

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

แนวคิดทฤษฎีการมีส่วนร่วม [1] (Participation) เป็นกระบวนการสื่อสารในระบบเปิด ซึ่งเป็นการสื่อสาร สองทางระหว่างบุคคล กลุ่มบุคคล ชุมชน หรือองค์การ ในการดำเนินกิจกรรมใดกิจกรรมหนึ่ง หรือ หลายกิจกรรมทั้งเป็นทางการและไม่เป็นทางการ เป็นทฤษฎีที่มีส่วนเกี่ยวข้องและเชื่อมโยงกับการมีส่วนร่วม ทฤษฎีปัจจัยของ เฮอ์เบิร์ก (Herzberg) เป็นทฤษฎีการจูงใจที่เกี่ยวข้องและสามารถโยงไปสู่กระบวนการมีส่วนร่วมได้ เป็นแนวคิดเกี่ยวกับชักชวนให้ผู้คนเกิดความพึงพอใจ กระตือรือร้นที่จะทำกิจกรรมซึ่งเป็นการเพิ่มความสัมพันธ์ให้มากขึ้นในสังคม ตรงกันข้ามหากผู้คนไม่พึงพอใจในการทำกิจกรรมก็จะเกิดปัญหาการเข้าร่วมสังคม ทฤษฎีดังกล่าวสอดคล้องกับการเข้ามามีส่วนร่วมในกิจกรรม กล่าวคือ ถ้าผู้คนได้เข้ามามีส่วนร่วมในการดำเนินกิจกรรมได้ร่วมคิดตัดสินใจจะส่งผลให้บุคคลนั้นเกิดความรู้สึกเป็นผู้สร้างสรรค์ในกิจกรรมมากขึ้น และอาจจะประสบผลสำเร็จในการวางแผนดำเนินกิจกรรมได้ นอกจากนี้ทฤษฎีการสร้างผู้นำก็มีความสำคัญ คือ ผู้นำที่ดีมักจะนำการเคลื่อนไหวในการวางแผนอยู่เสมอ ในขณะที่ผู้นำที่ไม่ดีจะไม่มี การวางแผนงานที่สร้างสรรค์เลย การสร้างผู้นำจะช่วยจูงใจให้ผู้คนเต็มใจที่จะทำกิจกรรมเพื่อให้กิจกรรมตรงตามวัตถุประสงค์ที่ร่วมกัน เนื่องจากผู้นำเป็นผู้ที่มีความสำคัญในการวางแผนจูงใจและรวมกลุ่มคน ดังนั้นทฤษฎีนี้จึงมีส่วนเกี่ยวข้องกับการมีส่วนร่วมของผู้คน เพราะทำให้เกิดการมีส่วนร่วมในการช่วยเหลือร่วมมือร่วมแรงกันอย่างมีคุณภาพ แสดงให้เห็นถึงการมีความคิดริเริ่มสร้างสรรค์ของผู้ตามและผู้นำร่วมกัน ซึ่งกระบวนการนี้จะต้องมีผู้นำที่ดีอันจะนำไปสู่ความสำเร็จของกิจกรรมได้

2.1.1 ความหมายของการมีส่วนร่วม

ความหมายของการมีส่วนร่วม มีนักวิชาการทั้งชาวไทย และชาวต่างประเทศที่มีความรู้ความสามารถ และทำวิจัย แต่งตำรา เกี่ยวกับเรื่องนี้มานาน ได้ให้ความหมายของการมีส่วนร่วม ดังรายละเอียด ดังนี้

Aronstein (1969) การมีส่วนร่วม หมายถึง การเข้าไปมีส่วนร่วมโดยไม่มีบทบาทอะไรเลย ย่อมไม่ได้ผลการมีส่วนร่วมที่มีคุณภาพนั้น ผู้เข้าร่วมจะต้องรู้จักใช้อำนาจและสามารถควบคุมกิจกรรมนั้น ได้จึงจะทำให้เกิดผลอย่างมีประสิทธิภาพ

Putti (1987) การมีส่วนร่วม หมายถึง พื้นฐานของกิจกรรมต่างๆ ที่จะส่งผลให้การบริหารจัดการมีลักษณะกว้าง ซึ่งเป็นทางหนึ่งที่จะทำให้การมีส่วนร่วมขยายไปสู่การปฏิบัติงานในระดับล่างขององค์การ

จิราภรณ์ ศรีคำ (2547) การมีส่วนร่วม หมายถึง การที่บุคคลที่มีความสนใจหรือมีส่วนเกี่ยวข้องในเรื่องเดียวกันเข้ามาร่วมกัน เพื่อปฏิบัติภารกิจ ไม่ว่าจะเป็นการวางแผน การดำเนินงาน การรับทราบผลการดำเนินงาน การติดตามประเมินผล หรือร่วมกันทำกิจกรรมต่างๆ เพื่อบรรลุเป้าหมายตามที่ได้ตกลงกันไว้

นรินทร์ชัย พัฒนพงศา (2546) การมีส่วนร่วม หมายถึง การที่ฝ่ายหนึ่งฝ่ายใดที่ไม่เคยได้เข้าร่วมในกิจกรรมต่างๆ หรือเข้าร่วมการตัดสินใจหรือเคยเล็กน้อยได้เข้าร่วมด้วยมากขึ้น เป็นไปอย่างมีอิสรภาพ เสมอภาค มิใช่เพียงมีส่วนร่วมอย่างผิวเผินแต่เข้าร่วมด้วยแท้จริงยิ่งขึ้น

เมตต์ เมตต์การุณจิต (2553) การมีส่วนร่วม หมายถึง การเปิดโอกาสให้บุคคลหรือกลุ่มบุคคลเข้ามามีส่วนร่วมในกิจกรรม ไม่ว่าจะเส้นทางตรงหรือทางอ้อม ในลักษณะของการร่วมรับรู้ ร่วมคิด ร่วมทำ ร่วมตัดสินใจ ร่วมติดตามผล

