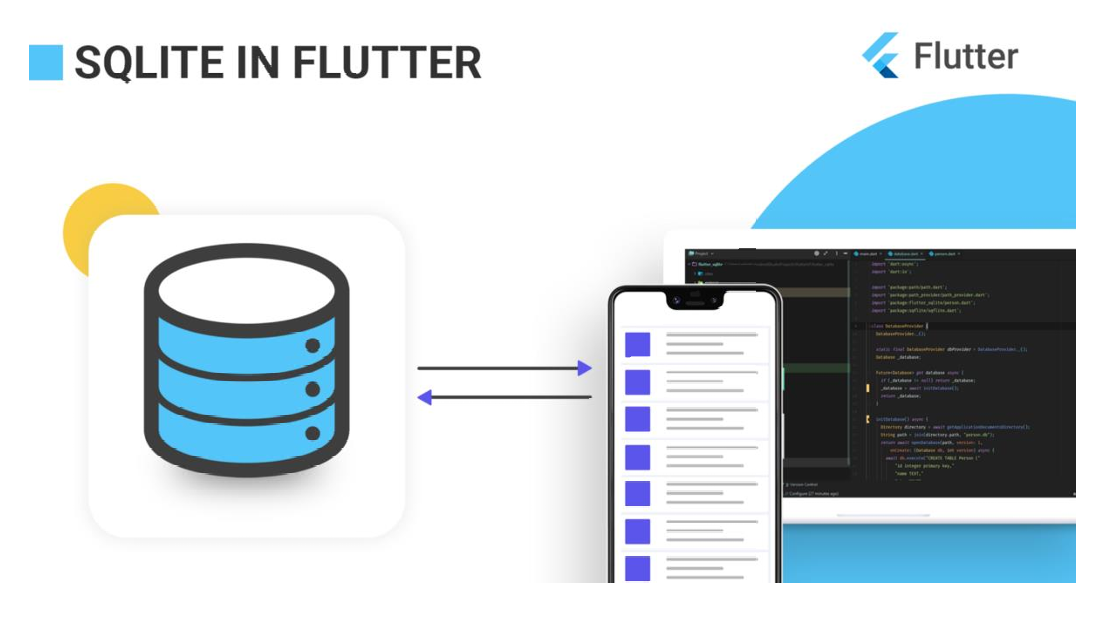
**Flutter SQLite CRUD**



**Database in Flutter**

**• ใน Flutter การจัดเก็บข้อมูลจะมี 3 รูปแบบที่นิยมใช้คือ**

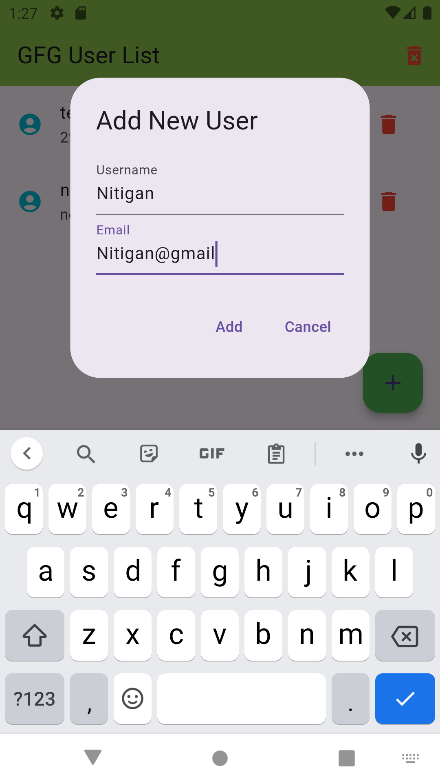
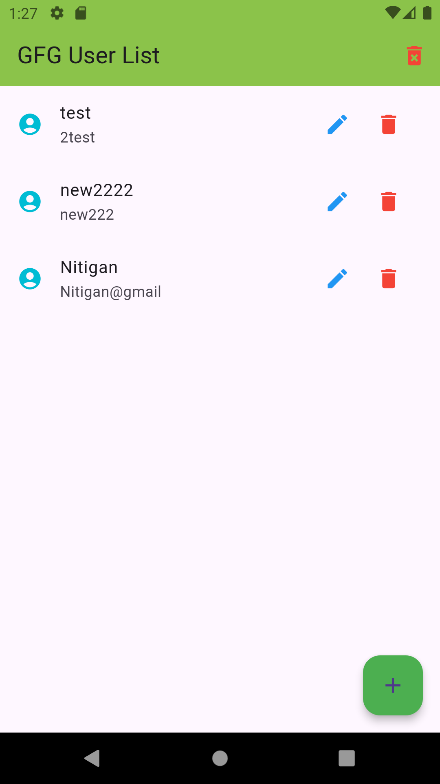
**1. Relational** เป็นฐานข้อมูลแบบเป็น Table มี Relational เป็น database มี primary key ทุก table จะสามารถกำหนด Relation เชื่อมกันได้ใน Flutter จะใช้ Sqflite เป็น ฐานข้อมูลที่พัฒนาต่อยอดจาก sqlite นั้นเอง

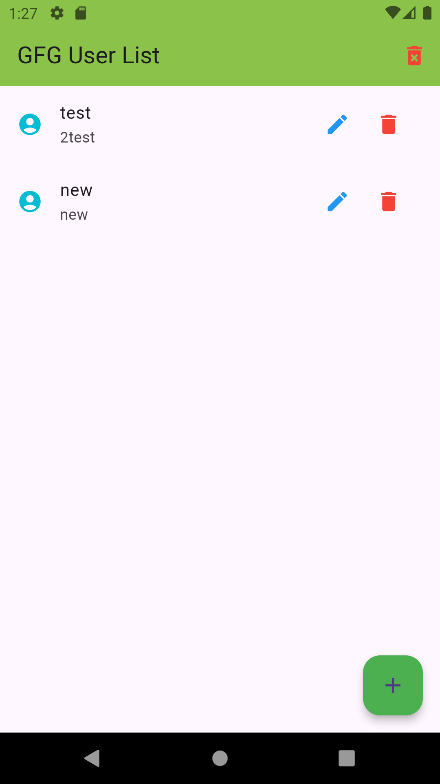
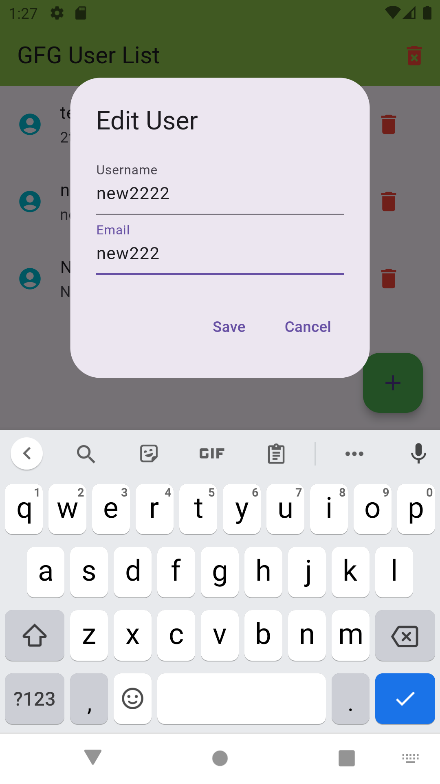
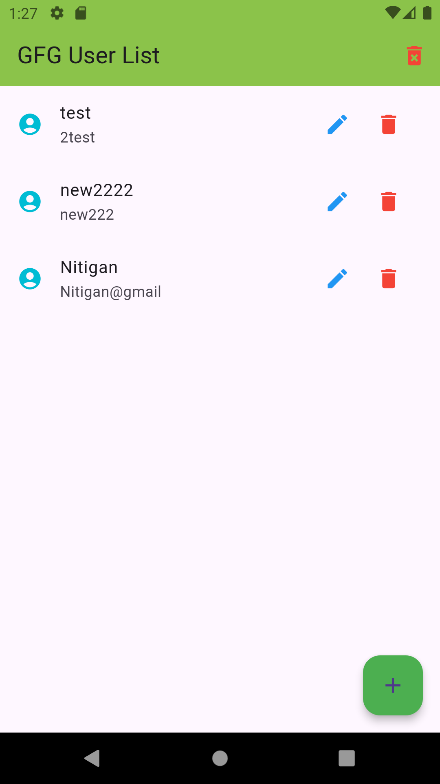
**2. NoSQL** ฐานข้อมูลที่ไม่อาศัย Relational เหมาะกับข้อมูลขนาดใหญ่ และสามารถขยายได้เรื่อย ๆ โดยไม่ต้องใช้หลักการ Table แบบ Relational ใน Flutter จะใช้ Firebase - online NoSQL storage โดยใช้ Service เชื่อมไปยังฐานข้อมูลเช่น MongoDB

**3. Individually tailored data storage** เก็บเป็นไฟล์ไว้บน Device เช่นการบันทึกไฟล์ json เก็บไว้บนเครื่องเลย

SQLite เป็นฐานข้อมูลเชิงสัมพันธ์ที่สามารถใช้เพื่อเก็บข้อมูลออฟไลน์สำหรับแอปพลิเคชันมือถือ Process   
การทำงาน แบบ CRUD หมายถึง Create, Read, Update and Delete

**สร้าง Application Flutter**

**Database Structure**

สร้าง database สำหรับเก็บข้อมูล กำหนดชื่อได้ตามความต้องการ ในที่นี้ ตั้งชื่อ databaseapp.db

สร้างตาราง ชื่อ tbUsers สำหรับเก็บข้อมูล มีโครงสร้างดังนี้

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| Id | INTEGER | id of user is Primary Key |
| username | TEXT | The username |
| email | TEXT | The email of user |

\* หากต้องการเพิ่มวันเวลาที่บันทึก ให้ทดลองสร้าง attribute เพิ่มเติม

createdAt TIMESTAMP The time that the item was created. It will be automatically added by SQLite.

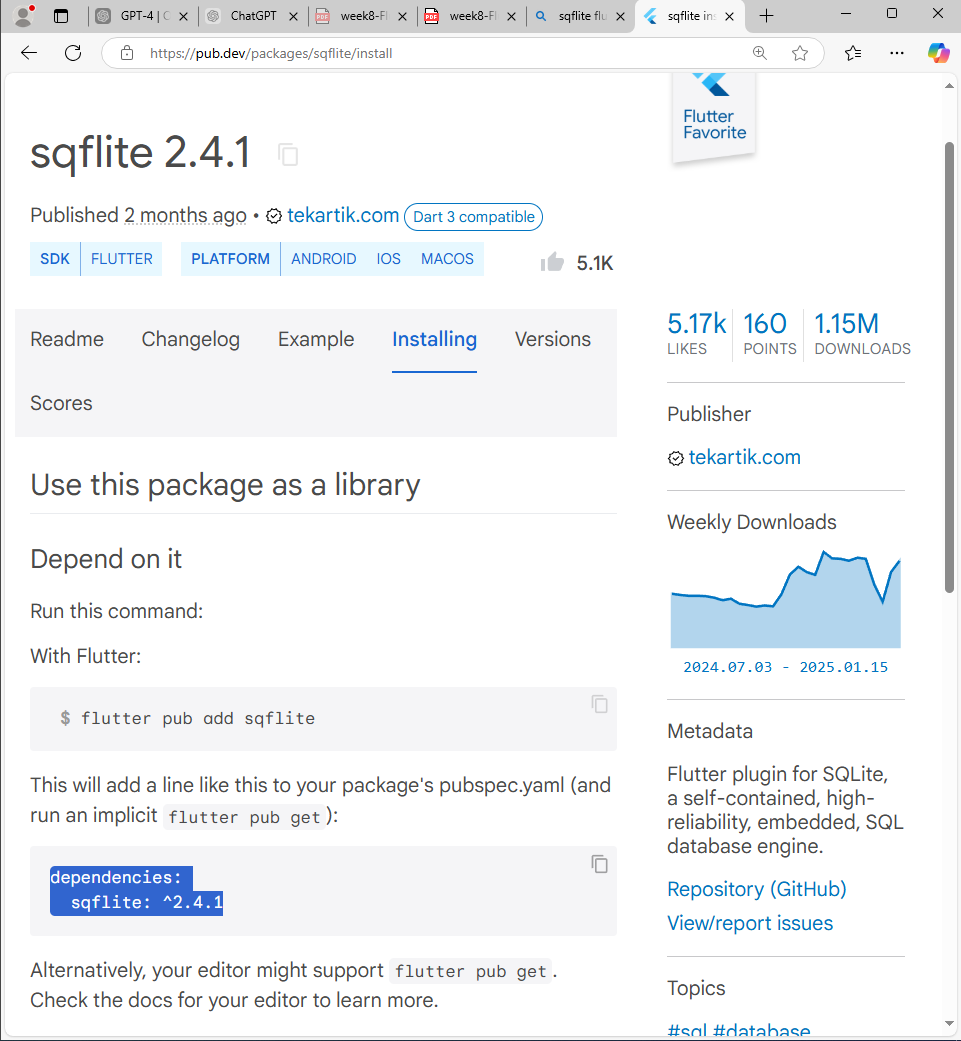
**โครงสร้างของไฟล์ อยู่ในภายใต้ folder lib**

