Technical Case Study – Open Layers

Agile Technical Design

Contents

[Document Control 1](#_Toc131376346)

[Introduction 1](#_Toc131376347)

[Solution Overview 1](#_Toc131376348)

[Client / End User Overview & Requirements 2](#_Toc131376349)

[Solution Architecture 2](#_Toc131376350)

[Software Required 3](#_Toc131376351)

# Document Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Comments |
| 03/04/2022 | 1 | Anand Devarajan | Technical case study |

# Introduction

The purpose of this document is to define the specific technology design, configuration, and testing metrics for the technology components of the technical case study.

It is intended that this document captures the solution design details pertaining to the deployment of technology in question as well as configuration build details and test details and outcomes.

# Solution Overview

The Main objective is to locate the user in Open Layer map and thereby pointing grid stations within radius of given range (default 50KM). The user can also change his latitude and longitude.

# Client / End User Overview & Requirements

Functional requirements:

1. Foremost the core feature is that 50Hertz grid planner can search the nearby grid station in a radius of 50km by default.
2. 50Hertz Grid Planner can add new grid stations on certain Lat, Log and save their basic info and see on the web app.
3. The planner can increase the radius from the UI, search the grid station, and view their contact info. (optional)

Non-Functional Requirements:

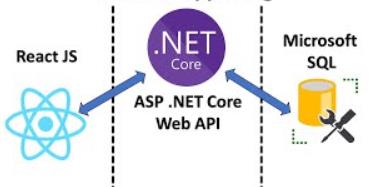
1. The search for nearby grid stations should be fast and have low latency.
2. The design of the web map solution should be very simple

The solution should contain:

1. The data model.
2. Implement APIs in any desired software stack (following an API first approach).
3. On the front end, you can choose any desired software stack and use the open layer https://openlayers.org/ to visualize the map.
4. Documentation is very important for a solution so update the readme with details.
5. Bonus points if you containerize your solution with Docker file.

# Solution Architecture

The application was developed using Microsoft .NET core 6.0 with React JS library along with Microsoft SQL server as a database.



As of now,

* <https://localhost:44485/> has been used for UI
* <https://localhost:44381/api/openlayers> for API.

Database Connection String:

Data Source=LAPTOP-JU3072BK\\SQLEXPRESS;Initial Catalog=OpenLayers;User ID=testserver;Password=\*\*\*\*\*\*\*\*;TrustServerCertificate=True

I have used Code first approach in the solution and DB needs to be configured accordingly.

**Note:**

The code for nearby grid stations has been written. Since we require paid authentication from google place API, the paid key has to be updated in the code for pointing the markers in Open Layers.

# Software Required

|  |
| --- |
| Visual Studio 2022 |
| Microsoft SQL Server |
| Node js |

Screenshots:

