

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA



LẬP TRÌNH NÂNG CAO

BÁO CÁO BÀI TẬP LỚN

**Đề tài: Ứng dụng quản lý ngân hàng
(Bank Management System)**

LỚP L03

Giảng viên hướng dẫn: ThS Lê Đình Thuận

Sinh viên thực hiện	Mã số sinh viên
Võ Ngọc Duy Nghiêm	2220036

Mục lục

Danh mục hình ảnh	2
1. Mô tả hệ thống	3
2. Mô tả chi tiết các lớp	3
3. Sơ đồ lớp (class diagram) của hệ thống	5
4. Chạy thử	5
5. Đề xuất hướng phát triển	12
6. Kết luận	12
Tài liệu tham khảo.....	13

Danh mục hình ảnh

Hình 1. Class diagram của hệ thống	5
Hình 2. Đoạn code khởi tạo tài khoản khi bắt đầu chương trình	5
Hình 3. Giao diện khi đăng nhập vào hệ thống.....	6
Hình 4. Giao diện dành cho tài khoản quản lý.....	6
Hình 5. Tạo tài khoản quản lý thành công	7
Hình 6. Danh sách tài khoản quản lý	7
Hình 7. Phương thức tạo tài khoản người dùng	8
Hình 8. Phương thức chuyển tiền đến một tài khoản dựa trên số tài khoản	8
Hình 9. Phương thức chuyển tiền từ một tài khoản đến tài khoản khác	9
Hình 10. Phương thức xem thông tin người dùng	9
Hình 11. Phương thức khoá tài khoản người dùng	10
Hình 12. Hạn chế của tài khoản quản lý cấp 2.....	10
Hình 13. Đăng nhập với tài khoản người dùng.....	11
Hình 14. Giao diện của tài khoản người dùng	11
Hình 15. Phương thức kiểm tra số dư của tài khoản người dùng	12

1. Mô tả hệ thống

Ứng dụng quản lý ngân hàng (bank management system) là một ứng dụng được tạo ra để hiện thực các tác vụ của một tài khoản khi truy cập vào hệ thống ngân hàng. Trong đó, tài khoản (Account) được chia thành hai loại: tài khoản dành cho người quản lý (Manager) và tài khoản dành cho người sử dụng (User). Với mỗi loại tài khoản, chúng sẽ có các quyền hạn khác nhau, do đó mỗi Manager và User sẽ có các phương thức (method) riêng.

Khi chương trình được chạy, chúng ta sẽ dùng những tài khoản (Manager hoặc User) để truy cập vào hệ thống ngân hàng (Bank) và thực hiện các tác vụ. Trong đó class Bank gồm những thuộc tính như danh sách người quản lý, danh sách người dùng và danh sách các tài khoản bị khoá. Để đảm bảo có thể thực hiện được việc đăng nhập, chương trình sẽ khởi tạo một tài khoản quản lý (Manager) nếu trong hệ thống chưa có tài khoản quản lý nào.

Ngoài những lớp được nêu trên, chúng ta cần dùng một lớp database để thực hiện việc đọc và ghi dữ liệu lại trên các tệp, các tệp này dùng để lưu các danh sách kể trên sau mỗi lần chạy chương trình.

Ngôn ngữ thực hiện: Java

Môi trường phát triển tích hợp (IDE): Eclipse 2023

2. Mô tả chi tiết các lớp

Account:

- Thuộc tính:
 - name, id, password, phoneNumber là các thuộc tính biểu diễn tên người dùng, số tài khoản, mật khẩu và số điện thoại của người dùng. Hệ thống giới hạn 1000 người quản lý, do đó id của tài khoản quản lý sẽ tự động được tạo từ 1 đến 1000, và id của người dùng sẽ tự động được tạo từ 1001. Mật khẩu là một số nguyên có 4 chữ số.
- Phương thức :
 - Gồm các getter và setter của những thuộc tính trên.

Bank:

- Thuộc tính:
 - name thể hiện tên ngân hàng.
 - listManager, listUser, listblock là các danh sách được sử dụng để quản lý những tài khoản có trong ngân hàng.
- Phương thức:
 - Constructor có tham số.
 - Getter của các danh sách trên.
 - login(): phương thức được dùng để đăng nhập vào hệ thống ngân hàng. Tùy theo loại tài khoản đăng nhập mà chương trình sẽ hiện ra giao diện quản lý hay giao diện người dùng.
 - managerMenu(): giao diện thể hiện danh sách tính năng mà tài khoản quản lý có thể thao tác.
 - userMenu(): giao diện thể hiện danh sách tính năng mà tài khoản người dùng có thể thao tác.

Manager: kế thừa lớp Account

- Thuộc tính:
 - position là thuộc tính dùng để phân chia cấp bậc của người quản lí, giả sử trong hệ thống chỉ có người quản lí cấp 1 và cấp 2 tương ứng với giá trị 1 và 2. Có một số tác vụ chỉ có người quản lí cấp 1 mới thực hiện được.
- Phương thức:
 - Constructor không tham số và constructor có tham số.
 - Getter và setter của thuộc tính position.
 - createManager(): phương thức để tạo ra thêm một tài khoản quản lí mới. Tài khoản được tạo thành công sẽ được thêm vào danh sách tài khoản quản lí. Đây là phương thức mà chỉ có người quản lí cấp 1 mới thực hiện được.
 - createUser(): là phương thức để tạo ra một tài khoản người dùng mới. Tài khoản được tạo thành công sẽ được thêm vào danh sách người dùng.
 - deposit(): là phương thức gửi tiền vào một tài khoản (dựa trên số tài khoản) được thực hiện bởi người quản lí. Phương thức này mô phỏng lại thực tế, khi khách hàng đến ngân hàng và nhờ gửi tiền, nhân viên ngân hàng sẽ thực hiện tác vụ này.
 - withdraw(): tương tự deposit(), phương thức này dùng để rút tiền từ một tài khoản khi có yêu cầu từ khách hàng.
 - transfer(): phương thức chuyển tiền từ một tài khoản đến một tài khoản khác dựa trên số tài khoản. Ở đây, ta giả sử chỉ có thể chuyển được giữa những tài khoản có trong ngân hàng.
 - showInfo(): phương thức xem thông tin của người dùng dựa trên số tài khoản.
 - blockUser(): khi có yêu cầu khoá tài khoản từ người dùng hoặc tài khoản vi phạm các chính sách của ngân hàng, người quản lí sẽ thực hiện tác vụ này. Tài khoản bị khoá sẽ không còn nằm trong danh sách người dùng và sẽ không thực hiện được các tác vụ của mình nữa. Đây là phương thức chỉ có người quản lí cấp 1 mới thực hiện được.
 - unblockUser(): là phương thức mở khoá một tài khoản người dùng đang bị khoá.

