**SourceCode**

import java.util.ArrayList;

import java.util.List;

// Product class representing the structure of a product

class Product {

private String productId;

private String productName;

private String category;

// Constructors, getters, and setters

// Method to check if two products are similar

public boolean isSimilar(Product otherProduct) {

// Implement your logic to determine similarity (e.g., same category)

return this.category.equals(otherProduct.getCategory());

}

}

// Database class to store and retrieve products

class ProductDatabase {

private List<Product> products;

public ProductDatabase() {

this.products = new ArrayList<>();

// Initialize the database with some sample data

products.add(new Product("1", "Laptop", "Electronics"));

products.add(new Product("2", "Smartphone", "Electronics"));

products.add(new Product("3", "Running Shoes", "Fashion"));

// Add more sample data as needed

}

// Method to fetch similar products based on product details

public List<Product> getSimilarProducts(Product inputProduct) {

List<Product> similarProducts = new ArrayList<>();

for (Product product : products) {

if (inputProduct.isSimilar(product) && !inputProduct.getProductId().equals(product.getProductId())) {

similarProducts.add(product);

}

}

return similarProducts;

}

}

// E-commerce application class

public class ECommerceApplication {

public static void main(String[] args) {

// Create an instance of the product database

ProductDatabase productDatabase = new ProductDatabase();

// Create an example product

Product inputProduct = new Product("4", "Gaming Laptop", "Electronics");

// Fetch similar products

List<Product> similarProducts = productDatabase.getSimilarProducts(inputProduct);

// Display the results

System.out.println("Similar products for " + inputProduct.getProductName() + ":");

for (Product similarProduct : similarProducts) {

System.out.println(similarProduct.getProductName());

}

}

}

Product details project:

================================

**package** lesson3TDD;

**import** org.testng.Assert;

**import** org.testng.annotations.Test;

**public** **class** TestProductDetails {

@Test

**public** **void** findProductdetails()

{

String product = "camlin";

String expectedProductType = "pencil";

ProductSearch ps = **new** ProductSearch();

String actualProductType = ps.getproductType(product);

System.***out***.println(actualProductType);

Assert.*assertEquals*(actualProductType, expectedProductType);

}

@Test

**public** **void** findproductdetailswithEmptyInput()

{

**try** {

String product = "";

//String expectedProductType = "";

ProductSearch ps = **new** ProductSearch();

String actualProductType = ps.getproductType(product);

}

**catch**(NullPointerException e1)

{

System.***out***.println("product name should not be empty");

}

}

@Test

**public** **void** invalidproductdetails()

{

**try** {

String product = "calindemo";

String expectedProductType = "pencil";

       ProductSearch ps = **new** ProductSearch();

String actualProductType = ps.getproductType(product);

}

**catch** (NullPointerException e1)

{

System.***out***.println("product details doesnot exist");

}

}

}

================

package lesson3TDD;

import java.util.HashMap;

import java.util.Map;

public class ProductSearch {

public Map<String, String> getproductdetails()

{

Map<String, String> productMap = new HashMap<>();

productMap.put("SeleniumBook", "books");

productMap.put("coach", "handbag");

productMap.put("camlin", "pencil");

productMap.put("mac", "laptop");

productMap.put("samsung", "mobile");

return productMap;

}

public String getproductType(String product) {

Map<String, String> productMap = null;

String producttype = null;

if(product.isEmpty()){

throw new NullPointerException("Exception: Product name should not be empty");

}

productMap = getproductdetails();

if(!productMap.containsKey(product))

{

throw new NullPointerException("Exception: Product name is invalid. Details donot exist");

}

else {

producttype = productMap.get(product);

}

return producttype ;

}

}