**1. Giới thiệu**

**Đề tài:** Xây dựng ứng dụng phân chia công việc

**Mục tiêu:** Phát triển một ứng dụng giúp nhóm phân chia và quản lý công việc hiệu quả.

**2. Phân tích yêu cầu**

**Hệ thống gồm hai thành phần chính:**

**1. Server:**

* **Chức năng:**
  + Lưu trữ danh sách công việc và thông tin thành viên.
  + Quản lý kết nối với các client.
  + Cập nhật trạng thái công việc theo hành động của client.
* **Yêu cầu:**
  + Khả năng lưu trữ và quản lý dữ liệu hiệu quả.
  + Khả năng xử lý đồng thời nhiều kết nối từ client.
  + Bảo mật dữ liệu và đảm bảo tính toàn vẹn.

**2. Client:**

* **Chức năng:**
  + Hiển thị danh sách công việc cho thành viên.
  + Cho phép thành viên đăng nhập để nhận công việc.
  + Cho phép thành viên chọn và nhận công việc.
  + Cập nhật trạng thái công việc (đang làm, hoàn thành, trả lại).
  + Giao tiếp với server để gửi/nhận dữ liệu.
* **Yêu cầu:**
  + Giao diện người dùng đơn giản, dễ sử dụng.
  + Hiệu suất hoạt động mượt mà, ổn định.
  + Khả năng kết nối và giao tiếp với server hiệu quả.

**3. Thiết kế hệ thống**

**Mô hình:** Client-server

* **Server:**Chạy trên máy chủ, quản lý dữ liệu và kết nối.
* **Client:**Chạy trên máy cá nhân, tương tác với server để thực hiện thao tác.

**Giao thức truyền tin:** TCP

* **Hướng kết nối:** TCP thiết lập kết nối giữa hai thiết bị trước khi truyền dữ liệu. Kết nối này được duy trì trong suốt quá trình truyền và được đóng sau khi dữ liệu được truyền xong.
* **Đáng tin cậy:** TCP đảm bảo rằng dữ liệu được truyền chính xác và theo đúng thứ tự. Nó sử dụng các cơ chế như kiểm tra lỗi, xác nhận và retransmission để phát hiện và sửa lỗi truyền tải.
* **Có kiểm soát lưu lượng:** TCP điều chỉnh tốc độ truyền dữ liệu dựa trên khả năng xử lý của thiết bị nhận. Điều này giúp tránh tắc nghẽn mạng và đảm bảo rằng dữ liệu được truyền với tốc độ phù hợp cho cả hai thiết bị.
* **Có hỗ trợ nhiều luồng:** TCP cho phép truyền nhiều luồng dữ liệu song song giữa hai thiết bị trên một kết nối duy nhất.

**Kiến trúc phần mềm:**

**Server:**

* Gồm các lớp: Quản lý kết nối, Quản lý công việc, Quản lý thành viên.
* Sử dụng socket server để lắng nghe và xử lý yêu cầu từ client.
* Kết nối với database MySQL để lưu trữ dữ liệu.

**Client:**

* Gồm các lớp: Giao diện người dùng, Giao tiếp mạng, Xử lý nghiệp vụ.
* Giao diện người dùng: Hiển thị danh sách công việc, nút thao tác, thông tin trạng thái.
* Giao tiếp mạng: Gửi/nhận dữ liệu từ server thông qua socket.
* Xử lý nghiệp vụ: Thực hiện các chức năng như nhận công việc, cập nhật trạng thái.

**Cơ sở dữ liệu:**

* Sử dụng MySQL để lưu trữ thông tin công việc, thành viên và trạng thái công việc.
* Thiết kế các bảng dữ liệu:
  + TaiKhoan (tenDangNhap, MatKhau)
  + CongViec (id, tenCongViec, nguoiThucHien, trangThai, version)

**Quản lý đa luồng:**

**Phía client:**

* Sử dụng thư viện SwingWorker trong Java để xử lý các tác vụ tốn thời gian (như cập nhật database).
* Đảm bảo giao diện người dùng mượt mà, không bị gián đoạn.