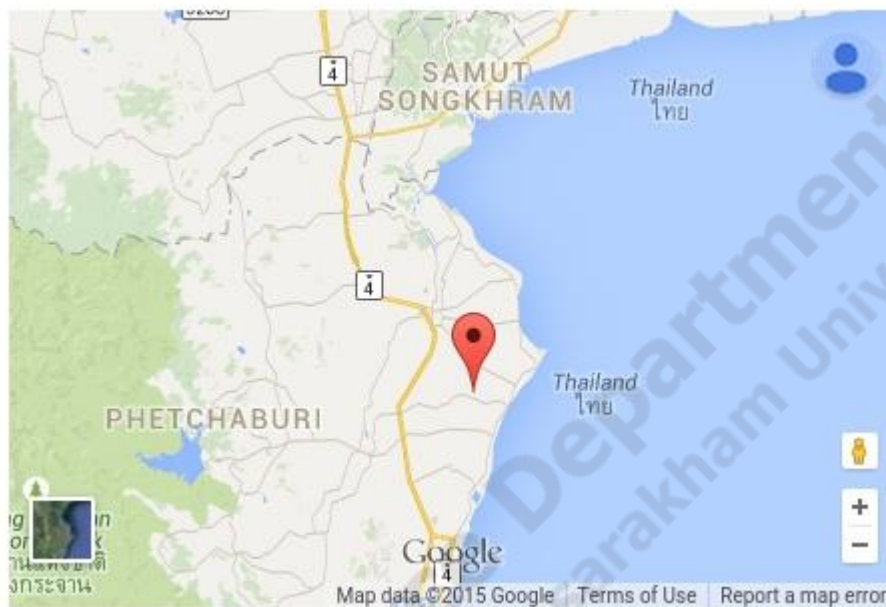
วิลเลียม (William, 1980 อ้างถึงใน ธนวัฒน์ ค ภาสิานนท์, 2550, หน้า 9) กล่าวว่า การมีส่วนร่วม หมายถึง การเกี่ยวข้องทางสังคมด้านจิตใจและอารมณ์ (Mental and Emotional Involvement) ของบุคคลหนึ่งในสถานการณ์ (Group Situation) ซึ่งผลของการเกี่ยวข้องดังกล่าวเป็นเหตุเร้าใจให้กระทำ การให้ (Contribution) บรรลุจุดมุ่งหมายของกลุ่มนั้น ทาให้เกิดความรู้สึกมีส่วนร่วมรับผิดชอบกับกลุ่ม

2.1.2 ลักษณะของการมีส่วนร่วม

ลักษณะการมีส่วนร่วม คือ การมีส่วนร่วมในระดับกิจกรรม ได้แก่ การรับรู้ข่าวสาร การปรึกษาหารือ การประชุมรับฟังความคิดเห็น การลงประเมิน และการมีส่วนร่วมในระดับการวางแผน ได้แก่ การมีส่วนร่วมในการตัดสินใจ ในประเด็นการมีส่วนร่วมในระดับการวางแผนนี้ ยังจะต้องพิจารณาจากสถานการณ์ ทุกอย่างจะมีความเท่าเทียมกันในกิจกรรมและกิจกรรมเป็นการมีส่วนร่วมตามความสมัครใจ

2.2 เทคโนโลยีที่เกี่ยวข้อง

2.2.1 Google Maps API



ภาพประกอบที่ 2.1 Google Maps API

ที่มา : <https://swiftlet.co.th/google-api->

Google Maps API [2] ทำให้แอปพลิเคชันของผู้ใช้งานสามารถแสดงตำแหน่งของบาร์ ร้านอาหาร คาเฟ่ สนามบิน หรือร้านขายของชำที่อยู่ใกล้กับผู้ใช้ก็สามารถให้รายชื่อสถานที่ที่ผ่านการกรอง ซึ่งมีความเกี่ยวข้องกับผู้ใช้มากที่สุด Google Maps API เป็นชุด API ของ Google สำหรับพัฒนา web application และ mobile application (Android, iOS) ไว้สำหรับเรียกใช้แผนที่และชุด service ต่างๆ ของ Google เพื่อพัฒนา Application ได้เหมือนกับที่ Google โดยแผนที่ยัง features ต่างๆ มากมาย ให้เรียกใช้ เช่น

- ชุดควบคุมแผนที่ (Map Control)
- การปรับแต่งแผนที่ (Styled Map)
- การนำทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Directions Service) Street View
- ชุดเครื่องมือวาดภาพบนแผนที่ (Drawing)
- การคำนวณความสูงของจุดพิกัด (Elevation Service)
- การดึงข้อมูล POI (Point of Interest) คือข้อมูลสถานที่ต่าง ๆ ที่ Google รวบรวมไว้ให้ เช่น

โรงแรมห้างสรรพสินค้า

- การแปลงที่อยู่เป็นพิกัด Latitude และ Longitude (Geocoding Service)
โรงเรียน สถานที่ราชการต่างๆ และอื่นๆ อีกมากมาย (Places API) มาใช้งานใน application เรา

2.2.1.1 การติดตั้ง Google Maps API

1. อันดับแรก [3] ต้องทำการขอ API Key ที่ <https://cloud.google.com/maps-platform/> เมื่อเข้ามาหน้าเว็บไซต์แล้วให้เข้าไปที่ Console

2. สร้าง Project ขึ้นมาก่อน เมื่อทำการสร้างเสร็จแล้วให้เปิดแท็บด้านขวามือ แล้วเลือกเมนูที่ชื่อว่า Google Maps

New Project

You have 6 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
My Project 68529

Project ID: vaulted-program-281304. It cannot be changed later. [EDIT](#)

Location *
No organisation [BROWSE](#)

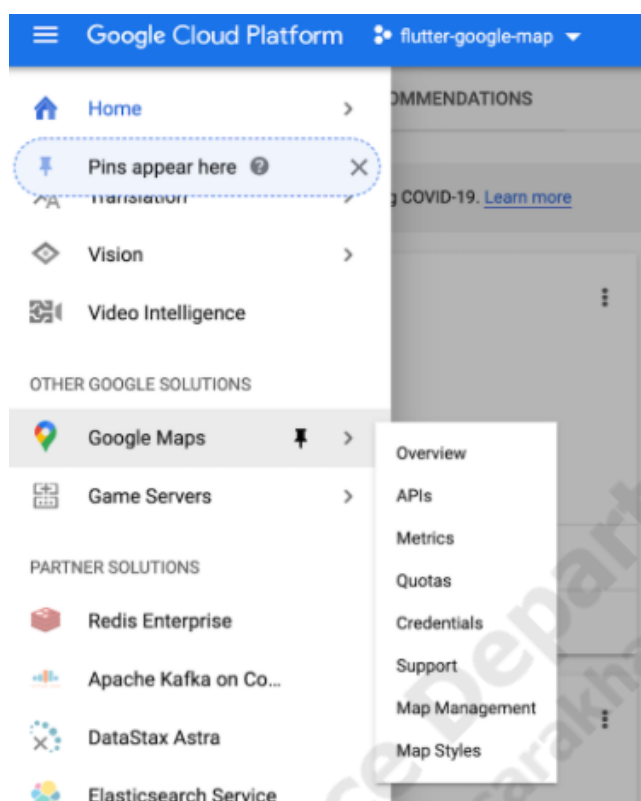
Parent organisation or folder

[CREATE](#) [CANCEL](#)

ภาพประกอบที่ 2.2 การติดตั้ง Google Maps API

ที่มา : [\[How To\] การติดตั้ง Google Maps for Flutter เบื้องต้น | Centillion Technology](#)

