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

**Thời gian: 2 tiếng**

**1. Giới thiệu chung (30 phút)**

* **Giới thiệu về hệ thống quản lý quy trình yêu cầu bồi thường bảo hiểm**
  + Mục đích và ý nghĩa của hệ thống.
  + Các vấn đề chính mà hệ thống sẽ giải quyết.
  + Lợi ích của việc sử dụng hệ thống này cho tổ chức.
* **Các công nghệ chính sử dụng**
  + **Java**
    - Tầm quan trọng và vai trò trong dự án.
    - Các phiên bản và lý do chọn Java.
  + **Spring Boot**
    - Giới thiệu về Spring Boot và các tính năng chính.
    - Lợi ích khi sử dụng Spring Boot cho dự án.
  + **MySQL**
    - Giới thiệu về MySQL và lý do chọn MySQL.
    - Các tính năng nổi bật của MySQL.
  + **Docker**
    - Khái niệm Docker và containerization.
    - Lợi ích của việc sử dụng Docker trong triển khai ứng dụng.

**2. Phân tích yêu cầu (30 phút)**

* **Xác định các yêu cầu nghiệp vụ và chức năng**
  + **Yêu cầu nghiệp vụ**
    - Các quy trình bồi thường bảo hiểm hiện tại.
    - Các vấn đề và nhu cầu của người sử dụng hệ thống.
  + **Yêu cầu chức năng**
    - Các tính năng cần có trong hệ thống.
    - Quy trình và luồng công việc của hệ thống.
* **Lập bản đặc tả yêu cầu**
  + **Mô tả chi tiết yêu cầu**
    - Các yêu cầu chức năng và không chức năng.
    - Các điều kiện và giới hạn của hệ thống.
  + **Tài liệu đặc tả yêu cầu**
    - Cấu trúc tài liệu yêu cầu.
    - Các thành phần chính của bản đặc tả yêu cầu.

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

* **Mô hình hóa quy trình yêu cầu bồi thường**
  + **Sơ đồ quy trình**
    - Xác định các bước chính trong quy trình yêu cầu bồi thường.
    - Các điểm kiểm tra và quyết định trong quy trình.
  + **Các đối tượng và vai trò**
    - Các đối tượng chính trong quy trình yêu cầu bồi thường.
    - Vai trò của từng đối tượng và sự tương tác giữa chúng.
* **Thiết kế các thành phần chính của hệ thống**
  + **Kiến trúc tổng quan**
    - Các thành phần chính của hệ thống và mối quan hệ giữa chúng.
    - Diagrama các lớp và mô-đun trong hệ thống.
  + **Xác định các dịch vụ và API**
    - Các dịch vụ chính cần có và các API tương ứng.
    - Quy trình giao tiếp giữa các dịch vụ và các API.

**Tổng kết buổi 1** (10 phút)

* Tóm tắt các điểm chính đã được trình bày.
* Nhắc lại các yêu cầu cần chuẩn bị cho buổi tiếp theo.

**Tài liệu tham khảo và chuẩn bị cho buổi tiếp theo** (10 phút)

* Cung cấp tài liệu tham khảo cho học viên.
* Hướng dẫn chuẩn bị cho Buổi 2: Thiết kế database và Giao diện người dùng.

