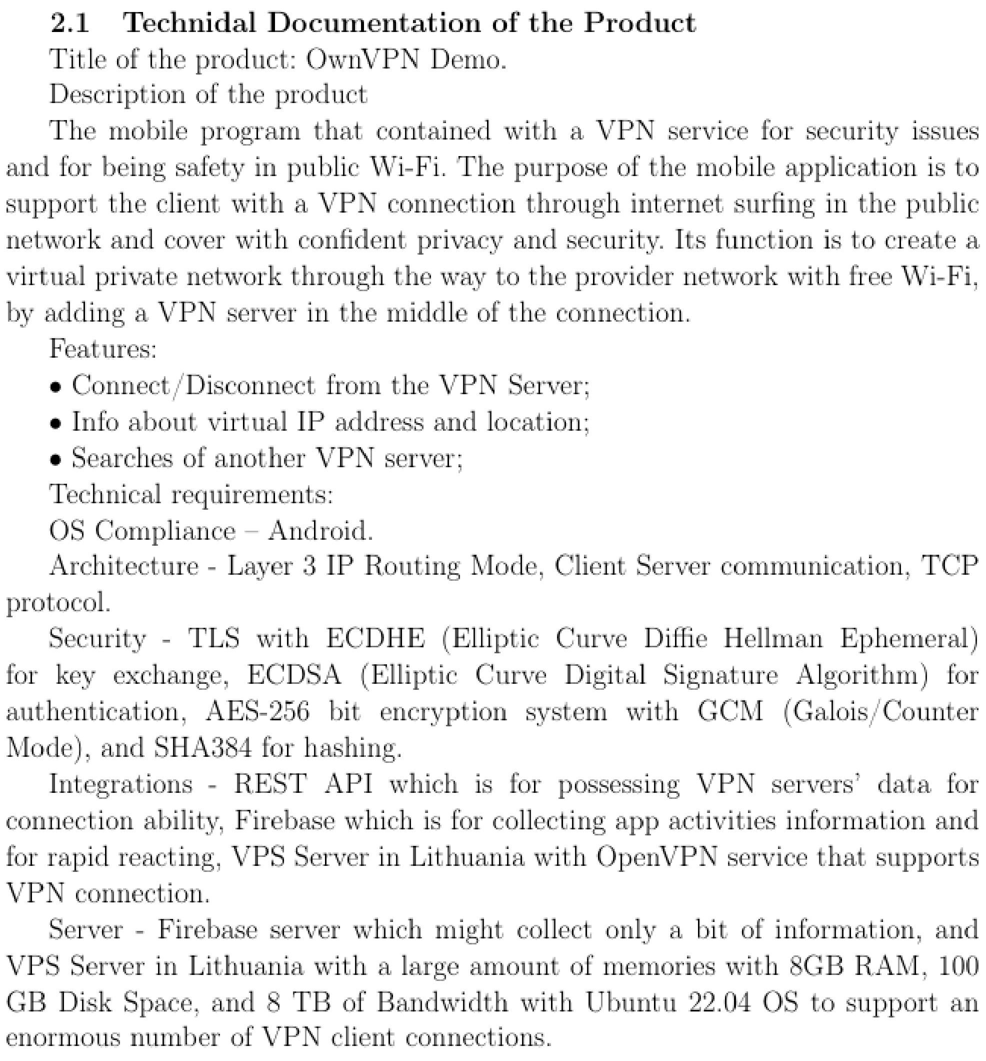
Technidal Documentation of the Product



2.1 Technidal Documentation of the Product

Title of the product: OwnVPN Demo.

Description of the product

The mobile program that contained with a VPN service for security issues and for being safety in public Wi-Fi. The purpose of the mobile application is to support the client with a VPN connection through internet surfing in the public network and cover with confident privacy and security. Its function is to create a virtual private network through the way to the provider network with free Wi-Fi, by adding a VPN server in the middle of the connection.

Features:

- Connect/Disconnect from the VPN Server;
- Info about virtual IP address and location;
- Searches of another VPN server;

Technical requirements:

OS Compliance – Android.

Architecture - Layer 3 IP Routing Mode, Client Server communication, TCP protocol.