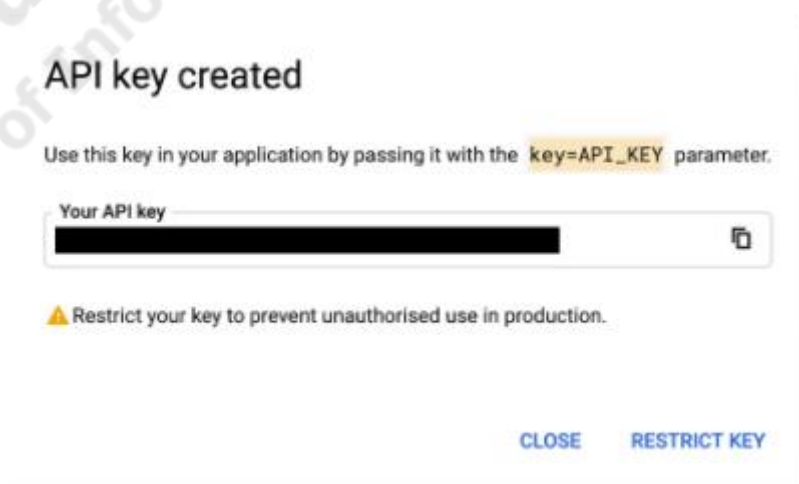
3. Google Maps จะมี API ให้ใช้หลายตัว เราสามารถเข้าไป Enable เพื่อที่จะเรียกใช้ API เหล่านั้นได้แต่ตอนนี้เราจะเลือก Enable Maps SDK for Android / IOS เพื่อที่จะเอา API Key ไปเรียกใช้งาน API



ภาพประกอบที่ 2.3 Google Maps มี API

ที่มา : [\[How To\] การติดตั้ง Google Maps for Flutter เบื้องต้น | Centillion Technology](#)

4. ต่อมาให้เข้าไปที่เมนู Credentials เพื่อที่จะสร้าง API Key เอาไปแปะในโค้ดให้เชื่อมต่อกับ Google Cloud Platform กดเลือก Create Credentials -> API Key เราก็จะได้ API Key มาเท่านี้ก็จะสามารถเชื่อมต่อกับ Google Map API ได้



ภาพประกอบที่ 2.4 API Key created

ที่มา : [\[How To\] การติดตั้ง Google Maps for Flutter เบื้องต้น | Centillion Technology](#)

5. สำหรับ Android ให้เข้าไปเพิ่ม API Key ที่ไฟล์ AndroidManifest.xml ในโฟลเดอร์ Android (android/app/src/main/AndroidManifest.xml) และเพิ่ม เพื่อขอ permission กับ เพื่อเพิ่ม API KEY เข้าไป

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.fluttergooglemap">

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>

    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="fluttergooglemap"
        android:icon="@mipmap/ic_launcher">

        <meta-data android:name="com.google.android.geo.API_KEY"
            android:value="YOUR API KEY"/>

        <activity
            android:name=".MainActivity"
            ...
```

ภาพประกอบที่ 2.5 เพิ่ม API Key ที่ไฟล์ AndroidManifest.xml

ที่มา : [\[How To\] การติดตั้ง Google Maps for Flutter เบื้องต้น | Centillion Technology](#)

6. เข้าไปที่ไฟล์ pubspec.yaml ติดตั้ง google_maps_flutter: ^0.5.28+1 (เวอร์ชันล่าสุดเมื่อ May 21, 2020) และ import Library

7. สร้างตัวแปรประเภท GoogleMapController คือ class ที่เอาไว้สร้าง Future อีกที่ เอาไว้ควบคุมการทำงาน

```
class _MyHomePageState extends State<MyHomePage> {
  Completer<GoogleMapController> _controller = Completer();
```

ภาพประกอบที่ 2.6 สร้างตัวแปรประเภท GoogleMapController

ที่มา : [\[How To\] การติดตั้ง Google Maps for Flutter เบื้องต้น | Centillion Technology](#)

8. Library จะมี Widget มาให้ชื่อว่า Google Map () เริ่มต้น Google Map จะต้อง initialCameraPosition เสมอ โดยกำหนดพิกัดจุดเริ่มต้นสถานที่ที่จะให้ Google Map แสดงขึ้นมา แล้วลอง Run

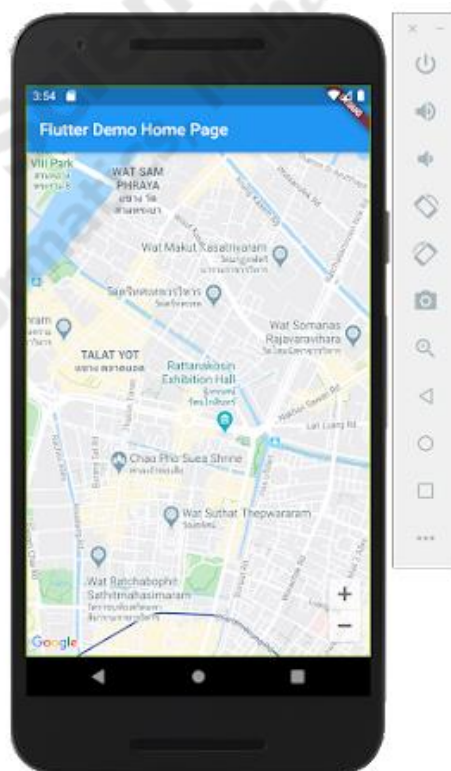
```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ),
    body: GoogleMap(
      initialCameraPosition: CameraPosition(
        target: LatLng(13.757429, 100.502465), //กำหนดพิกัดเริ่มต้นบนแผนที่
        zoom: 15, //กำหนดระยะการซูม สามารถกำหนดค่าได้ 0-20
      ),
      onMapCreated: (GoogleMapController controller) {
        _controller.complete(controller);
      },
    ),
  );
}

```

ภาพประกอบที่ 2.7 การกำหนดพิกัดจุดเริ่มต้นให้กับสถานที่

ที่มา : [\[How To\] การติดตั้ง Google Maps for Flutter เบื้องต้น | Centillion Technology](#)



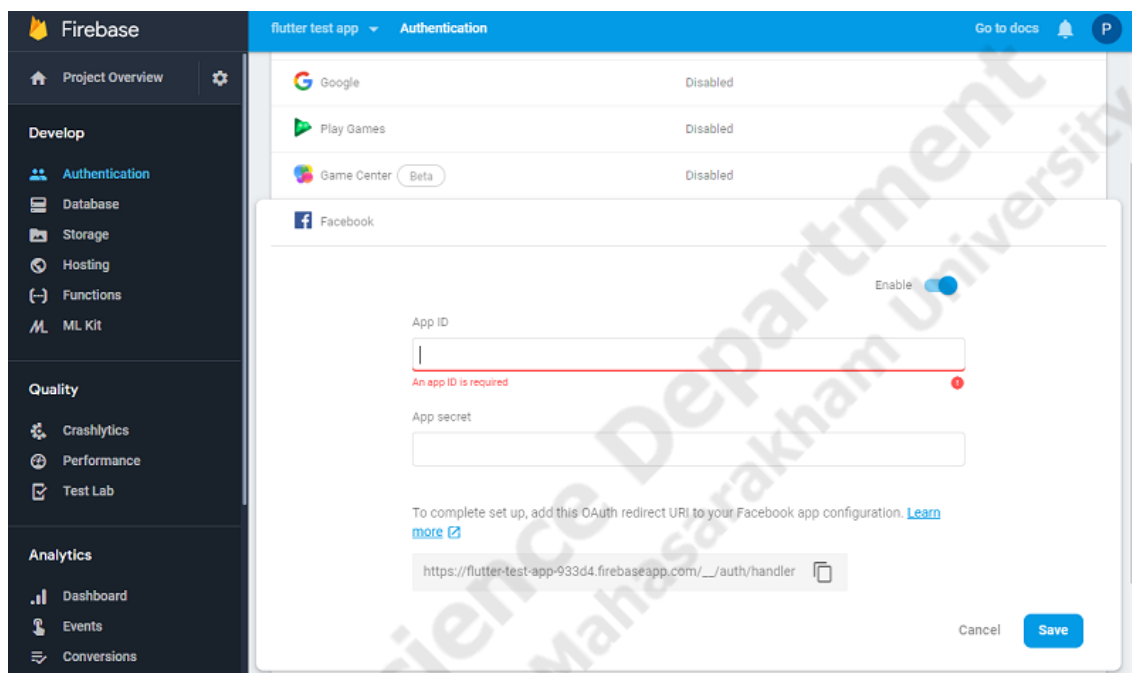
ภาพประกอบที่ 2.8 ตัวอย่างหน้าแสดง Maps

ที่มา : [\[How To\] การติดตั้ง Google Maps for Flutter เบื้องต้น | Centillion Technology](#)

2.2.2 Flutter Facebook Login

2.2.2.1 เปิดใช้งาน Facebook ใน Firebase Auth

ก่อนอื่นเปิดใช้งาน [4] Facebook ใน Firebase โดยไปที่ Firebase Auth > Sign in method > Facebook กด enable จะให้เรากรอก App ID ถ้าเรายังไม่มีให้ไปสร้างในขั้นตอนถัดไป



ภาพประกอบที่ 2.9 เปิดใช้งาน Facebook ใน Firebase Auth

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

2.2.2.2 สร้างแอปใน Facebook

ให้เข้าไปที่ <https://developers.facebook.com/apps/> จากนั้นกด create app แล้วกรอกข้อมูล

สร้าง ID ของแอปใหม่

เริ่มรวมระบบ Facebook เข้ากับแอปหรือเว็บไซต์ของคุณ

ชื่อที่แสดง

My Firebase Flutter

อีเมลติดต่อ

benz102@hotmail.com

เมื่อดำเนินการต่อ คุณยินยอมต่อ นโยบายแพลตฟอร์มของ Facebook

ยกเลิก

สร้าง ID ของแอป

ภาพประกอบที่ 2.10 สร้างแอปใน Facebook

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

จากนั้นไปที่แอปของเราดูที่ ดูแอปทั้งหมดแล้วเลือกแอปที่เราเพิ่งสร้างเมื่อสักครู่ที่เมนูข้อมูลพื้นฐานจะมี App ID และ secret ที่เราต้องการอยู่ให้ copy ไปใส่ใน Firebase ได้เลย

2.2.2.3 ตั้งค่าใน Facebook Developer

หลังจากกำหนด App ID แล้ว ต่อมาคือต้องทำให้ Facebook ระบุตัวตน app ของเรา โดยการกำหนดค่าในข้อมูลพื้นฐานที่ Facebook ต้องการให้ครบ เช่น Android คือระบุ package name , class , hash key ส่วนของ iOS ก็ระบุ App ID , iPhone Store IDhash key ห้ามลืมกดเปิด app ให้ Publish

2.2.2.4 กำหนด Facebook App ID ใน Project

จากนั้นเราต้องกำหนด App Id ใน project แยกตาม Platformเริ่มจากให้ copy resource ข้างล่างนี้ไปไว้ในโปรเจกต์ของเรา ไฟล์ที่ชื่อ strings.xml หากไม่มีไฟล์ strings.xml ให้สร้างใหม่ได้เลย แล้วใส่ App id ของเรา แทน 000000000000

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Your App Name here.</string>

  <!-- Replace "000000000000" with your Facebook App ID here. -->
  <string name="facebook_app_id">000000000000</string>

  <!--
  Replace "000000000000" with your Facebook App ID here.
  **NOTE**: The scheme needs to start with `fb` and then your ID.
  -->
  <string name="fb_login_protocol_scheme">fb000000000000</string>
</resources>