**Phía server:**

* Chia làm nhiều luồng.
* Sử dụng "optimistic lock" và kết hợp Lock table trên Mysql để tránh xung đột khi thao tác với cơ sở dữ liệu.

**Cách thức hoạt động của "optimistic lock":**

1. Khi một giao dịch muốn cập nhật dữ liệu, nó sẽ đọc giá trị phiên bản hiện tại của dữ liệu đó.
2. Giao dịch thực hiện các thay đổi cần thiết cho dữ liệu.
3. Khi giao dịch cố gắng ghi dữ liệu đã thay đổi, nó sẽ ghi lại cả giá trị phiên bản mới.
4. Hệ thống sẽ kiểm tra xem giá trị phiên bản mới có khớp với giá trị phiên bản hiện tại hay không.
   * Nếu khớp: Giao dịch thành công và dữ liệu được cập nhật.
   * Nếu không khớp: Giao dịch thất bại do dữ liệu đã bị thay đổi bởi giao dịch khác. Giao dịch có thể được thử lại hoặc thông báo lỗi cho người dùng.

**4. Triển khai hệ thống**

**Môi trường phát triển:**

* IDE NetBeans: Môi trường phát triển tích hợp mạnh mẽ cho Java và nhiều ngôn ngữ khác
  + NetBeans là một môi trường phát triển tích hợp (IDE) mã nguồn mở, miễn phí, được sử dụng rộng rãi cho phát triển phần mềm, đặc biệt là các ứng dụng Java. Nó cung cấp nhiều tính năng hỗ trợ lập trình viên viết code, debug, test và triển khai ứng dụng hiệu quả.
* Ngôn ngữ lập trình Java (JDK 17): Phiên bản với nhiều cải tiến

**Thư viện:**

* JDBC (Java Database Connectivity): là API chuẩn giúp ứng dụng Java giao tiếp với database SQL. Nó cung cấp bộ lớp và giao diện để kết nối, truy vấn, thao tác dữ liệu trên nhiều database khác nhau.
  + Ưu điểm:
    - Tiêu chuẩn, dễ sử dụng, linh hoạt
    - Hỗ trợ nhiều database phổ biến (MySQL, PostgreSQL, Oracle, SQL Server, v.v.)
    - Mạnh mẽ với các thao tác dữ liệu phức tạp
* GSON: Chuyển đổi cấu trúc dữ liệu sang dạng chuỗi và ngược lại.
  + GSON (Google SON for Java) là thư viện mã nguồn mở, miễn phí được cung cấp bởi Google, giúp chuyển đổi linh hoạt giữa cấu trúc dữ liệu Java và định dạng JSON. Nó được sử dụng rộng rãi trong lập trình Android, Java backend để đơn giản hóa quá trình trao đổi dữ liệu giữa ứng dụng và API, database.
    - Ưu điểm:
      * Dễ sử dụng: Cung cấp API đơn giản, trực quan để thực hiện chuyển đổi dữ liệu.
      * Hiệu quả: Tốc độ chuyển đổi nhanh chóng, tối ưu hóa cho ứng dụng.
      * Hỗ trợ đa dạng: Khả năng chuyển đổi nhiều loại cấu trúc dữ liệu Java sang JSON và ngược lại.
      * Tích hợp sẵn: Hỗ trợ tích hợp sẵn với các framework phổ biến như Spring MVC, Jackson.
* Socket: Giao tiếp mạng.

**Cấu trúc hệ thống:**

**Server:**

* Lớp Server: Lắng nghe kết nối từ client, khởi tạo luồng riêng cho mỗi kết nối.
* Lớp DatabaseHelper: Kết nối và thao tác với cơ sở dữ liệu.
* Lớp ClientHandler: Xử lý thông tin khi client gửi đến.
* Lớp TaiKhoan: Chuyển đổi cấu trúc dữ liệu tài khoản sang JSON.
* Lớp CongViec: Tương tự lớp TaiKhoan với dữ liệu công việc.
* Interface ClientHandlerDisconnected: Ngắt kết nối client.