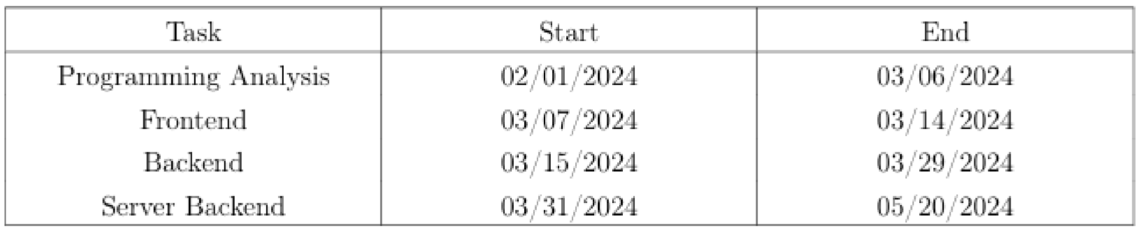
Security - TLS with ECDHE (Elliptic Curve Diffie Hellman Ephemeral) for key exchange, ECDSA (Elliptic Curve Digital Signature Algorithm) for authentication, AES-256 bit encryption system with GCM (Galois/Counter Mode), and SHA384 for hashing.

Integrations - REST API which is for possessing VPN servers' data for connection ability, Firebase which is for collecting app activities information and for rapid reacting, VPS Server in Lithuania with OpenVPN service that supports VPN connection.

Server - Firebase server which might collect only a bit of information, and VPS Server in Lithuania with a large amount of memories with 8GB RAM, 100 GB Disk Space, and 8 TB of Bandwidth with Ubuntu 22.04 OS to support an enormous number of VPN client connections.

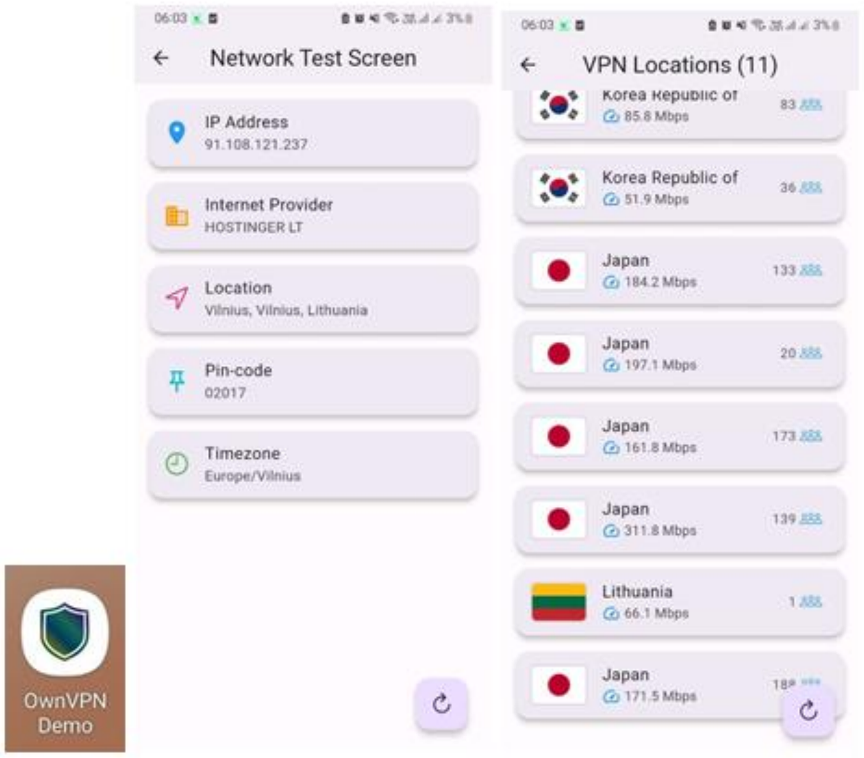
Schedule:

Table 2.3 – Product Calendar Plan



Task	Start	End
Programming Analysis	02/01/2024	03/06/2024
Frontend	03/07/2024	03/14/2024
Backend	03/15/2024	03/29/2024
Server Backend	03/31/2024	05/20/2024

UX/UI:


OwnVPN Demo

Network Test Screen

IP Address
91.108.121.237

Internet Provider
HOSTINGER LT

Location
Vilnius, Vilnius, Lithuania

Pin-code
02017

Timezone
Europe/Vilnius

VPN Locations (11)

Korea Republic of	85.8 Mbps	83
Korea Republic of	51.9 Mbps	36
Japan	184.2 Mbps	133
Japan	197.1 Mbps	20
Japan	161.8 Mbps	173
Japan	311.8 Mbps	139
Lithuania	66.1 Mbps	1
Japan	171.5 Mbps	18

Figure 2.9 – VPN Mobile Application Features

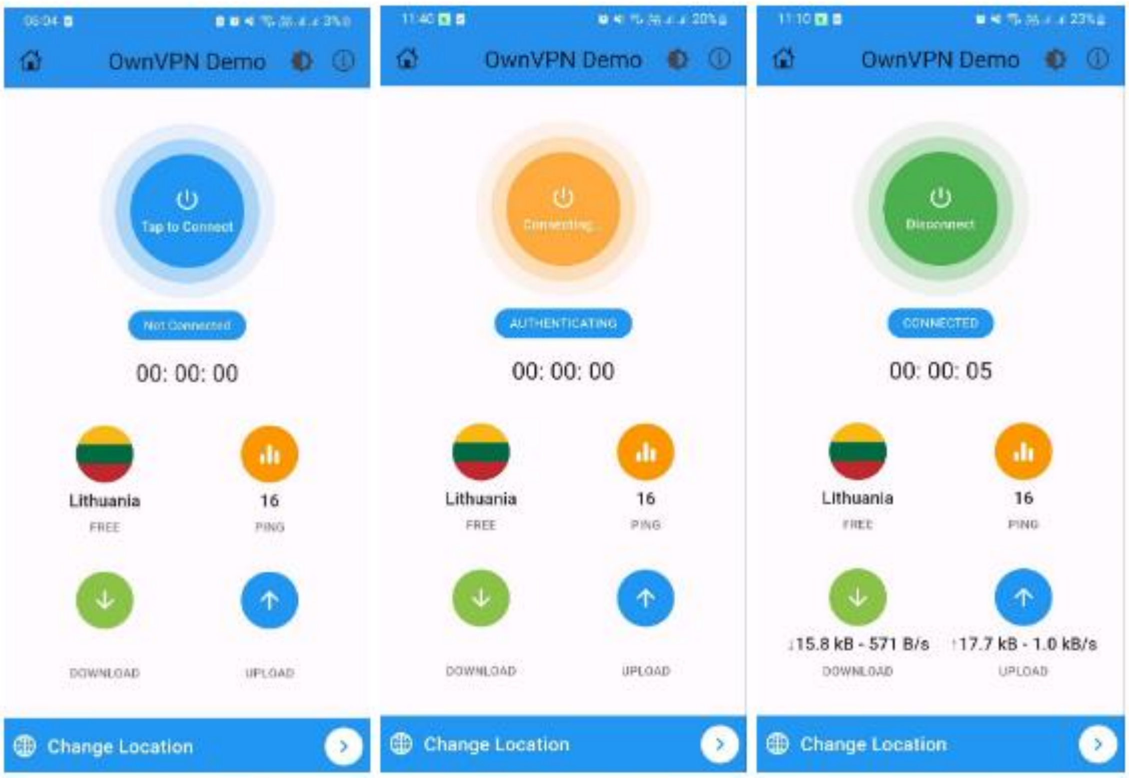


Figure 2.10 – VPN Mobile Application Functions

Backend:

Appendix B home_controller.dart

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:vpn_basic_project/helpers/pref.dart';
import 'package:vpn_basic_project/models/vpn.dart';
import 'package:vpn_basic_project/models/vpn_config.dart';
import 'package:vpn_basic_project/services/vpn_engine.dart';

class HomeController extends GetxController {
  final Rx<Vpn> vpn = Pref.vpn.obs;
  final vpnState = VpnEngine.vpnDisconnected.obs;

  void connectToVpn() {
    ///Stop right here if user not select a vpn
    if (vpn.value.openVPNConfigDataBase64.isEmpty) return;

    if (vpnState.value == VpnEngine.vpnDisconnected) {
      print('\nBefore: ${vpn.value.openVPNConfigDataBase64}');

      final data = Base64Decoder().convert(
      vpn.value.openVPNConfigDataBase64);
      final config = Utf8Decoder().convert(data);
      final vpnConfig = VpnConfig(
          country: vpn.value.countryLong,
          username: 'connect',
          password: '',
          config: config,
          certIsRequired: true);
    final data = Base64Decoder().convert(
    vpn.value.openVPNConfigDataBase64);
    final config = Utf8Decoder().convert(data);
    final vpnConfig = VpnConfig(
        country: vpn.value.countryLong,
        username: 'connect',
        password: '',
        config: config,
        certIsRequired: true);

    print('\nAfter: $config');