```

ภาพประกอบที่ 2.11 กำหนด Facebook App ID ใน Project

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

ขั้นตอนต่อมา copy ด้านล่างนี้ไปเพิ่มในโปรเจกของเรา ไฟล์ที่ชื่อว่า AndroidManifest.xml โดยจะอยู่ใน <application> ... </application>

```

<meta-data android:name="com.facebook.sdk.ApplicationId"
  android:value="@string/facebook_app_id"/>

<activity android:name="com.facebook.FacebookActivity"
  android:configChanges=
    "keyboard|keyboardHidden|screenLayout|screenSize|orientation"
  android:label="@string/app_name" />

<activity
  android:name="com.facebook.CustomTabActivity"
  android:exported="true">
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    <data android:scheme="@string/fb_login_protocol_scheme" />
  </intent-filter>
</activity>

```

ภาพประกอบที่ 2.12 กำหนด Facebook App ID ใน Project

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

2.2.2.5 Flutter Facebook Login

มาถึงส่วนของการ login จริงๆ ใน Flutter ให้เพิ่ม dependencies ของ flutter facebook login ใน pubspec.yaml ถ้ายังไม่มี firebase core และ firebase auth ให้เพิ่มเข้าไป

```
dependencies:
  ..|
  firebase_core: 0.3.1+1
  firebase_auth: 0.8.1+4
  flutter_facebook_login: 2.0.0 // here
```

ภาพประกอบที่ 2.13 Flutter Facebook Login

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

ที่หน้าจอของเราเพิ่มปุ่ม login with Facebook ในหน้าจอของเรา โดยจะใช้ UI จากตอนก่อนหน้านี้

```
Widget buildButtonFacebook(BuildContext context) {
  return InkWell(
    child: Container(
      constraints: BoxConstraints.expand(height: 50),
      child: Text("Login with Facebook ",
        textAlign: TextAlign.center,
        style: TextStyle(fontSize: 18, color: Colors.white)),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(16),
        color: Colors.blue[400]),
      margin: EdgeInsets.only(top: 12),
      padding: EdgeInsets.all(12)),
    onTap: () => loginWithFacebook(context));
}
```

ภาพประกอบที่ 2.14 Flutter Facebook Login

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>



ภาพประกอบที่ 2.15 loginWithFacebook

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

เขียน method ส่วนการทำงาน loginWithFacebook ซึ่งหลักการคือ พอกดที่ปุ่ม login Facebook แล้ว Facebook มันจะแสดงหน้าจอให้ user อนุญาตก่อน หลังจาก user ตกลงแล้วเราก็จะได้ access token เราก็เอา access token ไปให้กับ Firebase จากนั้น Firebase จะดึงข้อมูลจาก Facebook เช่นอีเมล มาสร้างเป็น user ใน Firebase

