### ****Buổi 1: Giới thiệu và Phân tích thiết kế hệ thống****

**Thời gian: 2 tiếng**

#### ****1. Giới thiệu chung (30 phút)****

**1.1. Giới thiệu về hệ thống quản lý quy trình yêu cầu bồi thường bảo hiểm (15 phút)**

* **Mục đích và ý nghĩa của hệ thống:**
  + **Mục đích**: Hệ thống được thiết kế để tự động hóa và quản lý quy trình yêu cầu bồi thường bảo hiểm, giúp tiết kiệm thời gian, giảm sai sót và tăng hiệu quả trong xử lý yêu cầu.
  + **Ý nghĩa**: Giúp tổ chức cải thiện quy trình xử lý yêu cầu, từ việc tiếp nhận yêu cầu, xác minh thông tin, xử lý bồi thường đến việc lưu trữ dữ liệu và báo cáo.
* **Các vấn đề chính mà hệ thống sẽ giải quyết:**
  + **Quản lý yêu cầu bồi thường**: Theo dõi và quản lý các yêu cầu bồi thường từ khi được gửi đến khi hoàn tất.
  + **Xử lý dữ liệu**: Tự động hóa việc xử lý và xác minh thông tin yêu cầu.
  + **Báo cáo và phân tích**: Cung cấp công cụ để theo dõi và phân tích quy trình yêu cầu bồi thường.
* **Lợi ích của việc sử dụng hệ thống này cho tổ chức:**
  + **Tăng cường hiệu quả**: Giảm thời gian xử lý và các lỗi thủ công.
  + **Cải thiện dịch vụ khách hàng**: Cung cấp phản hồi nhanh chóng và chính xác cho khách hàng.
  + **Quản lý tốt hơn**: Theo dõi quy trình yêu cầu và quản lý dữ liệu hiệu quả hơn.

**1.2. Các công nghệ chính sử dụng (15 phút)**

* **Java:**
  + **Tầm quan trọng và vai trò trong dự án:**
    - **Ngôn ngữ lập trình chính**: Java sẽ được sử dụng để viết mã nguồn của ứng dụng.
    - **Đặc điểm**: An toàn, hiệu suất cao, và hỗ trợ mạnh mẽ cho các ứng dụng doanh nghiệp.
  + **Các phiên bản và lý do chọn Java:**
    - **Phiên bản**: Java 17 (hoặc phiên bản mới hơn nếu cần).
    - **Lý do chọn**: Tính ổn định, hỗ trợ lâu dài, và cộng đồng lớn.
* **Spring Boot:**
  + **Giới thiệu về Spring Boot và các tính năng chính:**
    - **Khái niệm**: Spring Boot là một framework giúp dễ dàng phát triển ứng dụng Java với cấu hình tự động và tích hợp sẵn.
    - **Tính năng**: Cấu hình tự động, tích hợp dễ dàng với các công nghệ khác, quản lý phụ thuộc.
  + **Lợi ích khi sử dụng Spring Boot cho dự án:**
    - **Tiết kiệm thời gian**: Cấu hình đơn giản và dễ dàng khởi tạo dự án.
    - **Tính linh hoạt**: Tích hợp dễ dàng với các thư viện và công nghệ khác.
* **MySQL:**
  + **Giới thiệu về MySQL và lý do chọn MySQL:**
    - **Khái niệm**: MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở.
    - **Lý do chọn**: Độ tin cậy cao, dễ sử dụng, và hiệu suất tốt cho các ứng dụng web.
  + **Các tính năng nổi bật của MySQL:**
    - **Khả năng mở rộng**: Hỗ trợ lưu trữ và quản lý khối lượng dữ liệu lớn.
    - **Hiệu suất cao**: Tối ưu hóa các truy vấn và quản lý dữ liệu.
* **Docker:**
  + **Khái niệm Docker và containerization:**
    - **Khái niệm**: Docker là nền tảng để xây dựng, chạy, và quản lý các container.
    - **Containerization**: Tách biệt ứng dụng và các phụ thuộc của nó vào các container để dễ triển khai và quản lý.
  + **Lợi ích của việc sử dụng Docker trong triển khai ứng dụng:**
    - **Tính di động**: Chạy ứng dụng trên bất kỳ môi trường nào.
    - **Nhất quán**: Đảm bảo môi trường phát triển và sản xuất giống nhau.

