**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A picture containing room

Description automatically generated**

**BÁO CÁO MÔN CÁC CÔNG NGHỆ MỚI TRONG PHÁT TRIỂN PHẦN MỀM**

**Giáo viên hướng dẫn**: GS.TS. Phan Viết Hoàng

**Sinh viên thực hiện**: 2012353 – Đoàn Cao Nhật Hạ

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Đà Lạt, ngày 19 tháng 12 năm 2023

Giáo viên hướng dẫn

[Ký tên và ghi rõ họ tên]

**LỜI CẢM ƠN**

Trong quá trình học tập và thực hiện bài tập, em đã nhận được sự giúp đỡ của thầy cô. Em xin được bày tỏ sự cảm ơn sâu sắc tới thầy cô đã tạo điều kiện và giúp đỡ em trong quá trình học tập và nghiên cứu môn “Các công nghệ mới trong phát triển phần mềm”.

Trong quá trình làm bài tập này mặc dù em đã gặp nhiều khó khăn, kèm theo đó kiến thức còn hạn hẹp và thời gian thực hiện hạn chế nhưng em đã cố gắng hoàn thiện. Tuy nhiên, sẽ không tránh khỏi những thiếu sót khi tìm kiếm, tổng hợp, đánh giá và trình bày về bài tập. Em mong nhận được sự quan tâm, góp ý của thầy cô. Những ý kiến đóng góp đó là những kiến thức quý giá giúp em có thể nhận ra những hạn chế còn tồn đọng, qua đó có thể rút ra những kinh nghiệm và hoàn thiện được kỹ năng của mỗi bản thân sau này.

Với tình cảm chân thành và lòng biết ơn sâu sắc, em xin gửi lời cảm ơn tới Thầy Phan Viết Hoàng, Cô Trần Thị Phương Linh – người đã trực tiếp hướng dẫn và giúp đỡ trong suốt quá trình học tập, nghiên cứu và hoàn thành bài tập.

Em xin chân thành cảm ơn!

**ĐỀ CƯƠNG THỰC HIỆN BÀI TẬP**

**Tên đề tài**: Xây dựng ứng dụng DevOps ATM trên Docker.

Giáo viên hướng dẫn: GS.TS. Phan Viết Hoàng

**Sinh viên thực hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Email liên hệ** |
| 1 | Đoàn Cao Nhật Hạ | 2012353 | CTK44-PM | 2012353@dlu.edu.vn |

**I. Mục tiêu đề tài:**

Áp dụng những kiến thức đã học để xây dựng ứng dụng DevOps ATM trên Docker.

**II. Nội dung đề tài:**

* Phân tích các class MVC trong ATM.java
* Review và phân tích các bài TH Docker
* Lập trình (development) và triển khai (deployment) một ứng dụng DevOps ATM trên Docker. Ứng dụng này:
  + Dùng cấu trúc client-server tương tự như bài thực tập ‘client-server’
  + Dùng mô hình MVC (như ATM.java): frontend (View), backend (Control + Model), database (Model, Data).
  + Thay thế frontend React thành NodeJS.
  + Có giao diện tương tự bài Thực hành 'Thiết kế giao diện' trong môn 20CT4106.
* Xây dựng ứng dụng

**III. Phần mềm và công cụ sử dụng:**

* Visual Studio Code
* Android Studio
* NodeJS
* Docker
* GitHub

**IV. Dự kiến kết quả đạt được:**

* Nắm được những kiến thức chung về ứng dụng DevOps ATM
* Xây dựng được một ứng dụng DevOps ATM trên Docker
* Hoàn thành báo cáo bài tập

**V. Kế hoạch thực hiện**

* Xác định yêu cầu chức năng và phạm vi của ứng dụng
* Thiết kế giao diện ứng dụng client
* Xây dựng ứng dụng client
* Xây dựng server
* Triển khai ứng dụng trên Docker
* Viết báo cáo bài tập

|  |  |
| --- | --- |
| **Giáo viên hướng dẫn**  (Ký tên) | *Đà Lạt, ngày … tháng … năm 2023*  **SV Thực hiện**  (Ký tên) |
| **BCN Khoa**  (Ký tên) | **Tổ trưởng Bộ môn**  (Ký tên) |

**MỤC LỤC**

[**I. Phân tích các class MVC trong ATM.java. 1**](#_lnxbz9)

[**II. Review và phân tích các bài TH Docker. 5**](#_3cdcc32gqr1o)

[1. TH4-3 (goal) 5](#_rr4oqjlq7bhh)

[2. TH4-6 (feedback) 5](#_85oekcs6qtb2)

[3. TH4-7 (network) 6](#_rz8p6kmxjsph)

[**III. Lập trình (development) và triển khai (deployment) một ứng dụng DevOps ATM trên Docker. 6**](#_bg27o269duw)

[1. Tổng quan 6](#_2i9hkao1nr82)

[1.1. Giới thiệu đề tài 6](#_q20atjz83q4d)

[1.2. Đối tượng và phạm vi nghiên cứu 7](#_to3tk14amal9)

[1.2.1. Đối tượng 7](#_4y28zp8u0lee)

[1.2.2. Phạm vi nghiên cứu 7](#_r2oxm0gbaj9l)

[1.3. Mục tiêu nghiên cứu 7](#_9fq3d6uvio3s)

[2. Phân tích và thiết kế 8](#_9fq3d6uvio3s)

[2.1. Yêu cầu hệ thống 8](#_lodm45c7zzh9)

[2.2. Yêu cầu chức năng 8](#_peb890azeh7z)

[2.3. Cấu trúc thư mục 9](#_bafabiw0gslq)

[2.3.1. Cấu trúc thư mục hệ thống 9](#_7sktk92bnyl)

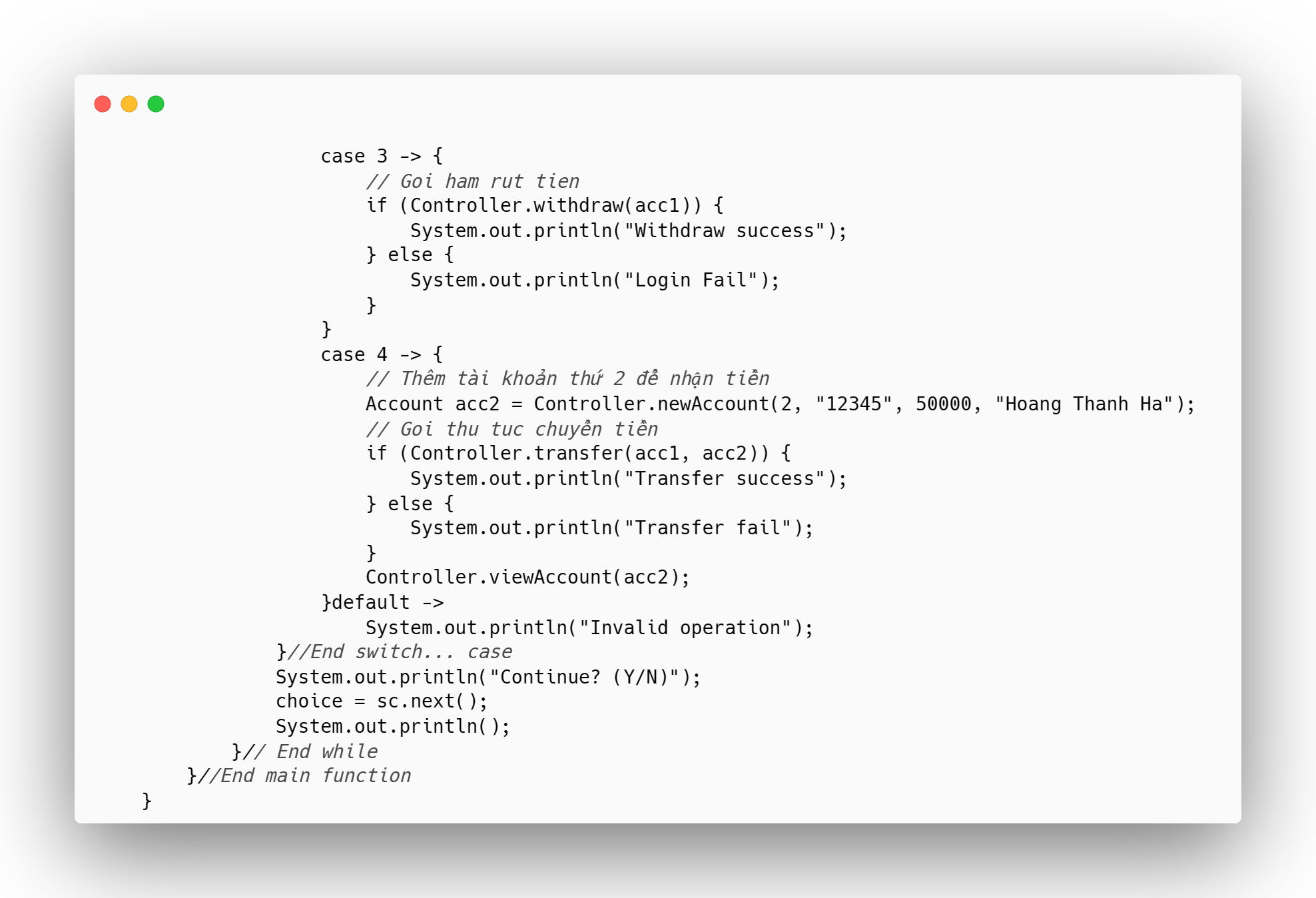
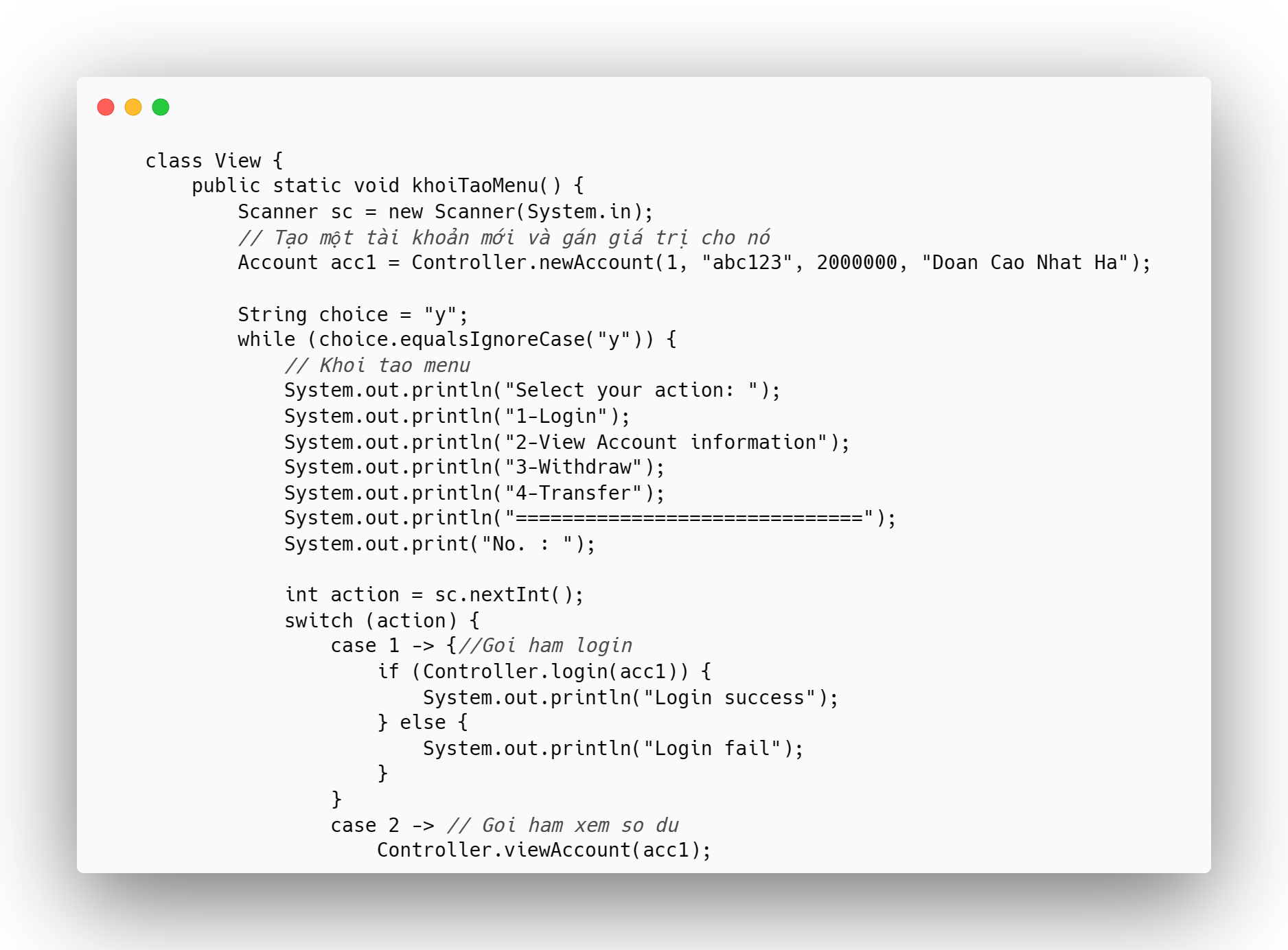
[2.3.2. Front - End 9](#_ovkjr8ntqwif)

