TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA KHOA HỌC CƠ BẢN

⸎⸎⸎⸎⸎

A logo of a university

Description automatically generated with low confidence

báo cáo tổng hợp

Giảng viên hướng dẫn: Vũ Quang Dũng

Sinh viên : NGUYỄN TUẤN ANH

NGUYỄN NGỌC AN

Lớp : KĨ THUẬT PHẦM MỀM (N06)

NĂM HỌC 2022-2023

## **I. Mở đầu**

## **1.1 Giới thiệu dự án**

## Trong bối cảnh quản lý dân cư ngày càng phức tạp tại các khu chung cư, việc xây dựng một phần mềm quản lý thông tin dân cư trở nên vô cùng cần thiết. Dự án này được thực hiện nhằm đáp ứng nhu cầu quản lý nhân khẩu, hộ khẩu, và các hoạt động đoàn thể tại một tòa nhà chung cư. Phần mềm sẽ hỗ trợ ban quản lý trong việc ghi nhận, theo dõi và quản lý thông tin cư dân một cách hiệu quả và chính xác.

## 1.2 Mục tiêu của dự án

## Mục tiêu chính của dự án là:

## Phát triển phần mềm quản lý thông tin dân cư trong một tòa nhà chung cư.

## Quản lý thông tin nhân khẩu, hộ khẩu, và các nghiệp vụ biến động nhân khẩu (thêm mới, tách hộ, chuyển đi, tạm trú, tạm vắng,...).

## Cung cấp giao diện dễ sử dụng cho ban quản lý để nhập liệu, tra cứu thông tin, và xuất báo cáo.

## 1.3 Phân công nhiệm vụ

## • Trưởng nhóm: Quản lý dự án, điều phối công việc và kiểm tra tiến độ.

## • Phân tích yêu cầu: Khảo sát biểu mẫu và phân tích yêu cầu nghiệp vụ.

## • Phát triển phần mềm: Lập trình và xây dựng cơ sở dữ liệu.

## • Kiểm thử và triển khai: Thực hiện kiểm thử hệ thống và triển khai phần mềm.

## **II. Giao diện**

2.1 Yêu cầu nghiệp vụ

Dự án có giao diện chia thành các phần chính sau:

1. **Trang chủ:** Hiển thị thông tin tổng quan về dự án và liên kết đến các phần khác.
2. **Căn hộ:** Hiển thị danh sách các căn hộ và cho phép tìm kiếm bằng mã căn hộ. Khi nhập mã căn hộ hợp lệ, thông tin căn hộ sẽ được hiển thị ở bảng phía dưới.
3. **Hóa đơn:** Hiển thị danh sách các hóa đơn, bao gồm hóa đơn điện, nước, dịch vụ và tổng tiền. Có chức năng thanh toán và chỉnh sửa cho quản trị viên.
4. **Quản lý xe:** Dành cho quản lý thông tin về xe đỗ trong khu căn hộ.
5. **Quản lý nhân viên:** Cho phép quản lý thông tin về nhân viên làm việc trong khu căn hộ.
6. **Chăm sóc khách hàng:** Dành cho quản lý thông tin và liên hệ với khách hàng.
7. **Trung tâm người dùng:** Hiển thị thông tin cá nhân của người dùng, cho phép thay đổi mật khẩu và cập nhật thông tin cá nhân.
8. **Đăng nhập:** Cho phép người dùng đăng nhập bằng tài khoản và mật khẩu hoặc đăng ký tài khoản mới nếu chưa có.

### **Người dùng:**

Về phần đầu tiên mà người dùng nhìn thấy thì sẽ là giao diện của đăng nhập, Người dùng sẽ được cấp tên đăng nhập và mật khẩu từ quản trị viên tòa nhà dùng để truy cập vào hệ thống của tòa nhà. Nếu người dùng quên mật khẩu của mình thì có thể vào phần quên mật khẩu để lấy lại mật khẩu.