#### ****2. Phân tích yêu cầu (30 phút)****

**2.1. Xác định các yêu cầu nghiệp vụ và chức năng (15 phút)**

* **Yêu cầu nghiệp vụ:**
  + **Các quy trình bồi thường bảo hiểm hiện tại:**
    - Mô tả quy trình từ việc khách hàng gửi yêu cầu bồi thường đến khi yêu cầu được xử lý và thanh toán.
    - **Các bước trong quy trình:**
      1. **Tiếp nhận yêu cầu**: Khách hàng gửi yêu cầu bồi thường qua hệ thống hoặc trực tiếp. Hệ thống tiếp nhận yêu cầu và tạo hồ sơ yêu cầu bồi thường.
      2. **Xác minh thông tin**: Nhân viên xử lý yêu cầu xác minh thông tin do khách hàng cung cấp. Kiểm tra các tài liệu và chứng từ liên quan. Xác nhận tính chính xác và đầy đủ của thông tin.
      3. **Đánh giá yêu cầu**: Nhân viên đánh giá yêu cầu dựa trên thông tin và tài liệu đã xác minh. Xác định mức độ hợp lệ và khả năng bồi thường. Có thể yêu cầu thêm thông tin hoặc tài liệu bổ sung từ khách hàng.
      4. **Quyết định bồi thường**: Quyết định bồi thường dựa trên đánh giá yêu cầu. Xác định số tiền bồi thường và các điều kiện đi kèm. Gửi thông báo quyết định cho khách hàng.
      5. **Thanh toán (nếu yêu cầu được chấp nhận)**: Thực hiện thanh toán cho khách hàng theo quyết định bồi thường. Cập nhật trạng thái yêu cầu trong hệ thống. Lưu trữ thông tin và tài liệu liên quan đến giao dịch bồi thường.
      6. **Cập nhật trạng thái (nếu yêu cầu bị từ chối)**: Cập nhật trạng thái yêu cầu là "Từ chối" trong hệ thống và lưu trữ thông tin liên quan đến việc từ chối yêu cầu.
  + **Các vấn đề và nhu cầu của người sử dụng hệ thống:**
    - **Vấn đề**: Thủ tục phức tạp, thời gian xử lý dài, khả năng quản lý dữ liệu kém.
    - **Nhu cầu**: Một hệ thống dễ sử dụng, tự động hóa quy trình, báo cáo và phân tích chính xác.
* **Yêu cầu chức năng:**
  + **Các tính năng cần có trong hệ thống:**
    - **Tiếp nhận yêu cầu**: Giao diện để khách hàng gửi yêu cầu bồi thường.
    - **Xử lý yêu cầu**: Các chức năng để nhân viên xử lý và cập nhật trạng thái yêu cầu.
    - **Theo dõi trạng thái**: Cung cấp báo cáo về trạng thái của các yêu cầu.
    - **Quản lý dữ liệu**: Lưu trữ và truy xuất thông tin yêu cầu bồi thường.
  + **Quy trình và luồng công việc của hệ thống:**
    - Mô tả chi tiết các bước trong quy trình xử lý yêu cầu và cách thức hoạt động của hệ thống.

**2. Phân tích yêu cầu (30 phút)**

**2.1. Mô tả chi tiết yêu cầu (15 phút)**

* **Các yêu cầu chức năng và không chức năng:**
  + **Yêu cầu chức năng:**
    - **Tạo yêu cầu bồi thường:**
      * Khách hàng có thể gửi yêu cầu bồi thường qua hệ thống.
      * Hệ thống sẽ tiếp nhận và tạo hồ sơ yêu cầu bồi thường.
      * Khách hàng có thể đính kèm các tài liệu liên quan (hóa đơn, hình ảnh, biên bản tai nạn, v.v.).
    - **Xử lý yêu cầu bồi thường:**
      * Nhân viên xử lý có thể xem và xác minh thông tin yêu cầu.
      * Nhân viên có thể yêu cầu bổ sung thông tin hoặc tài liệu từ khách hàng nếu cần.
      * Nhân viên đánh giá yêu cầu dựa trên thông tin đã xác minh.
      * Quyết định chấp nhận hoặc từ chối yêu cầu.
      * Gửi thông báo quyết định cho khách hàng qua hệ thống.
    - **Thanh toán bồi thường:**
      * Thực hiện thanh toán cho khách hàng nếu yêu cầu được chấp nhận.
      * Cập nhật trạng thái yêu cầu trong hệ thống.
    - **Báo cáo và thống kê:**
      * Hệ thống cung cấp báo cáo về số lượng yêu cầu, trạng thái yêu cầu, tổng số tiền bồi thường, v.v.
      * Nhân viên có thể truy xuất các báo cáo theo nhiều tiêu chí (ngày tháng, khách hàng, trạng thái, v.v.).
  + **Yêu cầu không chức năng:**
    - **Hiệu suất:**
      * Hệ thống phải có khả năng xử lý ít nhất 1000 yêu cầu bồi thường đồng thời mà không làm giảm hiệu suất.
      * Thời gian phản hồi của hệ thống không quá 2 giây cho mỗi yêu cầu.
    - **Bảo mật:**
      * Dữ liệu khách hàng và yêu cầu bồi thường phải được mã hóa khi lưu trữ và truyền tải.
      * Hệ thống phải có cơ chế xác thực và phân quyền để đảm bảo chỉ những người có thẩm quyền mới có thể truy cập và xử lý yêu cầu.
    - **Khả năng mở rộng:**
      * Hệ thống phải có khả năng mở rộng để đáp ứng nhu cầu tăng trưởng số lượng yêu cầu bồi thường trong tương lai.
      * Dễ dàng tích hợp với các hệ thống khác (như CRM, hệ thống kế toán, v.v.).
* **Các điều kiện và giới hạn của hệ thống:**
  + **Điều kiện hoạt động của hệ thống:**
    - Hệ thống phải hoạt động liên tục 24/7, đảm bảo tính sẵn sàng cao.
    - Hệ thống phải tuân thủ các quy định và tiêu chuẩn về bảo mật dữ liệu.
  + **Các ràng buộc kỹ thuật và tổ chức:**
    - Phải sử dụng công nghệ Java, Spring Boot, MySQL, và Docker.
    - Phải tuân theo quy trình phát triển phần mềm Agile.
    - Phải đảm bảo hệ thống hoạt động tốt trên cả môi trường thử nghiệm và sản xuất.

#### ****3. Thiết kế kiến trúc hệ thống (1 giờ)****

### 3.1. Mô hình hóa quy trình yêu cầu bồi thường (30 phút)

**Sơ đồ quy trình:**

**Xác định các bước chính trong quy trình yêu cầu bồi thường:**

* 1. **Tiếp nhận yêu cầu:**
     + **Vai trò:** Khách hàng
     + **Mô tả:** Khách hàng gửi yêu cầu bồi thường qua hệ thống hoặc trực tiếp tại công ty bảo hiểm. Hệ thống tiếp nhận yêu cầu và tạo hồ sơ yêu cầu bồi thường, bao gồm các thông tin chi tiết về yêu cầu và các tài liệu đính kèm (hóa đơn, hình ảnh, biên bản tai nạn, v.v.).
  2. **Xác minh thông tin:**
     + **Vai trò:** Nhân viên xử lý
     + **Mô tả:** Nhân viên xử lý tiếp nhận hồ sơ yêu cầu bồi thường từ hệ thống. Họ kiểm tra tính chính xác và đầy đủ của thông tin và tài liệu do khách hàng cung cấp. Nếu cần thiết, nhân viên xử lý có thể yêu cầu bổ sung thêm thông tin hoặc tài liệu từ khách hàng.
  3. **Đánh giá yêu cầu:**
     + **Vai trò:** Nhân viên xử lý
     + **Mô tả:** Nhân viên xử lý xem xét và phân tích yêu cầu bồi thường dựa trên các thông tin và tài liệu đã được xác minh. Họ xác định mức độ hợp lệ và khả năng bồi thường cho yêu cầu. Trong trường hợp cần thiết, họ có thể tham khảo thêm ý kiến từ các chuyên gia hoặc bộ phận khác để đưa ra quyết định chính xác.
  4. **Quyết định bồi thường:**
     + **Vai trò:** Quản lý
     + **Mô tả:** Quản lý nhận thông tin từ nhân viên xử lý và đưa ra quyết định cuối cùng về việc chấp nhận hoặc từ chối yêu cầu bồi thường. Quản lý xác định số tiền bồi thường và các điều kiện đi kèm nếu yêu cầu được chấp nhận. Quyết định này sau đó được thông báo cho khách hàng thông qua hệ thống.
  5. **Thanh toán (nếu yêu cầu được chấp nhận):**
     + **Vai trò:** Nhân viên xử lý
     + **Mô tả:** Nếu yêu cầu bồi thường được chấp nhận, nhân viên xử lý thực hiện thanh toán cho khách hàng theo quyết định của quản lý. Thông tin về giao dịch thanh toán được cập nhật trong hệ thống.
  6. **Cập nhật trạng thái (nếu yêu cầu bị từ chối):**
     + **Vai trò:** Nhân viên xử lý
     + **Mô tả:** Nếu yêu cầu bồi thường bị từ chối, nhân viên xử lý cập nhật trạng thái yêu cầu là "Từ chối" trong hệ thống và lưu trữ thông tin liên quan đến việc từ chối yêu cầu. Thông báo về việc từ chối cũng được gửi đến khách hàng.

