Flutter COM Automation Client
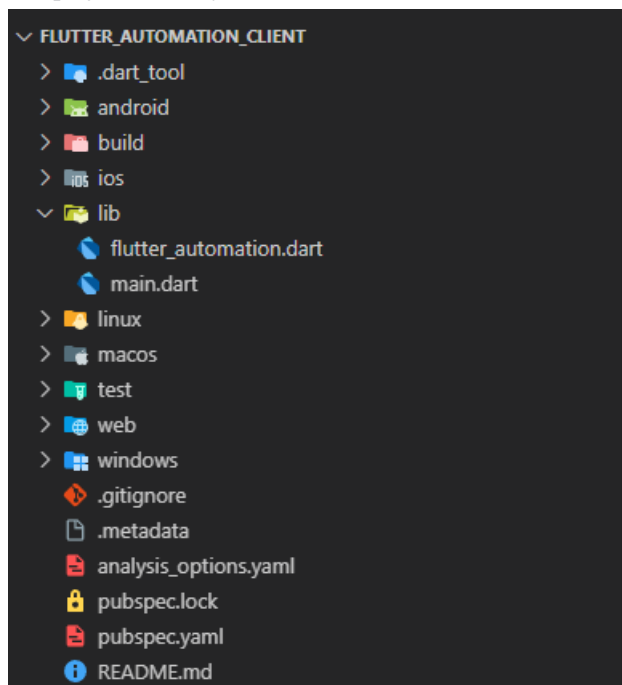
Steps

1. Download and install **Flutter SDK** - Flutter

2. Create a project using the following command

   ```
   flutter create flutter_automation_client
   ```

3. Create a file in `lib` folder named as `flutter_automation.dart`

   The project directory structure should look like this
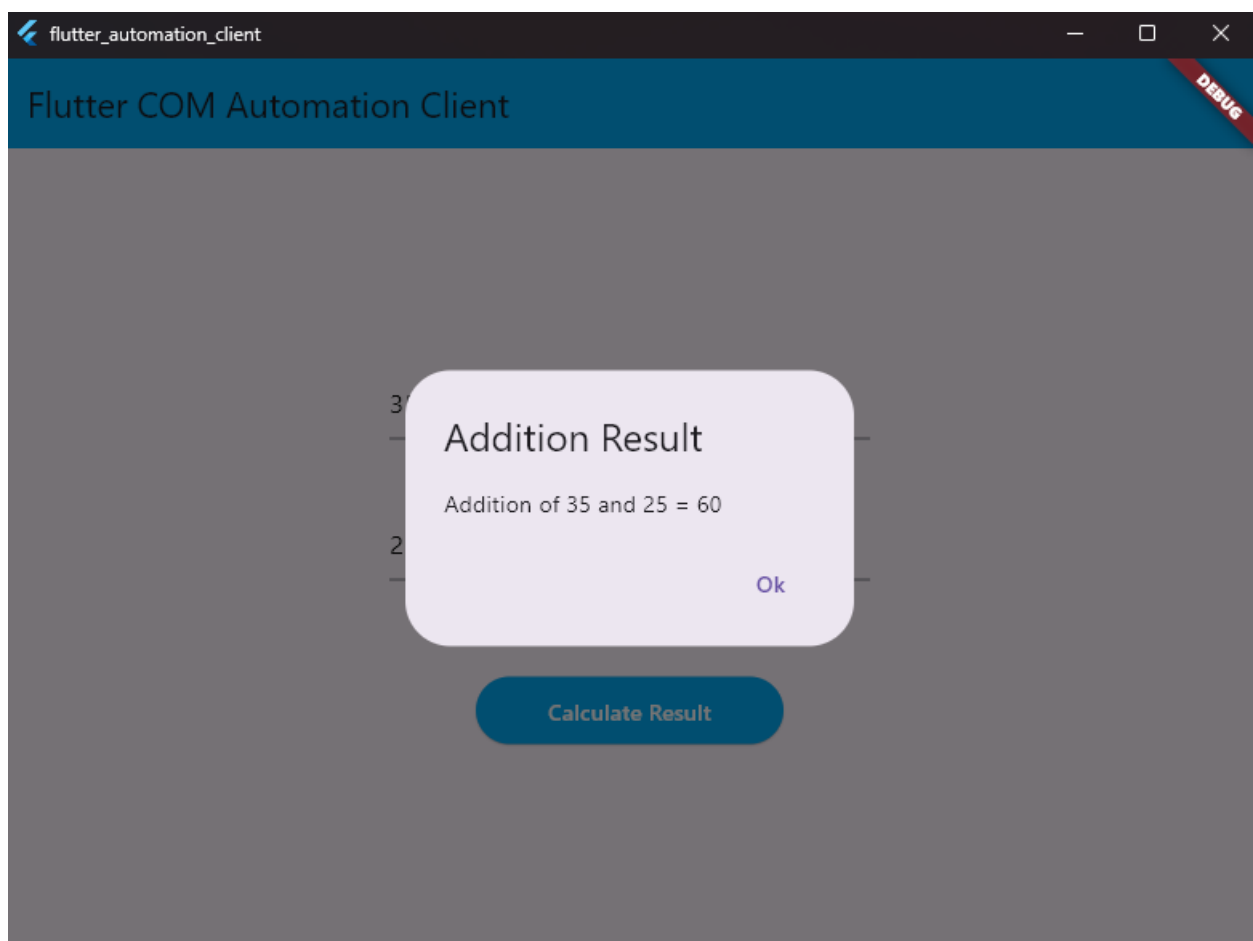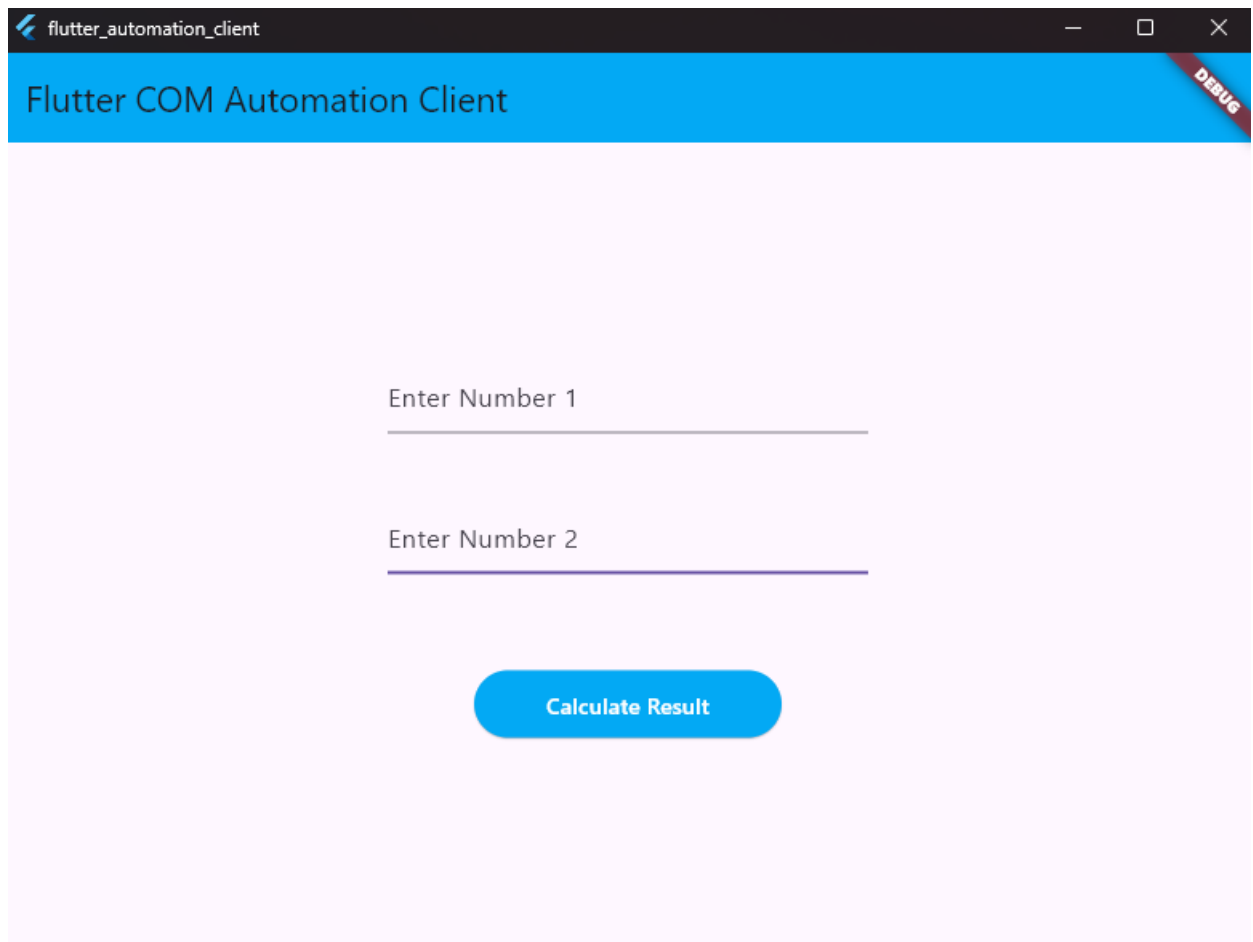

3. Install the required packages using the following command

   ```
   flutter pub add ffi win32
   ```

4. Type the code and replace **CLSID** with your **MyMath's CLSID** in `lib/flutter_automation.dart` . Also write the code for `lib/main.dart`