[2.3.3. Back - End 10](#_xz721nmzai6w)

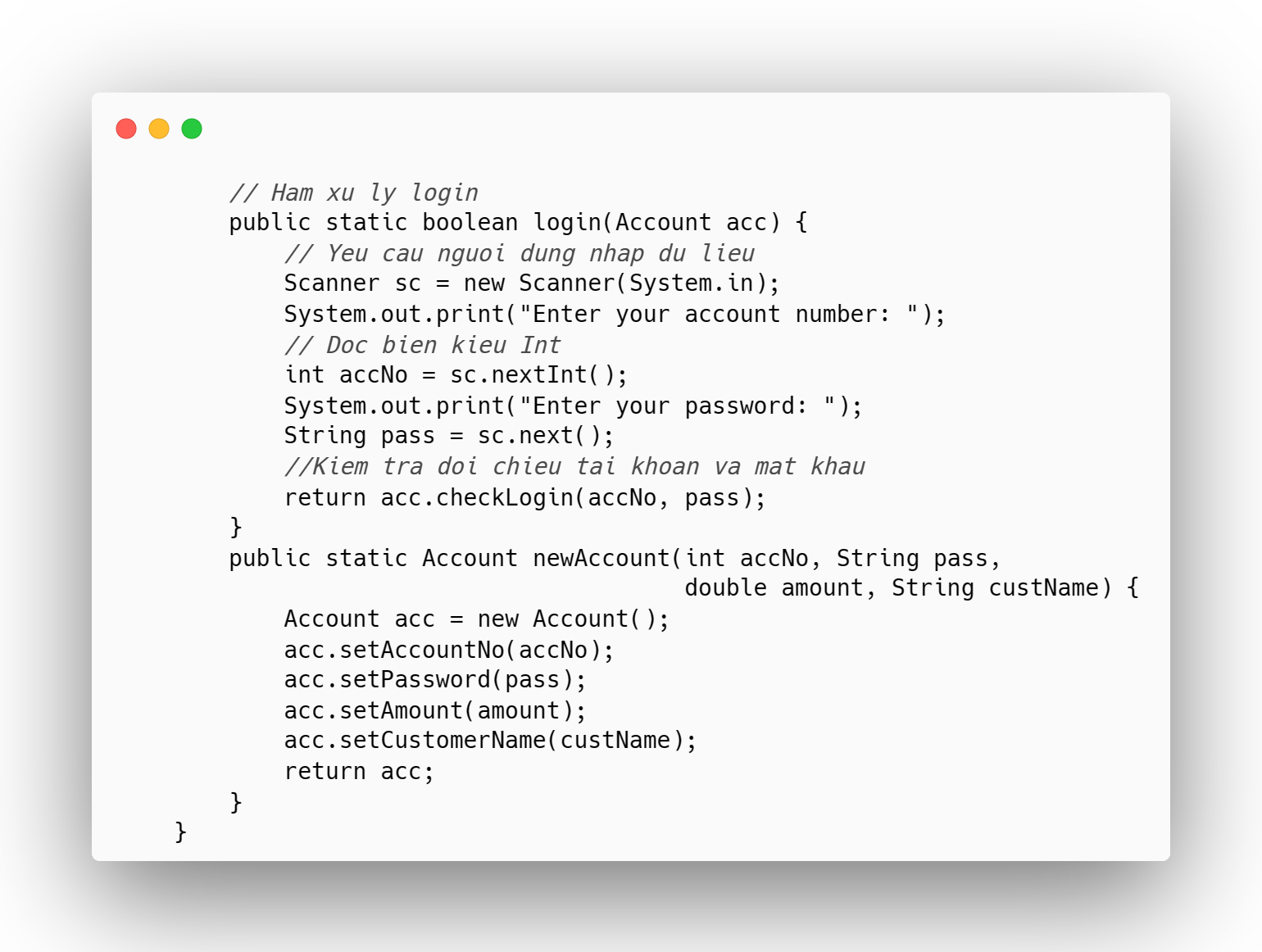
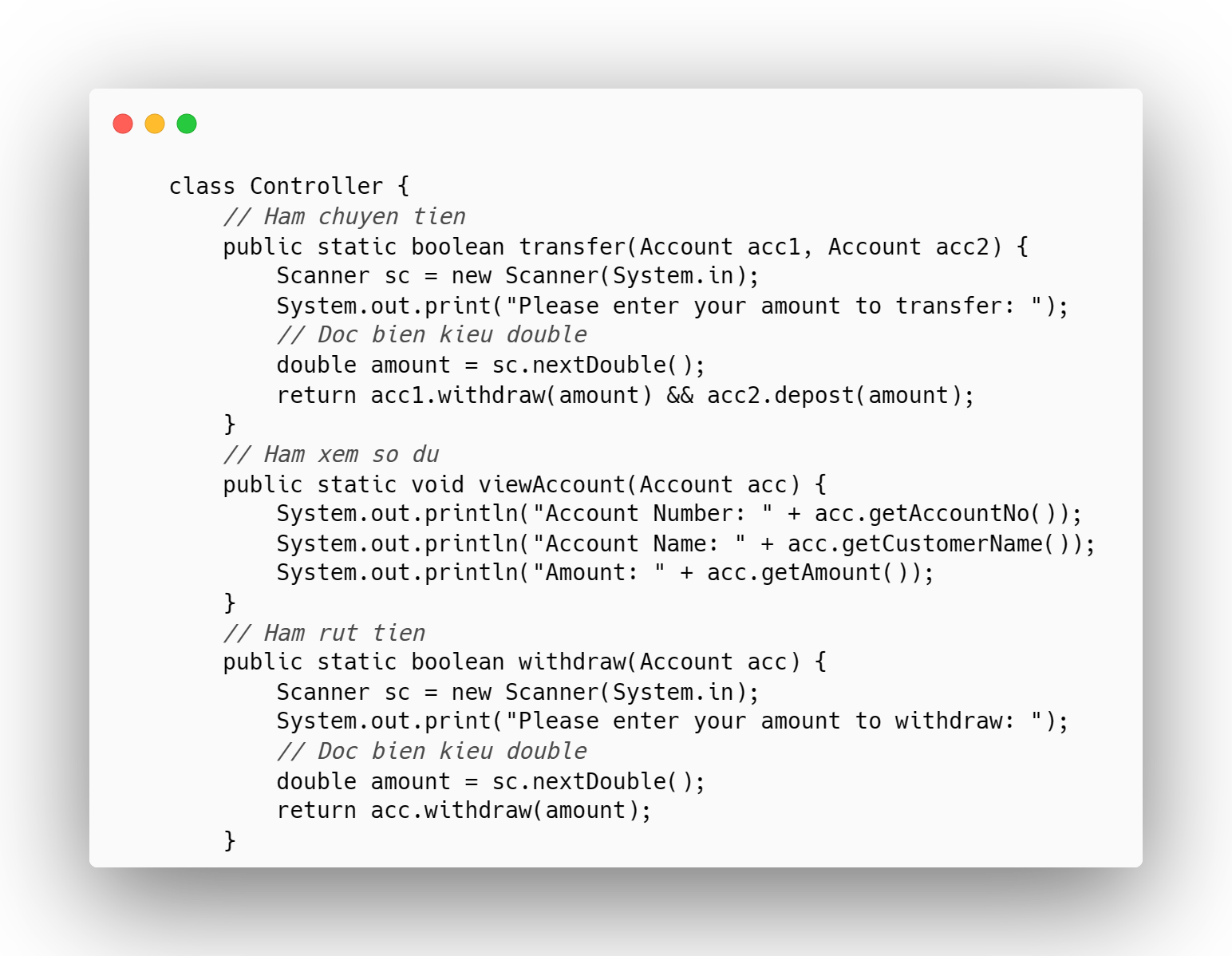
[2.3.4. Database 10](#_x6mhqoetd90d)