**Các điểm kiểm tra và quyết định trong quy trình:**

* **Điểm xác minh:**
  + **Vai trò:** Nhân viên xử lý
  + **Mô tả:** Tại điểm này, nhân viên xử lý kiểm tra tính chính xác và đầy đủ của thông tin và tài liệu do khách hàng cung cấp. Đây là bước quan trọng để đảm bảo rằng mọi thông tin cần thiết đều được xác thực trước khi tiếp tục quy trình.
* **Điểm quyết định:**
  + **Vai trò:** Quản lý
  + **Mô tả:** Đây là điểm mà quản lý đưa ra quyết định cuối cùng về việc chấp nhận hoặc từ chối yêu cầu bồi thường. Quyết định này dựa trên đánh giá của nhân viên xử lý và các thông tin, tài liệu liên quan.

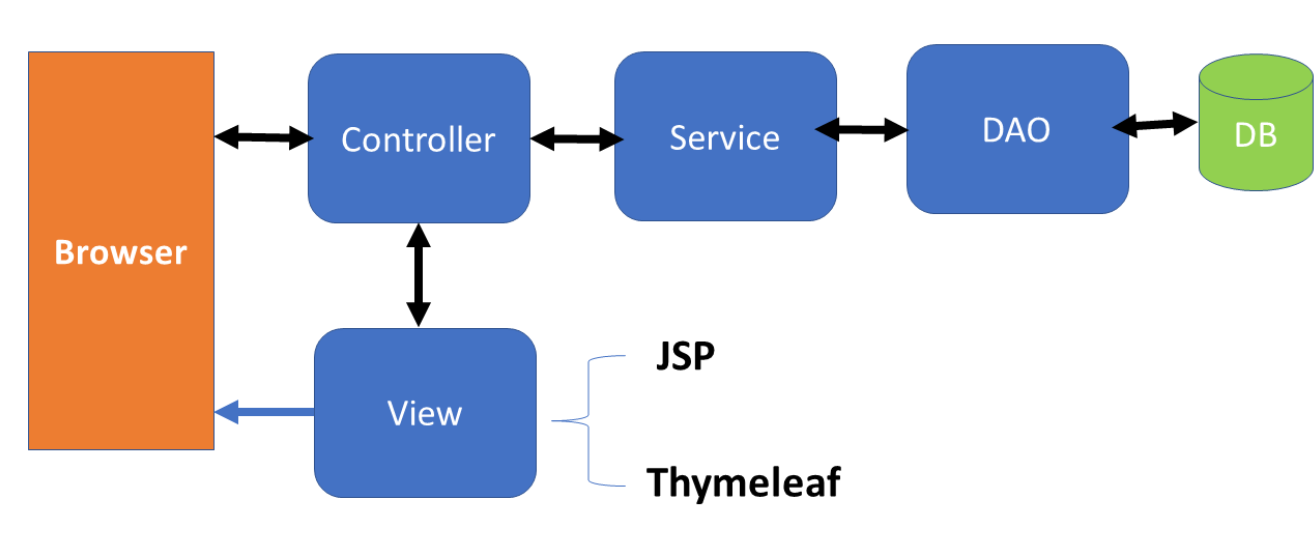
**Các đối tượng và vai trò:**

* **Vai trò của từng đối tượng và sự tương tác giữa chúng:**
  + **Khách hàng gửi yêu cầu:**
    - Khách hàng sử dụng hệ thống để gửi yêu cầu bồi thường, bao gồm thông tin chi tiết và tài liệu đính kèm.
  + **Nhân viên xử lý tiếp nhận, xác minh, và đánh giá yêu cầu:**
    - Nhân viên xử lý kiểm tra và xác minh thông tin từ yêu cầu của khách hàng, sau đó đánh giá mức độ hợp lệ và khả năng bồi thường.
  + **Quản lý đưa ra quyết định cuối cùng và thực hiện thanh toán:**