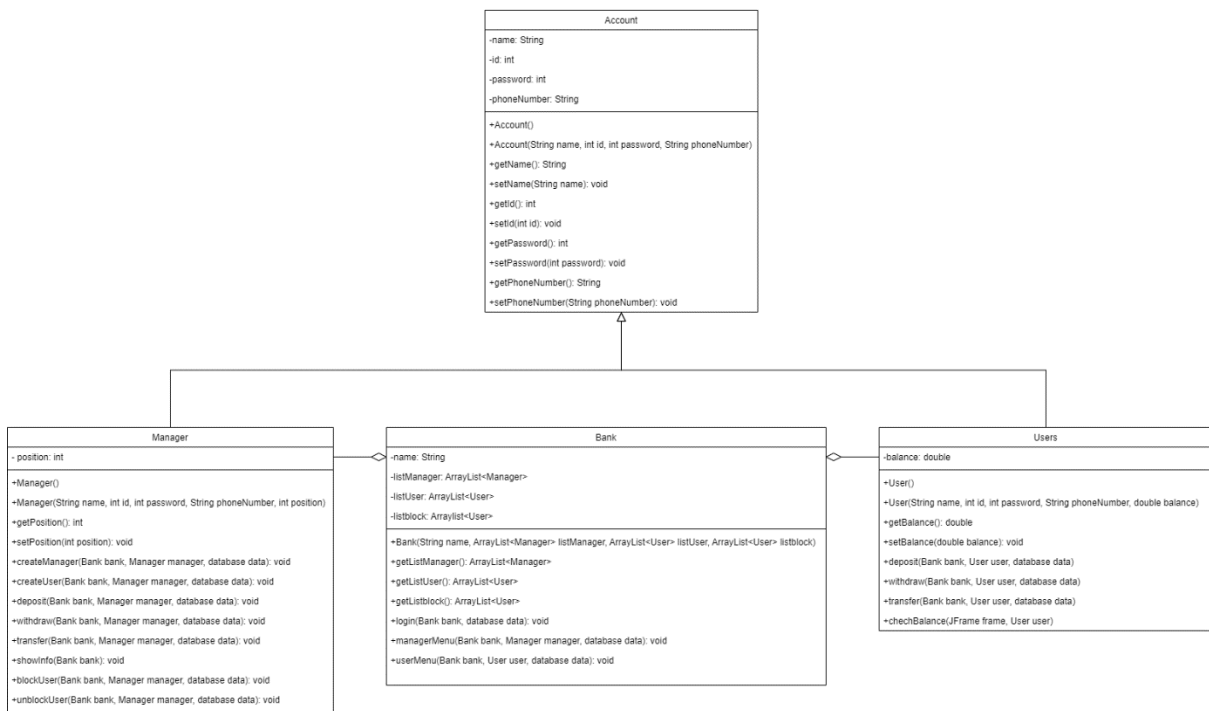
User: kế thừa lớp Account

- Thuộc tính:
 - balance: biểu thị số dư trong tài khoản của người dùng.
- Phương thức:
 - Constructor không tham số và constructor có tham số.
 - Getter và setter của thuộc tính balance.
 - deposit(): phương thức gửi tiền vào tài khoản cá nhân được thực hiện bởi người dùng. Phương thức này mô phỏng dựa trên hoạt động gửi tiền tại ATM trong thực tế.
 - withdraw(): phương thức rút tiền từ tài khoản cá nhân của người dùng. Phương thức này cũng dựa trên hoạt động rút tiền từ ATM.
 - transfer(): phương thức chuyển tiền từ tài cá nhân sang một tài khoản khác dựa trên số tài khoản.

- `checkBalance()`: phương thức theo dõi số dư của tài khoản cá nhân.

Ở tất cả các phương thức được mô tả trên, chương trình đều có những ràng buộc cụ thể, do đó bất kì thao tác nhập thông tin nào sai kiểu dữ liệu của thuộc tính hoặc sai thông tin (số tài khoản, mật khẩu xác nhận,...) chương trình sẽ đưa ra thông báo và người dùng sẽ phải nhập lại.