### **Admin:**

Quản trị viên sẽ có tài khoản và mật khẩu đặc biệt để truy cập vào tài khoản admin của hệ thống cho phép chỉnh sửa nhiều tác vụ bên trong đó, admin có thể tạo tài khoản, mật khẩu cho người dùng hoặc cho 1 admin khác

## **Quản lý Căn hộ**

Thông tin chi tiết về căn hộ gồm có:

* Mã căn hộ
* Giá
* Tầng
* Diện tích
* Loại căn hộ (thường hay xịn)
* Tình trạng đã có người thuê hay chưa

III.Thiết kế hệ thống

* 1. Kiến trúc hệ thống

Hệ thống sẽ gồm các thành phần chính sau:

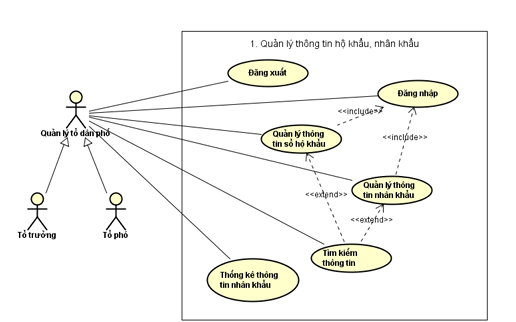
Giao diện người dùng (UI): Được phát triển bằng javaswing

### • Server Backend: Sử dụng ngôn ngữ lập trình như Java (Spring Boot)

### • Cơ sở dữ liệu: Sử dụng MySQL để lưu trữ thông tin hộ khẩu, nhân khẩu.

* 1. Use-case diagram

Use-case diagram mô tả cách người dùng tương tác với hệ thống.



* Actors:
  + Tổ trưởng/tổ phó: Thực hiện tất cả các chức năng quản lý.
  + Cán bộ quản lý: Phụ trách một số nghiệp vụ cụ thể (quản lý tạm trú, quản lý nhân khẩu).
* Use-cases chính:
  + Quản lý hộ khẩu.
  + Quản lý nhân khẩu.
  + Quản lý biến động dân cư.
  + Quản lý tạm trú/tạm vắng.
  + Thống kê và báo cáo

3.3 Thiết kế cơ sở dữ liệu

Bảng Hộ Khẩu (Households):

o household\_id (PK): Mã số hộ khẩu.

o head\_of\_household: Họ tên chủ hộ.

o address: Địa chỉ nhà.

• Bảng Nhân Khẩu (Residents):

o resident\_id (PK): Mã số nhân khẩu.

o name: Họ và tên.

o dob: Ngày sinh.

o relationship: Quan hệ với chủ hộ.

o household\_id (FK): Mã số hộ khẩu liên kết.

• Bảng Biến Động (Changes):

o change\_id (PK): Mã số sự kiện biến động.

o resident\_id (FK): Mã số nhân khẩu.

o change\_type: Loại biến động (thêm mới, chuyển đi, qua đời,...).

o date\_of\_change: Ngày xảy ra biến động.

o details: Chi tiết biến động.

4.1 Phát triển phần mềm

• Lập trình giao diện: Xây dựng các màn hình nhập liệu và báo cáo.

• Lập trình backend: Xây dựng API để xử lý yêu cầu từ giao diện và truy cập cơ sở dữ liệu.

• Tích hợp hệ thống: Liên kết các module giao diện với backend.

4.2 Kiểm thử phần mềm

• Kiểm thử đơn vị (Unit Testing): Kiểm tra từng module riêng lẻ.

• Kiểm thử tích hợp (Integration Testing): Đảm bảo các module làm việc chính xác khi kết hợp với nhau.

• Kiểm thử hệ thống (System Testing): Chạy thử toàn bộ hệ thống để tìm lỗi và sửa chữa.