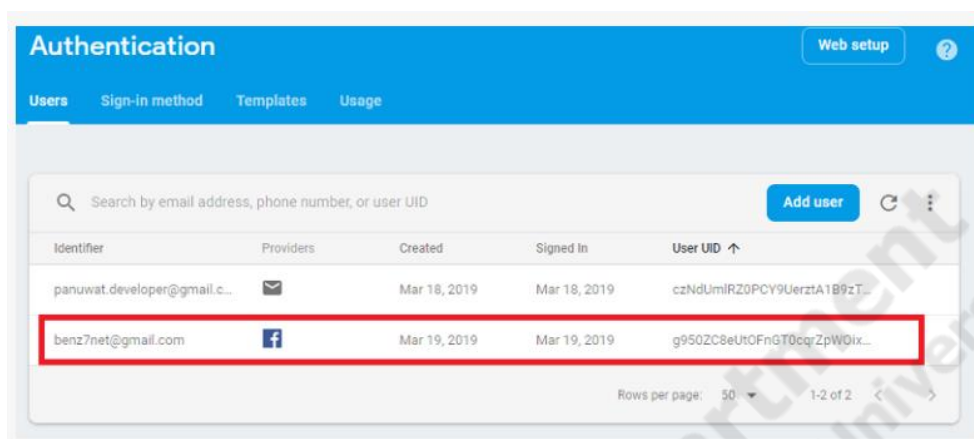
```
Future loginWithFacebook(BuildContext context) async {
  FacebookLogin facebookLogin = FacebookLogin();
  FacebookLoginResult result = await facebookLogin
    .loginWithReadPermissions(['email', "public_profile"]);

  String token = result.accessToken.token;
  print("Access token = $token");
  await _auth.signInWithCredential(
    FacebookAuthProvider.getCredential(accessToken: token));
  checkAuth(context); // after success, navigate to home.
}
```

ภาพประกอบที่ 2.16 loginWithFacebook

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

หลังจากกดตกลง ให้ดูที่เว็บ Firebase Auth ที่เมนู user ก็จะมี user อันใหม่ที่มีสัญลักษณ์ Facebook



ภาพประกอบที่ 2.17 Authentication

ที่มา : <https://benzneststudios.com/blog/flutter/facebook-login-with-firebase-auth-in-flutter/>

2.2.3 Firebase Authentication

Firebase Authentication [5] เป็น Service ตัวหนึ่งของ Firebase ที่ช่วยเรื่องของการทำ Authentication ให้เป็นเรื่องง่าย มีความปลอดภัยสูง โดยที่ตัว Firebase จะจัดการพวก Login, Register หรือ Session การลือคอินของ User ให้เรา มี UI พื้นฐานไว้ให้ใช้ รองรับการ Login ด้วย Email กับ Password แบบปกติ หรือจะเป็นการ Login เข้าสู่ระบบด้วย Provider ต่างๆ ไม่ว่าจะเป็น Gmail, Twitter, Facebook, GitHub, ผ่านมือถือ หรืออื่นๆอีกมากมาย เรียกได้ว่าครบจบในที่เดียว



ภาพประกอบที่ 2.18 Firebase Authentication

ที่มา : <https://firebase.google.com/docs/auth>

ในการลงชื่อเข้าใช้แอปของคุณ คุณต้องรับข้อมูลรับรองการตรวจสอบสิทธิ์จากผู้ใช้อก่อน ข้อมูลประจำตัวเหล่านี้อาจเป็นที่อยู่อีเมลและรหัสผ่านของผู้ใช้ หรือ Token OAuth จากผู้ให้บริการ ข้อมูลประจำตัวที่ติดต่อกับภายนอกจากนั้น ส่งข้อมูลรับรองเหล่านี้ไปยัง Firebase Authentication SDK การทำงานเบื้องหลังจะตรวจสอบข้อมูลรับรองเหล่านั้นและตอบกลับ

หลังจากลงชื่อเข้าใช้สำเร็จ เราจะเข้าถึงข้อมูลโปรไฟล์พื้นฐานของผู้ใช้ได้ และควบคุมการเข้าถึงของผู้ใช้ในการเข้าถึงข้อมูลที่จัดเก็บไว้ในผลิตภัณฑ์ Firebase อื่นๆ ได้ เรายังสามารถใช้ Token การตรวจสอบสิทธิ์ที่นำมาเพื่อยืนยันตัวตนของผู้ใช้ในการทำงานเบื้องหลังของเราเอง

หากผู้ลงชื่อเข้าใช้เป็นครั้งแรก บัญชีผู้ใช้ใหม่จะถูกสร้างขึ้นและเชื่อมโยงกับข้อมูลประจำตัว นั่นคือ ชื่อผู้ใช้และรหัสผ่าน หมายเลขโทรศัพท์ หรือข้อมูลผู้ให้บริการตรวจสอบสิทธิ์ ซึ่งผู้ลงชื่อเข้าใช้ บัญชีใหม่นี้จัดเก็บเป็นส่วนหนึ่งของโปรเจกต์ Firebase และใช้เพื่อระบุผู้ใช้ในทุกแอปในโปรเจกต์ ไม่ว่าผู้ลงชื่อเข้าใช้ จะลงชื่อเข้าใช้ด้วยวิธีใดก็ตามในแอปของคุณ คุณสามารถรับข้อมูลโปรไฟล์พื้นฐานของผู้ใช้จาก Firebase User

2.2.4 Firebase Realtime Database

จัดเก็บและซิงค์ข้อมูลกับฐานข้อมูลคลาวด์ NoSQL ของเรา [6] ข้อมูลจะซิงค์กับไคลเอนต์ ทั้งหมดแบบเรียลไทม์ และยังคงใช้งานได้เมื่อแอปของคุณออฟไลน์

ฐานข้อมูลเรียลไทม์ของ Firebase เป็นฐานข้อมูลที่โฮสต์บนคลาวด์ ข้อมูลจะถูกจัดเก็บเป็น JSON และซิงโครไนซ์แบบเรียลไทม์กับการเชื่อมต่อกับผู้ใช้ เมื่อสร้างแอปข้ามแพลตฟอร์มด้วย iOS, Android และ JavaScript SDK ผู้ใช้ทั้งหมดของคุณจะแชร์ตัวอย่างฐานข้อมูลเรียลไทม์หนึ่งตัวอย่างและรับการอัปเดตด้วยข้อมูลใหม่ล่าสุดโดยอัตโนมัติ

2.2.4.1 Installation & Setup on Android

1. Create a Database

ไปที่ส่วน Realtime Database ของคอนโซล Firebase จะได้รับแจ้งให้เลือกโปรเจกต์ Firebase ที่มีอยู่ ติดตามเวิร์กโฟลว์การสร้างฐานข้อมูล และเลือกโหมดเริ่มต้นสำหรับความปลอดภัยของ Firebase

Test mode เหมาะสำหรับการเริ่มต้นใช้งานไลบรารีบนมือถือและเว็บ แต่อนุญาตให้ทุกคนอ่านและเขียนทับข้อมูลของคุณได้ หลังจากการทดสอบ อย่าลืมอ่านส่วนทำความเข้าใจกฎฐานข้อมูลเรียลไทม์ของ Firebase

Locked mode ปฏิเสธการอ่านและเขียนทั้งหมดจากไคลเอนต์มือถือและเว็บ แอปพลิเคชันเซิร์ฟเวอร์ที่ผ่านการตรวจสอบสิทธิ์ยังคงสามารถเข้าถึงฐานข้อมูลของได้