3. Sơ đồ lớp (class diagram) của hệ thống



Hình 1. Class diagram của hệ thống

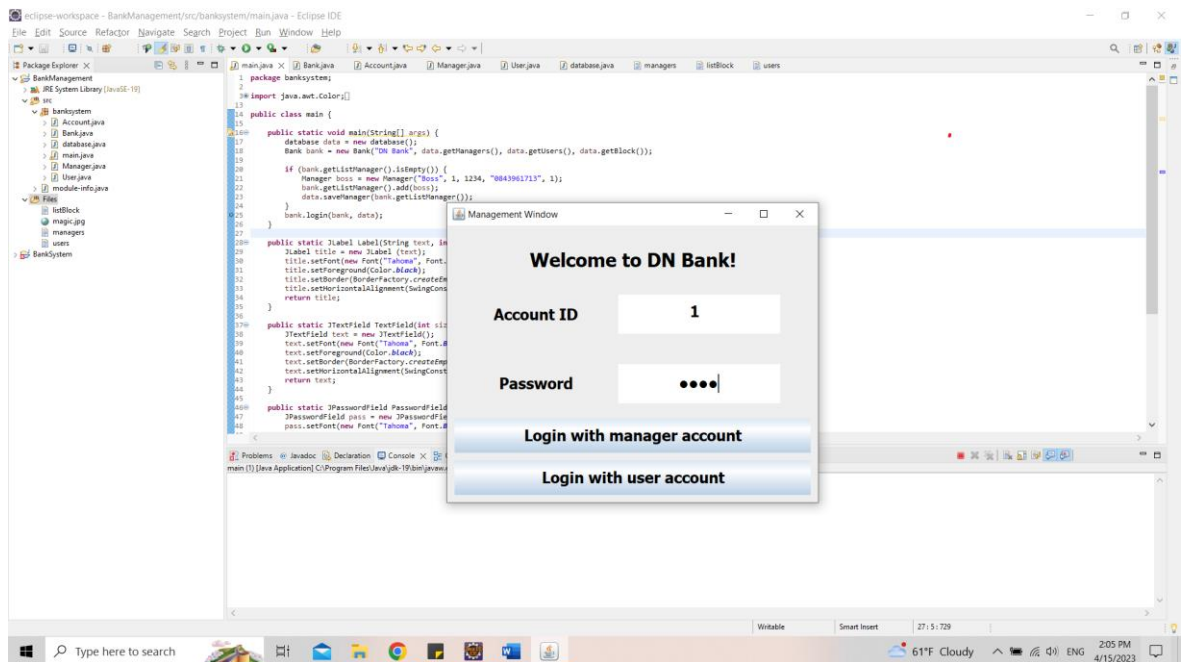
4. Chạy thử

Chương trình sẽ khởi tạo một tài khoản quản lý đầu tiên để đăng nhập vào hệ thống ngân hàng. Tài khoản quản lý này sẽ có số tài khoản (id là 1), mật khẩu (password là 1234) và cấp quản lý (position là 1)

```

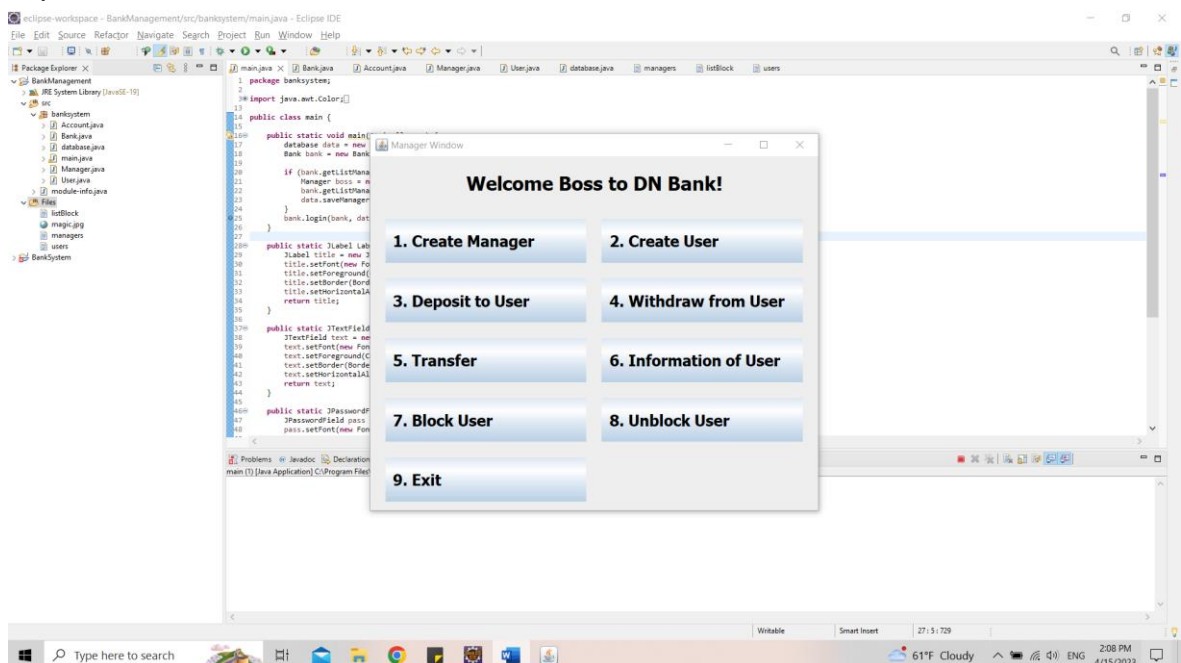
if (bank.getListManager().isEmpty()) {
    Manager boss = new Manager("Boss", 1, 1234, "0843961713", 1);
    bank.getListManager().add(boss);
    data.saveManager(bank.getListManager());
}
bank.login(bank, data);
  
```

Hình 2. Đoạn code khởi tạo tài khoản khi bắt đầu chương trình



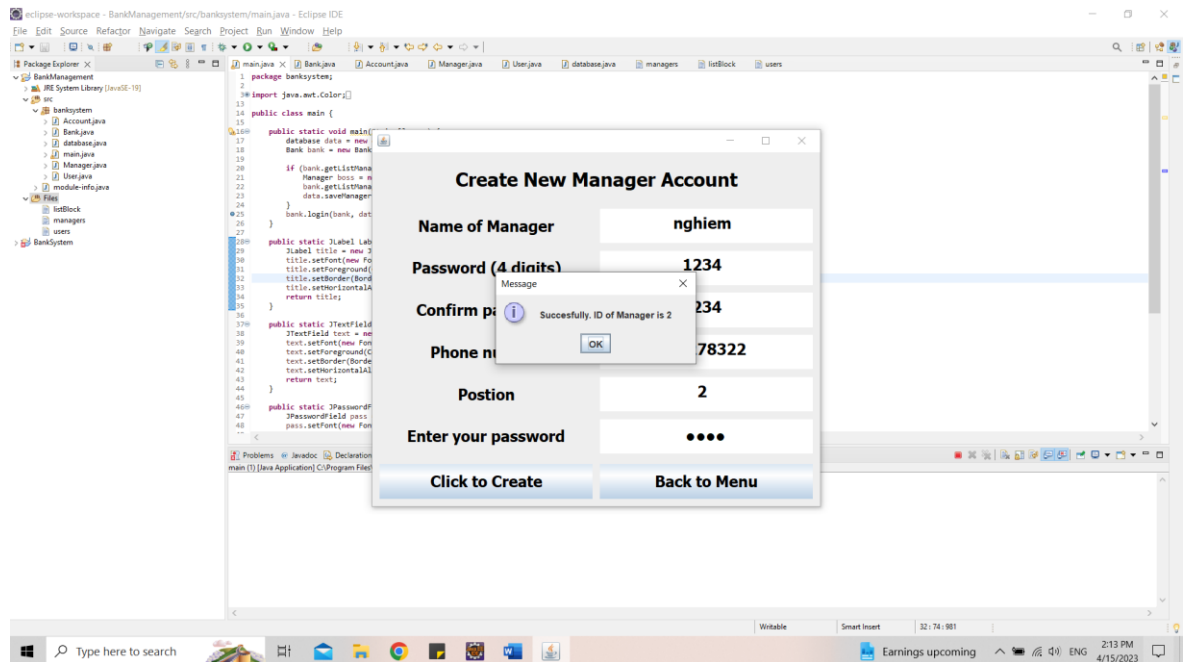
Hình 3. Giao diện khi đăng nhập vào hệ thống

