

คู่มือการใช้งานแอป CoordinateTracker

เวอร์ชันของ Dependencies ที่ใช้ :

```
"@expo/vector-icons": "^14.0.3",
"@react-native-picker/picker": "2.7.5",
"expo": "~51.0.39",
"expo-file-system": "~17.0.1",
"expo-intent-launcher": "~11.0.1",
"expo-location": "~17.0.1",
"expo-sharing": "~12.0.1",
"expo-status-bar": "~1.12.1",
"react": "18.2.0",
"react-native": "0.74.5",
"react-native-table-component": "^1.2.2",
"react-native-vector-icons": "^9.0.0",
},
"devDependencies": {
  "@babel/core": "^7.20.0"
```

อธิบายโค้ด

App.js

ส่วนของหน้าจอหลัก

```
export default function App() {
  const [location, setLocation] = useState(null);
  const [errorMsg, setErrorMsg] = useState(null);
  const [options, setOptions] = useState({
    accuracy: Location.Accuracy.Highest,
    interval: 5000,
    distanceInterval: 1,
  });
  const [showOptions, setShowOptions] = useState(false);
  const [locationData, setLocationData] = useState([]);
  const [showSheet, setShowSheet] = useState(false);
  const [showCollectScreen, setShowCollectScreen] = useState(false);
```

ประกาศตัวแปร

location: ใช้เก็บข้อมูลตำแหน่งปัจจุบันของผู้ใช้

errorMsg: ใช้เก็บข้อความเมื่อไม่สามารถดึงข้อมูลตำแหน่งได้

options: ใช้เก็บการตั้งค่าของแอป โดยค่าเริ่มต้นคือ

- accuracy (ความแม่นยำ) ระดับสูงสุด
- interval(ความถี่เวลาในการอัปเดต) 5000 มิลลิวินาที
- distanceInterval (ความถี่ระยะทางในการอัปเดต) 1 เมตร

showOptions: ใช้สำหรับควบคุมการแสดงผลหน้าการตั้งค่า (OptionsScreen)

locationData : ใช้เก็บข้อมูลพิกัดทั้งหมดของผู้ใช้
showSheet : ใช้แสดงหน้าต่าง Sheet
showCollectScreen : ใช้แสดงหน้าต่าง CollectScreen

```
export default function App() {  
  useEffect(() => {  
  
    (async () => {  
      let { status } = await Location.requestForegroundPermissionsAsync();  
      if (status !== 'granted') {  
        setErrorMsg('Permission to access location was denied');  
        return;  
      }  
  
      locationSubscription = await Location.watchPositionAsync(  
        {  
          accuracy: options.accuracy,  
          timeInterval: options.timeInterval,  
          distanceInterval: options.distanceInterval,  
        },  
(newLocation) => {  
          setLocation(newLocation);  
  
          setLocationData((prevData) => [  
            ...prevData,  
            {  
              lat: newLocation.coords.latitude,  
              lon: newLocation.coords.longitude,  
              accuracy: newLocation.coords.accuracy,  
              timeInterval: options.timeInterval,  
              distanceInterval: options.distanceInterval,  
              time: new Date().toLocaleString(),  
            },  
          ])  
        })  
      );  
    })();  
  
    return () => {  
      if (locationSubscription) {  
        locationSubscription.remove();  
      }  
    };  
  }  
), [options]);  
}
```

useEffect

เมื่อคอมโพเนนต์เริ่มทำงาน และเมื่อการตั้งค่าใน Option มีการเปลี่ยนแปลง จะทำงาน ดังนี้

1. ขอสิทธิ์ในการเข้าถึงตำแหน่งของผู้ใช้ ถ้าสิทธิ์ถูกปฏิเสธ จะแสดงข้อความที่ปรากฏใน errorMsg.

