High Level Design for Mobile Search API

Document Revision:

Revision #	Date	Author	Description
1.0	June 6, 2021	Sameh Elsayed	Write HLD for the API

Table of Contents:

1. **Introduction**
    a. **What and Why HLD**
        i. The HLD presents the structure of the system, such as the database architecture if any, application architecture (layers), application routes if any.

        ii. This document will help others to know how the layers/modules/systems interact with each other at high level.

    b. **What is Mobile Search API**
        i. This API is designed and developed to enable the customers to search for the mobile devices they want.

        ii. This API has been developed using Java programming language, Spring boot framework that contains embedded web server to serve the client requests.

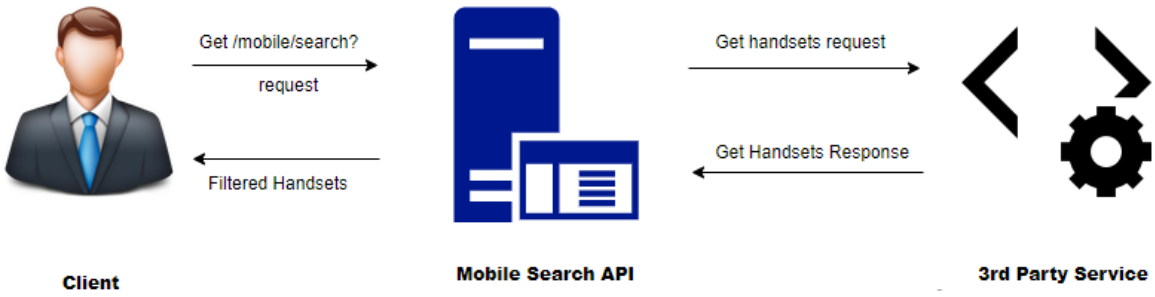
        iii. This API will call 3rd party service to get the handsets which acts as DB.

2. **Design Details**

    a. **Software Environment**
        i. The mobile search API is java based application, so the JRE should be installed on the machine.

        ii. The internet should be available to be able to get the handsets data from the 3rd party service.

b. Application architecture



i. Th mobile search API consists of the below layers.

1. **Controller Layer**
    a. It is used to handle the client request, and map the HTTP paths to methods

2. **Service Layer**
    a. It is used to handle the business logic.

3. **Cache Layer**
    a. It is used to cache the static data to increase the performance.

4. **DAO Layer**
    a. It is used to handle CRUD operations on the models/ entities.

c. Performance

i. Performance is important factor to speed up the request processing.
ii. Because the handsets are static data, so it is better to cache them then start reusing the cached list of handsets.

d. Portability

i. This API is portable so it can be run on any machine that is having JRE installed.

e. Maintainability

i. This API is designed and developed using separate layers to follow the SOLID principles and this will speed up the development process, and if there is an issue, it will be fixed fast.

f. Test Cases

i. This API has test cases to make sure that every single unit is working as expected before sending the artifacts to the staging/ production environments.

g. Response Handling

i. The API will return

1. HTTP status code **200** along with the list of handsets in case there are handsets matched the search criteria.

2. HTTP status code **404** with proper error message in case no handsets matched the search criteria.

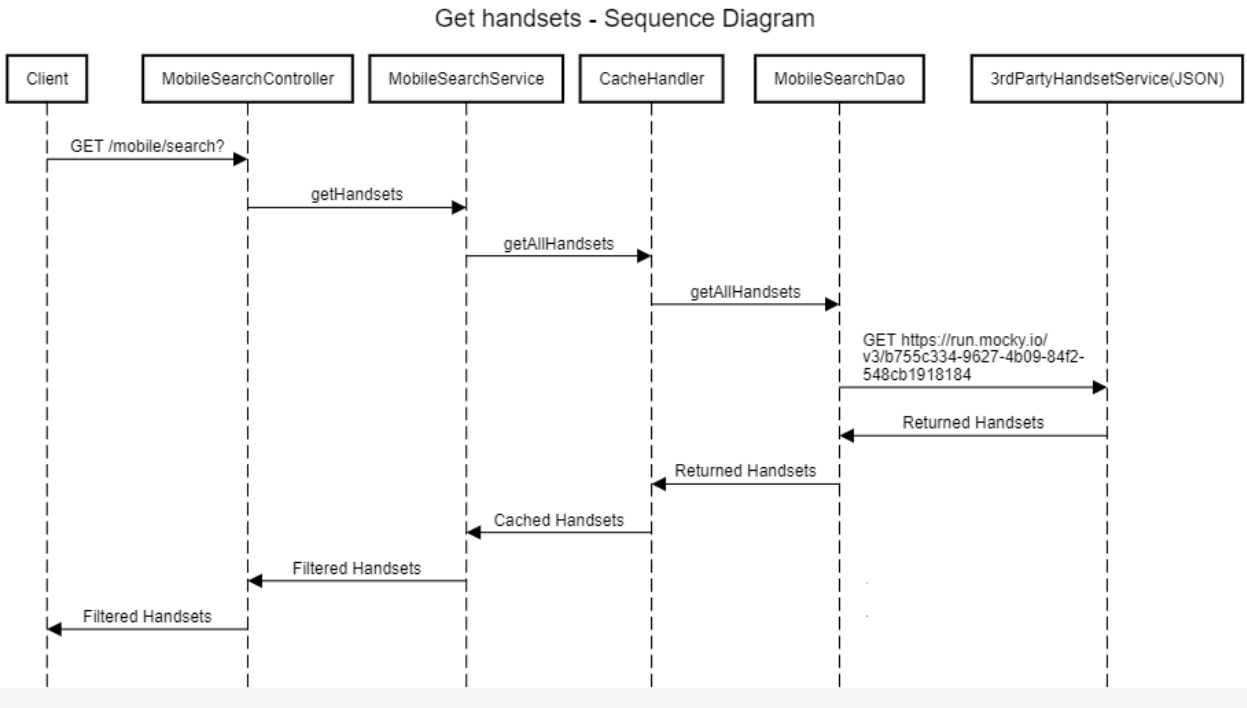
3. HTTP status code **500** with error message “Something went wrong” in case there is a runtime exception.

3. API Sequence Diagram

a. Get Handsets endpoint - Sequence Diagram

i. The below picture shows the sequence diagram for handling the get handsets request.

ii. If the handsets are not cached, so below SD will be followed.



iii. If the handsets are cached, so below SD will be followed.