5. Run the project using the following command, and select `Windows` from the connected device list

   ```
   flutter run
   ```

   You should be able to see the following output

Flutter COM Automation Client

Enter Number 1

Enter Number 2

Calculate Result

Flutter COM Automation Client

3

Addition Result

Addition of 35 and 25 = 60

2

Ok

Calculate Result

Flutter COM Automation Client

3

2

Subtraction Result

Subtraction of 35 and 25 = 10

Ok

Calculate Result

Code

lib/flutter_automation.dart

```dart
import 'dart:ffi';

import 'package:ffi/ffi.dart';

import 'package:win32/win32.dart';


const clsIdMyMath = '{69DEFCBC-00E1-4817-A8D1-C608EE5C20F3}';


(int, int) calculateResult(int num1, int num2)
{
  // Code
  int sum = 0, subtraction = 0;


  // Initialize COM
  CoInitializeEx(
    nullptr,
    COINIT_APARTMENTTHREADED,
  );


  Dispatcher dispatcher = Dispatcher.fromCLSID(clsIdMyMath);


  final dispParams = calloc<DISPPARAMS>();

  final vArg = calloc<VARIANT>(2);


  VariantInit(vArg);
  {
    vArg[0].vt = VT_INT;
    vArg[0].intVal = num2;


    vArg[1].vt = VT_INT;
    vArg[1].intVal = num1;


    dispParams.ref.cArgs = 2;
```

```
    dispParams.ref.rgvarg = vArg;


    final vRetVal = calloc<VARIANT>();
    VariantInit(vRetVal);
    {
      dispatcher.invoke("SumOfTwoIntegers", dispParams, vRetVal);
      sum = vRetVal.ref.intVal;


      dispatcher.invoke("SubtractionOfTwoIntegers", dispParams, vRetVal);
      subtraction = vRetVal.ref.intVal;
    }
    VariantClear(vRetVal);
    free(vRetVal);
  }
  VariantClear(vArg);


  free(vArg);
  free(dispParams);


  dispatcher.dispose();


  // Uninitialize COM
  CoUninitialize();


  return (sum, subtraction);
}
```

lib/main.dart

```dart
import "package:flutter/material.dart";

import "package:flutter_automation_client/flutter_automation.dart";


void main() {
  runApp(const MyApp());
}


class MyApp extends StatelessWidget {
  const MyApp({super.key});


  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Flutter COM Automation Client",
      theme: ThemeData(
        useMaterial3: true,
      ),
      home: const MyHomePage(title: "Flutter COM Automation Client"),
    );
  }
}


class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});


  final String title;


  @override
  State<MyHomePage> createState() => _MyHomePageState();
}


class _MyHomePageState extends State<MyHomePage> {
  //* Variables
  final TextEditingController num1Controller = TextEditingController();
  final TextEditingController num2Controller = TextEditingController();
```

```dart
//* Code
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      backgroundColor: Colors.lightBlue,
      title: Text(widget.title),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          SizedBox(
            width: 300,
            child: TextField(
              controller: num1Controller,
              style: const TextStyle(color: Colors.black),
              expands: false,
              keyboardType: TextInputType.number,
              decoration: const InputDecoration(
                hintText: "Enter Number 1",
              ),
            ),
          ),
          const SizedBox(height: 40),
          SizedBox(
            width: 300,
            child: TextField(
              controller: num2Controller,
              expands: false,
              style: const TextStyle(color: Colors.black),
              keyboardType: TextInputType.number,
              decoration: const InputDecoration(
                hintText: "Enter Number 2",
              ),
            ),
          ),
```

```dart
      ),
      const SizedBox(height: 60),
      ElevatedButton.icon(
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.lightBlue,
          foregroundColor: Colors.white,
          minimumSize: const Size(200, 50),
        ),
        label: const Text("Calculate Result"),
        onPressed: () {
          // Call COM Functions using IDispatch
          int num1 = int.parse(num1Controller.text);
          int num2 = int.parse(num2Controller.text);

          var (addition, subtraction) = calculateResult(num1, num2);

          // Show Addition Result
          showDialog<String>(
            context: context,
            builder: (BuildContext context) => AlertDialog(
              title: const Text("Addition Result"),
              content: Text("Addition of $num1 and $num2 = $addition"),
              actions: <Widget>[
                TextButton(
                  onPressed: () => {
                    Navigator.pop(context, "Ok"),
                    // Show Subtraction Result
                    showDialog<String>(
                      context: context,
                      builder: (BuildContext context) => AlertDialog(
                        title: const Text("Subtraction Result"),
                        content: Text(
                            "Subtraction of $num1 and $num2 = $subtraction"),
                        actions: <Widget>[
                          TextButton(
                            onPressed: () => Navigator.pop(context, "Ok"),
                            child: const Text("Ok"),
```

```
                    ),
                  ],
                ),
              ),
            },
            child: const Text("Ok"),
          ),
        ],
      ),
    },
  ),
  ],
  ),
  ),
  );
}


@override
void dispose() {
  num2Controller.dispose();
  num1Controller.dispose();
  super.dispose();
}
}
```