├── main.dart

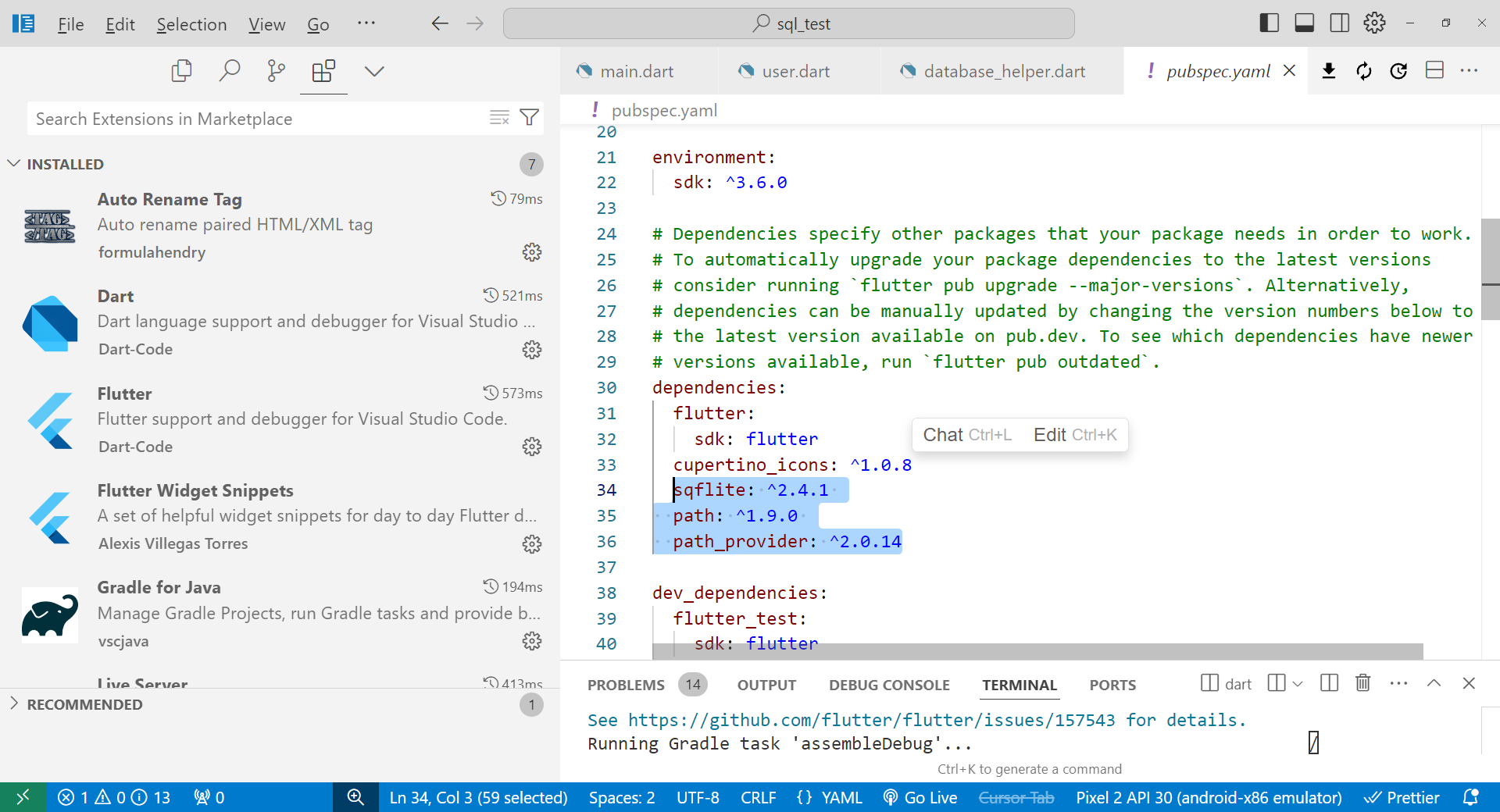
├── user.dart

├── database\_helper.dart

**Install sqflite**



**Pubspec.yaml**

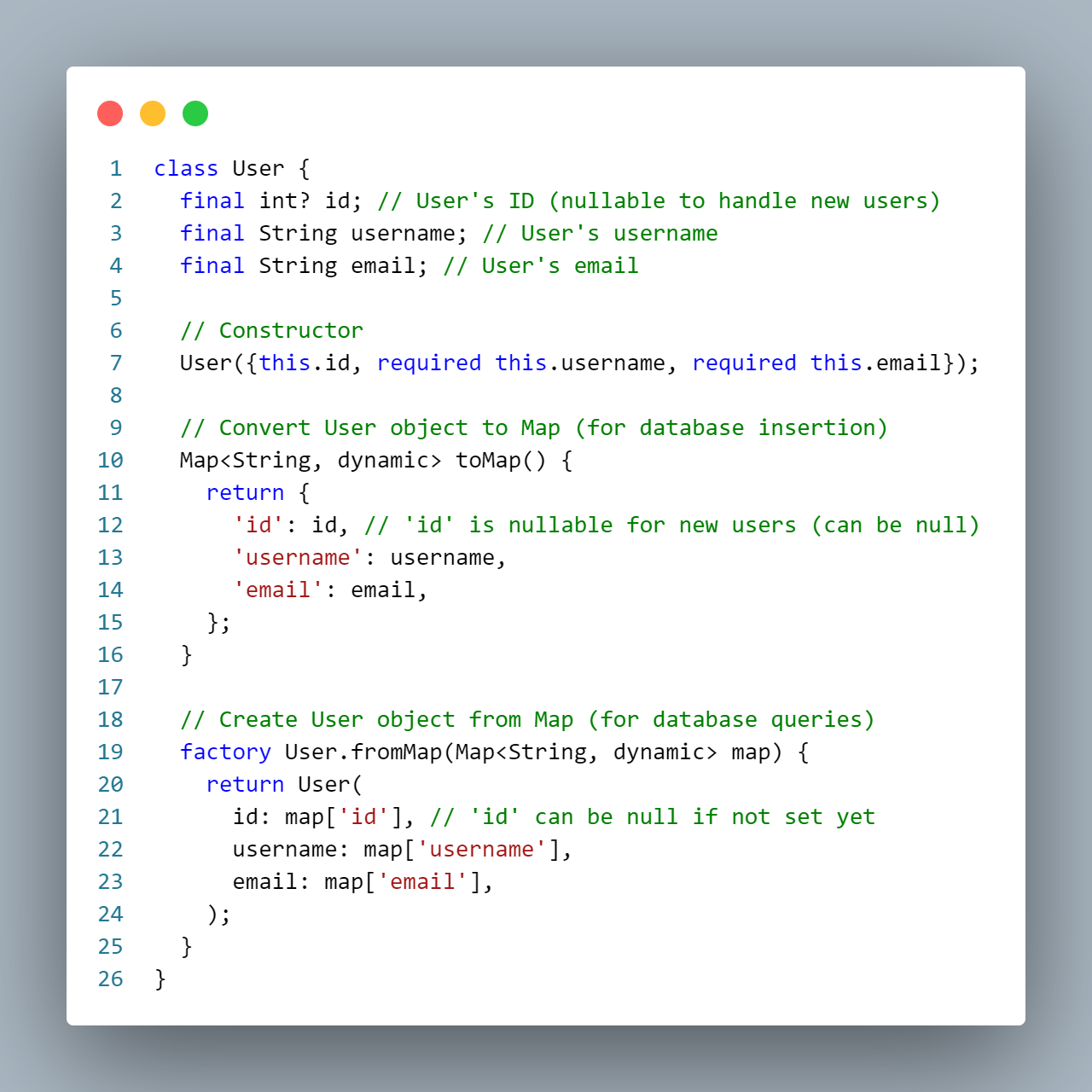


* **Sqflite** คือแพ็คเกจที่ให้การรวม SQLite สำหรับ Flutter
* **path**เป็นแพ็คเกจที่ช่วยในการค้นหาเส้นทางของไฟล์ฐานข้อมูล

***user.dart***

สร้างไฟล์ใน ' **lib/user.dart'**เพื่อกำหนดคลาสโมเดลเพื่อแสดงข้อมูลผู้ใช้ นี่คือตัวอย่างของคลาสโมเดล

ที่มี attribute id , username , email พร้อมด้วยคอนสตรัคเตอร์ที่กำหนดค่าเริ่มต้น ของข้อมูล

****

***database\_helper.dart***

สร้างคลาสเพื่อจัดการฐานข้อมูล โดยมีฟังก์ชันทั้งหมดที่ใช้งานที่นี่ คลาสฐานข้อมูลมีวิธีการดังต่อไปนี้

**initDb()** ใช้ในการเริ่มต้นฐานข้อมูล โดยจะตรวจสอบว่าตาราง 'tbUsers' มีอยู่หรือไม่   
และถ้าไม่มีตาราง ฟังก์ชันจะสร้างตารางนั้นขึ้นมา

**\_onCreate()** เป็นฟังก์ชันคอลแบ็กที่เรียกใช้งานเมื่อสร้างฐานข้อมูล วัตถุประสงค์คือเพื่อกำหนดโครงร่าง

ของตาราง 'tbUsers'

