



Харківський національний університет радіоелектроніки  
Кафедра ЕОМ



## Комп'ютерна система автоматичного контролю локалізації клієнтів оздоровчих та відпочивальних закладів

КВАЛІФІКАЦІЙНА РОБОТА  
ДРУГИЙ (МАГІСТЕРСЬКИЙ) РІВЕНЬ

Автор:  
Глушко Д.В.,  
студ. гр. КСМм-20-1

Керівник:  
Завізіступ Ю.Ю.,  
проф. каф. КОМ

## Мета кваліфікаційної роботи



Метою кваліфікаційної роботи є розробка комп'ютерної система автоматичного контролю локалізації клієнтів оздоровчих та відпочивальних закладів. А саме розробка бездротової мережі, яка здатна покрити сигналом WiFi всю територію оздоровчої установи, розробка портативного пристрою, за допомогою якого пацієнт зможе передати сигнал про необхідну допомогу шляхом натискання на кнопку, розробка програми контролю локалізації пристрою, що надіслав сигнал.

**Мета:** розробка десктопної програми, яка за допомогою простого та зрозумілого інтерфейсу дозволить диспетчеру:

- швидко побачити, що пацієнт відправив сигнал про необхідну допомогу;
- побачити місцезнаходження цього пацієнту на карті закладу.

## Постановка задачі



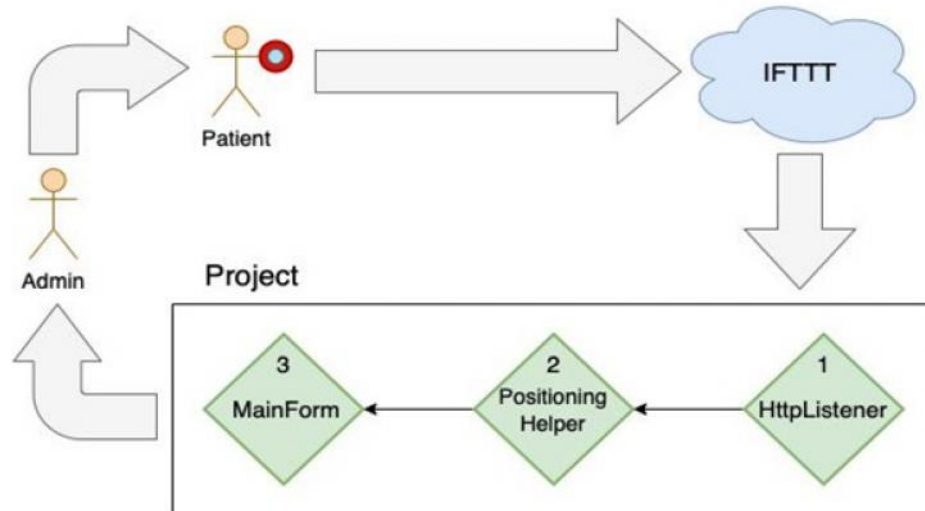
Існує реальний оздоровчий заклад, у якому перебувають люди, які мають серцеві захворювання. Площа цієї установи складає двадцять п'ять гектарів лісової місцевості, а кількість пацієнтів досягає чотирьохсот чоловік і вони можуть вільно пересуватися по всій території. Оскільки серйозність захворювання пацієнтів може бути високою, то стан будь-якого пацієнта може різко погіршитися. У такому разі йому негайно має бути надана медична допомога. Звідси виникає потреба у створенні комп'ютерної системи, яка б надала пацієнтові можливість подати сигнал про допомогу і розрахувала б місце його розташування в межах території установи.