    ///Start if stage is disconnected
    VpnEngine.startVpn(vpnConfig);
  } else {
    ///Stop if stage is "not" disconnected
    VpnEngine.stopVpn();
  }
}
```

```
  // vpn buttons color
  Color get getButtonColor {
    switch (vpnState.value) {
      case VpnEngine.vpnDisconnected:
        return Colors.blue;

      case VpnEngine.vpnConnected:
        return Colors.green;

      default:
        return Colors.orangeAccent;
    }
  }

  // vpn button text
  String get getButtonText {
    switch (vpnState.value) {
      case VpnEngine.vpnDisconnected:
        return 'Tap to Connect';

      case VpnEngine.vpnConnected:
        return 'Disconnect';

      default:
        return 'Connecting...';
    }
  }
}
```

API Server (list of VPN Servers)

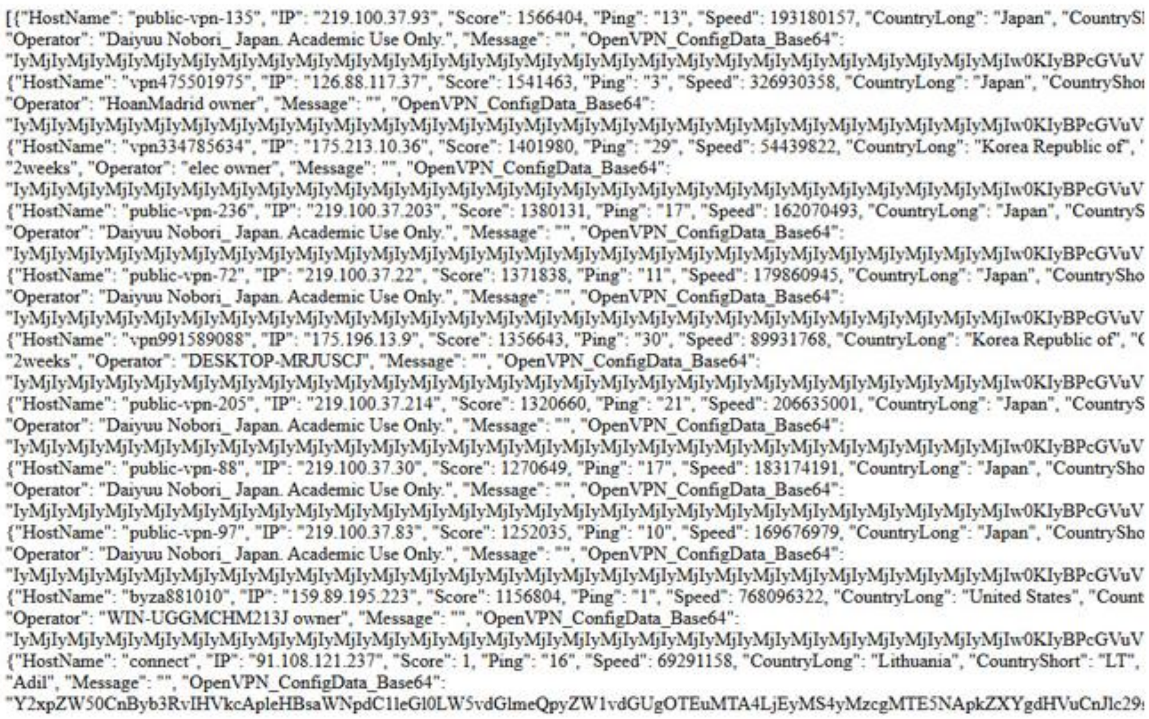
Appendix A apis.dart

```
import 'dart:convert';
import 'package:get/get.dart';
import 'package:http/http.dart';
import 'package:vpn_basic_project/helpers/pref.dart';
import 'package:vpn_basic_project/models/ip_details.dart';
import 'package:vpn_basic_project/models/vpn.dart';

class APIs {
  static Future<List<Vpn>> getVPNServers() async {
    final List<Vpn> list_vpn = [];
    try {
      final res = await get(Uri.parse(
      'https://swan03.pythonanywhere.com/')
      );
      final List<dynamic> decoded_body = jsonDecode(res.body);
      if (decoded_body.isNotEmpty) {
        for (int i = 0; i < decoded_body.length; ++i) {
          list_vpn.add(Vpn.fromJson(decoded_body[i]));
        }
      }
    } catch (e) {
      print('\ngetVPNServerse: $e');
    }
    list_vpn.shuffle();