2. ทำการติดตามตำแหน่งของผู้ใช้โดยอิงตามตัวเลือกใน Option ซึ่งค่าเริ่มต้นจะเป็น
 - accuracy (ความแม่นยำ) ระดับสูงสุด
 - setInterval(ความถี่เวลาในการอัปเดต) 5000 มิลลิวินาที
 - distanceInterval (ความถี่ระยะทางในการอัปเดต) 1 เมตร
3. เก็บข้อมูลลงตาราง โดยจะเก็บข้อมูล ดังนี้
 - Latitude ละติจูด
 - Longitude ลองจิจูด
 - Accuracy ความคลาดเคลื่อนของพิกัดในหน่วยเมตร
 - Time Interval ระยะเวลาที่จะอัปเดตข้อมูลพิกัด
 - Distance Interval ระยะทางที่จะอัปเดตข้อมูลพิกัด
 - Time เวลาที่บันทึกข้อมูล
4. เมื่อได้รับตำแหน่งใหม่ จะอัปเดตสถานะ location ด้วยค่าตำแหน่งใหม่
5. เมื่อมีการเปลี่ยนแปลงของ Option จะทำการติดตามตำแหน่งของผู้ใช้ในปัจจุบัน และทำการติดตามตำแหน่งของผู้ใช้ใหม่ตาม Option ใหม่ที่เลือก

```
const updateOptions = (newOptions) => {
  setOptions(newOptions);
};
```

updateOptions(newOptions)

ฟังก์ชันสำหรับอัปเดตการตั้งค่าให้เป็นไปตามที่เราต้องการ

```
let text = 'Waiting..';
if (errorMsg) {
  text = errorMsg;
} else if (location) {
  text = `Latitude: ${location.coords.latitude} \n Longitude: ${location.coords.longitude}`;
}
```

เริ่มแรกนั้น จะขึ้นข้อความว่า Waiting หลังจากนั้น หากมีข้อผิดพลาด จะขึ้น errorMsg แต่ถ้าได้รับ location มาเรียบร้อยแล้ว จะทำการแสดงละติจูดและลองจิจูดของตำแหน่งนั้น

```
return (
  <View style={styles.container}>
    {showOptions ? (
      <OptionsScreen options={options} updateOptions={updateOptions} setShowOptions={setShowOptions} />
    ) : showSheet ? (
      <SheetScreen data={locationData} setLocationData={setLocationData} setShowSheet={setShowSheet} />
    ) : showCollectScreen ? (
      <CollectScreen options={options} setShowCollectScreen={setShowCollectScreen} />
    ) : (
      <>
        <Text style={styles.paragraph}>{text}</Text>
        <TouchableOpacity style={styles.settingsButton} onPress={() => setShowOptions(true)}>
          <Icons name="settings" size={30} color="black" />
        </TouchableOpacity>
        <TouchableOpacity style={styles.saveButton} onPress={() => setShowSheet(true)}>
          <Icons name="document-text" size={30} color="black" />
        </TouchableOpacity>
        <TouchableOpacity style={styles.collectButton} onPress={() => setShowCollectScreen(true)}>
          <Icons name="timer" size={30} color="black" />
        </TouchableOpacity>
      </>
    )}
  </View>
);
```

ส่วนการแสดงผลทางหน้าจอ

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  paragraph: {
    fontSize: 18,
    textAlign: 'center',
  },
  settingsButton: {
    position: 'absolute',
    bottom: 20,
    right: 20,
    backgroundColor: '#fff',
    borderRadius: 25,
    padding: 10,
    elevation: 5,
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.3,
    shadowRadius: 3,
  },
  saveButton: {
    position: 'absolute',
    bottom: 80,
    right: 20,
    backgroundColor: '#fff',
    borderRadius: 25,
    padding: 10,
    elevation: 5,
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.3,
    shadowRadius: 3,
  },
  collectButton: {
    position: 'absolute',
    bottom: 140,
    right: 20,
    backgroundColor: '#fff',
    borderRadius: 25,
    padding: 10,
    elevation: 5,
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.3,
    shadowRadius: 3,
  },
});

```

ส่วนการตกแต่งภายในหน้าหลักแอป

OptionsScreen.js

เป็นหน้าต่างตั้งค่าของแอป

```
export default function OptionsScreen({ options, updateOptions, setShowOptions }) {  
  const [accuracy, setAccuracy] = useState(options.accuracy);  
  const [timeInterval, setTimeInterval] = useState(String(options.timeInterval));  
  const [distanceInterval, setDistanceInterval] = useState(String(options.distanceInterval));
```

ประกาศตัวแปร

- accuracy: ใช้เก็บค่าความแม่นยำของตำแหน่ง (สูง, สมดุล, ต่ำ)
- timeInterval: ใช้เก็บค่าเวลาในการอัปเดตตำแหน่ง
- distanceInterval: ใช้เก็บค่าระยะทางที่ต้องเคลื่อนที่ก่อนอัปเดตตำแหน่ง

```
const saveOptions = () => {  
  updateOptions({  
    accuracy: accuracy,  
    timeInterval: parseInt(timeInterval, 10),  
    distanceInterval: parseFloat(distanceInterval),  
  });  
  setShowOptions(false);  
};
```

saveOption

ฟังก์ชันสำหรับเซฟการตั้งค่าที่เราตั้งไว้ และปิดหน้าต่าง Safe

```
return (  
  );  
}
```

ส่วนการแสดงผลหน้าจอ

```
> const styles = StyleSheet.create({ ...  
  });
```

ส่วนการตกแต่งหน้าจอ

Sheet.js

```
export default function SheetScreen({ data, setLocationData, setShowSheet }) {  
  const generateCSV = () => {  
    const header = 'ID, Latitude, Longitude, Accuracy, Time Interval, Distance Interval, Time\n';  
    const rows = data  
      .map((item, index) => {  
        `${index + 1}, ${item.lat}, ${item.lon}, ${item.accuracy}, ${item.timeInterval}, ${item.distanceInterval}, ${item.time}`  
      })  
      .join('\n');  
    return header + rows;  
  };  
};
```

generateCSV

ฟังก์ชันสำหรับสร้าง CSV โดยจะมีการเก็บข้อมูล ดังนี้

- Lat ละติจูด
- Lon ลองจิจูด
- accuracy ความคลาดเคลื่อนของพิกัด
- timeInterval ระยะเวลาที่อัปเดต
- distanceInterval; ระยะทางที่อัปเดต
- time เวลาที่บันทึก

```
const saveCSV = async () => {  
  const csv = generateCSV();  
  const fileUri = FileSystem.documentDirectory + 'locationData.csv';  
  
  await FileSystem.writeAsStringAsync(fileUri, csv, {  
    encoding: FileSystem.EncodingType.UTF8,  
  });  
  
  Sharing.shareAsync(fileUri);  
};
```

saveCSV

ฟังก์ชันสำหรับส่งออกไฟล์ CSV

```
const clearData = () => {  
  setLocationData([]);  
  alert("Cleared!!");  
};
```

clearData

ฟังก์ชันสำหรับเคลียร์ข้อมูลที่บันทึกทั้งหมด

```
> return (...  
);  
}
```

ส่วนของการแสดงผลหน้าจอ

```
> const styles = StyleSheet.create({  
});
```

ส่วนของการตกแต่งหน้าจอ

CollectionScreen.js

หน้าแสดงค่าเฉลี่ยของพิกัด

```
export default function CollectScreen({ options, setShowCollectScreen }) {  
  const [duration, setDuration] = useState(1);  
  const [locationData, setLocationData] = useState([]);  
  const [averageLocation, setAverageLocation] = useState(null);  
  const [isCollecting, setIsCollecting] = useState(false);
```

ประกาศตัวแปร

- duration ระยะเวลาที่ต้องการใช้ในหน่วยนาที
- locationData ข้อมูลพิกัดทั้งหมดของตำแหน่งปัจจุบัน
- averageLocation ข้อมูลเฉลี่ยของพิกัดปัจจุบัน
- isCollecting สถานะว่ากำลังเก็บข้อมูลอยู่หรือไม่