# Phân tích các class MVC trong ATM.java.

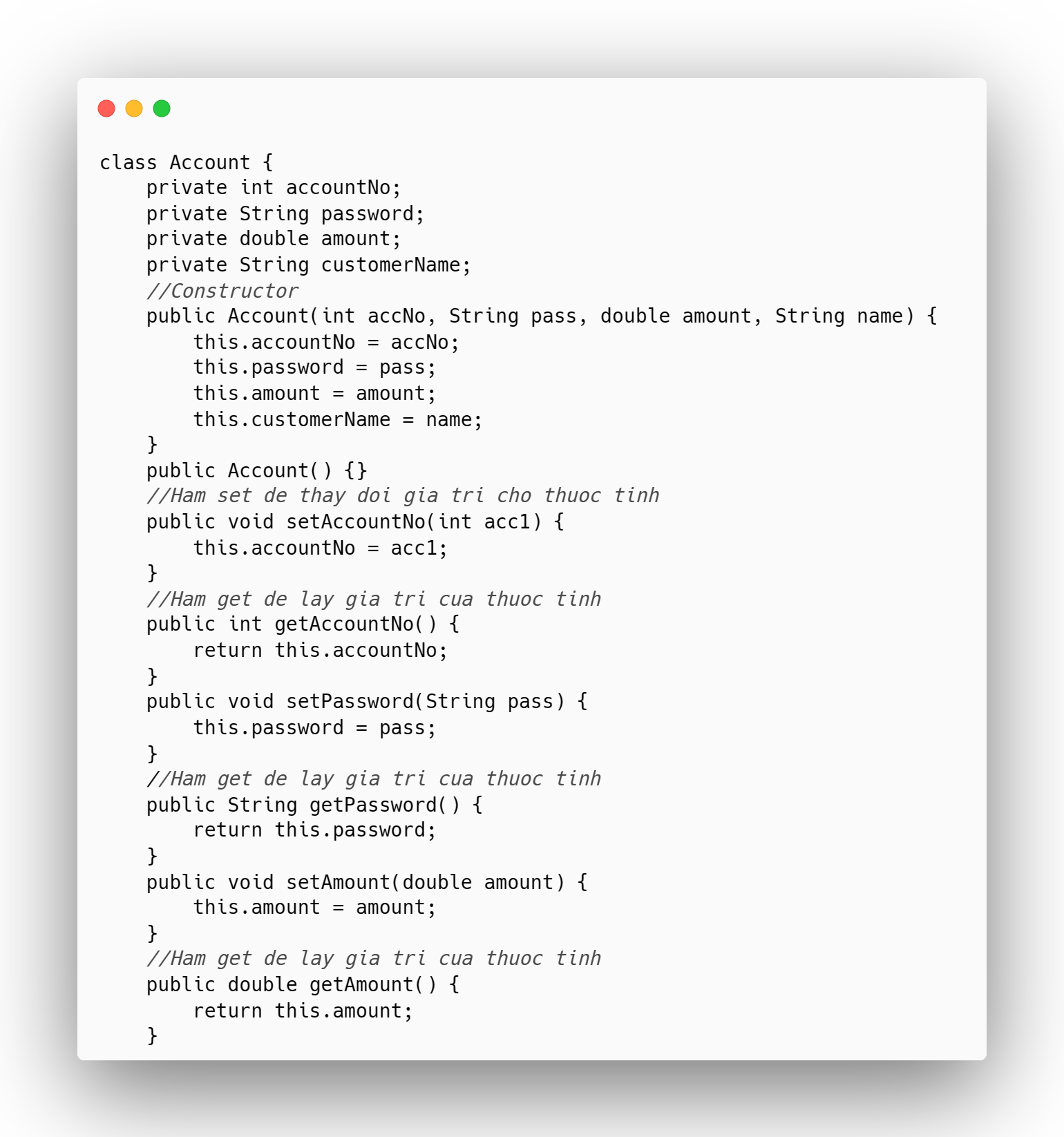
* Class View chứa các chức năng như:
* Hiển thị menu chức năng: Cho phép người dùng chọn các tùy chọn như rút tiền, kiểm tra số dư, chuyển khoản,...
* Hiển thị thông báo lỗi: Hiển thị thông báo khi có lỗi xảy ra trong quá trình giao dịch hoặc tương tác với người dùng
* Hiển thị thông báo thành công: Cung cấp phản hồi cho người dùng khi giao dịch hoặc hành động thành công.
* Hiển thị số dư: Hiển thị số dư hiện tại của tài khoản người dùng.
* Yêu cầu nhập thông tin chuyển khoản: Yêu cầu người dùng nhập thông tin người nhận và số tiền muốn chuyển khoản.
* Yêu cầu nhập số tiền rút: Yêu cầu người dùng nhập số tiền muốn rút từ tài khoản.