    if (list_vpn.isNotEmpty) Pref.list_vpn = list_vpn;

    return list_vpn;
  }
```

```
static Future<void> getIPDetails({
required Rx<IPDetails> ipData
}) async {
  try {
    final res = await get(Uri.parse('http://ip-api.com/json/'));
    final data = jsonDecode(res.body);
    ipData.value = IPDetails.fromJson(data);
  } catch (e) {
    print('\ngetIPDetailsE: $e');
```

```
    }
  }
}
```

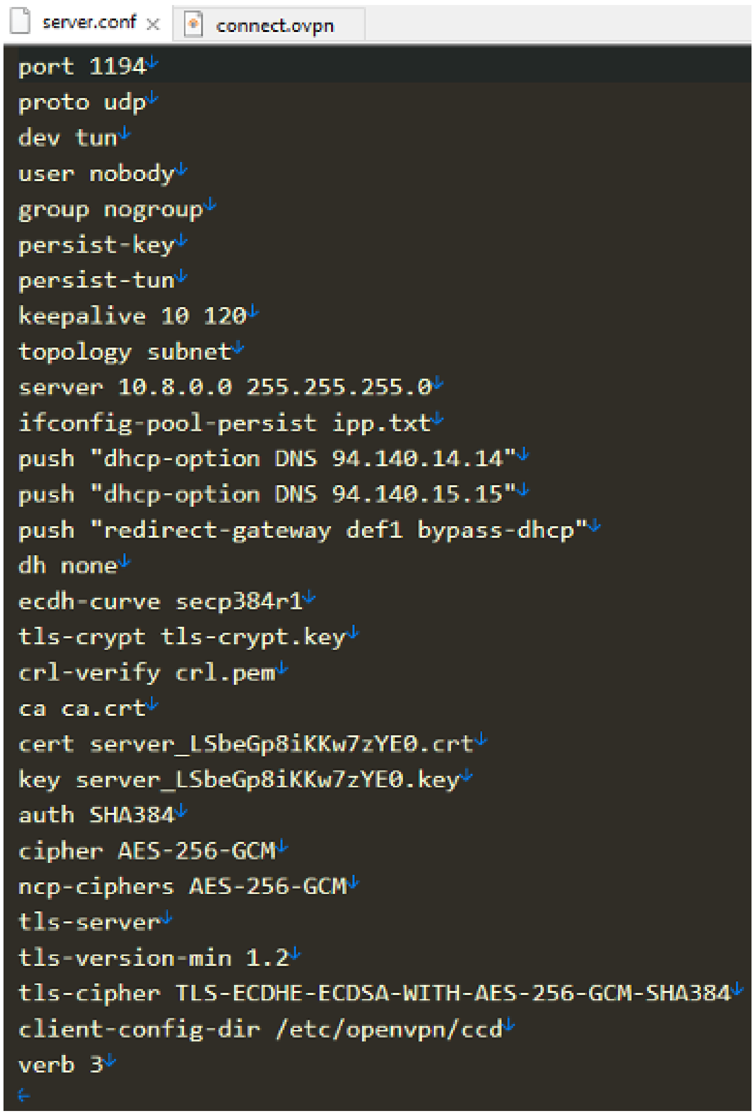
VPN Servers List:



[{"HostName": "public-vpn-135", "IP": "219.100.37.93", "Score": 1566404, "Ping": "13", "Speed": 193180157, "CountryLong": "Japan", "CountrySl
"Operator": "Daiyuu Nobori_ Japan. Academic Use Only.", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "vpn475501975", "IP": "126.88.117.37", "Score": 1541463, "Ping": "3", "Speed": 326930358, "CountryLong": "Japan", "CountrySho
"Operator": "HoanMadrid owner", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "vpn334785634", "IP": "175.213.10.36", "Score": 1401980, "Ping": "29", "Speed": 54439822, "CountryLong": "Korea Republic of", '
"2weeks", "Operator": "elec owner", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "public-vpn-236", "IP": "219.100.37.203", "Score": 1380131, "Ping": "17", "Speed": 162070493, "CountryLong": "Japan", "CountryS
"Operator": "Daiyuu Nobori_ Japan. Academic Use Only.", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "public-vpn-72", "IP": "219.100.37.22", "Score": 1371838, "Ping": "11", "Speed": 179860945, "CountryLong": "Japan", "CountrySho