เลือกภูมิภาคสำหรับฐานข้อมูล ขึ้นอยู่กับการเลือกภูมิภาคของ ฐานข้อมูลจะอยู่ในรูปแบบ <database Name>.firebaseio.com หรือ <database Name>. <region>.firebaseio.com สำหรับข้อมูลเพิ่มเติม โปรดดูที่เลือกสำหรับโปรเจก และเสร็จสิ้นเมื่อเปิดใช้งาน Realtime Database ก็จะเปิดใช้งาน API ในตัวจัดการ Cloud API ด้วย

2. Add the Realtime Database SDK to your app

- ใช้ Firebase Android BoM ประกาศการพึ่งพาสำหรับไลบรารี Realtime Database Android ในไฟล์ Gradle ของโมดูล (โดยปกติคือ app/build.gradle)

3. Configure Realtime Database Rules

ฐานข้อมูลเรียลไทม์มีภาษาการประกาศที่ให้กำหนดว่าข้อมูลควรมีโครงสร้างอย่างไร ควรจัดทำดัชนีอย่างไร และเมื่อใดที่ข้อมูลสามารถอ่านและเขียนได้

4. Write to your database

ดึงตัวอย่างของฐานข้อมูลของคุณโดยใช้ getInstance() และอ้างอิงตำแหน่งที่คุณต้องการเขียนถึง

```

FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");

myRef.setValue("Hello, World!");

```

ภาพประกอบที่ 2.19 Write to your database

ที่มา : <https://firebase.google.com/docs/database>

สามารถบันทึกช่วงของชนิดข้อมูลลงในฐานข้อมูลได้ด้วยวิธีนี้ ซึ่งรวม Object Java เมื่อคุณ Object การตอบกลับจาก getters ใดๆ จะถูกบันทึกเป็นรายการย่อยของตำแหน่งนี้

5. Read from your database

หากต้องการอัปเดตข้อมูลแอปแบบเรียลไทม์ ควรเพิ่ม ValueEventListener ในข้อมูลอ้างอิงที่เพิ่งสร้างขึ้นเมธอด onDataChange() ในคลาสนี้จะถูกทริกเกอร์หนึ่งครั้งเมื่อมีการแนบ Listener และอีกครั้งทุกครั้งที่ข้อมูลเปลี่ยนแปลง รวมถึงรายการย่อย

```

myRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // This method is called once with the initial value and again
        // whenever data at this location is updated.
        String value = dataSnapshot.getValue(String.class);
        Log.d(TAG, "Value is: " + value);
    }

    @Override
    public void onCancelled(DatabaseError error) {
        // Failed to read value
        Log.w(TAG, "Failed to read value.", error.toException());
    }
});

```

ภาพประกอบที่ 2.20 Read from your database

ที่มา : <https://firebase.google.com/docs/database>

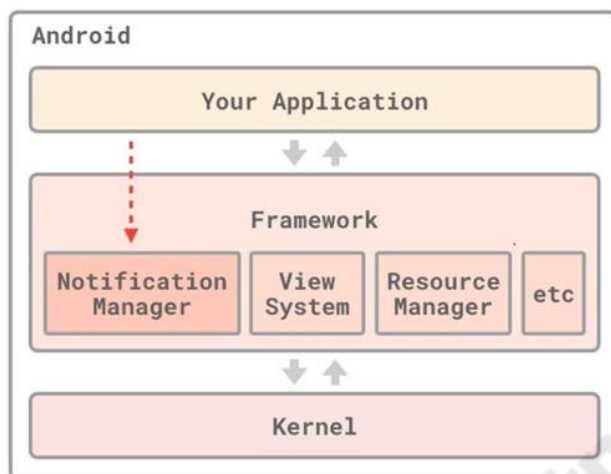
2.2.5 Notification

Notification [7] เป็นหนึ่งในช่องทางของแอนดรอยด์ที่เปิดให้แอปสามารถส่งข้อความให้ผู้ใช้เห็นได้ โดยผู้ใช้ไม่จำเป็นต้องเปิดแอปขึ้นมา และผู้ใช้ก็สามารถสั่งงานบางอย่างผ่าน Notification ตัวนั้นๆกลับมาได้อีกด้วย นี่คือการจำกัดความสั้นๆของ Notification ซึ่งเป็นหนึ่งในความสามารถของแอนดรอยด์ที่เปิดให้นักพัฒนาสามารถใช้งาน Notification ในสถานการณ์ต่างๆที่เหมาะสมกับแอปของตนเองได้

ตามธรรมชาติของ Notification บนอุปกรณ์แอนดรอยด์ที่เป็นแบบ Phone กับ Tablet จะแสดงผลอยู่ที่ด้านบนของหน้าจอ และสามารถดู Notification ที่เคยแสดงได้จาก System Bar ของแอนดรอยด์และเนื่องจาก Notification เป็นแค่ช่องทางหนึ่งในการส่งข้อความให้ผู้ใช้ จึงหมายความว่าตัวมันเองไม่ได้ยึดติดกับการทำงานของแอปเลย สิ่งที่ Notification สนใจก็มีแค่รายละเอียดของข้อมูลที่ต้องการให้แสดงใน Notification เท่านั้นเอง

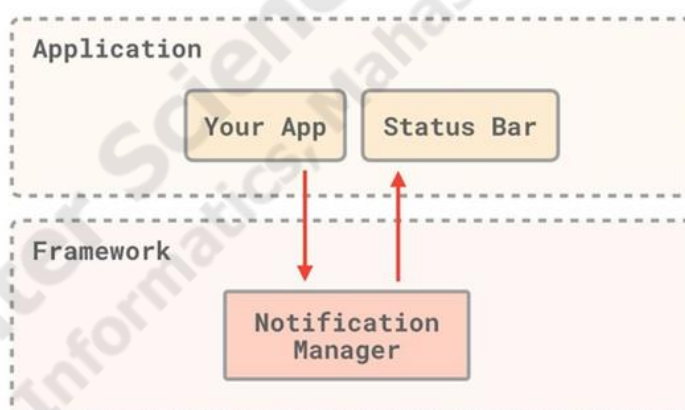
2.2.5.1 การทำงานของ Notification บนแอนดรอยด์

การทำงานจะทำงานอยู่บน Android Framework ซึ่งเป็นหัวใจหลักในการทำงานของระบบแอนดรอยด์ที่มีไว้ให้แอปต่างๆเรียกใช้งานซึ่งใน Android Framework ก็จะมีการแบ่งการทำงานแยกกันออกไปตามหน้าที่ และส่วนที่ทำหน้าที่เกี่ยวกับระบบ Notification ก็จะมีชื่อเรียกว่า Notification Manager



ภาพประกอบที่ 2.21 การทำงานของ Notification บนแอนดรอยด์

ที่มา : [Notification in Android ตอนที่ 1 — เรื่องพื้นฐานของ Notification ที่ควรรู้ \(akexorcist.dev\)](#)
และ Status Bar ของแอนดรอยด์ที่ทำหน้าที่แสดง Notification จากแอปต่างๆก็จะ
รับข้อมูลมาจาก Notification Manager ดังนั้นเมื่อใดก็ตามที่แอปในเครื่องส่ง Notification ไปให้
Notification Manager ก็จะถูกส่งไปเพื่อแสดงใน Notification Drawer ที่อยู่ใน Status Bar นั่นเอง

