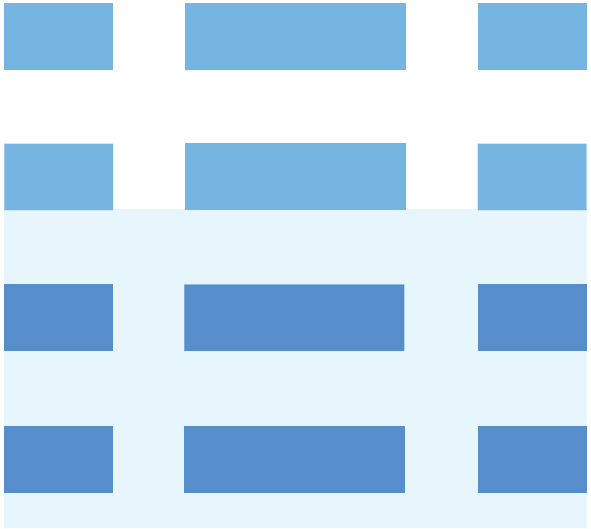
WEMORSE

Centralized Morse Code Communication

Group 3 | 4 ISA 3 | IoT project



TEAM

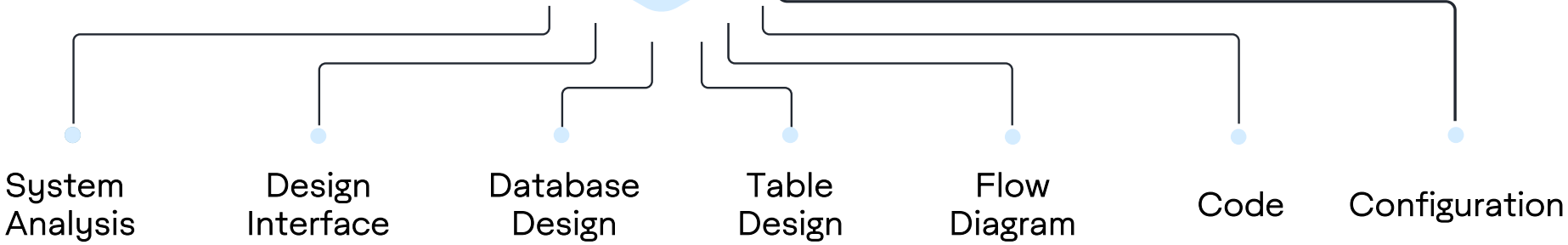
ESP Dev

Riki Awal Syahputra
(2120010136)

App Dev

Faza Rama Nugraha
(2120010291)

TABLE OF CONTENT



System Analysis

01

Java-based Android app
built using Kotlin framework

02

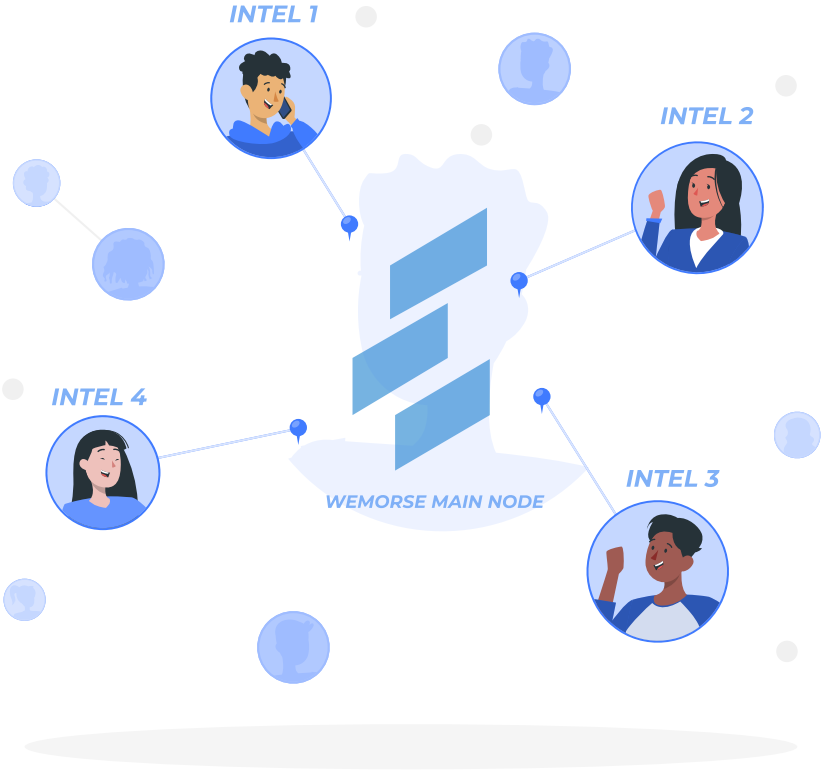
Secret reporting
communication

03

Flexible to upscale the
security of communication

04

Powered by ESP8266,
communicate from everywhere


INTEL 1
INTEL 2
INTEL 4
INTEL 3
WEMORSE MAIN NODE

[Text to Morse]

.- -... -.-.

[Text to Morse]

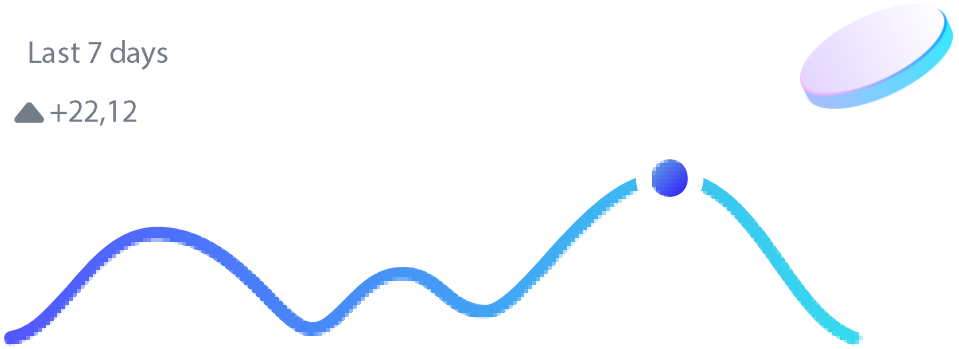
[Encrypted Text to Morse]

.- -... -.-.

[Encrypted Text to Morse]

Last 7 days

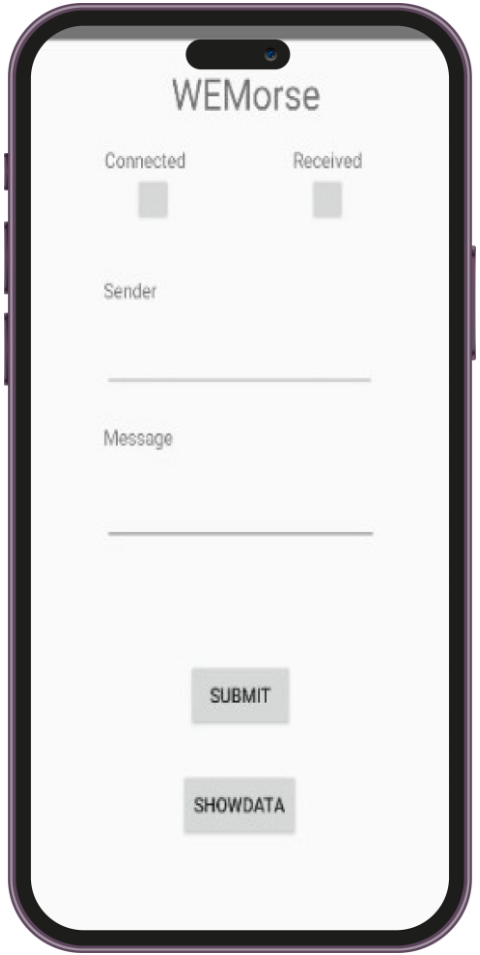
+22,12



Design Interface

Lorem ipsum dolor sit

amet consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.


WEMorse
Connected
Received
Sender
Message
SUBMIT
SHOWDATA

Main node

Database Design

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	sender	varchar(8)	utf8_unicode_ci		No	None			Change Drop More
3	msg	varchar(40)	utf8_unicode_ci		No	None			Change Drop More
4	timestamp	varchar(100)	utf8_unicode_ci		No	None			Change Drop More
5	flag	int(11)			No	None			Change Drop More

Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext

+ Options

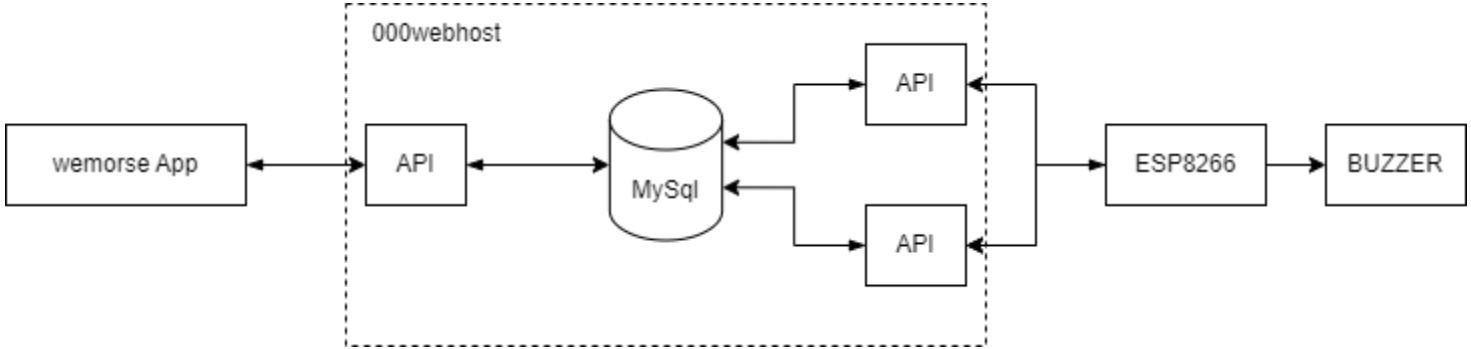
	id	sender	msg	timestamp	flag
Edit Copy Delete	1	X	hello world	18-05-2023 09:16:41	1
Edit Copy Delete	2	LEON	We are heading to airport	18-05-2023 09:18:09	1
Edit Copy Delete	4	Y	intel 3 join	18-05-2023 10:37:40	1
Edit Copy Delete	32	Y	target killed	18-05-2023 14:26:31	1
Edit Copy Delete	33	Y	we need aid kit at 0130`380	18-05-2023 14:29:21	1
Edit Copy Delete	39	LEONARDO	Based on Tempo's monitoring	19-05-2023 08:39:10	1

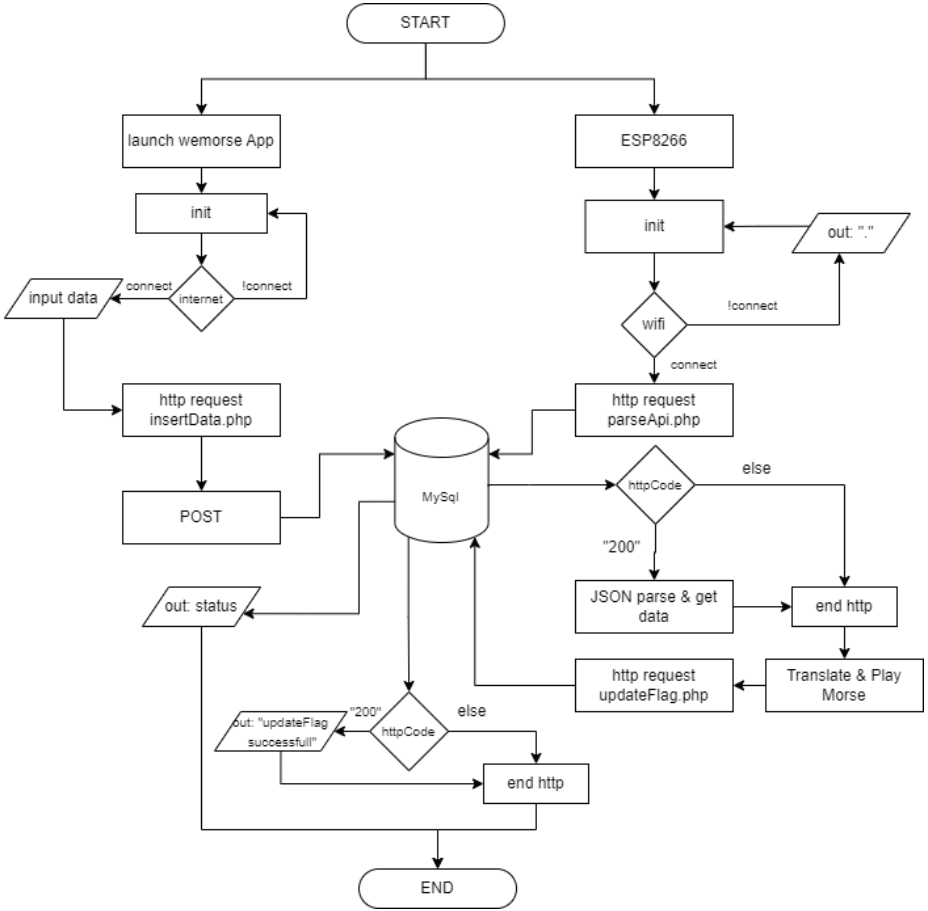
Table Design

NO	Field Name	Data Type	Length	Description
1	id	int	-	id number (primary key)
2	sender	varchar	8	sender's name
3	msg	varchar	40	user messages
4	timestamp	varchar	100	Message time stamp
5	flag	int	-	flag status (unread & read)

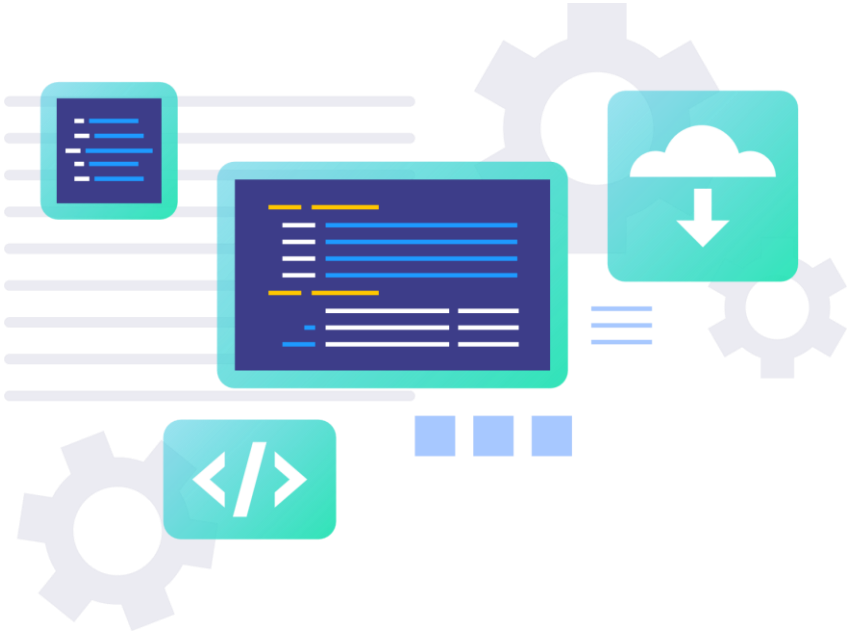
Flow Diagram

WEMORSE (General Flowchart)





WEMORSE (Detailed Flowchart)



Code

ESP CODE

Initialize

(Dictionary of conversion & variables)

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

const char* wifiName = "IPHONE 20 PRO";
const char* wifiPass = "mokorutoaitokotoz";
WiFiClient client;

// Web Server address to read/write from
const char* host =
"http://triall090909.000webhostapp.com/Monitor%206%20Controll/parseApi.php";
const int ledPin = 2; // Ganti dengan pin yang sesuai

// Mappings untuk huruf-huruf Morse
const char* morseCodeMapping[] = {
  ".-",   // A
  "-...", // B
  "-.-.", // C
  "-..",  // D
  ".",    // E
  "..-.", // F
  "--.",  // G
  "....", // H
  "..",   // I
  ".---", // J
  "-.-",  // K
  ".-..", // L
  "--",   // M
  "-.",   // N
  "---",  // O
  ".--.", // P
  "--.-", // Q
  ".-.",  // R
  "...",  // S
  "-",    // T
  "..-",  // U
  "...-", // V
  ".--",  // W
  "-..-", // X
  "-.--", // Y
  "--.."  // Z
};

int currentMessageId = 0;
```

Init & Wifi Connect

(Connect to Internet)

```
void setup() {
  Serial.begin(9600);
  delay(10);
  Serial.println();

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  Serial.print("Connecting to ");
  Serial.println(wifiName);
  WiFi.begin(wifiName, wifiPass);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