```
const startCollection = async () => {  
  setIsCollecting(true);  
  setLocationData([]);  
  setAverageLocation(null);  
  
  const endTime = Date.now() + duration * 60000;  
  
  const locationSubscription = await Location.watchPositionAsync(  
    {  
      accuracy: options.accuracy,  
      timeInterval: options.timeInterval,  
      distanceInterval: options.distanceInterval,  
    },  
    (newLocation) => {  
      setLocationData((prevData) => [...prevData, newLocation.coords]);  
    }  
  );  
  
  const stopCollection = () => {  
    locationSubscription.remove();  
    setIsCollecting(false);  
  
    if (locationData.length > 0) {  
      const avgLat = locationData.reduce((sum, loc) => sum + loc.latitude, 0) / locationData.length;  
      const avgLon = locationData.reduce((sum, loc) => sum + loc.longitude, 0) / locationData.length;  
      const avgAcc = locationData.reduce((sum, loc) => sum + loc.accuracy, 0) / locationData.length;  
  
      setAverageLocation({ lat: avgLat, lon: avgLon, accuracy: avgAcc });  
    } else {  
      alert("No location data collected.");  
    }  
  };  
  
  setTimeout(stopCollection, duration * 60000);  
};
```

startCollection / stopCollection

ฟังก์ชันสำหรับเก็บข้อมูลค่าพิกัดเฉลี่ย โดยจะรับตำแหน่งด้วยการตั้งค่าที่เราตั้งค่าไว้ ในระยะเวลาที่ระบุ และนำมาคำนวณค่าเฉลี่ย

```
const retryCollection = () => {
  setAverageLocation(null);
  setLocationData([]);
};
```

RetryCollection

ฟังก์ชันสำหรับกลับหน้าคำนวณค่าเฉลี่ยอีกครั้ง

```
return (
  <View style={styles.container}>
    {averageLocation ? (
      <View style={styles.resultContainer}>
        <Text style={styles.text}>Average Latitude: {averageLocation.lat}</Text>
        <Text style={styles.text}>Average Longitude: {averageLocation.lon}</Text>
        <Text style={styles.text}>Average Accuracy: {averageLocation.accuracy} meters</Text>
        <View style={styles.buttonContainer}>
          <Button title="Retry" onPress={retryCollection} />
        </View>
      </View>
    ) : (
      <>
        <Text style={styles.title}>Specify Duration (in minutes):</Text>
        <TextInput
          style={styles.input}
          keyboardType="numeric"
          value={duration.toString()}
          onChangeText={(value) => setDuration(parseInt(value) || 1)}
          editable={!isCollecting}
        />
        <View style={styles.buttonContainer}>
          <Button
            title={isCollecting ? "Collecting..." : "Start Collection"}
            onPress={startCollection}
            disabled={isCollecting}
          />
        </View>
      </>
    )}
    <View style={styles.buttonContainer}>
      <Button title="Back" onPress={() => setShowCollectScreen(false)} />
    </View>
  </View>
);
}
```

ส่วนการแสดงผลหน้าจอ


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    padding: 20,
  },
  title: {
    fontSize: 18,
    marginBottom: 10,
  },
  text: {
    fontSize: 18,
  },
  input: {
    height: 40,
    borderColor: '#ccc',
    borderWidth: 1,
    marginBottom: 20,
    paddingHorizontal: 10,
  },
  resultContainer: {
    marginTop: 20,
  },
  buttonContainer: {
    marginVertical: 10,
  },
});
```

ส่วนการตกแต่งหน้าจอ

App.json

```
{
  "expo": {
    "name": "CoordinateTracker",
    "slug": "CoordinateTracker",
    "version": "1.0.0",
    "orientation": "portrait",
    "icon": "./assets/icon.png",
    "userInterfaceStyle": "light",
    "splash": {
      "image": "./assets/splash.png",
      "resizeMode": "contain",
      "backgroundColor": "#ffffff"
    },
    "ios": {
      "supportsTablet": true
    },
    "android": {
      "adaptiveIcon": {
        "foregroundImage": "./assets/adaptive-icon.png",
        "backgroundColor": "#ffffff"
      },
      "package": "com.saviorwf.CoordinateTracker"
    },
    "web": {
      "favicon": "./assets/favicon.png"
    },
    "extra": {
      "eas": {
        "projectId": "566a188e-ce6b-45ab-b6a4-621acf5e4139"
      }
    }
  }
}
```