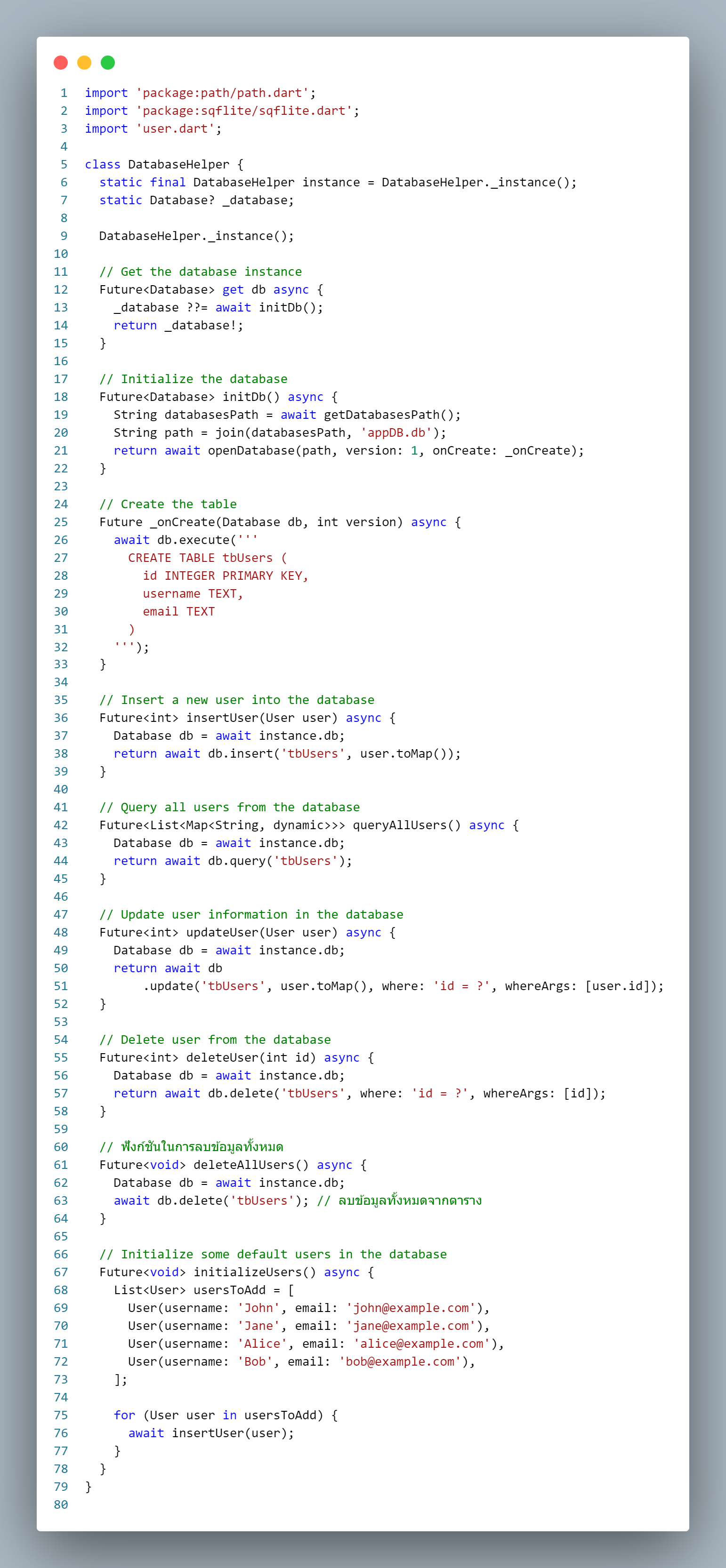
**insertUser()**  เพื่อแทรกผู้ใช้ใหม่เข้าไปในตาราง 'tbUsers'