Main part

(API Connection & Actuator Action)

```
void loop() {
  HTTPClient http;  // Declare object of class HTTPClient
  Serial.print("Request Link: ");
  Serial.println(host);
  http.begin(client, host);  // Specify request destination
  String statusResponse;

  int httpCode = http.GET();  // Send the request
  String payload = http.getString();  // Get the response payload from server

  Serial.print("Response Code: ");  // 200 is OK
  Serial.println(httpCode);  // Print HTTP return code

  Serial.print("Returned data from Server: ");
  Serial.println(payload);  // Print request response payload

  if (httpCode == 200) {
    // Allocate JsonBuffer
    // Use arduinojson.org/assistant to compute the capacity.
    const size_t capacity = JSON_OBJECT_SIZE(3) + JSON_ARRAY_SIZE(2) + 60;
    DynamicJsonDocument jsonBuffer(capacity);

    // Parse JSON object
    DeserializationError error = deserializeJson(jsonBuffer, payload);
    if (error) {
      Serial.print("Parsing failed: ");
      Serial.println(error.c_str());
      return;
    }

    // Decode JSON/Extract values
    const JsonObject& root = jsonBuffer["data"];
    currentMessageId = root["id"];
    const char* sender = root["sender"];
    const char* msg = root["msg"];

    String text = String(sender) + " " + String(msg);
    Serial.println("Received message: " + text);

    translateAndPlayMorse(text);
  } else {
    Serial.println("Error in response");
  }

  http.end();  // Close connection
  delay(1000);  // GET Data every 1 second
}
```

Function

(Conversion & updateFlag)

```
void translateAndPlayMorse(String text) {
  for (int i = 0; i < text.length(); i++) {
    char c = toupper(text.charAt(i));
    if (c == ' ') {
      // Jeda antara kata
      delay(700);
    } else {
      int index = c - 'A';
      if (index >= 0 && index < 26) {
        String morseCode = morseCodeMapping[index];
        playMorseCode(morseCode);
      }
    }

    // Jeda antara karakter
    delay(300);
  }

  // Setelah selesai memainkan Morse, panggil fungsi untuk mengupdate flag
  updateFlag(currentMessageId);
}

void playMorseCode(String morseCode) {
  for (int i = 0; i < morseCode.length(); i++) {
    char c = morseCode.charAt(i);
    if (c == '.') {
      // Bunyi pendek
      digitalWrite(ledPin, HIGH);
      delay(200); // Durasi bunyi pendek (ms)
      digitalWrite(ledPin, LOW);
    } else if (c == '-') {
      // Bunyi panjang
      digitalWrite(ledPin, HIGH);
      delay(600); // Durasi bunyi panjang (ms)
      digitalWrite(ledPin, LOW);
    }

    // Jeda antara tanda Morse
    delay(200);
  }

  // Jeda antara karakter Morse
  delay(200);
}

void updateFlag(int messageId) {
  String url =
"http://triall090909.000webhostapp.com/Monitor%20&%20Controll/updateFlag.php?id=" +
String(messageId);
  HTTPClient http;
  http.begin(client, url);

  int httpCode = http.GET();
  if (httpCode == 200) {
    String response = http.getString();
    Serial.println("Response from Server: " + response);
    if (response.indexOf("Flag update successful") != -1) {
      Serial.println("Flag update successful");
    } else {
      Serial.println("Failed to update flag");
    }
  } else {
    Serial.println("Failed to connect to server");
  }

  http.end();
}
```