Sử dụng nút “Login with manager account”, giao diện dành cho tài khoản quản lý sẽ hiện ra.



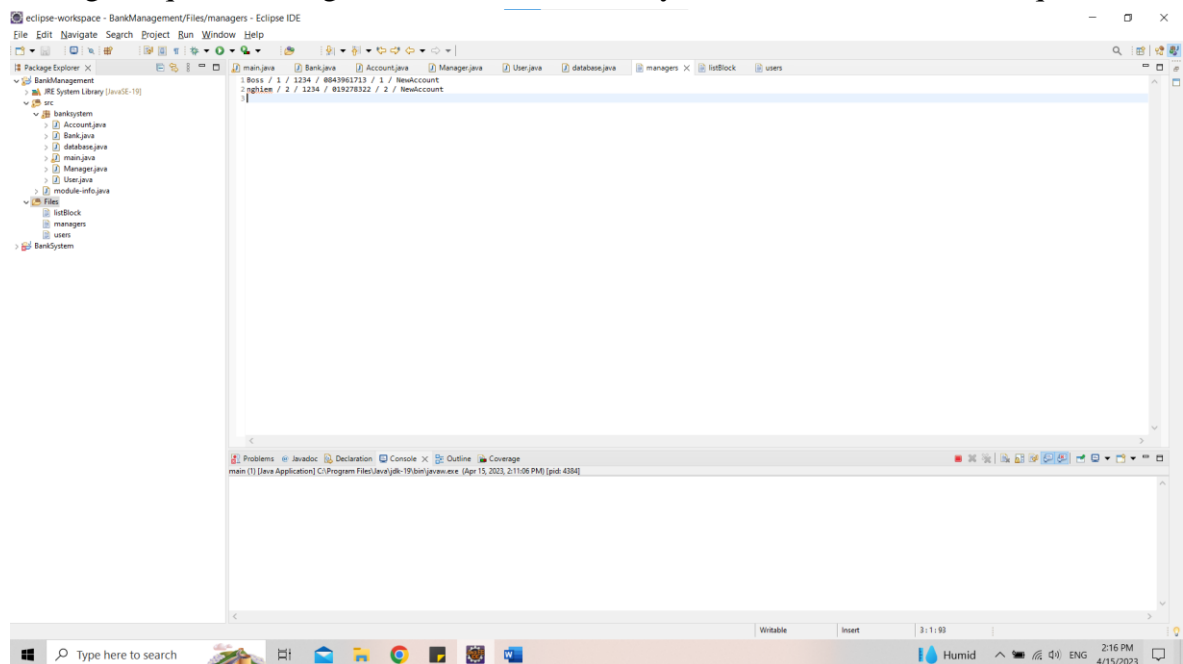
Hình 4. Giao diện dành cho tài khoản quản lý

Sau đây những hình ảnh chạy thử của các tác vụ được thực hiện bởi người quản lý:



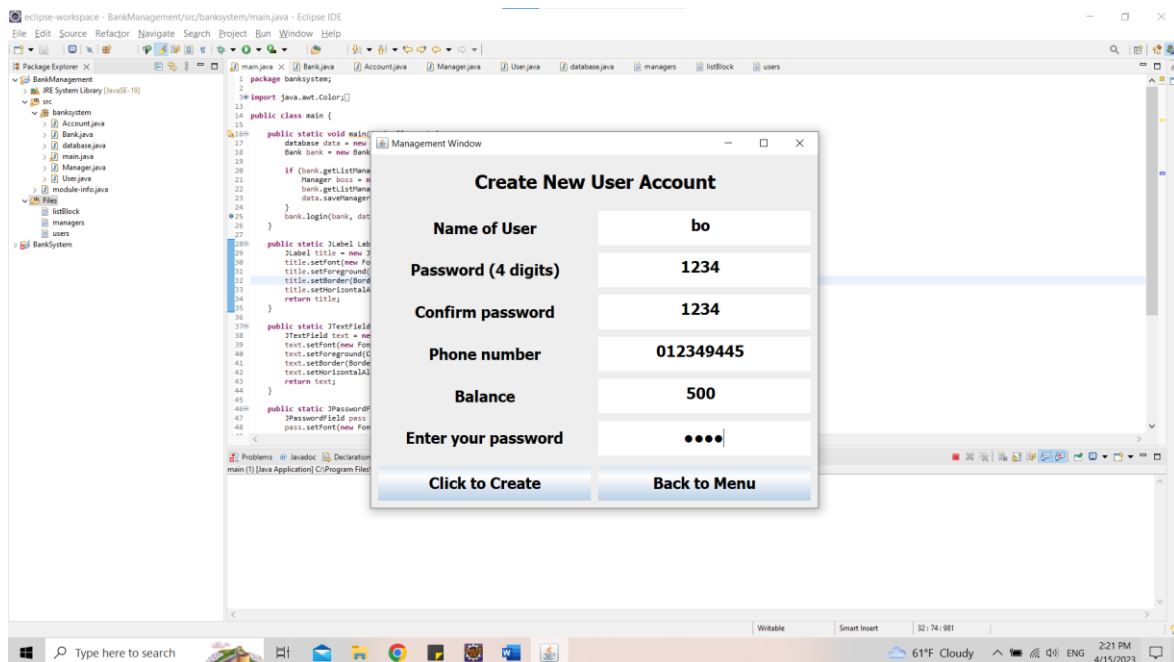
Hình 5. Tạo tài khoản quản lý thành công

Sau khi tạo thành công tài khoản quản lý, hệ thống sẽ báo cho chúng ta số tài khoản của người quản lý, đồng thời thêm tài khoản này vào danh sách tài khoản quản lý.