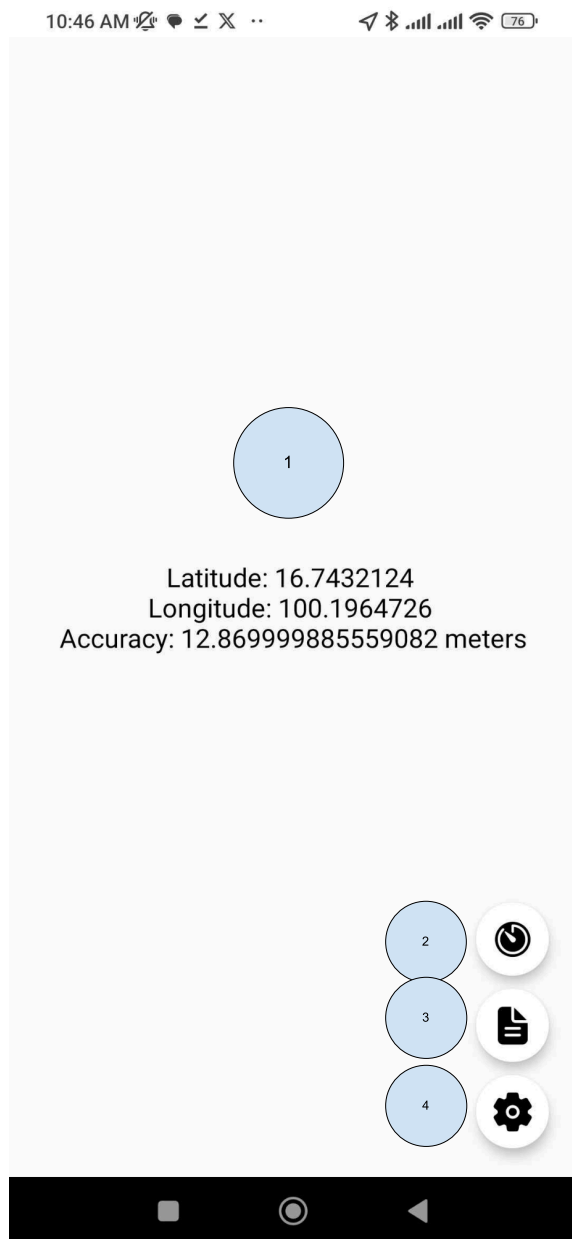
ส่วนสำหรับการตั้งชื่อและตั้งไอคอน

Package.json

```
{
  "name": "coordinatetracker",
  "version": "1.0.0",
  "main": "expo/AppEntry.js",
  ▶ Debug
  "scripts": {
    "start": "expo start",
    "android": "expo start --android",
    "ios": "expo start --ios",
    "web": "expo start --web"
  },
  "dependencies": {
    "expo": "~51.0.39",
    "expo-status-bar": "~1.12.1",
    "react": "18.2.0",
    "react-native": "0.74.5",
    "expo-location": "~17.0.1",
    "@react-native-picker/picker": "2.7.5",
    "react-native-vector-icons": "^9.0.0",
    "@expo/vector-icons": "^14.0.3"
  },
  "devDependencies": {
    "@babel/core": "^7.20.0"
  },
  "private": true
}
```

ส่วนสำหรับดูชื่อแอป เวอร์ชัน คำสั่งที่ใช้ทดสอบ เวอร์ชันของ Dependencies แต่ละตัว

อธิบาย UI ของแอป



หน้าหลัก

- 1.พื้นที่แสดงละติจูด ลองจิจูด และความคาดเคลื่อน
- 2.ปุ่มแสดงหน้าเก็บข้อมูลค่าเฉลี่ย
- 3.ปุ่มแสดงหน้า Sheet และส่งออกเป็น .csv
- 4.ปุ่มแสดงหน้าตั้งค่า

10:46 AM 76

Specify Duration (in minutes):

1

START COLLECTION

BACK

หน้าเก็บข้อมูลค่าเฉลี่ย

สามารถกรอกเวลาที่ต้องการเก็บค่าได้ เช่นในที่นี้ต้องการจะเก็บค่า 1 นาที เมื่อกรอกเสร็จให้กดปุ่ม Start Collecting รอ 1 นาที จะแสดงผลดังภาพต่อไป