**Buổi 2: Thiết kế database và Giao diện người dùng**

**Thời gian: 2 tiếng**

**1. Thiết kế database (1 giờ)**

* **Tạo sơ đồ ER (Entity-Relationship)** (20 phút)
  + **Khái niệm sơ đồ ER**
    - Giới thiệu các khái niệm cơ bản: thực thể, thuộc tính, mối quan hệ.
  + **Thiết kế sơ đồ ER cho hệ thống**
    - Vẽ sơ đồ ER cho các bảng đã xác định.
    - Xác định các mối quan hệ và thuộc tính của từng thực thể.
* **Xác định các bảng và mối quan hệ giữa các bảng** (20 phút)
  + **Các bảng chính cần có trong hệ thống**
    - Bảng yêu cầu bồi thường.
    - Bảng khách hàng.
    - Bảng sản phẩm bảo hiểm.
    - Bảng trạng thái yêu cầu.
  + **Mối quan hệ giữa các bảng**
    - Quan hệ một-một, một-nhiều và nhiều-nhiều.
    - Các ràng buộc toàn vẹn dữ liệu (foreign keys, unique constraints).
* **Thiết kế chi tiết các bảng và cột** (20 phút)
  + **Chi tiết các bảng**
    - Định nghĩa tên bảng, cột, kiểu dữ liệu và kích thước cột.
    - Các thuộc tính đặc biệt như primary key, auto-increment.
  + **Thiết kế chỉ mục và tối ưu hóa**
    - Xác định các chỉ mục cần thiết cho việc truy vấn dữ liệu hiệu quả.
    - Thực hành tối ưu hóa thiết kế bảng để giảm thiểu vấn đề hiệu suất.

**2. Thiết kế giao diện người dùng (1 giờ)**

* **Nguyên tắc thiết kế UI/UX** (20 phút)
  + **Nguyên tắc cơ bản của UI (User Interface)**
    - Đảm bảo tính nhất quán và dễ sử dụng.
    - Đáp ứng nhu cầu và mong đợi của người dùng.
  + **Nguyên tắc cơ bản của UX (User Experience)**
    - Tạo trải nghiệm người dùng mượt mà và dễ chịu.
    - Tối ưu hóa các yếu tố như tốc độ, độ tin cậy, và sự dễ dàng trong tương tác.
* **Thiết kế mockup và wireframe cho các màn hình chính** (25 phút)
  + **Khái niệm mockup và wireframe**
    - Sự khác biệt giữa mockup và wireframe.
    - Khi nào và tại sao sử dụng chúng.
  + **Thiết kế wireframe cho các màn hình chính**
    - Xác định các màn hình chính của hệ thống (ví dụ: màn hình đăng nhập, quản lý yêu cầu bồi thường, xem báo cáo).
    - Vẽ wireframe để thể hiện cấu trúc và bố cục cơ bản.
  + **Thiết kế mockup chi tiết**
    - Tạo mockup với các yếu tố thiết kế chi tiết như màu sắc, kiểu chữ, và hình ảnh.
* **Công cụ hỗ trợ thiết kế giao diện** (15 phút)
  + **Công cụ phổ biến**
    - Giới thiệu các công cụ thiết kế giao diện như Figma, Adobe XD, Sketch.
  + **Hướng dẫn cơ bản về công cụ**
    - Cách sử dụng công cụ để tạo wireframe và mockup.
    - Các tính năng chính và cách tận dụng chúng trong thiết kế giao diện người dùng.

**Tổng kết buổi 2** (10 phút)

* Tóm tắt các điểm chính đã được trình bày trong buổi học.
* Nhắc lại các yêu cầu cần chuẩn bị cho buổi tiếp theo.

**Tài liệu tham khảo và chuẩn bị cho buổi tiếp theo** (5 phút)

* Cung cấp tài liệu tham khảo cho học viên.
* Hướng dẫn chuẩn bị cho Buổi 3: Phát triển Backend và Tích hợp với Database.