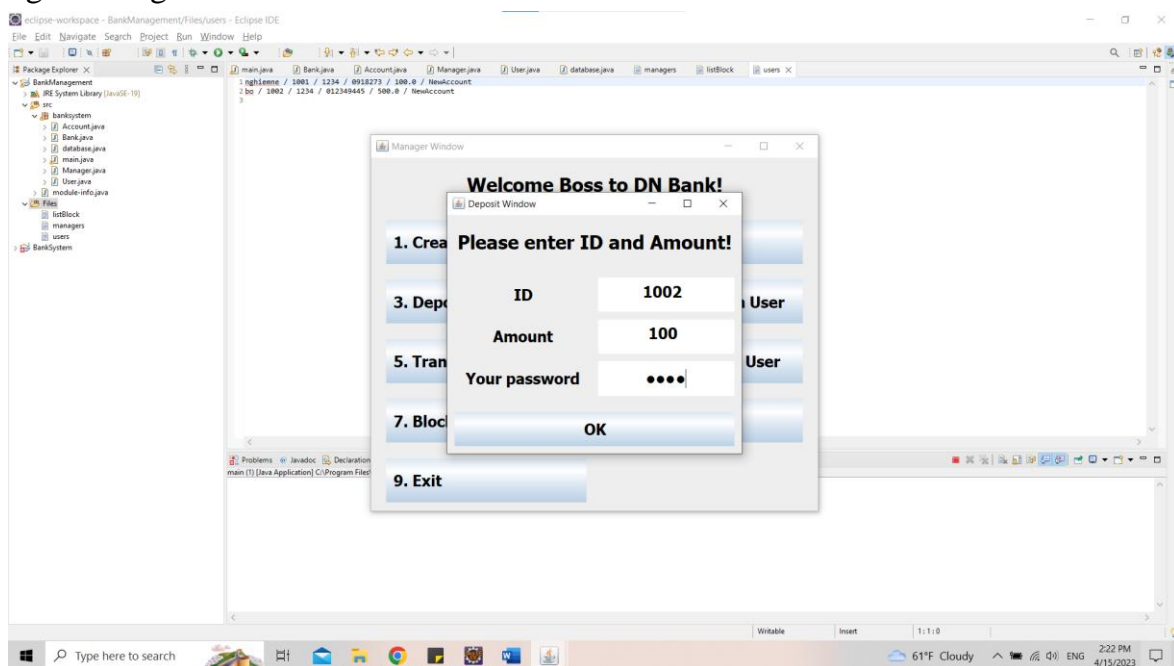
Hình 6. Danh sách tài khoản quản lý

Tương tự sẽ là hình ảnh của các tác vụ tiếp theo:



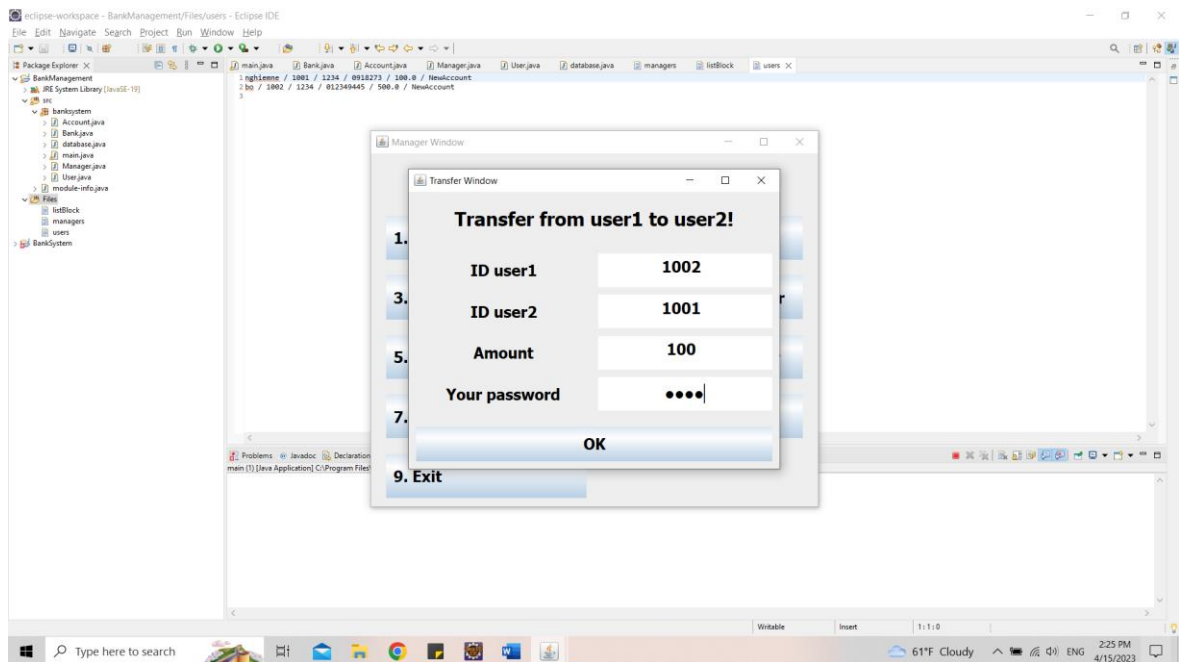
Hình 7. Phương thức tạo tài khoản người dùng

Sau khi tạo thành công tài khoản người dùng quản lí, hệ thống sẽ báo cho chúng ta số tài khoản của người dùng, đồng thời thêm tài khoản này vào danh sách tài khoản người dùng.



