```
#include <iostream>
#include <vector>
#include <algorithm>  // For sort and binary_search
#include <string>

using namespace std;

// Define a structure for personal record
struct PersonalRecord {
    string name;
    string dob;  // Date of Birth in format "DD-MM-YYYY"
    string phoneNumber;

    // Constructor to initialize a record
    PersonalRecord(string n, string d, string p) : name(n), dob(d), phoneNumber(p)
{}

    // Overload < operator to sort records by name
    bool operator<(const PersonalRecord& other) const {
        return name < other.name;
    }

    // Overload == operator to compare records for searching
    bool operator==(const PersonalRecord& other) const {
        return name == other.name;
    }
};

// Function to display a personal record
void displayRecord(const PersonalRecord& record) {
    cout << "Name: " << record.name << ", DOB: " << record.dob << ", Phone: " <<
record.phoneNumber << endl;
}

// Function to display all records
void displayAllRecords(const vector<PersonalRecord>& records) {
    for (const auto& record : records) {
        displayRecord(record);
    }
}

int main() {
    // Vector to store personal records
    vector<PersonalRecord> records;

    // Adding sample records
    records.push_back(PersonalRecord("John Doe", "15-06-1990", "123-456-7890"));
    records.push_back(PersonalRecord("Alice Smith", "20-03-1985", "987-654-3210"));
    records.push_back(PersonalRecord("Bob Johnson", "01-01-2000", "555-123-4567"));

    cout << "Original Records:\n";
    displayAllRecords(records);

    // Sorting the records by name using STL sort and the overloaded < operator
    sort(records.begin(), records.end());

    cout << "\nRecords after sorting by name:\n";
    displayAllRecords(records);
```

```
    // Search for a record (e.g., searching for "Alice Smith")
    PersonalRecord searchRecord("Alice Smith", "", "");
    auto it = find(records.begin(), records.end(), searchRecord);

    if (it != records.end()) {
        cout << "\nRecord found:\n";
        displayRecord(*it);
    } else {
        cout << "\nRecord not found.\n";
    }

    return 0;
}
```