```
package demo2;

import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.model.Filters;
import org.bson.Document;

import java.util.Scanner;

public class jdbc456 {

    private static MongoClient mongoClient;
    private static MongoDatabase database;
    private static MongoCollection<Document> collection;

    public static void main(String[] args) {

             String uri = "mongodb://127.0.0.1:27017";
             String dbName = "te31265_db";
             String collectionName = "items";


        // Create MongoClient with URI (modern 4.x driver)
```

```
mongoClient = MongoClients.create(uri);
database = mongoClient.getDatabase(dbName);
collection = database.getCollection(collectionName);

System.out.println("Connected to MongoDB database: " + dbName);

Scanner scanner = new Scanner(System.in);
int choice;

do {
    System.out.println("\nCRUD Operations Menu:");
    System.out.println("1. Add Item");
    System.out.println("2. View All Items");
    System.out.println("3. Update Item");
    System.out.println("4. Delete Item");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    choice = scanner.nextInt();
    scanner.nextLine();  // consume newline

    switch (choice) {
        case 1 -> addItem(scanner);
        case 2 -> viewItems();
        case 3 -> updateItem(scanner);
        case 4 -> deleteItem(scanner);
        case 5 -> System.out.println("Exiting...");
```

```
                default -> System.out.println("Invalid choice! Please try again.");
            }

        } while (choice != 5);

        mongoClient.close();
        scanner.close();
    }

    private static void addItem(Scanner scanner) {
        System.out.print("Enter item name: ");
        String name = scanner.nextLine();

        System.out.print("Enter item description: ");
        String description = scanner.nextLine();

        Document doc = new Document("name", name)
                .append("description", description);

        collection.insertOne(doc);
        System.out.println("Item added successfully!");
    }

    private static void viewItems() {
        System.out.println("Items in the collection:");
        try (MongoCursor<Document> cursor = collection.find().iterator()) {
            while (cursor.hasNext()) {
                System.out.println(cursor.next().toJson());
            }
        }
    }

    private static void updateItem(Scanner scanner) {
        System.out.print("Enter the name of the item to update: ");
        String name = scanner.nextLine();

        Document existing = collection.find(Filters.eq("name", name)).first();
        if (existing == null) {
            System.out.println("Item not found.");
            return;
        }

        System.out.print("Enter new description: ");
        String newDescription = scanner.nextLine();

        collection.updateOne(Filters.eq("name", name),
                new Document("$set", new Document("description", newDescription)));

        System.out.println("Item updated successfully!");
    }

    private static void deleteItem(Scanner scanner) {
        System.out.print("Enter the name of the item to delete: ");
        String name = scanner.nextLine();


        long deletedCount = collection.deleteOne(Filters.eq("name", name)).getDeletedCount();


        if (deletedCount > 0) {
            System.out.println("Item deleted successfully!");
        } else {
            System.out.println("Item not found.");
        }
    }
}
```