    - Quản lý xem xét đánh giá từ nhân viên xử lý và quyết định chấp nhận hoặc từ chối yêu cầu. Nếu yêu cầu được chấp nhận, quản lý phê duyệt thanh toán và nhân viên xử lý thực hiện thanh toán cho khách hàng.

### 3.2. Thiết kế các thành phần chính của hệ thống (30 phút)

**Kiến trúc tổng quan:**

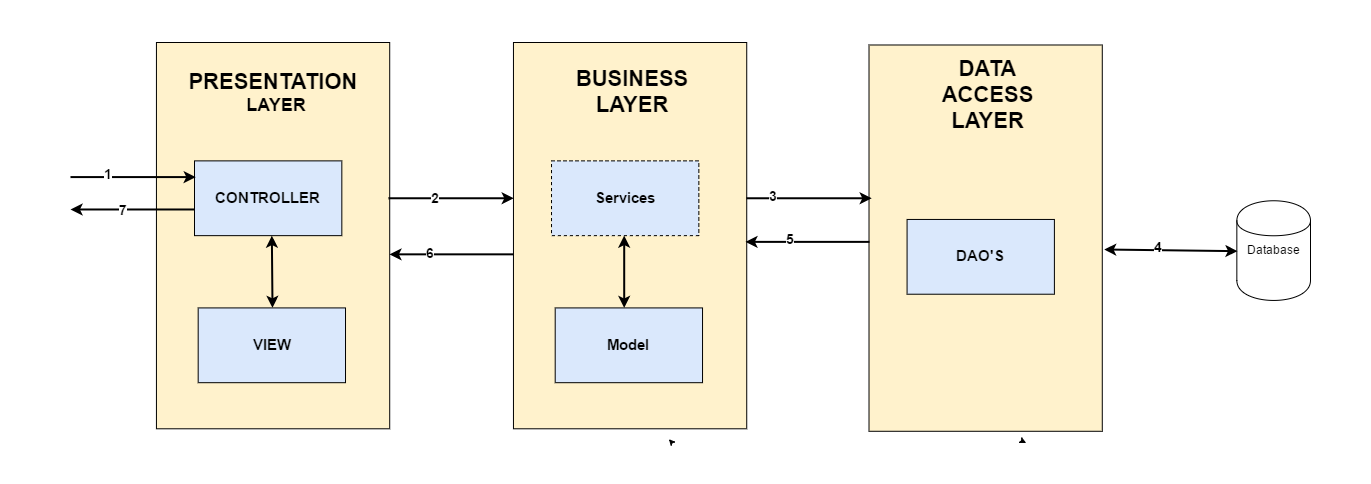
* **Các thành phần chính của hệ thống và mối quan hệ giữa chúng:**
  1. **Frontend:**
     + **Mô tả:** Giao diện người dùng để gửi yêu cầu bồi thường và xem trạng thái yêu cầu. Đây là nơi khách hàng tương tác trực tiếp với hệ thống.
     + **Công nghệ:** HTML, CSS, JavaScript, Thymeleaf.
     + **Vai trò:** Thu thập dữ liệu từ người dùng và hiển thị thông tin phản hồi từ backend.
  2. **Backend:**
     + **Mô tả:** Xử lý yêu cầu và quản lý dữ liệu. Đây là nơi xử lý logic nghiệp vụ, xác minh thông tin, đánh giá yêu cầu, và quyết định bồi thường.
     + **Công nghệ:** Spring Boot.
     + **Vai trò:** Nhận yêu cầu từ frontend, thực hiện xử lý logic và tương tác với cơ sở dữ liệu.
  3. **Database:**
     + **Mô tả:** Lưu trữ dữ liệu yêu cầu bồi thường và thông tin liên quan. Đây là nơi lưu trữ thông tin chi tiết về các yêu cầu bồi thường, trạng thái của chúng, và thông tin khách hàng.
     + **Công nghệ:** MySQL.
     + **Vai trò:** Cung cấp khả năng lưu trữ và truy xuất dữ liệu cho backend.