**Client:**

* Giao diện người dùng: Thiết kế với Swing.
* Gói com.acer.component: Chứa các thiết kế Header và Menu.
* Gói com.acer.event: Giao diện đăng ký sự kiện.
  + EventMenuSelected: Bắt sự kiện click vào menu.
  + ServerMessageListener: Lắng nghe thông tin từ server.
* Gói com.acer.form: Các form chính người dùng thao tác.
* Gói com.acer.glasspanepopup: Xử lý popup.
* Gói com.acer.icon: Chứa icon chương trình.
* Gói com.acer.model: Xử lý dữ liệu cho các class khác.
* Gói com.acer.socket: Lớp ClientSocket kết nối với socket server.
* Gói com.acer.swing: Thiết kế và logic giao diện người dùng.
* Lớp ClientSocket: Kết nối với server, khởi tạo luồng vào/ra để gửi dữ liệu.
* Lớp ServerListener: Lắng nghe thông tin từ server, cung cấp cho các lớp đã đăng ký.

**Cấu hình database:**

* 2 bảng: TaiKhoan (**tenDangNhap**, MatKhau) và CongViec (**id**, tenCongViec, nguoiThucHien, trangThai, version).

**5. Chương trình demo:**

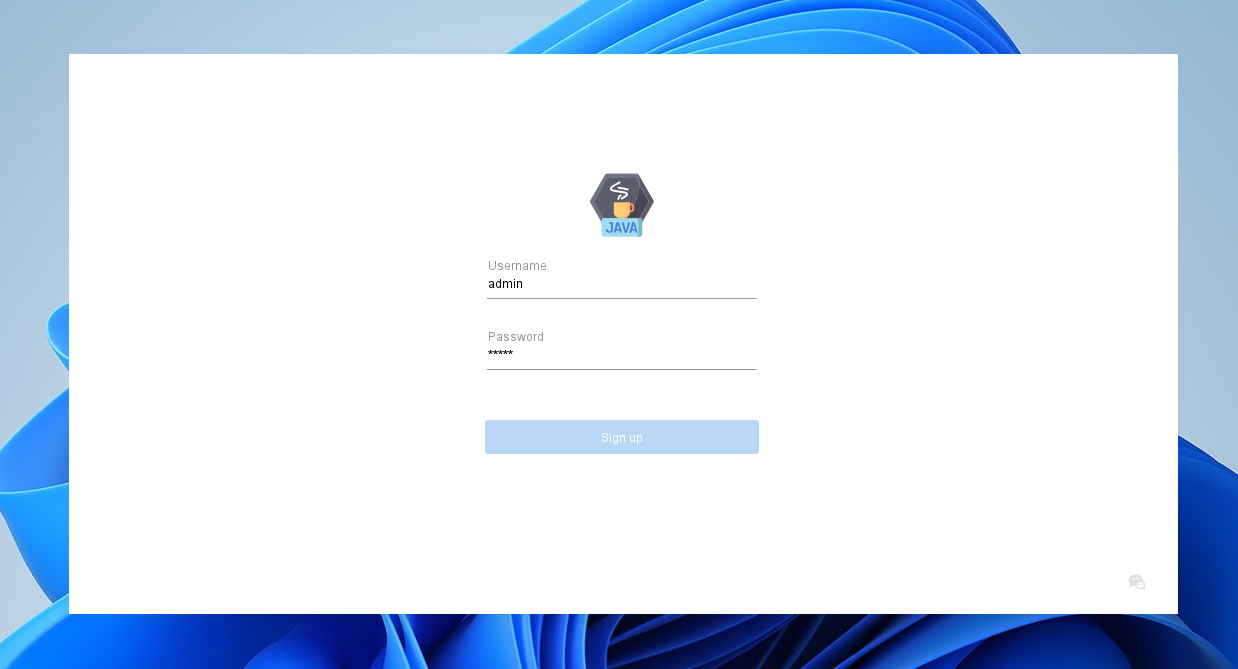


Figure 1Giao diện đăng nhập phía client

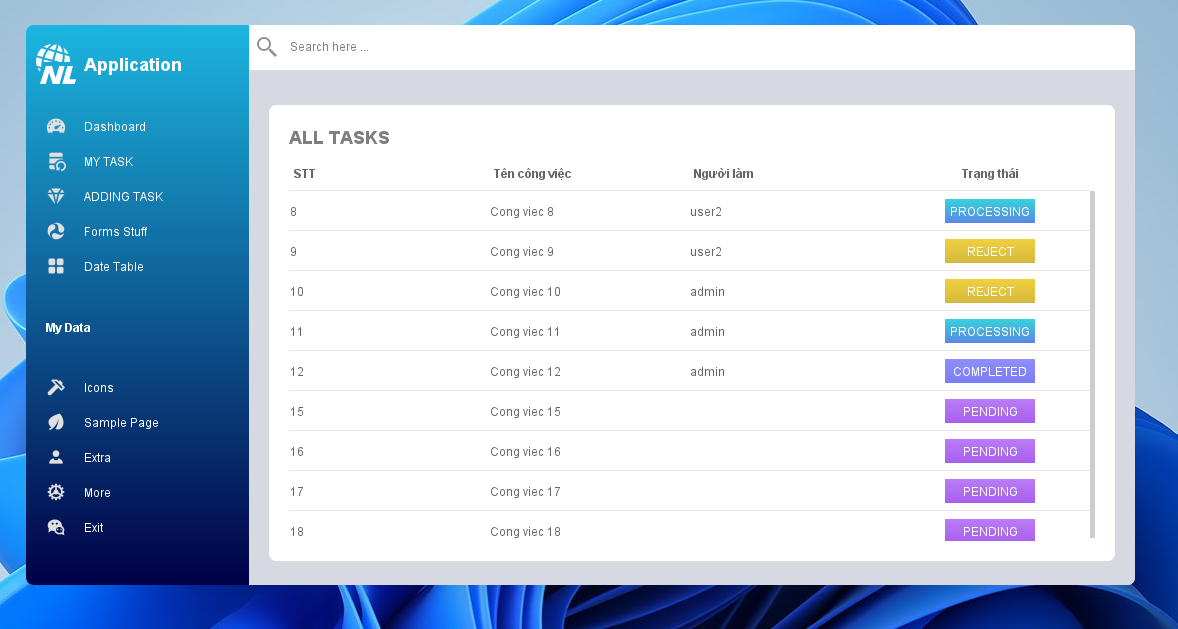


Figure 2Giao diện các công việc

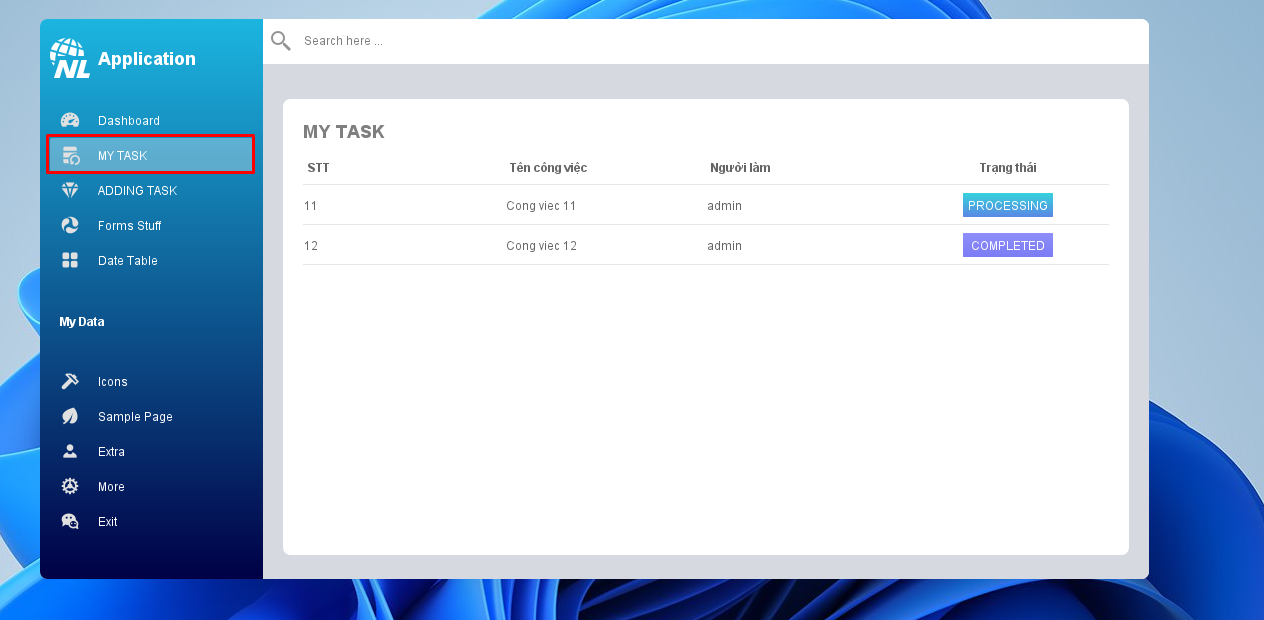


Figure 3Giao diện chức năng công việc của tôi

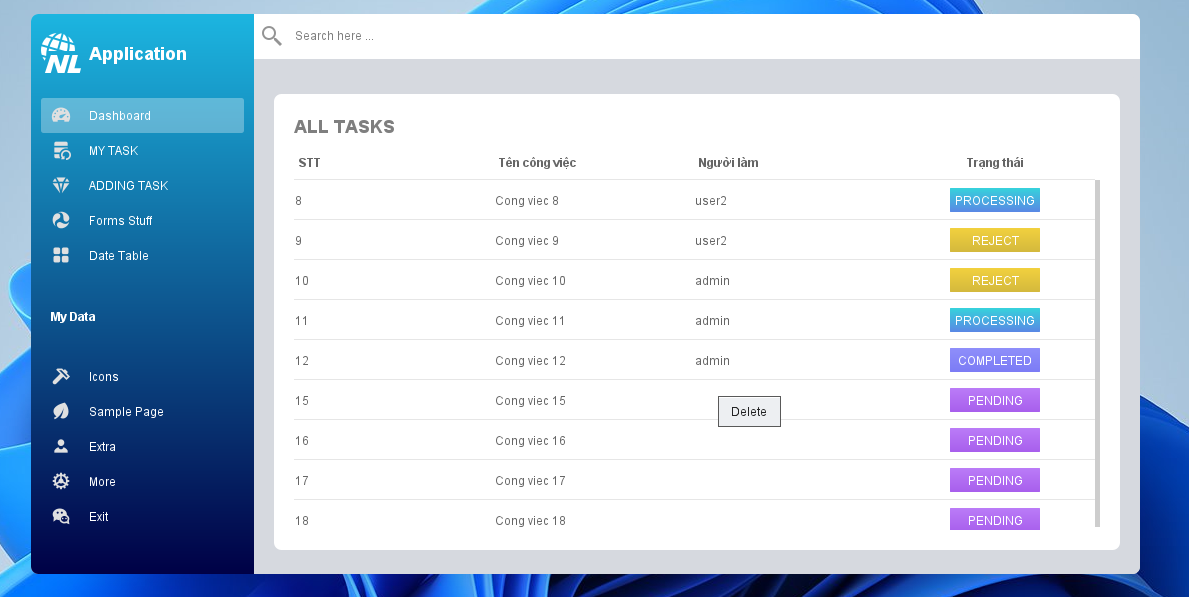