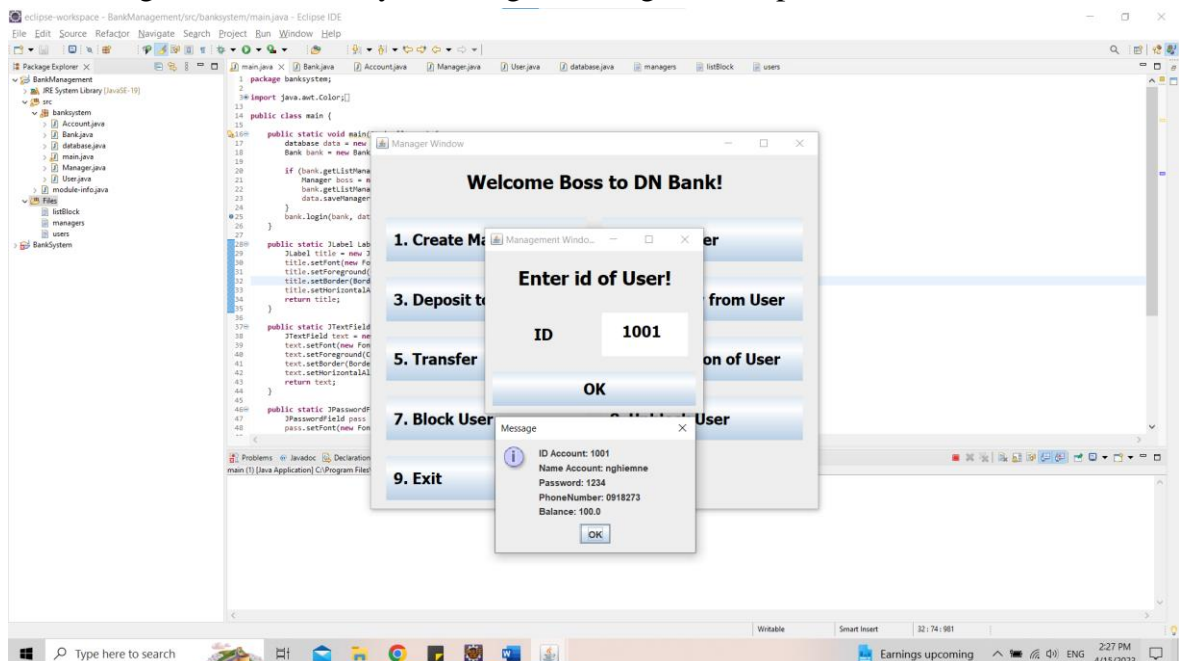
Hình 8. Phương thức chuyển tiền đến một tài khoản dựa trên số tài khoản

Phương thức rút tiền từ một tài khoản dựa trên số tài khoản (Withdraw from User) giống như phương thức gửi tiền vào (Deposit to User).

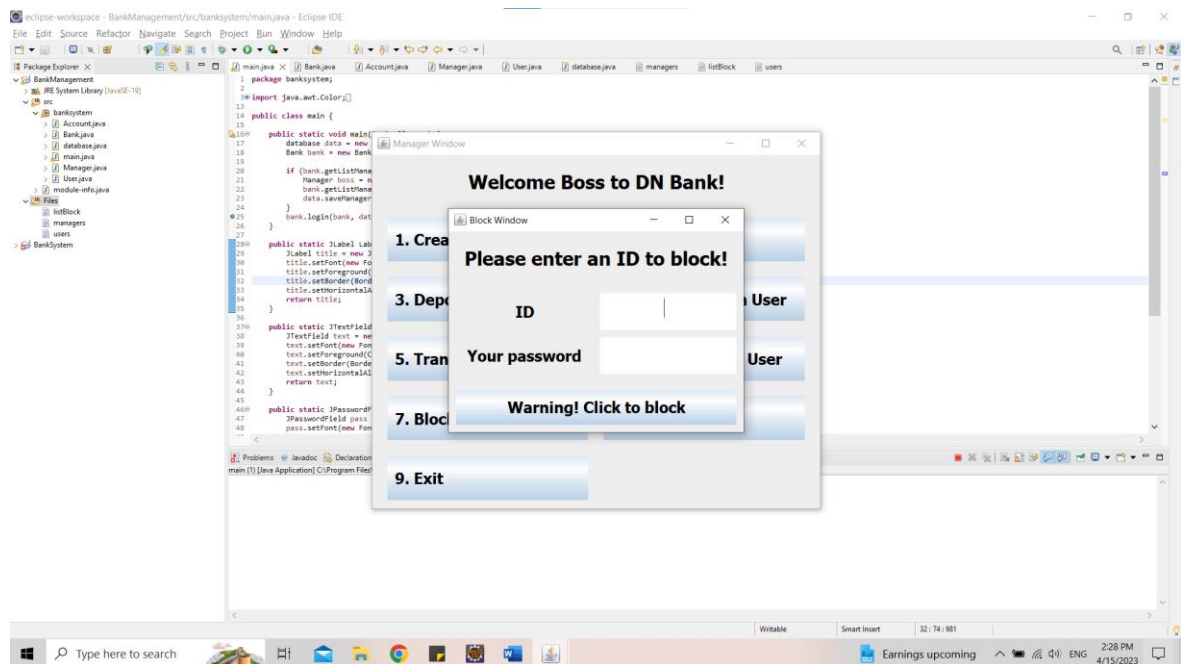


Hình 9. Phương thức chuyển tiền từ một tài khoản đến tài khoản khác

Ở những phương thức rút hoặc chuyển tiền, hệ thống sẽ báo lỗi nếu số dư tài khoản được dùng để rút hoặc chuyển không đủ lượng tiền nhập vào.