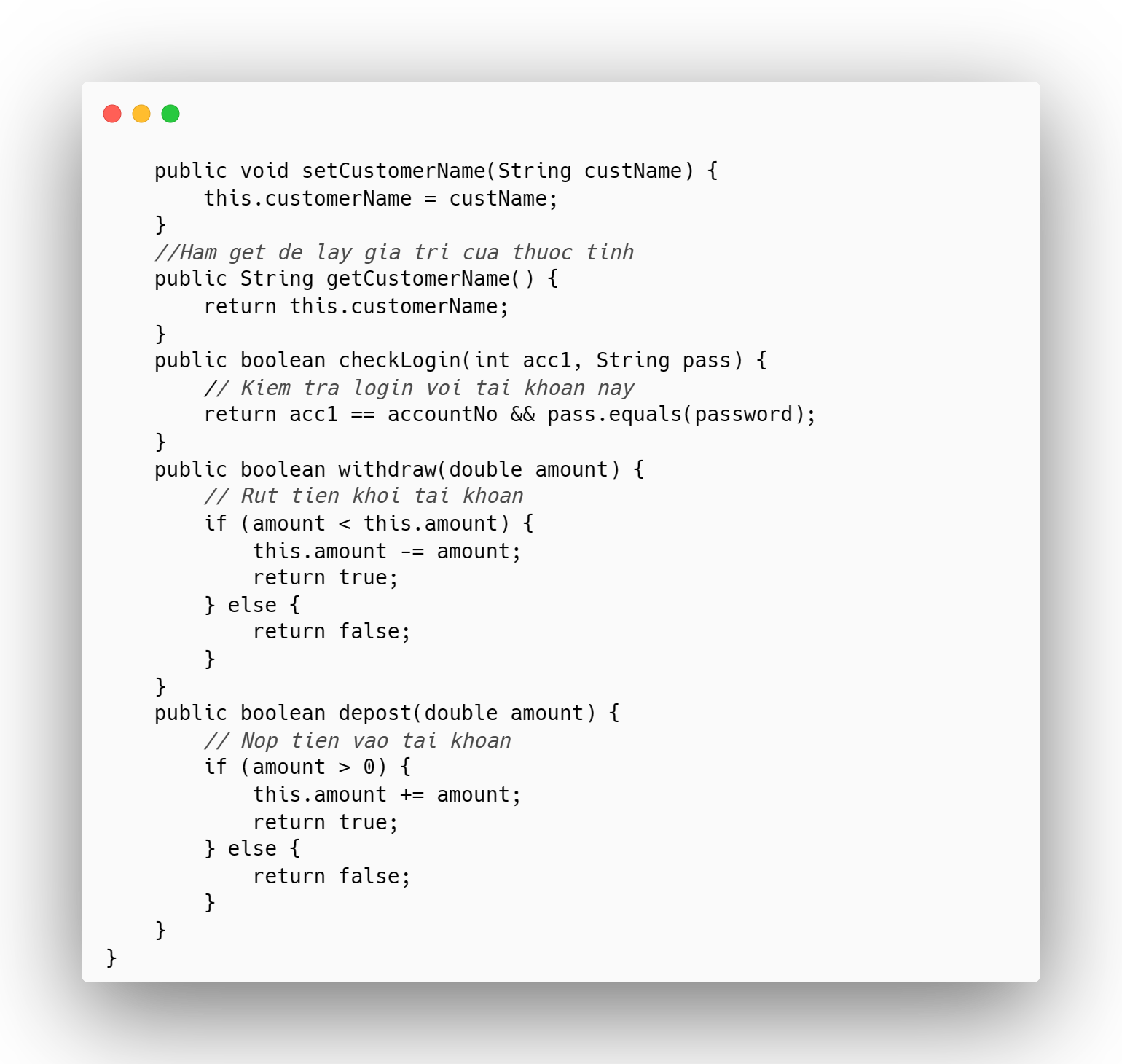


* Class Controller chứa các chức năng như:
* Xử lý yêu cầu rút tiền: Nhận yêu cầu rút tiền từ người dùng, kiểm tra số dư và thực hiện giao dịch rút tiền.
* Xử lý yêu cầu kiểm tra số dư: Nhận yêu cầu kiểm tra số dư từ người dùng và trả về số dư hiện tại của tài khoản.
* Xử lý yêu cầu chuyển khoản: Nhận yêu cầu chuyển khoản từ người dùng, kiểm tra thông tin và thực hiện giao dịch chuyển khoản.
* Xử lý các sự kiện từ View: Nhận thông tin từ giao diện người dùng và điều phối các hoạt động tương ứng.
* Tương tác với Model: Gọi các phương thức của lớp Model để thực hiện các thao tác cần thiết như cập nhật số dư, lịch sử giao dịch, v.v



* Class Account chứa các chức năng như:
* Quản lý thông tin tài khoản: Lưu trữ thông tin về tên tài khoản, số tài khoản, mật khẩu, số dư, lịch sử giao dịch, v.v.
* Cập nhật số dư: Cung cấp phương thức để cập nhật số dư sau mỗi giao dịch như rút tiền, chuyển khoản.
* Xác thực mật khẩu: Kiểm tra tính hợp lệ của mật khẩu khi người dùng đăng nhập hoặc thực hiện các giao dịch.
* Lịch sử giao dịch: Ghi lại lịch sử các giao dịch được thực hiện trên tài khoản.
* Kiểm tra số dư: Cung cấp phương thức để kiểm tra số dư hiện tại của tài khoản.





# Review và phân tích các bài TH Docker.

## TH4-3 (goal)

* Dự án “goal” (NodeJS) là dự án xây dựng ứng dụng đặt mục tiêu đơn giản có thể thiết lập mục tiêu, quản lý mục tiêu cá nhân: cho phép người dùng tạo, sửa đổi và xóa các mục tiêu cá nhân; tương tác với người dùng: giao diện thân thiện, dễ sử dụng để tạo trải nghiệm người dùng tốt và hiển thị mục tiêu đã đặt ra. Sau khi hoàn thành ứng dụng, triển khai dự án trên Docker và cập nhật yêu cầu của đề bài.