## Алгоритм роботи системи автоматичного контролю локалізації пацієнтів закладу



- 1) Якщо пацієнту стає погано, він натискає на кнопку індивідуального портативного WiFi-пристрою, підключеного до MESH-мережі WiFi закладу.
- 2) Запит про потрібну допомогу надсилається на сторонній онлайн-портал IFTTT, який потрібен для швидкого та зручного програмування портативного WiFi-пристрою.
- 3) IFTTT перенаправляє запит на спеціальну URL-адресу, яка відстежується сервісом десктопної програми локалізації пацієнтів - HttpListener за допомогою технології SimpleHTTP.
- 4) HttpListener приймає запит, з нього отримує сили сигналу і Mac-адреси найближчих до переносного пристрою маршрутизаторів і передає їх сервісу – PositioningHelper.
- 5) PositioningHelper на основі отриманих даних за допомогою формули триангуляції розраховує координати пацієнта, що шукається та відправляє їх у форму MainForm.
- 6) MainForm відображає мітку на карті оздоровчого закладу у головному вікні програми.
- 7) Диспетчер системи бачить мітку та відправляє туди санітарів для надання допомоги.

## Схема алгоритму роботи системи контролю локалізації пацієнтів закладу



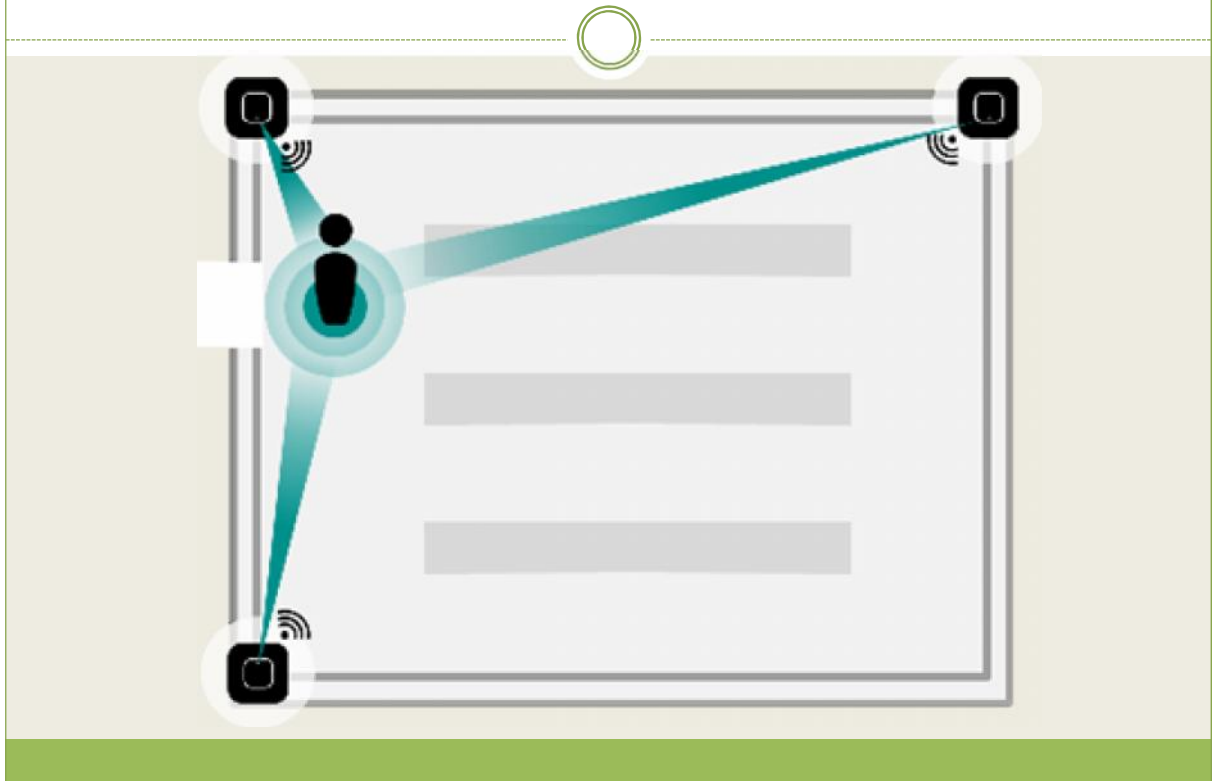
## Метод визначення місцезнаходження об'єкта: тріангуляція

Цей метод полягає в тому, щоб визначити силу сигналу від клієнта до 3-4х точок доступу WiFi і в зоні перетину можливого розташування клієнта щодо кожної точки спозиціонувати пристрій. При правильному розташуванні точок доступу він дозволяє з високою ймовірністю визначити координату клієнта з точністю 5-7м.