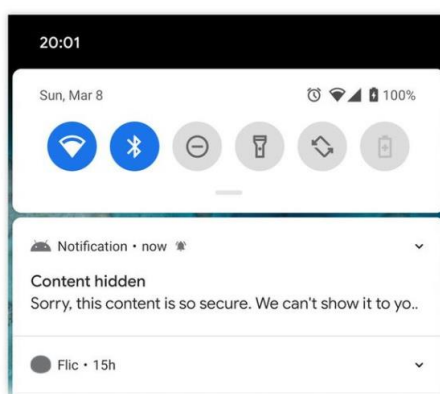


ภาพประกอบที่ 2.22 Status Bar ของแอนดรอยด์

ที่มา : [Notification in Android ตอนที่ 1 — เรื่องพื้นฐานของ Notification ที่ควรรู้ \(akexorcist.dev\)](#)
ด้วยเหตุนี้ จึงเป็นที่มาว่าทำไมทุกครั้งที่นักพัฒนาต้องการแสดงข้อความแบบ
Notification ให้กับผู้ใช้ จะต้องเรียกคำสั่งผ่าน Notification Manager อยู่เสมอ

2.2.5.2 ความแตกต่างของ Notification ในแต่ละเวอร์ชัน/ยี่ห้อ

เมื่อแอนดรอยด์มีการปล่อยอัปเดตเวอร์ชันใหม่ๆ ก็ย่อมมาพร้อมกับการเปลี่ยนแปลง
UI ซึ่งรวมไปถึง Notification Drawer ด้วยเช่นกัน ดังนั้นอย่ามองข้ามตรงจุดนี้ไป เพราะอาจจะทำให้
Notification ของแอปแสดงผลได้ไม่เหมาะสมกับเวอร์ชันนั้นๆ



ภาพประกอบที่ 2.23 ความแตกต่างของ Notification ในแต่ละเวอร์ชัน/ยี่ห้อ

ที่มา : [Notification in Android ตอนที่ 1 — เรื่องพื้นฐานของ Notification ที่ควรรู้ \(akexorcist.dev\)](#)

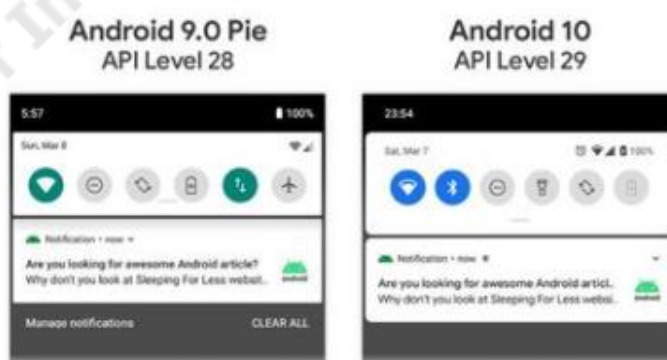
แต่ยังถือว่าโชคดีที่ Notification Drawer บนแอนดรอยด์ในเวอร์ชันหลังๆมานี้เริ่มมีความแตกต่างกันน้อยมาก รวมไปถึงฟีเจอร์ที่มีให้เรียกใช้งานด้วยเช่นกันและนอกจากเวอร์ชันแล้วอุปกรณ์แอนดรอยด์ที่มียี่ห้อต่างกันก็อาจจะทำให้ UI ของ Notification Drawer แตกต่างเช่นกัน

2.2.5.1 Notification Template

รูปแบบในการแสดงผลของ Notification นั้นจะขึ้นอยู่กับโค้ด ซึ่งแอนดรอยด์ได้เตรียมคำสั่งง่ายๆเพื่อแสดงผล Notification ในรูปแบบต่างๆ โดยเรียกว่า Template เพื่อให้ นักพัฒนาสามารถนำ Template ไปใช้งานให้ตรงกับความต้องการได้ง่ายขึ้น ซึ่งมีทั้งหมด 6 Template ด้วยกัน

2.2.5.2 การแสดง Notification บน Lock Screen

ตั้งแต่ Android 5.0 Lollipop (API 21) ขึ้นไป ผู้ใช้กำหนดได้ว่าจะให้แสดงรายละเอียดของ Notification ทั้งหมด ในขณะที่ยังไม่ได้ปลดล็อกหน้าจอหรือไม่ ถ้าต้องการให้ซ่อนเนื้อหา ผู้ใช้ก็จะไม่เห็นรายละเอียดของ Notification เมื่อเปิด Lock Screen



ภาพประกอบที่ 2.24 การแสดง Notification บน Lock Screen

ที่มา : [Notification in Android ตอนที่ 1 — เรื่องพื้นฐานของ Notification ที่ควรรู้ \(akexorcist.dev\)](#)

ซึ่งนักพัฒนาสามารถกำหนด Notification อีกตัวหนึ่งที่เรียกว่า Public Notification เพื่อแสดงให้ผู้ใช้เห็นบน Lock Screen แทนได้ จนกว่าจะปลดล๊อคหน้าจอ

2.2.5.3 ใช้งาน Notification ร่วมกับ Service เพื่อให้เป็น Foreground Service

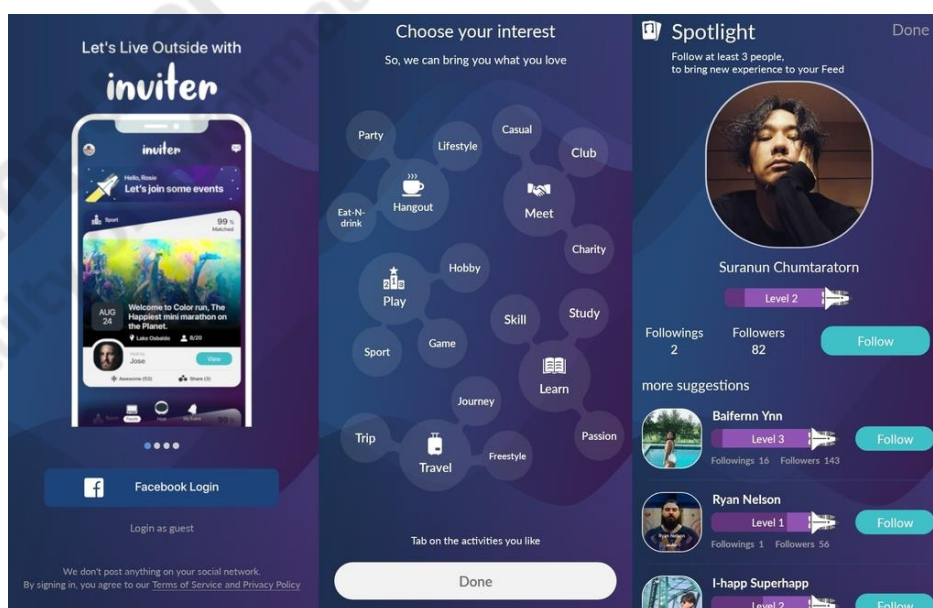
ในแอนดรอยด์เวอร์ชันใหม่ๆต้องการลดการทำงานไม่จำเป็นให้น้อยลง โดยเฉพาะการทำงานเบื้องหลังอย่าง Background Service จึงมีการปรับเปลี่ยนรูปแบบการทำงานใหม่ โดยให้ Background Service ทำงานได้แค่ระยะเวลาสั้นๆเท่านั้นแต่สำหรับแอปที่ต้องการสร้าง Background Service แต่อยากให้งานเป็นระยะเวลานานๆได้ จะต้องเปลี่ยนไปสร้างเป็น Foreground Service แทน