## TH4-6 (feedback)

* Dự án “feedback” (NodeJS) là dự án xây dựng ứng dụng nhận xét đơn giản có thể gửi phản hồi, đánh giá, hiển thị phản hồi và quản lý người dùng; tương tác với người dùng: giao diện thân thiện, dễ sử dụng để tạo trải nghiệm người dùng tốt. Triển khai dự án trên Docker và cập nhật chức năng yêu cầu.

## TH4-7 (network)

* Dự án “network”(NodeJS) là dự án xây dựng ứng dụng kết nối internet đơn giản có thể xác thực người dùng, truy xuất dữ liệu từ internet, gửi và nhận thông tin qua mạng. Giao diện người dùng: giao diện thân thiện, dễ sử dụng để tạo trải nghiệm người dùng tốt. Triển khai dự án trên Docker và cập nhật chức năng yêu cầu.

# Lập trình (development) và triển khai (deployment) một ứng dụng DevOps ATM trên Docker.

## Tổng quan

### **1.1. Giới thiệu đề tài**

DevOps là một phương pháp để phát triển và vận hành phần mềm một cách hiệu quả bằng cách kết hợp quy trình phát triển phần mềm (Dev) với quy trình vận hành hệ thống (Ops) nhằm tạo ra một quy trình liên tục hóa và tự động hóa, giúp tối ưu hóa thời gian phát triển, triển khai và vận hành phần mềm, đồng thời cải thiện sự hợp tác giữa các nhóm phát triển và vận hành. Docker là một nền tảng phần mềm cho việc tạo, triển khai và vận hành các ứng dụng trong môi trường container.

Việc tìm hiểu về DevOps và sử dụng Docker có ý nghĩa quan trọng trong việc tối ưu hóa quy trình phát triển và triển khai phần mềm. DevOps giúp tạo ra một quy trình làm việc liên tục, từ việc phát triển đến vận hành, giảm thời gian triển khai, tăng cường tính ổn định và khả năng phản hồi nhanh chóng. Sử dụng Docker giúp đóng gói ứng dụng và môi trường chạy của chúng thành các container độc lập, đơn giản hóa việc triển khai và đảm bảo tính nhất quán khi di chuyển giữa các môi trường khác nhau. Đồng thời, việc tìm hiểu và áp dụng DevOps cùng Docker cũng giúp nâng cao hiệu suất và hiệu quả làm việc của tổ chức phát triển phần mềm.

Ứng dụng DevOps về ATM là một hệ thống tự động hóa các tác vụ liên quan đến việc phát triển và triển khai một ứng dụng ATM. DevOps tối ưu hóa quy trình phát triển, triển khai và vận hành của ứng dụng quản lý ATM, giảm thiểu thời gian triển khai các cập nhật phần mềm trên hệ thống ATM. Hệ thống sử dụng Docker để tạo môi trường triển khai và phát triển, giúp đóng gói và triển khai phần mềm một cách nhẹ nhàng và hiệu quả hơn, giúp giảm thiểu sự cố vận hành và tối ưu hoá quản lý cấu hình của hệ thống ATM.

### **1.2. Đối tượng và phạm vi nghiên cứu**

### **1.2.1. Đối tượng**

* Ứng dụng DevOps vào hệ thống.
* Xây dựng ứng dụng ATM gồm: Frontend, Backend và Database.
* Sử dụng Docker là môi trường phát triển và triển khai hệ thống.

### **1.2.2. Phạm vi nghiên cứu**

* Tìm hiểu nghiên cứu, áp dụng các kiến thức về DevOps, sử dụng môi trường Docker và các công nghệ khác .
* Phân tích yêu cầu của hệ thống như: Tạo tài khoản yêu cầu phải có ký hiệu trong tài khoản, đăng nhập vào hệ thống, đăng xuất khỏi hệ thống, xem thông tin tài khoản và số dư trong tài khoản.
* Thiết kế và xây dựng hệ thống ATM gồm: Trang đăng ký, đăng nhập hệ thống, đăng xuất khỏi hệ thống, xây dựng trang chủ của hệ thống, thiết kế chức năng xem thông tin chi tiết tài khoản, chức năng quay trở lại thao tác trước.
* Tích hợp và kiểm thử hệ thống ATM.
* Triển khai hệ thống ATM có ứng dụng DevOps.

### **1.3. Mục tiêu nghiên cứu**

* Nghiên cứu và xây dựng hệ thống DevOps ATM sử dụng Docker nhằm tối ưu hóa quá trình phát triển và triển khai ứng dụng.
* Hiểu rõ về các khái niệm cơ bản của DevOps, Docker và các công nghệ liên quan.
* Phân tích các ưu và nhược điểm của việc triển khai ứng dụng hệ thống DevOps ATM trên Docker so với các phương pháp truyền thống.
* Hiểu rõ về quá trình triển khai, phát triển và ứng dụng trong môi trường DevOps.
* Triển khai ứng dụng ATM sử dụng Docker đảm bảo sự đồng nhất và di động của môi trường triển khai.
* So sánh về hiệu suất của phương pháp triển khai sử dụng Docker với các phương pháp truyền thống khác hiện nay trên thị trường.
* Đánh giá hiệu suất và hiệu quả của hệ thống DevOps ATM dựa trên Docker.
* Xác định hướng phát triển tiếp theo cho môi trường triển khai DevOps, chú trọng vào tích hợp công nghệ mới và các phương tiện tối ưu hóa.