Існує два види тріангуляції: трилатерація та ангуляція. Метод трилатерації визначає місцезнаходження об'єкта, вимірюючи відстань від нього до кількох точок доступу. У методі ангуляції розташування об'єкта визначається як перетин прямих, кожна з яких спрямована під певним кутом від відповідної точки доступу до мобільного об'єкта.

В умовах системи, що розробляється найкращим методом буде трилатерація на основі оцінювання відстані залежно від рівня прийнятого сигналу. Оскільки, на відміну від ангуляції, не потребує в додатковому обладнанні.

## Схема роботи тріангуляції



## Формули розрахунку координат

$$R_1 = \frac{S_1}{(S_1 + S_2 + S_3)}$$

$$R_2 = \frac{S_2}{(S_1 + S_2 + S_3)}$$

$$R_3 = \frac{S_3}{(S_1 + S_2 + S_3)}$$

$R_N$  - N відносний сигнал

$S_N$  - сила сигналу N-го маршрутизатора

$$x = (R_1 * Xr_1)(R_2 * Xr_2)(R_3 * Xr_3)$$

$$y = (R_1 * Yr_1)(R_2 * Yr_2)(R_3 * Yr_3)$$

$x, y$  – координати пристрою, що шукається

$Sr_x_N$  - x координата N-го маршрутизатора

$Sr_y_N$  - y координата N-го маршрутизатора

## Варіанти технологій локалізації



- **Супутникові радіонавігаційні системи.** Наприклад, GPS та ГЛОНАСС (глобальні навігаційні супутникові системи) створені спеціально для цілей навігації та дозволяють отримати інформацію про положення у просторі: географічні координати та висоту. Принцип роботи базується на вимірі відстані від супутника до антени на пристрої за допомогою радіосигналів.
- **Позиціонування Wi-Fi.** Система позиціонування у режимі реального часу на базі даної технології є найбільш доступним методом у зв'язку з розвинутою інфраструктурою мереж на базі Wi-Fi точок доступу і не вимагає додаткового обладнання. Практично всі бездротові приймачі вимірюють рівень вхідного сигналу і повертають параметр індикатору сили прийнятого сигналу, на підставі якого обчислюються координати.
- **Позиціонування Bluetooth.** Система використовує радіомаячки з функцією Bluetooth Low Energy (BLE) – енергоефективних пристроїв, здатних подібно до пожежного датчика періодично подавати сигнали. Радіомаячки здатні визначити наближення до них пристроїв з увімкненим BLE і повідомити про це сервер мобільного додатка, який цю інформацію використовує як координату пристрою.

## Переваги WiFi-позиціонування в умовах поставленого завдання



Серед наведених методів позиціонування було обрано саме WiFi-позиціонування, тому що в умовах поставленого завдання дана технологія має переваги над конкурентами:

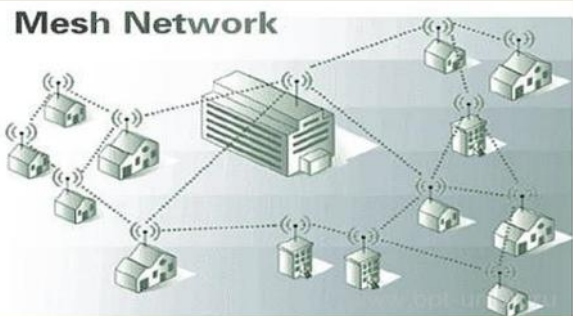
- Через те, що оздоровчий заклад знаходиться в лісовій місцевості, густа крона дерев стає серйозною перешкодою для GPS-сигналу, що може позначитися на точності локалізації пацієнта, так само на точність можуть вплинути погодні умови. У разі ж WiFi-позиціонування, при грамотній побудові системи та розташуванню точок доступу, описані фактори впливають на силу WiFi-сигналу значно менше, залишаючи необхідну точність.
- Взявши до уваги те, що площа оздоровчої установи становить 25 гектарів, виникає проблема і в позиціонуванні на основі Bluetooth, оскільки дальність роботи BLE-радіомаячків становить близько 30 метрів. У разі ж WiFi-позиціонування ця цифра приблизно у 2-3 рази більша.
- Окрім цього система WiFi-позиціонування є найекономічнішою та найпростішою у реалізації серед проведених.

## MESH-мережа



MESH-мережа формується з багатьох з'єднань точка-точка вузлів, які знаходяться в зоні радіопокриття один одного. Ключова властивість самоорганізуючих сітчастих мереж полягає в тому, що, по-перше, зв'язки між вузлами встановлюються автоматично; по-друге, будь-який вузол може виконувати функції транзиту (маршрутизації) пакетів для інших учасників мережі. Мережа характеризується високою надійністю, високою пропускнуою здатністю та зниженим споживанням електроенергії. Висока надійність забезпечується резервуванням вузлів (якщо один вузол виходить з ладу, дані можуть обійти іншим шляхом).

Зменшення енергоспоживання досягається за рахунок зменшення потужності сигналу за рахунок передачі даних через більшу кількість вузлів, розділених меншими відстанями.



## Вибір обладнання для розгортання MESH-мережі



Точками доступу були обрані маршрутизатори від фірми Ubiquiti, а саме: Ubiquiti UniFi AP AC Mesh Pro (UAP-AC-M-PRO) та Ubiquiti UniFi AP AC Mesh (UAP-AC-M). Оскільки ця фірма є однією з найпопулярніших та найпередовіших представників даного сегменту, а представлені вище маршрутизатори були створені спеціально для побудови MESH мереж. UniFi AP AC Mesh Pro та UniFi AP AC Mesh ідеально підходять для виконання поставленого завдання, тому що вони мають досить великий радіус покриття, відмінну швидкість передачі даних, здатні одночасно обслуговувати велику кількість клієнтів і є невибагливими до погодних умов, що дозволяє розміщувати їх поза приміщеннями. Також для побудови мережі необхідним є комутатор UniFi Switch 8 POE-60W.