10:49 AM



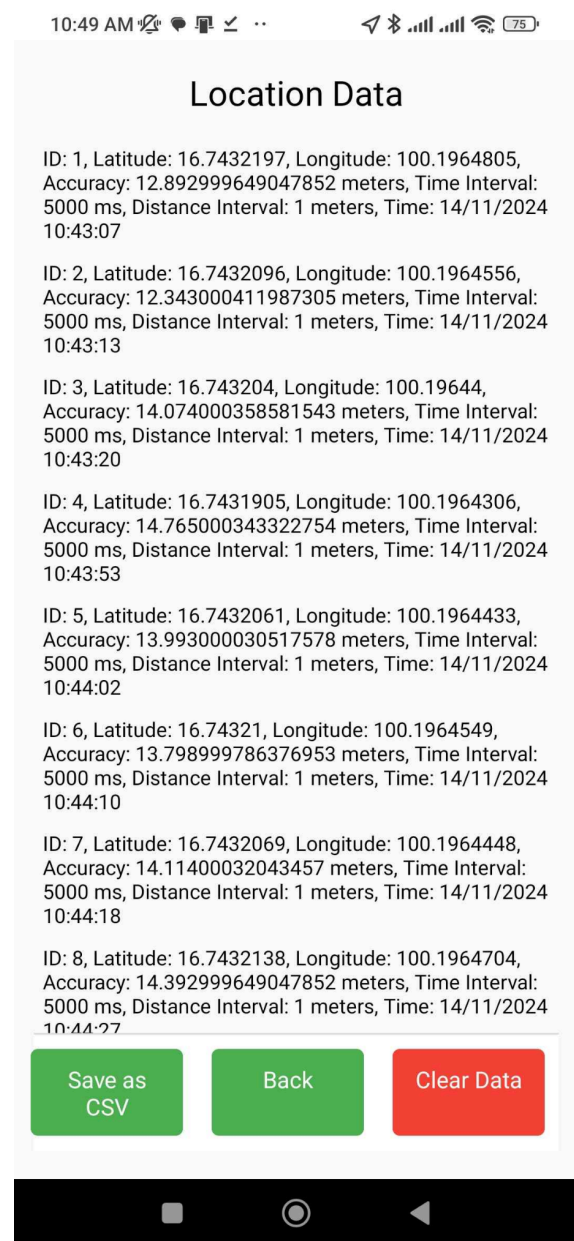
Average Latitude: 16.7431991125
Average Longitude: 100.19643105
Average Accuracy: 16.026249885559082
meters

RETRY

BACK

โดยสามารถกดปุ่ม Retry เพื่อกลับไปบันทึกค่าเฉลี่ยอีกครั้งได้

ส่วนของ Sheet และการส่งออก CSV



ภายในหน้าจอก็จะมีการแสดงรายละเอียด ดังนี้

- ID หมายเลขของการเก็บข้อมูล
- Latitude ละติจูด
- Longitude ลองจิจูด
- Accuracy ความคลาดเคลื่อนของพิกัดในหน่วยเมตร
- Time Interval ระยะเวลาที่จะอัปเดตข้อมูลพิกัด
- Distance Interval ระยะทางที่จะอัปเดตข้อมูลพิกัด
- Time เวลาที่บันทึกข้อมูล

โดยที่เราสามารถกด Save as CSV เพื่อบันทึกข้อมูลออกมาเป็นไฟล์ CSV ได้ และสามารถ Clear Data เพื่อล้างข้อมูลได้

ส่วนการตั้งค่า

10:49 AM 75%

Adjust Location Settings

Accuracy

Highest (5)

Time Interval (ms)

5000

Distance Interval (meters)

1

←

📁

Accuracy: ค่าความแม่นยำของตำแหน่ง (สูงสุด, สูง, สมดุล, ต่ำ, ต่ำสุด)

Time Interval: กำหนดให้พิกัดอัปเดตทุกๆ กี่มิลลิวินาที

Distance Interval: กำหนดให้พิกัดอัปเดตทุกๆ กี่เมตร

ทางด้านขวาล่าง ปุ่มลูกศรคือย้อนกลับไปที่หน้าแรกโดยไม่บันทึกการตั้งค่า รูปแผ่นดิสก์คือบันทึกการตั้งค่าและกลับไปหน้าแรก