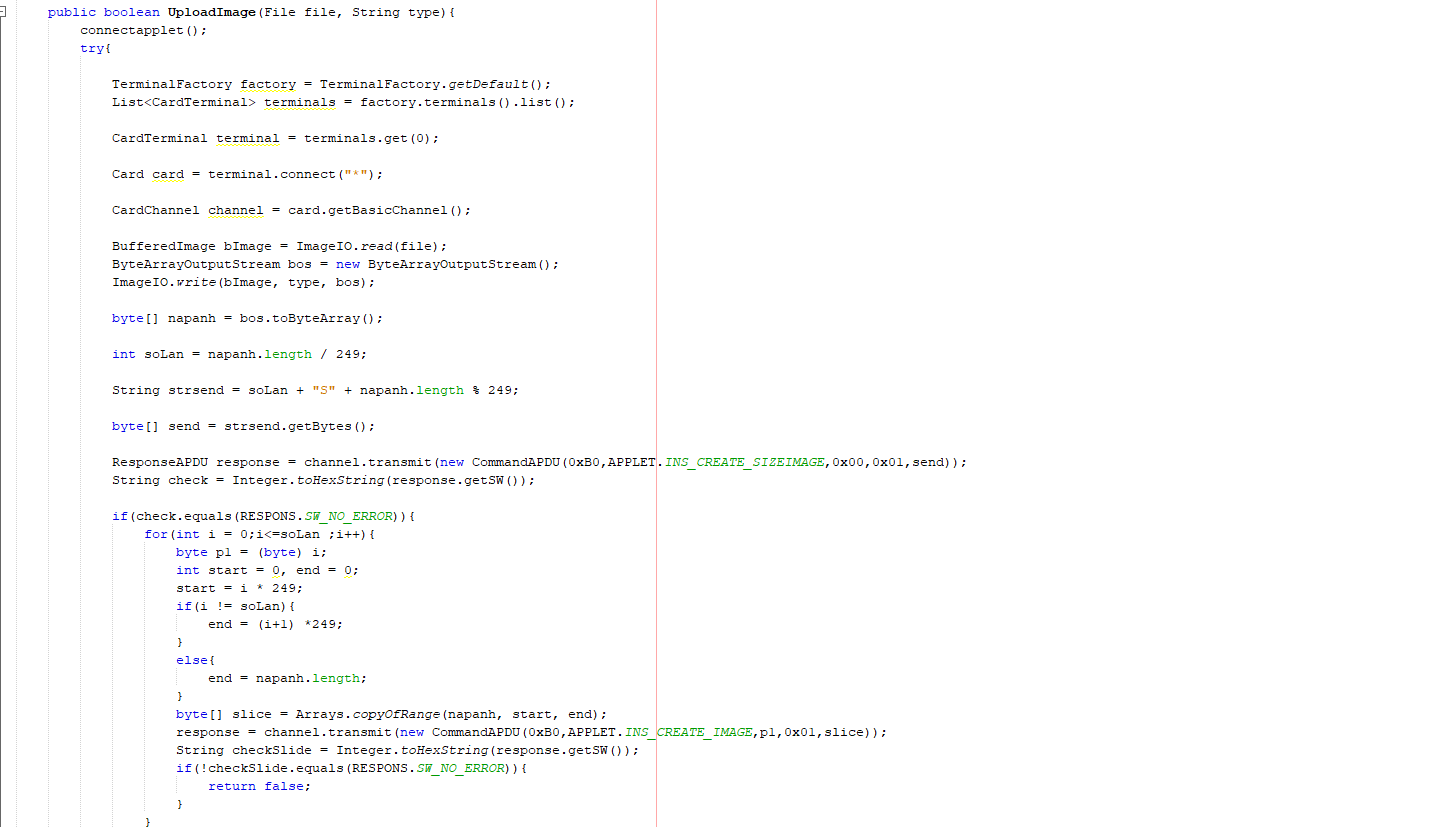
### Xử lý ảnh



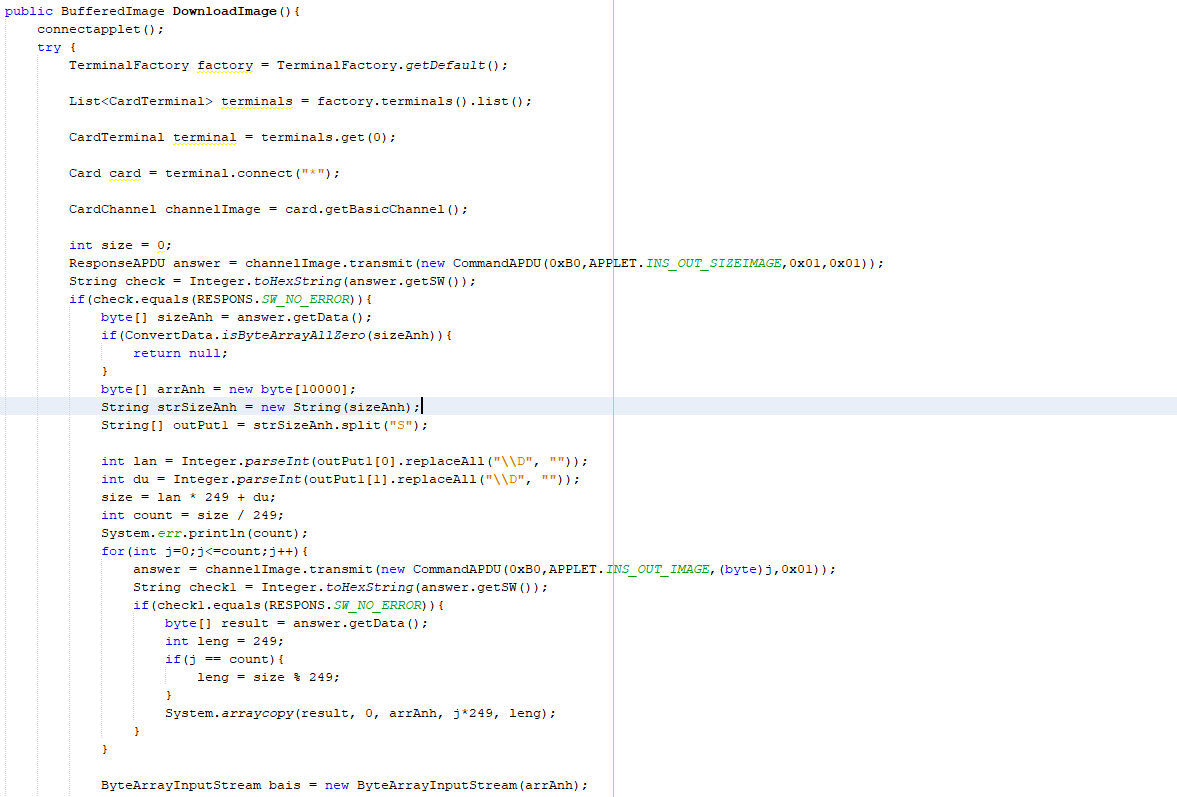
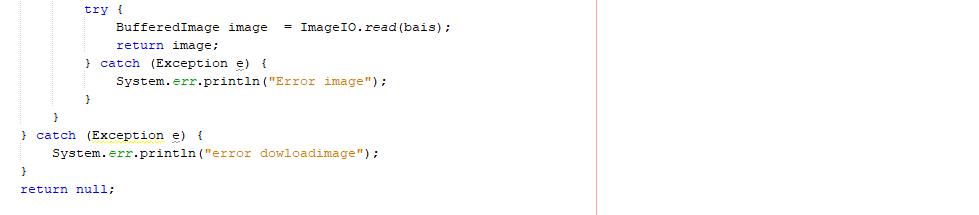
Hình 2‑12 Xứ lý input ảnh phía Applet