UAP-AC-M-PRO

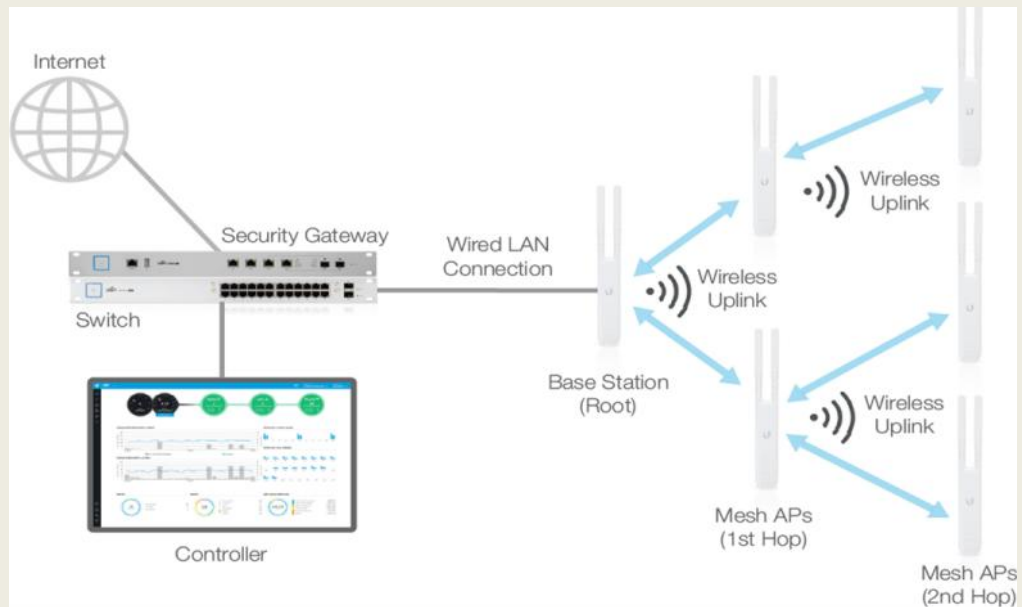


UniFi Switch 8 POE-60W



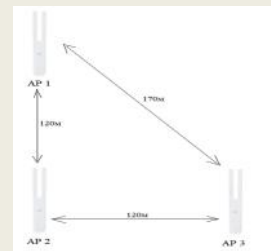
UAP-AC-M

## Схема мережі MESH на основі обладнання Ubiquiti



## Розрахунок розташування та вартості обладнання

Площа оздоровчої установи становить 25 га. Припустимо, що вона має прямокутну форму зі сторонами рівними 720 та 360 метрам. Відомо, що обрані точки доступу мають радіус покриття до 183 метрів, однак радіусом дії, що рекомендується, в якому сигнал буде досить хорошим незважаючи на перешкоди і погодні умови є приблизно 80 метрів. Отже, щоб забезпечити стійкий сигнал між AP1 і AP3 (рисунок 2.10), відстань між AP1 та AP2, AP2 та AP3 повинна складати 120 метрів, а відстань між AP1 та AP3 - 170.



Таким чином, щоб покрити всю територію комплексу необхідно встановити 18 точок доступу: одну UniFi AC Mesh Pro та 17 UniFi AC Mesh.

Вартість обладнання:

- UniFi AC Mesh Pro коштує 5920 грн.;
- UniFi AC Mesh коштує 3260 грн.;
- Ubiquiti UniFi Switch 8-60W коштує 4200 грн.

В такому разі, загальна вартість обладнання складатиме 65540 грн.

## Схема розташування точок доступу на карті



## Розробка портативного WiFi-пристрою для передачі сигналу про допомогу

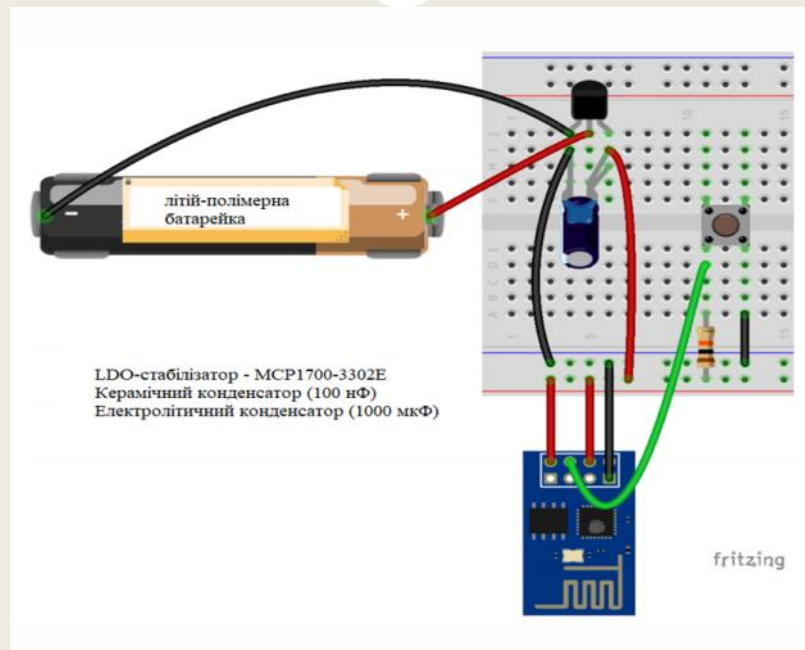
Розглянемо варіант розробки WiFi-кнопки за допомогою мікроконтролеру ESP8266 та WiFi-модулю ESP-01 - пристрій, при натисканні на кнопку якого буде переданий сигнал про необхідну допомогу в комп'ютерну систему, яка, у свою чергу, запускатиме алгоритм пошуку розташування цього пристрою.