## Phân tích và thiết kế

### **2.1. Yêu cầu hệ thống**

Xây dựng hệ thống ATM gồm:

* Xây dựng Frontend
* Xây dựng Backend
* Xây dựng Database cho hệ thống
* Triển khai trên Docker
* Sử dụng công nghệ Ionic FrameWork

### **2.2. Yêu cầu chức năng**

Các chức năng chính bao gồm:

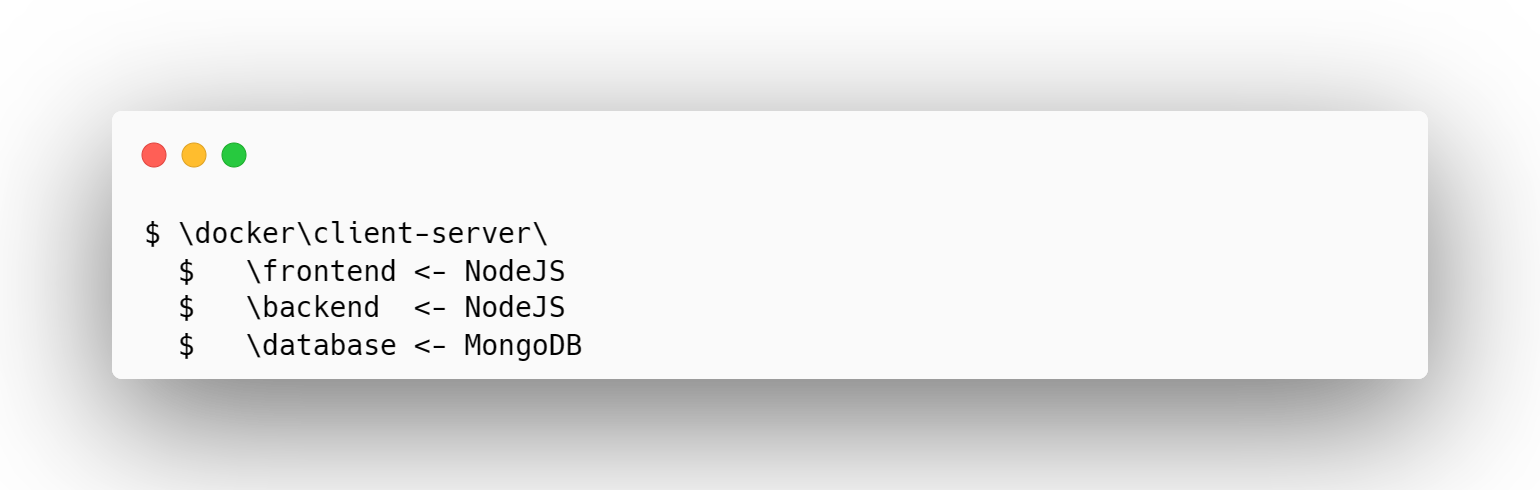
* Đăng ký tài khoản
* Đăng nhập tài khoản
* Đăng xuất tài khoản
* Hiển thị thông tin tài khoản
* Hiển thị chi tiết thông tin tài khoản

Ngoài ra còn tích hợp các chức năng như:

* Chức năng chụp hình
* Chức năng bản đồ
* Gọi API

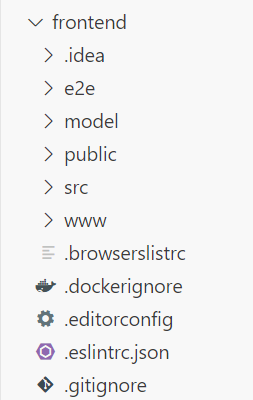
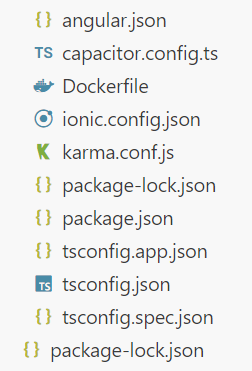
### **2.3. Cấu trúc thư mục**

#### **2.3.1. Cấu trúc thư mục hệ thống**

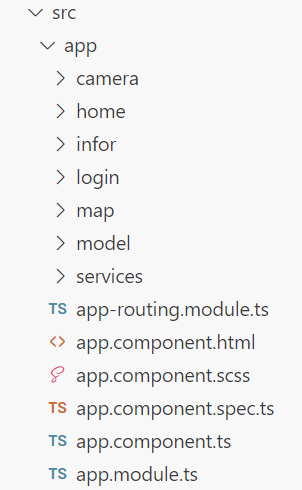
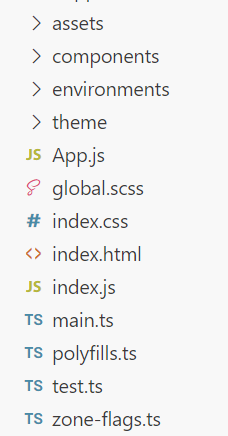


#### **2.3.2. Front - End**

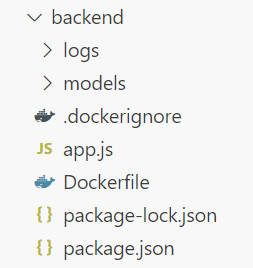
* Tổng quan

* Chi tiết thư mục src

#### **2.3.3. Back - End**



#### **2.3.4. Database**