**Diagrama các lớp và mô-đun trong hệ thống back-end:**

1. **Lớp Presentation:**
   * **Mô tả:** Lớp giao diện người dùng và xử lý các yêu cầu từ người dùng.
   * **Thành phần chính:**
     + Controllers: Nhận yêu cầu từ người dùng và gửi phản hồi từ backend.
     + View Templates: Các trang HTML/Thymeleaf để hiển thị thông tin cho người dùng.
2. **Lớp Service:**
   * **Mô tả:** Lớp xử lý logic ứng dụng và tương tác với dữ liệu.
   * **Thành phần chính:**
     + Services: Chứa logic nghiệp vụ, thực hiện xác minh, đánh giá, và quyết định bồi thường.
     + Business Logic: Các quy tắc nghiệp vụ và quy trình xử lý yêu cầu.
3. **Lớp Data Access:**
   * **Mô tả:** Lớp kết nối và tương tác với cơ sở dữ liệu.
   * **Thành phần chính:**
     + Repositories: Quản lý truy vấn và thao tác dữ liệu trong cơ sở dữ liệu.
     + Entities: Định nghĩa các bảng và quan hệ trong cơ sở dữ liệu.

**Sơ đồ tổng quan:**



**Xác định các dịch vụ và API:**

* **Các dịch vụ chính cần có và các API tương ứng:**
  1. **Dịch vụ yêu cầu bồi thường:**
     + **API tạo yêu cầu bồi thường:**
       - **HTTP Method:** POST
       - **Endpoint:** /api/claims
       - **Mô tả:** Nhận dữ liệu từ người dùng để tạo một yêu cầu bồi thường mới.
     + **API cập nhật yêu cầu bồi thường:**
       - **HTTP Method:** PUT
       - **Endpoint:** /api/claims/{id}
       - **Mô tả:** Cập nhật thông tin của yêu cầu bồi thường dựa trên ID.
     + **API truy xuất yêu cầu bồi thường:**
       - **HTTP Method:** GET
       - **Endpoint:** /api/claims/{id}
       - **Mô tả:** Truy xuất thông tin chi tiết của yêu cầu bồi thường dựa trên ID.
     + **API xóa yêu cầu bồi thư ờng:**
       - **HTTP Method:** DELETE
       - **Endpoint:** /api/claims/{id}
       - **Mô tả:** Xóa yêu cầu bồi thường dựa trên ID.
  2. **Dịch vụ khách hàng:**
     + **API tạo khách hàng:**
       - **HTTP Method:** POST
       - **Endpoint:** /api/customers
       - **Mô tả:** Nhận dữ liệu từ người dùng để tạo một khách hàng mới.
     + **API cập nhật thông tin khách hàng:**
       - **HTTP Method:** PUT
       - **Endpoint:** /api/customers/{id}
       - **Mô tả:** Cập nhật thông tin của khách hàng dựa trên ID.
     + **API truy xuất thông tin khách hàng:**
       - **HTTP Method:** GET
       - **Endpoint:** /api/customers/{id}
       - **Mô tả:** Truy xuất thông tin chi tiết của khách hàng dựa trên ID.
* **Quy trình giao tiếp giữa các dịch vụ và các API:**
  1. **Giao tiếp đồng bộ:**
     + **API RESTful:** Các dịch vụ sử dụng các API RESTful để trao đổi dữ liệu theo mô hình request-response. Ví dụ, frontend gửi yêu cầu tạo yêu cầu bồi thường đến backend qua API, và backend trả về phản hồi cho frontend.
  2. **Giao tiếp không đồng bộ:**
     + **Message Queues:** Sử dụng message queues (như RabbitMQ hoặc Kafka) để xử lý các quy trình phức tạp hoặc các tác vụ yêu cầu thời gian dài. Ví dụ, khi có yêu cầu bồi thường lớn, hệ thống có thể sử dụng message queue để xử lý yêu cầu này không đồng bộ và thông báo cho khách hàng khi hoàn tất.