Необхідне обладнання:

- ESP-01 – 1шт.;
- Кнопка – 1шт.;
- Резистор на 10 ком - 1шт.;
- Літій-іонна або літій-полімерна батарея - 1шт.;
- Стабілізатор напруги:
  - МСР1700-3302Е - 1шт.;
  - Керамічний конденсатор на 100 нФ – 1шт.;
  - Електролітичний конденсатор на 1000 мкФ – 1шт.;
- Пластиковий корпус – 1шт.

Для програмування подібних портативних WiFi пристроїв існує безліч різних рішень. Найпростішим є безкоштовний онлайн-ресурс під назвою IFTTT.

## Схема портативного WiFi-пристрою для передачі сигналу про допомогу



## Висновки

В даній кваліфікаційній роботі було проведено дослідження існуючих технологій розробки бездротових мереж, здатних покрити мережею WiFi територію площею в 25 гектарів, локалізації клієнтів за допомогою WiFi-сигналів, розробки простих портативних пристроїв для передачі сигналу по мережі WiFi.

В результаті виконання даної роботи запропоновано оптимальну для поставленого завдання топологію WiFi-мереж – MESH, яка значного спрощую процес розгортання мережі на велику площу та забезпечує більшу відмовостійкість та менше енергоспоживання системи; оптимальний алгоритм локалізації на основі трилатерація з пошуком сили сигналу, оскільки він не вимагає ніякого додаткового обладнання і надає достатньо точний результат; простий, дешевий та водночас повністю задовольняючий вимоги варіант розробки портативного пристрою на основі мікроконтролеру ESP8266 та онлайн-сервісу IFTTT.

Була розроблена десктопна програма, за допомогою якої диспетчер може швидко реагувати на отриманий сигнал про необхідну допомогу та бачити безпосередньо мітку місцезнаходження пацієнта на карті.

## .1 – Program.cs

```
using SimpleHttp;

namespace Diploma
{
    internal static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            HttpListener.Host();

            HttpListener.Listen();

            ApplicationConfiguration.Initialize();
            Application.Run(new Form2());
        }
    }
}
```

## .2 – MainForm.cs

```
using SimpleHttp;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Diploma
{
    public static class HttpListener
    {
        public static void Host()
        {
            HttpServer.ListenAsync(
                2705,
                Cancellation.Token.None,
                Route.OnHttpRequestAsync)
        }
    }
}
```

```

        .Wait();
    }

    public static void Listen()
    {
        Route.Add("/", (req, res, porps) =>
        {
            MainForm.CalculateCoordinates(req);
        });
    }
}
}
}

```

### .3 – MainForm.cs

```

using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Net;

namespace Diploma
{
    public partial class MainForm : Form
    {
        private SqlConnection connection = null;
        private SqlDataAdapter adapter = null;
        private DataSet dataSet = null;
        private FormModal2 modal2 = null;
        public MainForm()
        {
            InitializeComponent();
            pictureBox2.Parent = pictureBox1;
            pictureBox2.BackColor = Color.Transparent;
        }

        private void Main_Load(object sender, EventArgs e)
        {
            connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["TestDB"].C
onnectionString);
            connection.Open();
        }

        public static void
CalculateCoordinates(HttpListenerRequest req)
        {
            List<KeyValuePair<float, float>>
routerCoordinates = new
System.Collections.Generic.List<KeyValuePair<float, float>>();