****

Figure 4Giao diện chức năng xóa công việc

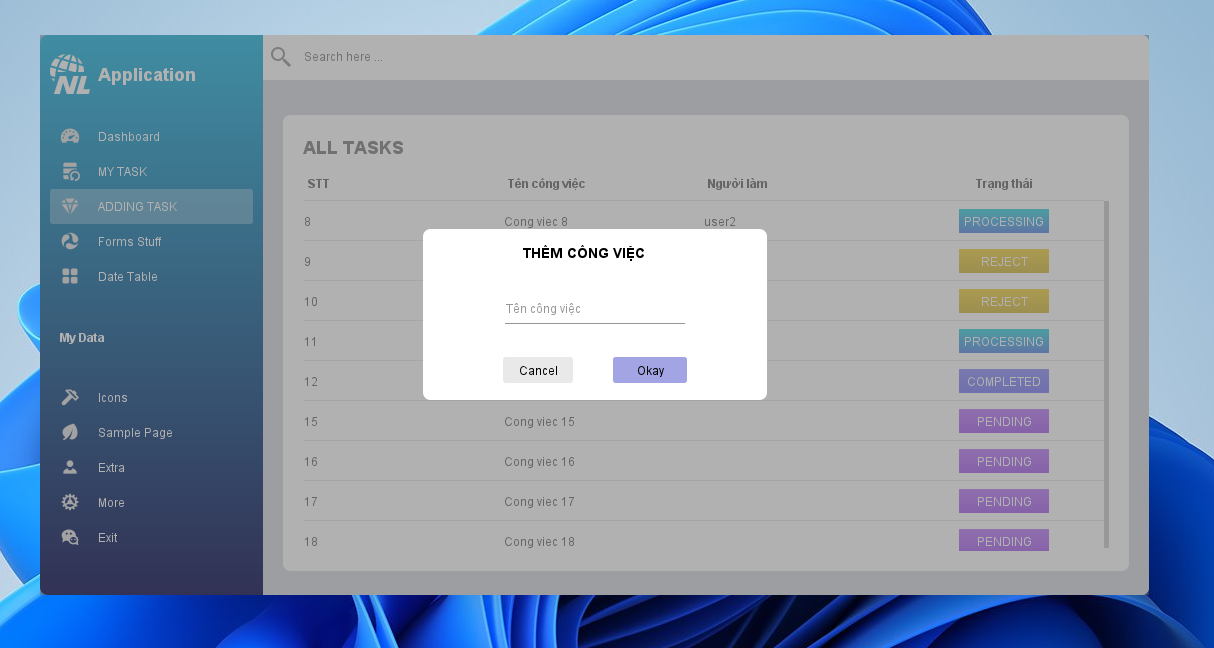


Figure 5Giao diện chức năng thêm công việc

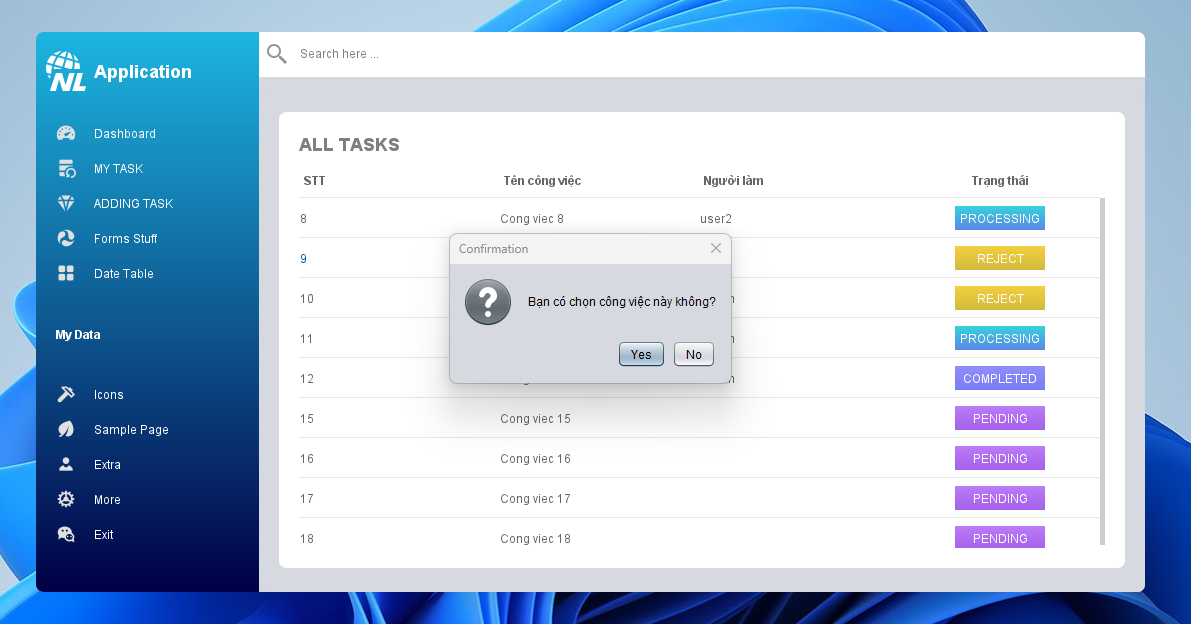


Figure 6Giao diện chức năng chọn công việc

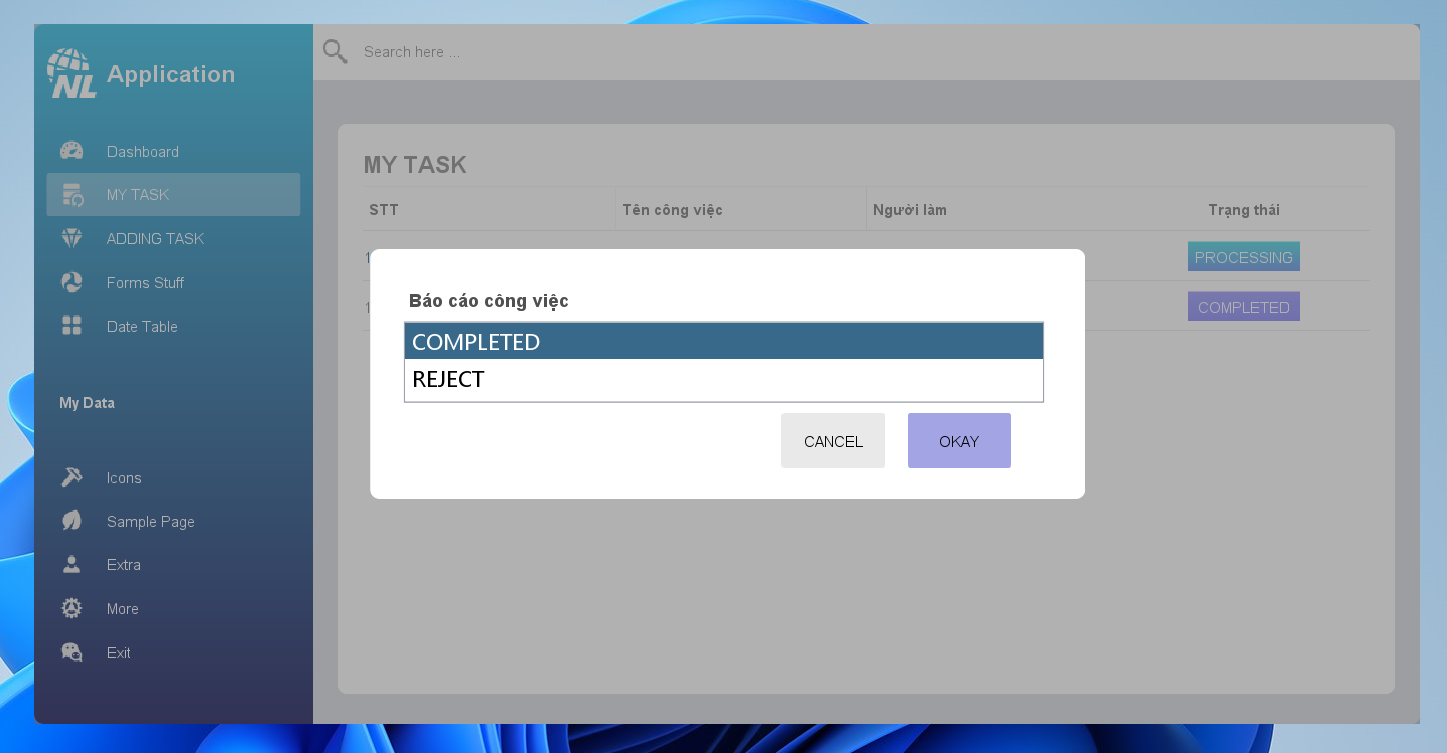


Figure 7Giao diện chức năng báo cáo công việc

**6. Kết luận**

**Qua quá trình thực hiện dự án "Xây dựng ứng dụng phân chia công việc" bằng công nghệ Java, em đã gặt hái được nhiều kiến thức và kinh nghiệm quý giá trong lĩnh vực lập trình và phát triển phần mềm.**

**1. Nâng cao kỹ năng Java:**

* Thành thạo hơn về khái niệm interface và ứng dụng của nó để tăng tính linh hoạt và tái sử dụng code.
