

```
import java.util.*;
import java.util.stream.Collectors;

public class ElectricityBoardMain {
    private Map<String, String> electricityMap;

    public ElectricityBoardMain() {
        this.electricityMap = new HashMap<>();
    }

    public Map<String, String> getElectricityMap() {
        return electricityMap;
    }

    public void setElectricityMap(Map<String, String> electricityMap) {
        this.electricityMap = electricityMap;
    }

    public int findCountOfConnectionsBasedOnTheConnectionType(String connectionType) {
        if (connectionType == null || connectionType.isEmpty()) {
            return -1;
        }

        int count = 0;
        for (String type : electricityMap.values()) {
            if (type.equalsIgnoreCase(connectionType)) {
                count++;
            }
        }

        return count > 0 ? count : -1;
    }
}
```

```
}
```

```
public List<String> findConnectionIdsBasedOnTheConnectionType(String connectionType) {  
    if (connectionType == null || connectionType.isEmpty()) {  
        return new ArrayList<>();  
    }  
}
```

```
return electricityMap.entrySet().stream()  
    .filter(entry -> entry.getValue().equalsIgnoreCase(connectionType))  
    .map(Map.Entry::getKey)  
    .collect(Collectors.toList());  
}
```

```
public static void main(String[] args) {  
    ElectricityBoardMain electricityBoard = new ElectricityBoardMain();  
    Scanner scanner = new Scanner(System.in);  
  
    System.out.println("Enter the number of connection records to be added");  
    int numRecords = Integer.parseInt(scanner.nextLine());  
  
    System.out.println("Enter the connection records (ConnectionId:Connectiontype)");  
    Map<String, String> records = new HashMap<>();  
    for (int i = 0; i < numRecords; i++) {  
        String[] parts = scanner.nextLine().split(":");  
        if (parts.length == 2) {  
            records.put(parts[0], parts[1]);  
        }  
    }  
    electricityBoard.setElectricityMap(records);  
  
    System.out.println("Enter the Connection type to be searched");  
}
```

```

String searchType = scanner.nextLine();

int count = electricityBoard.findCountOfConnectionsBasedOnTheConnectionType(searchType);

if (count == -1) {
    System.out.println("No Connection Ids were found for " + searchType);
} else {
    System.out.println("The count of connection Ids based on " + searchType + " are " + count);
}

System.out.println("Enter the Connection type to identify the ConnectionIds");

String filterType = scanner.nextLine();

List<String> connectionIds =
electricityBoard.findConnectionIdsBasedOnTheConnectionType(filterType);

if (connectionIds.isEmpty()) {
    System.out.println("No Connection Ids were found for " + filterType);
} else {
    System.out.println("Connection Ids based on the " + filterType + " are");
    connectionIds.forEach(System.out::println);
}
}
}

```