```

```

        foreach (var item in req.Adapters)
        {
            adapter = new SqlDataAdapter(
                $"SELECT Coordinates from Router WHERE
MacAddress == {item.MacAddress}", connection);
            dataSet = new DataSet();
            adapter.Fill(dataSet);

            KeyValuePair<float, float> pair =
dataSet.Tables[0].Coordinates;
            routerCoordinates.Add(pair);
        }

        KeyValuePair<float, float> coordinates =
PositioningHelper.CalculateCoordinates(req, routerCoordinates);

        float xInPix = coordinates.Key * (float)113.57;
        float yInPix = coordinates.Value * (float)84.11;

        FormModall modal1 = new
FormModall(xInPix,yInPix).ShowDialog();

        if (modal2 == null)
        {
            modal2 = new FormModal2().ShowDialog();
        }
    }
}

```

#### .4 – PositioningHelper.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;

namespace Diploma
{
    public static class PositioningHelper
    {
        public static KeyValuePair<float, float>
CalculateCoordinates(HttpListenerRequest req,
List<KeyValuePair<float, float>> coordinates)
        {
            List<float> signalStrengths = new List<float>();
            foreach (var adapter in req.Adapters)
            {
                foreach (var network in
adapter.NetworkReport.AvailableNetworks)

```

```

        {
            signalStrengths.Add(network.SignalBars);
        }
    }
    signalStrengths.Sort();

    float weightedSignal1 = signalStrengths[0] /
(signalStrengths[0] + signalStrengths[1] + signalStrengths[2]);
    float weightedSignal2 = signalStrengths[1] /
(signalStrengths[0] + signalStrengths[1] + signalStrengths[2]);
    float weightedSignal3 = signalStrengths[2] /
(signalStrengths[0] + signalStrengths[1] + signalStrengths[2]);

    float x = weightedSignal1 * coordinates[0].Key +
weightedSignal2 * coordinates[1].Key + weightedSignal3 *
coordinates[2].Key;
    float y = weightedSignal1 * coordinates[0].Value
+ weightedSignal2 * coordinates[1].Value + weightedSignal3 *
coordinates[2].Value;

    KeyValuePair<float, float> res = new
KeyValuePair<float, float>(x, y);
    return res;
    }
}
}
}

```

### .5 – Form1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    public partial class Form1 : Form
    {
        private SqlConnection connection = null;
        private SqlDataAdapter adapter = null;
        private DataSet dataSet = null;
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

```

        private void Form1_Load(object sender, EventArgs e)
        {
            connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["TestDB"].C
onnectionString);
            connection.Open();

            this.GetPatients();
        }

        private void GetPatients()
        {
            adapter = new SqlDataAdapter(
                "SELECT * from Patient", connection);

            dataSet = new DataSet();
            adapter.Fill(dataSet);

            dataGridView1.DataSource = dataSet.Tables[0];
        }

        private void button1_Click(object sender, EventArgs
e)
        {
            int rowIndex = e.RowIndex;
            dataGridView1.Rows.RemoveAt(rowIndex);

            dataSet.Tables["Patient"].Rows[rowIndex].Delete();
            adapter.Update(dataSet, "Patient");
        }
    }
}

```

## .6 – Form2.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diploma
{
    public partial class Form2 : Form
    {

```

```

private SqlConnection connection = null;
public Form2()
{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs
e)
{
    SqlCommand sqlCommand = new SqlCommand($"INSERT
INTO [Patient] (Name, Information, DeviceNumber)" +
        $"          VALUES          (N'{textBox1.Text}',
N'{textBox2.Text}', N'{comboBox1.Text}')" , connection);

    sqlCommand.ExecuteNonQuery();
}

private void Form2_Load(object sender, EventArgs e)
{
    connection = new
SqlConnection(ConfigurationManager.ConnectionStrings["TestDB"].C
onnectionString);
    connection.Open();
}
}
}
}

```