**Buổi 3: Phát triển Backend và Tích hợp với Database**

**Thời gian: 2 tiếng**

**1. Phát triển Backend (1 giờ)**

* **Giới thiệu Spring Boot và các thành phần cơ bản** (20 phút)
  + **Khái niệm Spring Boot**
    - Giới thiệu về Spring Boot và mục đích của việc sử dụng nó.
    - Các tính năng nổi bật của Spring Boot như cấu hình tự động, tích hợp dễ dàng với các công nghệ khác.
  + **Các thành phần cơ bản**
    - **Spring Boot Starter Projects**: Các starter packages cơ bản cho các loại ứng dụng.
    - **Application Properties**: Cấu hình ứng dụng qua file application.properties hoặc application.yml.
    - **Auto-Configuration**: Cấu hình tự động dựa trên các thư viện có sẵn trong dự án.
* **Tạo project Spring Boot** (20 phút)
  + **Khởi tạo dự án**
    - Sử dụng Spring Initializr để tạo project mới.
    - Chọn các dependency cần thiết như Spring Web, Spring Data JPA, MySQL Driver.
  + **Cấu trúc dự án**
    - Giới thiệu cấu trúc thư mục cơ bản trong một ứng dụng Spring Boot.
    - Các thành phần chính như controllers, services, repositories.
* **Xây dựng các API chính của hệ thống** (20 phút)
  + **Tạo các Controller**
    - Tạo các REST controller để xử lý các yêu cầu HTTP.
    - Viết các endpoint cho các chức năng chính như tạo yêu cầu, cập nhật yêu cầu, lấy danh sách yêu cầu.
  + **Tạo các Service**
    - Xây dựng các service để xử lý logic nghiệp vụ.
    - Tạo các phương thức cho các chức năng của hệ thống.
  + **Tạo các Repository**
    - Tạo các repository để tương tác với database.
    - Sử dụng Spring Data JPA để thực hiện các thao tác CRUD.

**2. Tích hợp với Database (1 giờ)**

* **Cấu hình kết nối MySQL** (20 phút)
  + **Cấu hình database trong Spring Boot**
    - Cập nhật application.properties để cấu hình kết nối với MySQL.
    - Các thuộc tính cần thiết như URL, username, password.
  + **Tạo database và bảng**
    - Tạo database và các bảng cần thiết trong MySQL bằng các công cụ như MySQL Workbench hoặc dòng lệnh, intellij
* **Sử dụng JPA/Hibernate để tương tác với database** (20 phút)
  + **Giới thiệu JPA và Hibernate**
    - Khái niệm về JPA (Java Persistence API) và Hibernate.
    - Lợi ích của việc sử dụng JPA/Hibernate để quản lý dữ liệu.
  + **Tạo các Entity Classes**
    - Xây dựng các lớp entity tương ứng với các bảng trong database.
    - Sử dụng các annotation như @Entity, @Table, @Id, @GeneratedValue để cấu hình các lớp entity.
  + **Cấu hình các Repository Interface**
    - Tạo các interface repository kế thừa từ JpaRepository.
    - Viết các phương thức truy vấn tùy chỉnh nếu cần.
* **Thực hiện các thao tác CRUD (Create, Read, Update, Delete)** (20 phút)
  + **Tạo dữ liệu (Create)**
    - Viết mã để tạo các bản ghi mới trong database.
  + **Đọc dữ liệu (Read)**
    - Viết mã để lấy dữ liệu từ database và trả về cho client.
  + **Cập nhật dữ liệu (Update)**
    - Viết mã để cập nhật các bản ghi hiện có.
  + **Xóa dữ liệu (Delete)**
    - Viết mã để xóa các bản ghi không cần thiết.

**Tổng kết buổi 3** (10 phút)

* Tóm tắt các điểm chính đã được trình bày trong buổi học.
* Nhắc lại các yêu cầu cần chuẩn bị cho buổi tiếp theo.

**Tài liệu tham khảo và chuẩn bị cho buổi tiếp theo** (10 phút)

* Cung cấp tài liệu tham khảo cho học viên.
* Hướng dẫn chuẩn bị cho Buổi 4: Triển khai và Docker hóa ứng dụng.