การทำงานของ Foreground Service จะบังคับให้นักพัฒนาต้องสร้าง Notification ตัวหนึ่งขึ้นมาเพื่อแจ้งให้ผู้ใช้ทราบว่ามีการทำงานเบื้องหลังจากแอปอยู่นั่นเอง

2.3 ระบบงานที่เกี่ยวข้อง

2.3.1 INVITER

INVITER [8] เป็นแอปพลิเคชันสำหรับการหาเพื่อนที่ชอบทำกิจกรรมอะไรเหมือนกัน มาอยู่ที่เดียวกัน หรืออาจจะนัดเจอหน้างานได้ โดยที่แอปพลิเคชันนั้นทำงานโดยสมาชิกที่ใช้งาน หากเมื่อสนใจกิจกรรมที่มีคนสร้างไว้ก็สามารถเข้าร่วมกิจกรรมนั้นได้ หรือหากอยากสร้างตัวกิจกรรมเองก็สามารถสร้างได้โดยการกรอกรายละเอียดต่างๆลงไป ข้อมูลด้วย เช่น กิจกรรมที่อยากทำ สถานที่ เวลา เพศ อายุ เพื่อให้คนที่สนใจในสิ่งเดียวได้ร่วมทำกิจกรรมด้วยกัน



ภาพประกอบที่ 2.25 แอปพลิเคชัน Inviter

ที่มา : <https://www.eventpop.me/blogs/195-inviter-application>

ข้อดี

- แอปพลิเคชันมีฟังก์ชันการสอบถามความคิดเห็นก่อนเข้าร่วม
- แอปพลิเคชันมีการกวดติดตามสมาชิกที่เราสนใจได้ ในกรณีเมื่อเขาสร้างกิจกรรมเรา

จะได้รับการแจ้งเตือน

ข้อเสีย

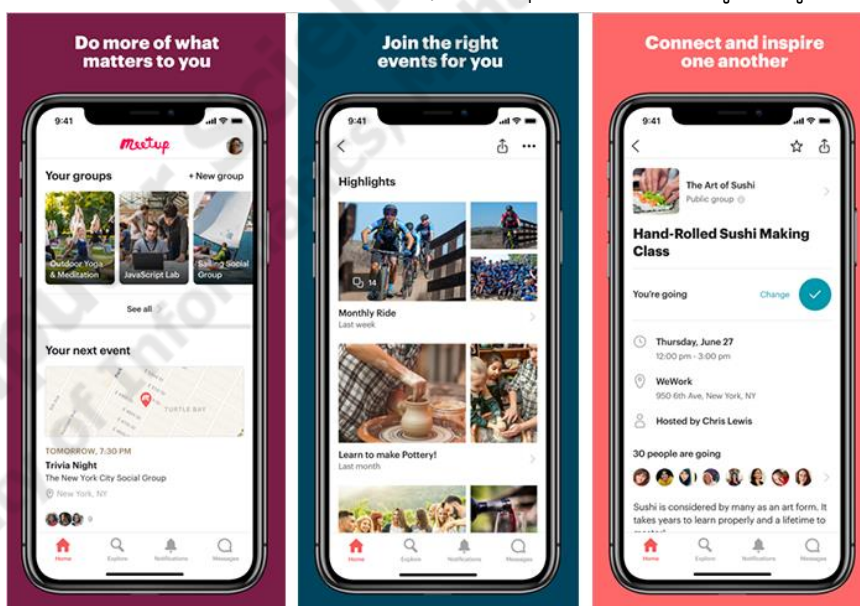
- แอปพลิเคชันไม่มีการค้นหาแบบเฉพาะเจาะจง แอปพลิเคชันมีการจัดสรรแค่เพียงสิ่งที่
ผู้ใช้สนใจเท่านั้น

- แอปพลิเคชันไม่มีการแจ้งพิกัดสถานที่ของตัวแอปพลิเคชันเพื่อให้สะดวกต่อการ

เดินทาง

2.3.2 Meetup

Meetup [9] เป็นแอปพลิเคชัน ในการเป็นสื่อกลางที่ประสานให้ผู้คนในแต่ละท้องถิ่นโลก
ใบนี้ มาพบปะและทำกิจกรรมดีๆ ร่วมกัน และในประเทศไทยก็มีกลุ่มผู้ใช้งาน Meetup อยู่จำนวนหนึ่ง
หัวข้อกิจกรรมที่น่าสนใจที่ทางระบบ Meetup จัดตั้งขึ้นมาก็มี อาทิ Running Meetup สำหรับการ
รวมกลุ่มหาเพื่อนวิ่งออกกำลังกาย Writing Meetup รวมกลุ่มกันเพื่อให้ความรู้ เพิ่มพูนทักษะงานเขียน



ภาพประกอบที่ 2.26 แอปพลิเคชัน Meetup

ที่มา : <https://software.thaiware.com/12944-Meetup-App.html>

ข้อดี

- แอปพลิเคชันมีกลุ่มกิจกรรมที่หลากหลาย
- แอปพลิเคชันมีการแชร์กิจกรรมที่ตัวสมาชิกเคยเข้าร่วม

ข้อเสีย

- แอปพลิเคชันไม่มีการสอบถามความคิดเห็นก่อนเข้าร่วมกิจกรรม
- แอปพลิเคชันไม่มีการตรวจสอบคะแนนผู้ใช้อีกก่อนการเข้าร่วมกิจกรรม
- แอปพลิเคชันไม่มีการให้คะแนนผู้ใช้

ตารางที่ 2.1 ตารางการเปรียบเทียบการทำงานของระบบ

การทำงานของระบบ	INVITER	Meetup	Playmate
ฟังก์ชันการเข้าสู่ระบบด้วย Facebook	/	/	/
ฟังก์ชันการจัดการข้อมูล	/	/	/
ฟังก์ชันการค้นหากิจกรรม		/	/
ฟังก์ชันการสร้างกิจกรรม	/	/	/
ฟังก์ชันการสอบถามก่อนเข้าร่วมกิจกรรม	/		/
ฟังก์ชันการแชร์รูปภาพกิจกรรม	/	/	/
ฟังก์ชันการแสดงพิกัดสถานที่		/	/
ฟังก์ชันการแสดงพิกัดบุคคล			/
ฟังก์ชันการสนทนา			/
ฟังก์ชันการให้คะแนนผู้เข้าร่วม			/