* Nắm vững Java Swing, xây dựng giao diện người dùng hiệu quả và tối ưu hóa trải nghiệm người dùng.
* Hiểu rõ về luồng trong Java và cách quản lý chúng, đảm bảo tính ổn định và xử lý đồng thời các tác vụ.
* Áp dụng thành thạo socket để giao tiếp giữa các thành phần mạng, tăng cường tính tương tác.
* Nâng cao hiểu biết về giao thức mạng TCP và tối ưu hóa truyền thông giữa các máy tính.
* Vận dụng lập trình hướng sự kiện để xây dựng các tính năng tương tác trong ứng dụng.

**2. Khắc phục khó khăn và học hỏi:**

* Trải nghiệm xử lý các vấn đề như quản lý luồng không tối ưu, rút ra kinh nghiệm cho dự án sau.

**3. Định hướng phát triển:**

* Mở rộng ứng dụng desktop sang ứng dụng web.
* Bổ sung tính năng tính lương, sửa thông tin sai, chát, thông báo.
* Nâng cao tính ứng dụng và tương tác.

**4. Tổng kết:**

Dự án "Xây dựng ứng dụng phân chia công việc" đã giúp em:

* Mở rộng kiến thức và kỹ năng lập trình, phát triển phần mềm.
* Hiểu rõ quy trình phát triển ứng dụng từ lập kế hoạch đến triển khai và duy trì.

**Kết luận:**

Với những kiến thức, kỹ năng và kinh nghiệm thu được, em tự tin có thể tiếp tục học hỏi và phát triển trong lĩnh vực lập trình và phát triển phần mềm, góp phần xây dựng những ứng dụng hữu ích và hiệu quả.

**7. Tài liệu tham khảo**

* **Laing'sRaven:** <https://www.youtube.com/channel/UCt6scxq1kdrgeH0YKMlDJdQ> - Kênh YouTube cung cấp các video hướng dẫn về lập trình Java Swing.
* **Java Socket Class in Java:** <https://www.geeksforgeeks.org/socket-programming-in-java/> - Bài viết trên GeeksforGeeks giải thích về lớp Socket trong Java, cách sử dụng nó để tạo kết nối mạng và truyền dữ liệu.
* **Record Locking và Optimistic Locking:** <https://github.com/tikv/tikv> - Bài viết giải thích về hai cơ chế khóa bản ghi trong cơ sở dữ liệu: Record Locking và Optimistic Locking.