**Buổi 4: Triển khai và Docker hóa ứng dụng**

**Thời gian: 2 tiếng**

**1. Triển khai ứng dụng (1 giờ)**

* **Giới thiệu về Docker và containerization** (20 phút)
  + **Khái niệm Docker**
    - Giới thiệu về Docker và nguyên lý hoạt động của containerization.
    - Các lợi ích của việc sử dụng Docker trong phát triển và triển khai ứng dụng.
  + **Các thành phần chính của Docker**
    - **Docker Images**: Định nghĩa và cách tạo Docker images.
    - **Docker Containers**: Khái niệm về containers và cách chúng hoạt động.
    - **Docker Hub**: Nơi lưu trữ và chia sẻ Docker images.
* **Tạo Dockerfile cho ứng dụng Spring Boot** (20 phút)
  + **Khái niệm Dockerfile**
    - Giới thiệu Dockerfile và mục đích của nó trong việc tạo Docker images.
  + **Viết Dockerfile cho ứng dụng Spring Boot**
    - Tạo file Dockerfile và cấu hình các bước để xây dựng image.
    - Ví dụ cơ bản về cấu hình Dockerfile cho một ứng dụng Spring Boot:

Dockerfile

Sao chép mã

FROM openjdk:17-jdk-slim

COPY target/myapp.jar /app.jar

ENTRYPOINT ["java", "-jar", "/app.jar"]

* **Tạo docker-compose file để quản lý nhiều container** (20 phút)
  + **Khái niệm docker-compose**
    - Giới thiệu về Docker Compose và mục đích của nó trong việc quản lý các dịch vụ đa container.
  + **Viết docker-compose.yml**
    - Tạo file docker-compose.yml để cấu hình và quản lý nhiều container.
    - Ví dụ cơ bản về cấu hình Docker Compose cho ứng dụng Spring Boot và MySQL:

yaml

Sao chép mã

version: '3.8'

services:

app:

image: myapp:latest

ports:

- "8080:8080"

depends\_on:

- db

db:

image: mysql:8.0

environment:

MYSQL\_ROOT\_PASSWORD: rootpassword

MYSQL\_DATABASE: mydb

ports:

- "3306:3306"

**2. Thực hành triển khai (1 giờ)**

* **Xây dựng và chạy ứng dụng trong container** (30 phút)
  + **Xây dựng Docker image**
    - Sử dụng lệnh docker build để xây dựng image từ Dockerfile.
    - Ví dụ: docker build -t myapp:latest .
  + **Chạy container**
    - Sử dụng lệnh docker run để chạy container từ image.
    - Ví dụ: docker run -p 8080:8080 myapp:latest
  + **Kiểm tra ứng dụng**
    - Truy cập ứng dụng qua trình duyệt hoặc công cụ API để đảm bảo nó hoạt động như mong đợi.
* **Quản lý container bằng Docker Compose** (20 phút)
  + **Khởi động các dịch vụ**
    - Sử dụng lệnh docker-compose up để khởi động tất cả các dịch vụ định nghĩa trong docker-compose.yml.
  + **Quản lý và kiểm tra container**
    - Sử dụng các lệnh như docker-compose ps, docker-compose logs để theo dõi trạng thái và logs của các container.
  + **Thực hiện các thao tác cơ bản**
    - Cập nhật Dockerfile và docker-compose.yml.
    - Thực hiện lệnh docker-compose down để dừng và xóa các container.
* **Kiểm tra và debug ứng dụng** (10 phút)
  + **Kiểm tra ứng dụng**
    - Đảm bảo rằng tất cả các thành phần của ứng dụng (backend, database) hoạt động đúng cách trong môi trường container.
  + **Debug ứng dụng**
    - Sử dụng các công cụ debug Docker để kiểm tra và khắc phục sự cố.

**Tổng kết buổi 4** (10 phút)

* Tóm tắt các điểm chính đã được trình bày trong buổi học.
* Nhắc lại các yêu cầu cần chuẩn bị cho buổi tổng kết.

**Q&A và Tổng kết khóa học** (10 phút)

* **Giải đáp thắc mắc của học viên**
  + Trả lời các câu hỏi và vấn đề còn tồn đọng của học viên.
* **Tổng kết và đánh giá lộ trình đào tạo**
  + Đánh giá kết quả học tập và sự tiến bộ của học viên.
* **Đưa ra hướng phát triển tiếp theo**
  + Hướng dẫn các bước tiếp theo cho học viên để tiếp tục học hỏi và phát triển kỹ năng.