VI. các Modun:

Lớp DangNhapJDialog

1. Phương thức kiemTraDangNhap(String username, String password)

Phương thức này thực hiện kiểm tra đăng nhập sử dụng JDBC (Java Database Connectivity). Cụ thể, nó thực hiện các bước sau:

Thực hiện kết nối tới cơ sở dữ liệu SQL Server sử dụng URL kết nối được định nghĩa trước đó.

java

Copy code

Connection con = DriverManager.getConnection(connectionUrl);

Chuẩn bị một truy vấn SQL, trong trường hợp này, là truy vấn SQL để kiểm tra thông tin đăng nhập (tên người dùng và mật khẩu).

java

Copy code

String query = "SELECT [dang] FROM [DANGNHAP].[dbo].[ACCOUNT] WHERE [username] = ? AND [pass] = ?";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, username);

ps.setString(2, password);

Thực hiện truy vấn SQL và kiểm tra kết quả. Nếu kết quả chứa một bản ghi, có nghĩa là đăng nhập hợp lệ, và phương thức trả về dang tương ứng.

java

Copy code

ResultSet resultSet = ps.executeQuery();

if (resultSet.next()) {

return resultSet.getInt("dang");

}

2. Phương thức quenMatKhau()

Phương thức này chịu trách nhiệm hiển thị giao diện để người dùng nhập tên tài khoản và địa chỉ Gmail để khôi phục mật khẩu nếu họ quên. Đoạn mã sau hiển thị hộp thoại yêu cầu người dùng nhập thông tin.

java

Copy code

Object[] message = {

"Tài khoản:", txtUsername,

"Địa chỉ Gmail:", txtEmail

};

int option = JOptionPane.showConfirmDialog(this, message, "Nhập thông tin tài khoản và Gmail", JOptionPane.OK\_CANCEL\_OPTION);

Nếu người dùng nhấp OK, phương thức sẽ kiểm tra thông tin tài khoản và địa chỉ Gmail từ cơ sở dữ liệu SQL Server sử dụng truy vấn SQL.

java

Copy code

String password = kiemTraTaiKhoanVaEmail(username, email);

Nếu thông tin đúng, phương thức sẽ hiển thị mật khẩu tương ứng.

Lớp MainJFrame

1. Phương thức openMenuBar() và closeMenuBar()

Các phương thức này thực hiện việc mở và đóng thanh menu bên trái. Khi người dùng nhấp vào nút "menu" (được thể hiện bằng biểu tượng "≡"), openMenuBar() được gọi để mở thanh menu. Khi người dùng nhấp vào biểu tượng đóng (X), closeMenuBar() được gọi để đóng lại thanh menu.

Cách hoạt động của các phương thức này dựa vào việc thay đổi kích thước của các thành phần giao diện để tạo hiệu ứng trượt ra vào.

2. Sự kiện jlbDangNhapMouseClicked

Sự kiện này thực hiện một hành động nào đó khi nhấp chuột vào nhãn "Đăng nhập". Tuy nhiên, phần thân hàm xử lý sự kiện này đã bị xóa.

3. Sự kiện jLabel10MouseClicked và closeMouseClicked

Các sự kiện này xử lý khi người dùng nhấp vào các biểu tượng "menu" và "đóng" bên trái màn hình để mở và đóng thanh menu.

4. Mô-đun DangnhapController

Trong ứng dụng, có một mô-đun quản lý đăng nhập có tên là DangNhapController. Đây là nơi xử lý quá trình đăng nhập của người dùng. Mô-đun này có một số phương thức quan trọng:

setEvent(): Phương thức này thiết lập sự kiện cho nút đăng nhập. Nó kiểm tra xem người dùng đã nhập thông tin đăng nhập chưa, sau đó gọi phương thức kiemTraDangNhap() để kiểm tra đăng nhập thông qua cơ sở dữ liệu. Nếu đăng nhập thành công, nó mở màn hình chính của ứng dụng.

kiemTraDangNhap(): Phương thức này thực hiện kiểm tra đăng nhập sử dụng JDBC. Nó truy vấn cơ sở dữ liệu để kiểm tra xem tên đăng nhập và mật khẩu có đúng hay không. Nếu đúng, nó trả về true, ngược lại trả về false.

5. Mô-đun ChuyenManHinhController

Mô-đun ChuyenManHinhController là một phần quan trọng của ứng dụng, quản lý việc chuyển đổi giữa các màn hình khác nhau. Nó cho phép người dùng điều hướng qua lại giữa các chức năng của ứng dụng. Mô-đun này gồm:

setView(): Phương thức này xác định màn hình hiển thị ban đầu và thiết lập sự kiện cho các mục trong menu.

setEvent(): Phương thức này thiết lập sự kiện cho các mục trong menu. Khi người dùng nhấp vào một mục, nó sẽ chuyển đổi màn hình tương ứng.