Hình 2‑13 Xứ lý output ảnh phía Applet

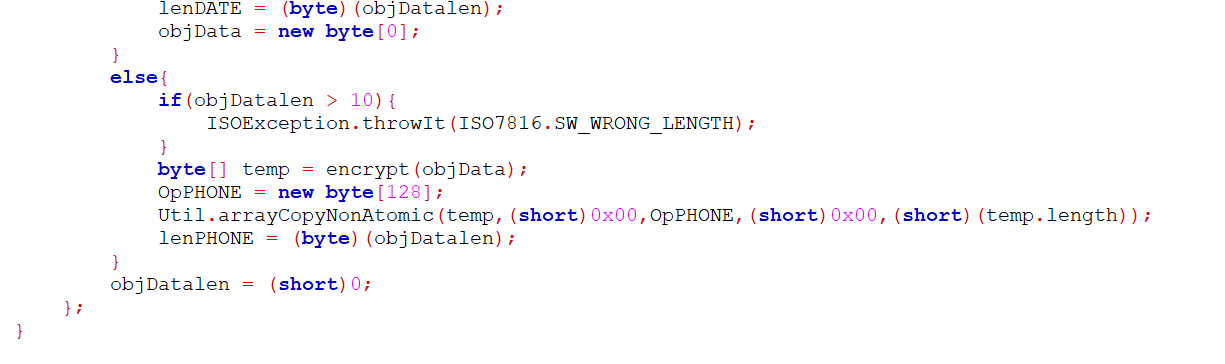


Hình 2‑14 Xứ lý tải lên ảnh phía Netbeans

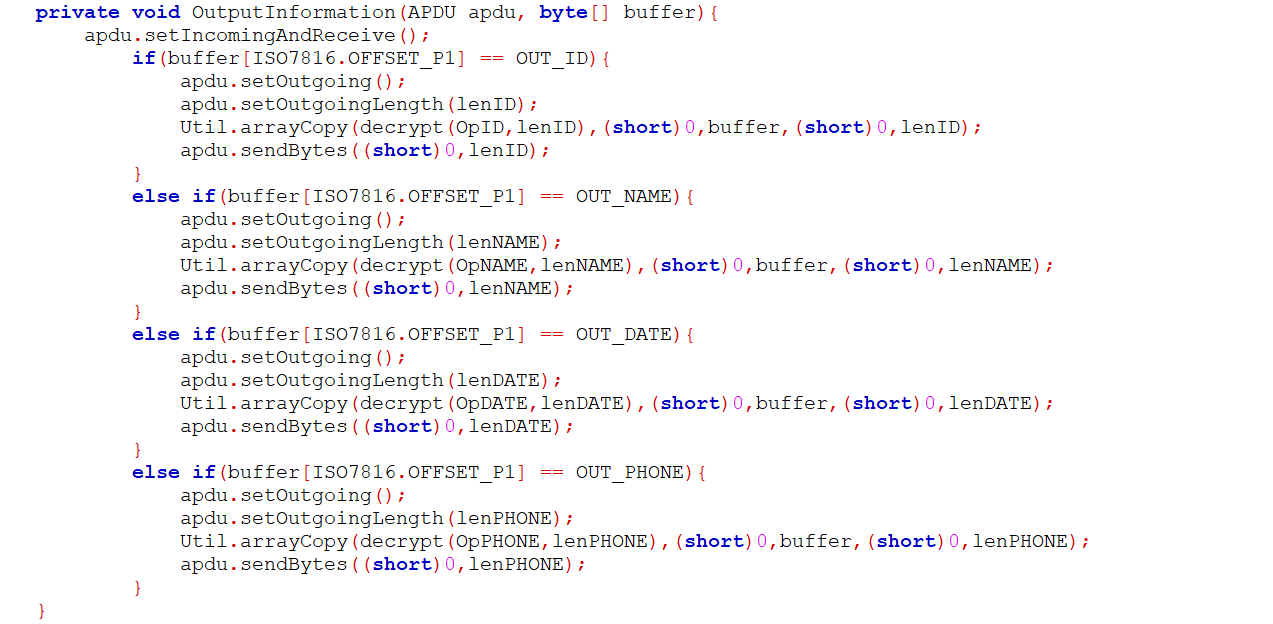
 

Hình 2‑15 Xứ lý tải xuống ảnh phía Netbeans

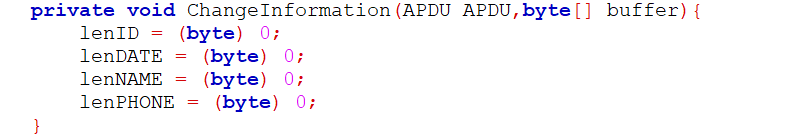
### Xử lý thông tin

Hình 2‑16 Xứ lý input thông tin



Hình 2‑17 Xứ lý output thông tin



Hình 2‑18 Xứ lý thay đổi thông tin

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1.Các kết quả đạt được

* Nghiên cứu về hệ mã RSA và thuật toán mã mã hóa AES.
* Nghiên cứu phát triển ứng dụng applet trên Netbean.
* Đã phát triển được một ứng dụng applet thẻ thư viện thông minh. Các chức năng của thẻ đã hoàn thành như dự kiến ban đầu. Giao diện và chức năng của phần mềm giao tiếp thẻ cũng đã hoàn thiện tương đối.
* Bên cạnh đó cũng tìm hiểu và hiểu cơ bản về các cấu trúc, ngôn ngữ và các phương thức xây dựng và phát triển phần mềm dựa trên thẻ thông minh smartcard.

## 2. Hạn chế

Vì thời gian có hạn cũng như trong quá trình nghiên cứu và làm việc không thể tránh khỏi sai xót nên chương trình vẫn còn nhiều lỗi chưa xử lý được.

Chỉ triển khai trên mô phỏng phần mềm nên còn hạn chế chưa thể tích hợp cũng như ứng dụng vào thực tế.

## 3. Hướng phát triển tiếp theo

Nghiên cứu và phát triển tích hợp thêm các chức năng mới hơn vào bảo mật hơn, xây dựng và triển khai luồng xử lý có nhiều ứng dụng cho thẻ. Đồng thời nâng cấp giao diện sao cho phù hợp và đẹp hơn nữa.

# TÀI LIỆU THAM KHẢO

[1] Slide bài giảng Phát triển phần mềm trong thẻ thông minh , ThS Cao Thanh Vinh, Học viện Kỹ thuật mật mã

[2] Giáo trình Phát triển phần mềm trong thẻ thông minh Chủ biên: ThS Cao Thanh Vinh & CN. Nguyễn Đức Hiếu