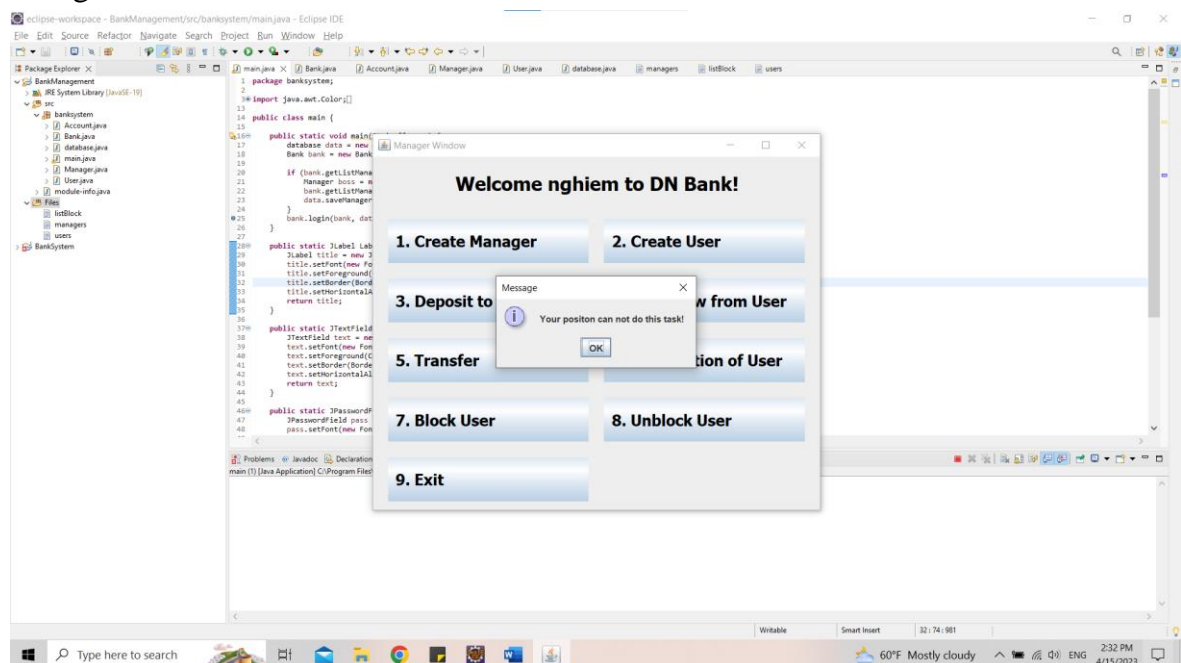


Hình 10. Phương thức xem thông tin người dùng

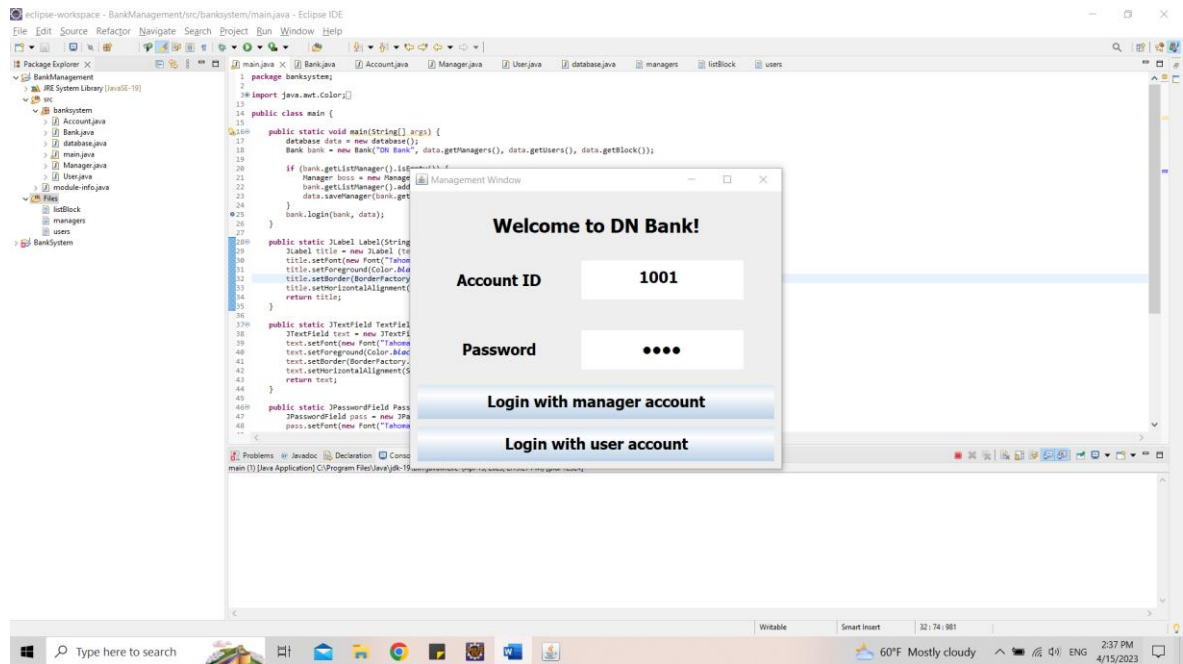


Hình 11. Phương thức khoá tài khoản người dùng

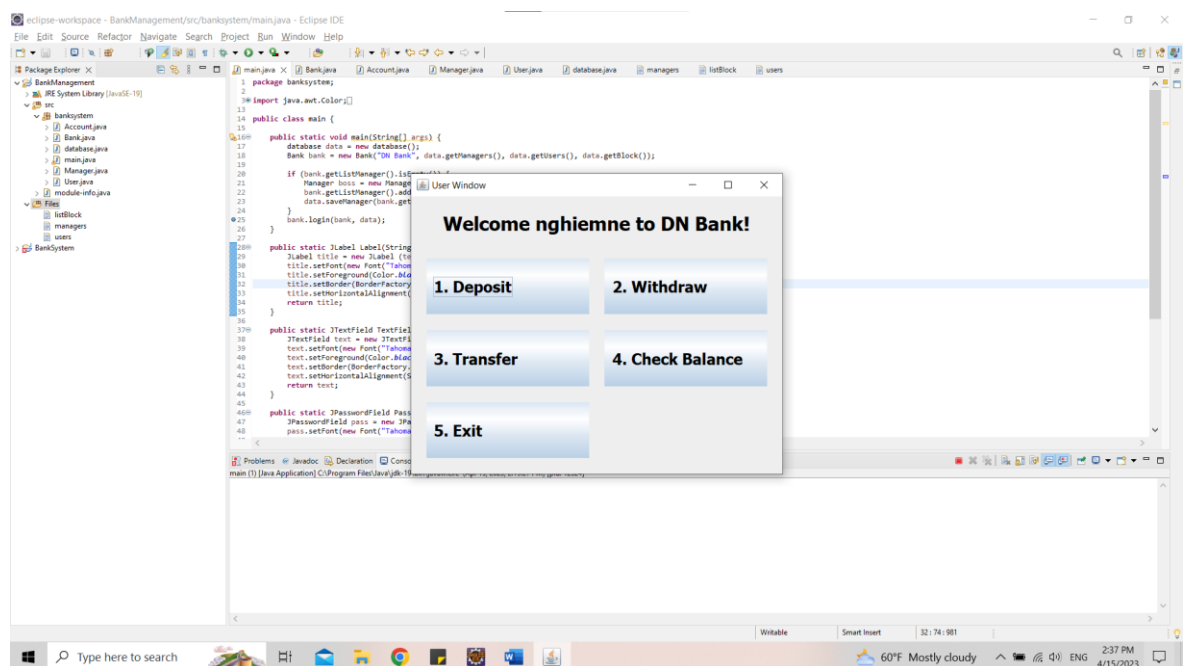
Phương thức mở khoá tài khoản tương tự như phương thức khoá, chỉ cần nhập số tài khoản và mật khẩu của tài khoản đang sử dụng. Sau khi khoá, tài khoản sẽ được chuyển từ danh sách người dùng sang danh sách khoá, sau khi mở khoá thì ngược lại. Khi đăng nhập bằng tài khoản có cấp bậc 2, sẽ không thực hiện được tác vụ mở tài khoản quản lý và khoá tài khoản người dùng, tương ứng với phương thức số 1 và 7 trên giao diện.



Hình 12. Hạn chế của tài khoản quản lý cấp 2

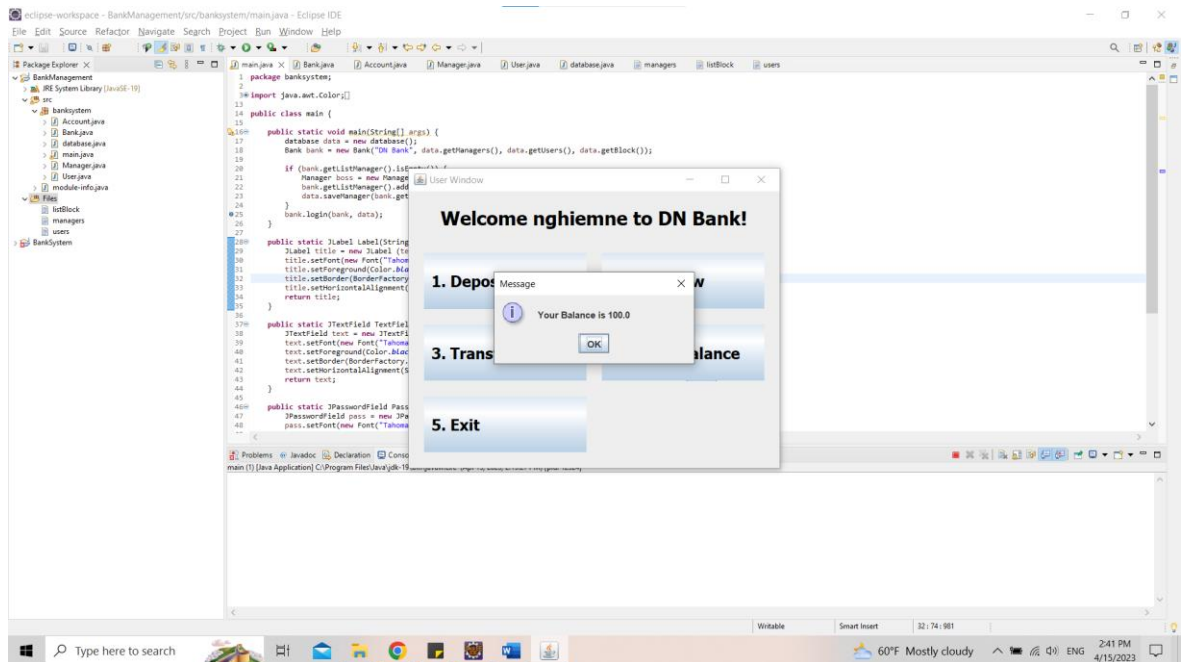


Hình 13. Đăng nhập với tài khoản người dùng



Hình 14. Giao diện của tài khoản người dùng

Ba phương thức gửi, rút và chuyển tiền của người dùng sẽ tương tự khi thao tác bằng tài khoản quản lý, chỉ cần nhập những thông tin đơn giản như số tiền gửi, rút hoặc chuyển, số tài khoản chuyển đến và mật khẩu xác nhận.



Hình 15. Phương thức kiểm tra số dư của tài khoản người dùng

5. Đề xuất hướng phát triển

Một số đề xuất để ứng dụng phát triển hoàn thiện hơn:

- Thêm phương thức cập nhật thông tin người dùng.
- Thêm phương thức ghi lại lịch sử giao dịch của người dùng.
- Các phương thức xóa tài khoản người dùng và tài khoản quản lí.

6. Kết luận

Qua quá trình thực hiện bài tập lớn, em đã có thêm cho mình nhiều kinh nghiệm trong việc sử dụng ngôn ngữ Java để lập trình, hiểu cách để tạo ra giao diện và nắm vững hơn về kiến thức lập trình hướng đối tượng. Tuy class diagram của hệ thống vẫn chưa bao quát hết kiến thức của lập trình hướng đối tượng, các đoạn code còn lặp khá nhiều và ứng dụng vẫn còn thiếu sót, nhưng em tự tin có thể hoàn thiện nó hơn nữa trong tương lai.

Cám ơn thầy đã tạo ra bài tập lớn này để em có cơ hội tìm hiểu và thực hiện các kiến thức mới, nâng cao khả năng lập trình của bản thân.

Tài liệu tham khảo

1. <https://codelearn.io/learning/object-oriented-programming-in-java>
2. <https://www.youtube.com/@techwithbob221>