LabelEvent: Là một lớp lồng bên trong ChuyenManHinhController, xử lý sự kiện khi người dùng nhấp vào một mục trong menu. Nó xác định mục được chọn và thực hiện chuyển đổi màn hình tương ứng.

Các Phần Khác

Ngoài ra, trong ứng dụng còn có các phần code khác như các JPanel tương ứng với mỗi màn hình, phần xử lý tương tác người dùng và các phần liên quan đến giao diện như các thành phần Swing.

VII. Kết nối java với mysql:

driver: Đây là tên của lớp driver JDBC (Java Database Connectivity) được sử dụng để kết nối với cơ sở dữ liệu SQL Server. Trong trường hợp này, bạn đang sử dụng driver của Microsoft SQL Server. Để sử dụng JDBC để kết nối với SQL Server, bạn cần chắc chắn rằng bạn đã bao gồm thư viện (JAR file) chứa driver này trong classpath của dự án Java của mình. Driver này được sử dụng để thiết lập kết nối giữa ứng dụng Java của bạn và cơ sở dữ liệu SQL Server.

connectionUrl: Đây là URL kết nối (connection URL) sử dụng để xác định cách kết nối với cơ sở dữ liệu SQL Server. URL này chứa các thông tin như địa chỉ máy chủ (ở đây là "localhost"), tên cơ sở dữ liệu (ở đây là "DANGNHAP"), tên người dùng (ở đây là "sa"), và mật khẩu (ở đây là "123456").

Với Statement là một giao diện (interface) trong JDBC được sử dụng để thực hiện các câu lệnh SQL đơn giản và không tham số. Các loại câu lệnh SQL mà Statement hỗ trợ bao gồm SELECT, INSERT, UPDATE và DELETE.

Và ResultSet là một đối tượng trong JDBC được sử dụng để lưu trữ kết quả của một truy vấn SELECT. Khi bạn thực hiện một câu lệnh SELECT bằng Statement hoặc PreparedStatement, kết quả được trả về dưới dạng ResultSet. ResultSet cho phép bạn lặp qua các dòng của kết quả và truy cập dữ liệu từ các cột.

Sau đây là đoạn code về phần đăng nhập dùng mysql để lưu trữ và kiểm tra dữ liệu là username và password :

Sau đây là giải thích về dòng code trên:

private void btnloginMouseClicked(java.awt.event.MouseEvent evt) { ... }: Đây là một phương thức xử lý sự kiện được gọi khi người dùng nhấp chuột vào nút đăng nhập (hoặc sự kiện tương tự).

try (Connection con = DriverManager.getConnection(connectionUrl)) { ... }: Trong khoặc sau khối try, một kết nối đến cơ sở dữ liệu được thiết lập. Biến con đại diện cho kết nối đến cơ sở dữ liệu và sử dụng DriverManager để lấy ra kết nối dựa trên URL kết nối (connectionUrl). Lưu ý rằng bạn cần đảm bảo rằng bạn đã tải driver của cơ sở dữ liệu phù hợp (ví dụ: MySQL JDBC driver) trước khi sử dụng DriverManager.

String sql ="select \* from ACCOUNT where USERNAME=? and PASS=?";: Đây là một câu truy vấn SQL. Câu truy vấn này chọn tất cả các trường từ bảng ACCOUNT trong cơ sở dữ liệu nơi cột USERNAME và cột PASS trùng khớp với giá trị bạn sẽ gửi vào dưới dạng tham số. PreparedStatement ps = con.prepareStatement(sql);: Đối tượng PreparedStatement được tạo ra từ câu truy vấn SQL. Đối tượng này cho phép bạn truyền vào tham số (sử dụng setString()) một cách an toàn mà không cần phải lo lắng về tấn công SQL injection.