KOTLIN CODE

Main Activity

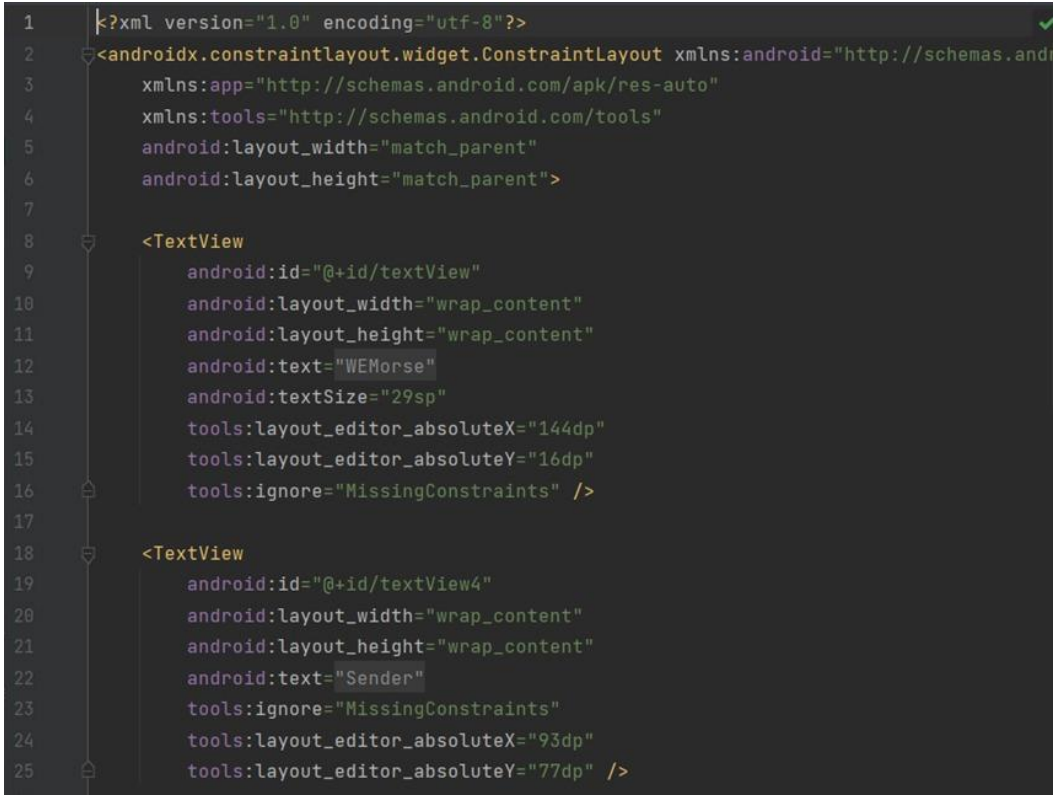
(Main Activity)

```
package com.example.morse

import ...

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MorseTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Greeting( name: "Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier
    )
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
```

```
@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    MorseTheme {
        Greeting( name: "User")
    }
```

Activity WeMorse

(Activity to create Mainmenu)

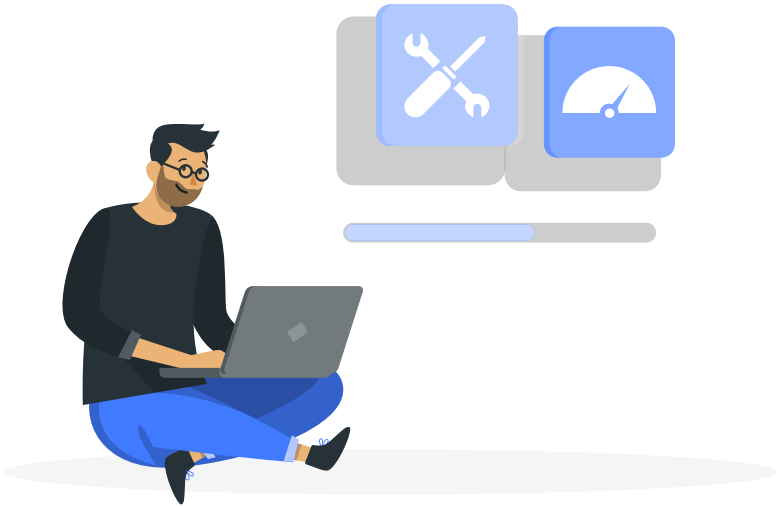


```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.andr
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="WEMorse"
        android:textSize="29sp"
        tools:layout_editor_absoluteX="144dp"
        tools:layout_editor_absoluteY="16dp"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sender"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="93dp"
        tools:layout_editor_absoluteY="77dp" />
```

WeMorse Activity

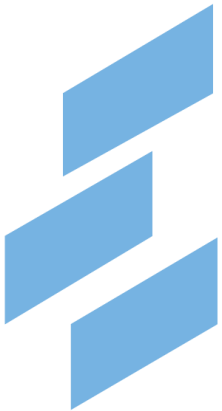
(Activity to create show data menu)

```
    <EditText
        android:id="@+id/editTextText2"
        android:layout_width="227dp"
        android:layout_height="66dp"
        android:autofillHints=""
        android:ems="10"
        android:inputType="text"
        android:text="Message"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.505"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="LabelFor,MissingConstraints,DuplicateSpeakableTextCheck"
        tools:layout_editor_absoluteY="195dp" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        tools:layout_editor_absoluteX="161dp"
        tools:layout_editor_absoluteY="336dp"
        tools:ignore="MissingConstraints" />


</androidx.constraintlayout.widget.ConstraintLayout>
```



Configuration

Minimum Configuration	
Hardware	Smartphone
Operating System	Android
Software	Wemorse



Thanks!