"Operator": "Daiyuu Nobori_ Japan. Academic Use Only.", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "vpn991589088", "IP": "175.196.13.9", "Score": 1356643, "Ping": "30", "Speed": 89931768, "CountryLong": "Korea Republic of", "(
"2weeks", "Operator": "DESKTOP-MRJUSCJ", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "public-vpn-205", "IP": "219.100.37.214", "Score": 1320660, "Ping": "21", "Speed": 206635001, "CountryLong": "Japan", "CountryS
"Operator": "Daiyuu Nobori_ Japan. Academic Use Only.", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "public-vpn-88", "IP": "219.100.37.30", "Score": 1270649, "Ping": "17", "Speed": 183174191, "CountryLong": "Japan", "CountrySho
"Operator": "Daiyuu Nobori_ Japan. Academic Use Only.", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "public-vpn-97", "IP": "219.100.37.83", "Score": 1252035, "Ping": "10", "Speed": 169676979, "CountryLong": "Japan", "CountrySho
"Operator": "Daiyuu Nobori_ Japan. Academic Use Only.", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "byza881010", "IP": "159.89.195.223", "Score": 1156804, "Ping": "1", "Speed": 768096322, "CountryLong": "United States", "Count
"Operator": "WIN-UGGMCHM213J owner", "Message": "", "OpenVPN_ConfigData_Base64":
"IyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIw0KIyBPcGVuV
{"HostName": "connect", "IP": "91.108.121.237", "Score": 1, "Ping": "16", "Speed": 69291158, "CountryLong": "Lithuania", "CountryShort": "LT",
"Adil", "Message": "", "OpenVPN_ConfigData_Base64":
"Y2xpZW50CnByb3RvIHVkcApleHBsaWNpdC1leGl0LW5vdGlmeQpyZW1vdGUgOTEuMTA4LjEyMS4yMzcgMTE5NApkZXYgdHVuCnJlc29

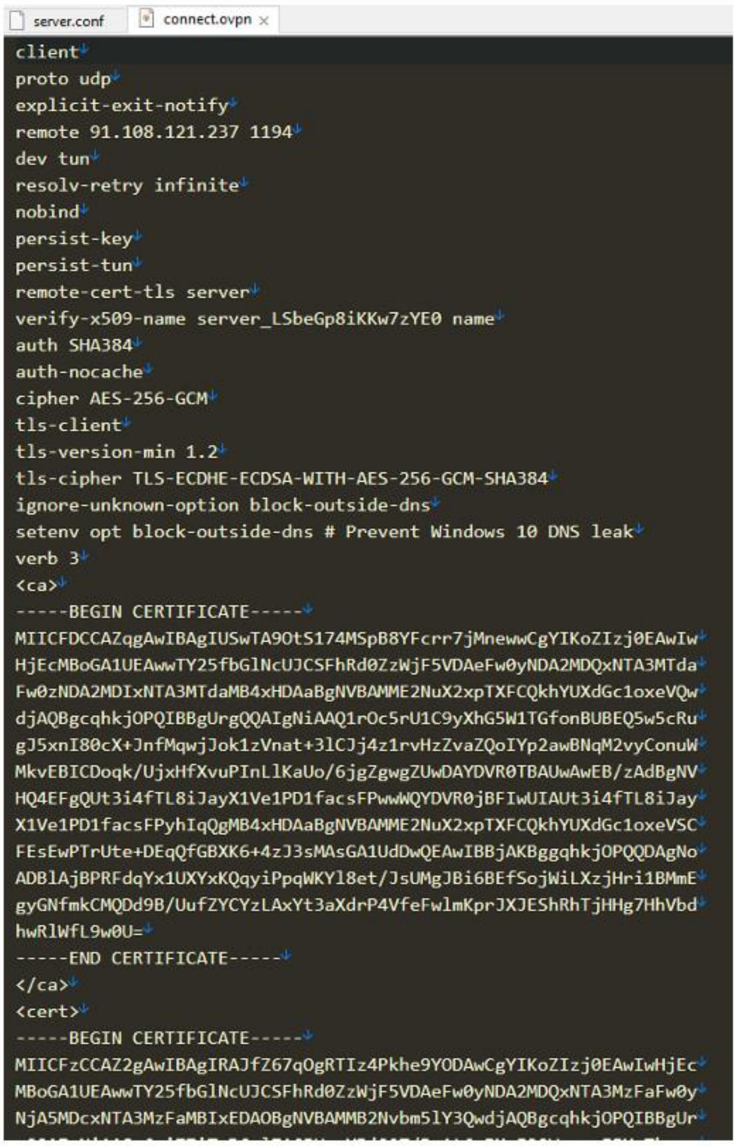
Figure 3.1 – API Server with VPN servers records

VPN Server:



```
port 1194
proto udp
dev tun
user nobody
group nogroup
persist-key
persist-tun
keepalive 10 120
topology subnet
server 10.8.0.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push "dhcp-option DNS 94.140.14.14"
push "dhcp-option DNS 94.140.15.15"
push "redirect-gateway def1 bypass-dhcp"
dh none
ecdh-curve secp384r1
tls-crypt tls-crypt.key
crl-verify crl.pem
ca ca.crt
cert server_LSbeGp8iKKw7zYE0.crt
key server_LSbeGp8iKKw7zYE0.key
auth SHA384
cipher AES-256-GCM
ncp-ciphers AES-256-GCM
tls-server
tls-version-min 1.2
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384
client-config-dir /etc/openvpn/ccd
verb 3
```

Figure 2.12 – Server configuration



```
client
proto udp
explicit-exit-notify
remote 91.108.121.237 1194
dev tun
resolv-retry infinite
nobind
persist-key
persist-tun
remote-cert-tls server
verify-x509-name server_LSbeGp8iKKw7zYE0 name
auth SHA384
auth-nocache
cipher AES-256-GCM
tls-client
tls-version-min 1.2
tls-cipher TLS-ECDHE-ECDSA-WITH-AES-256-GCM-SHA384
ignore-unknown-option block-outside-dns
setenv opt block-outside-dns # Prevent Windows 10 DNS leak
verb 3
<ca>
-----BEGIN CERTIFICATE-----
MIICFDCCAZqgAwIBAgIUSwTA9OtS174MSpB8YFcrr7jMnewwCgYIKoZIzj0EAwIw
HjEcMBoGA1UEAwwTY25fbGlNcUJCSFhRd0ZzWjF5VDAeFw0yNDA2MDQxNTA3MTda
Fw0zNDA2MDIxNTA3MTdaMB4xHDAaBgNVBAMME2NuX2xpTXFCQkhYUXdGc1oxeVQw
djAQBgcqhkjOPQIBBgUrgQQAIgNiAAQ1rOc5rU1C9yXhG5W1TGfonBUBEQ5w5cRu
gJ5xnI80cX+JnfMqwjJok1zVnat+3lCJj4z1rvHzZvaZQoIYp2awBNqM2vyConuW
MkvEBICDoqk/UjxHfXvuPInLlKaUo/6jgZgwgZUwDAYDVR0TBAUwAwEB/zAdBgNV
HQ4EFgQUt3i4fTL8iJayX1Ve1PD1facsFPwwWQYDVR0jBFIwUIAUt3i4fTL8iJay
X1Ve1PD1facsFPyhIqQgMB4xHDAaBgNVBAMME2NuX2xpTXFCQkhYUXdGc1oxeVSC
FEsEwPTrUte+DEqQfGBXK6+4zJ3sMAsGA1UdDwQEAwIBBjAKBggqhkjOPQQDAgNo
ADBlAjBPRFdqYx1UXYxKQqyiPpqWKYl8et/JsUMgJBi6BEfSojWiLXzjHri1BMmE
gyGNfmkCMQDd9B/UufZYCYzLAxYt3aXdrP4VfeFwlmKprJXJEShRhTjHHg7HhVbd
hwRlWfL9w0U=
-----END CERTIFICATE-----
</ca>
<cert>
-----BEGIN CERTIFICATE-----
MIICFzCCAZ2gAwIBAgIRAJfZ67qOgRTIz4Pkhe9YODAwCgYIKoZIzj0EAwIwHjEc
MBoGA1UEAwwTY25fbGlNcUJCSFhRd0ZzWjF5VDAeFw0yNDA2MDQxNTA3MzFaFw0y
NjA5MDcxNTA3MzFaMBIxEDAOBgNVBAMMB2Nvbm5lY3QwdjAQBgcqhkjOPQIBBgUr
```

Figure 2.13 – Client configuration