ps.setString(1, txtdn.getText()); và ps.setString(2, jfpass.getText());: Hai tham số của câu truy vấn được thiết lập bằng giá trị của hai trường văn bản txtdn (assumed là một trường nhập username) và jfpass (assumed là một trường nhập password) trên giao diện người dùng.

rs = ps.executeQuery();: Câu truy vấn được thực thi bằng cách gọi executeQuery(), và kết quả được trả về dưới dạng ResultSet (đối tượng này chứa kết quả từ câu truy vấn).

if(txtdn.getText().equals("") || jfpass.getText().equals("")) { ... }: Một điều kiện kiểm tra xem trường username hoặc password có rỗng không. Nếu một trong hai trường này rỗng, một hộp thoại thông báo sẽ được hiển thị thông báo cho người dùng.

else if (rs.next()) { ... }: Nếu không có trường nào rỗng, kết quả của câu truy vấn được kiểm tra. Nếu có ít nhất một kết quả (nghĩa là tồn tại một dòng trong bảng ACCOUNT mà username và password trùng khớp), một hộp thoại thông báo đăng nhập thành công sẽ hiển thị.

catch (SQLException e) { ... }: Bất kỳ lỗi nào trong quá trình kết nối cơ sở dữ liệu hoặc thực hiện câu truy vấn sẽ được bắt và xử lý ở đây. Trong trường hợp có lỗi, nó sẽ được in ra console (sử dụng e.printStackTrace()) và có thể được xử lý tiếp theo tùy ý.

Nhận Xét và Các Thư Viện Hỗ Trợ Trong Java:

Nhận Xét:

Tích Hợp Linh Hoạt: JDBC cho phép tích hợp với nhiều hệ quản trị cơ sở dữ liệu khác nhau, không chỉ MySQL. Điều này làm cho ứng dụng có khả năng chuyển đổi từ một hệ thống cơ sở dữ liệu sang hệ thống khác một cách dễ dàng.

Bảo Mật: Sử dụng PreparedStatement để ngăn chặn tấn công SQL injection. Điều này giúp bảo vệ ứng dụng của bạn khỏi các lỗ hổng bảo mật phổ biến.

Các Thư Viện Hỗ Trợ Trong Java:

Hibernate: Hibernate là một ORM (Object-Relational Mapping) framework cho Java. Nó giúp đơn giản hóa việc tương tác với cơ sở dữ liệu bằng cách ánh xạ các đối tượng Java sang bản ghi cơ sở dữ liệu. Hibernate cung cấp một lớp trừu tượng giúp quản lý kết nối và tạo truy vấn SQL.

Spring Data JPA: Spring Data JPA là một phần của Spring Data, nó cung cấp một cách dễ dàng để tương tác với cơ sở dữ liệu thông qua JPA (Java Persistence API). Nó tự động tạo các truy vấn SQL dựa trên tên phương thức, giảm bớt việc viết truy vấn SQL thủ công.

MyBatis: MyBatis là một framework persistence layer cho Java và .NET. Nó sử dụng các file XML hoặc các annotation để xác định cách ánh xạ các truy vấn SQL đến các phương thức Java.

Các thư viện trên giúp giảm bớt công việc lập trình và tăng tính linh hoạt, đồng thời giúp ứng dụng của bạn dễ dàng thích ứng với các thay đổi trong cấu trúc cơ sở dữ liệu hoặc chuyển đổi sang các hệ quản trị cơ sở dữ liệu khác.

VIII. Tổng kết Dự án

quản lý căn hộ và hóa đơn này giúp tạo ra một hệ thống hiệu quả để quản lý thông tin căn hộ, tạo và quản lý hóa đơn cho các loại dịch vụ cung cấp trong căn hộ, và cung cấp chức năng quản lý người dùng. Điều này giúp tối ưu hóa quá trình quản lý khu căn hộ và giúp quản trị viên thực hiện công việc một

Hướng phát triển tiếp theo

• Mở rộng chức năng: Tích hợp thêm các chức năng như quản lý tài sản chung cư, quản lý phí dịch vụ.

• Nâng cấp hệ thống: Tối ưu hóa hiệu suất và cải thiện giao diện người dùng để phù hợp hơn với nhu cầu thực tế.