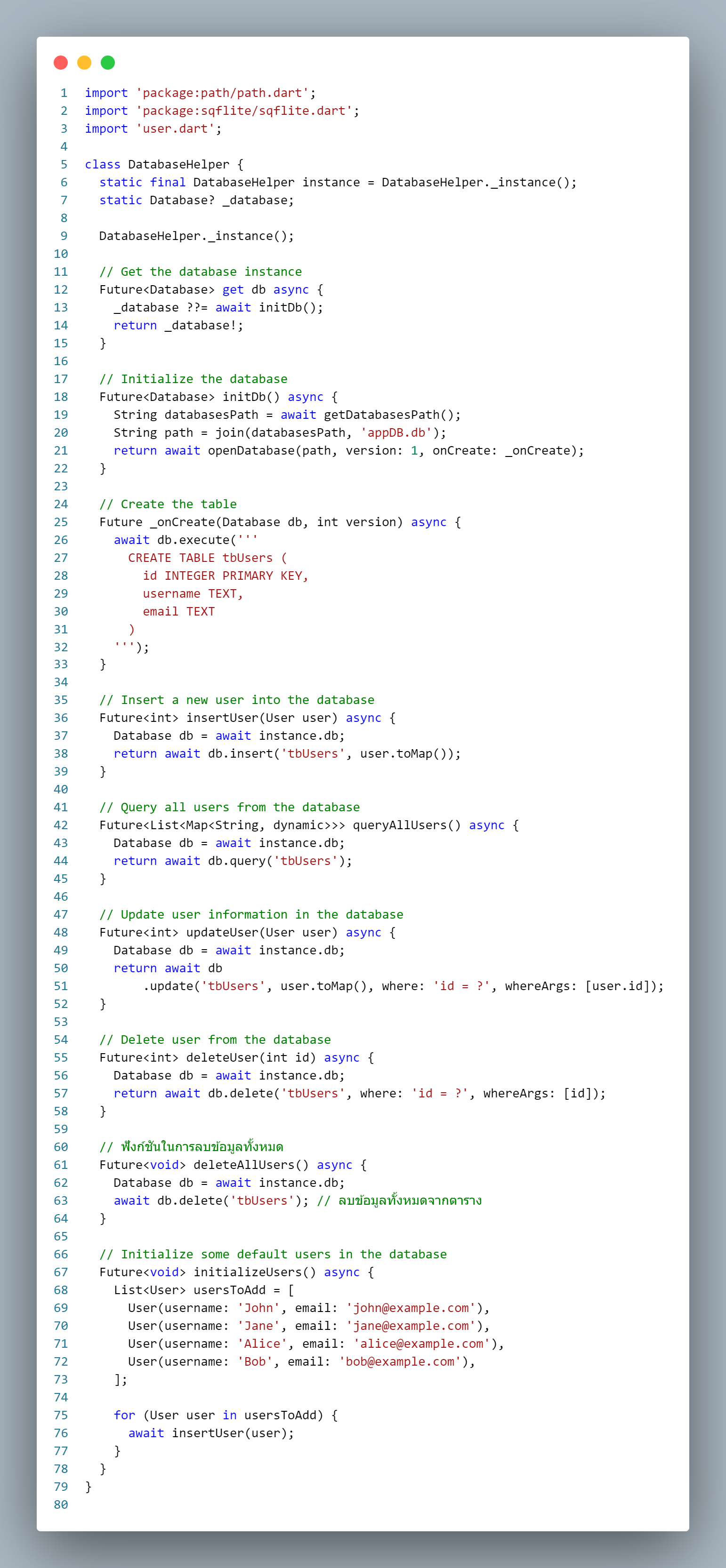
**queryAllUsers()** ดึงข้อมูลผู้ใช้ทั้งหมดจากตาราง 'tbUsers'

**updateUser()** เพื่ออัปเดตผู้ใช้ที่มีอยู่ในตาราง 'tbUsers'

**deleteUser()**  ลบผู้ใช้ออกจากตาราง 'tbUsers' โดยใช้ ID ของ User นั้น

**deleteAllUsers()**  ลบผู้ใช้ทั้งหมด ออกจากตาราง 'tbUsers' โดยใช้ ID ของ User นั้น





***main.dart***

**การเริ่มต้น** : ต้องแน่ใจว่าได้เริ่มต้นฐานข้อมูลและระบุข้อมูลโครงสร้างของ user ก่อนที่จะรันวิดเจ็ต **MyApp**

**การแสดงรายชื่อผู้ใช้**: วิดเจ็ต UserList รับผิดชอบในการแสดงรายชื่อผู้ใช้ที่ดึงมาจากฐานข้อมูลโดยใช้คลาส DatabaseHelper

**การจัดการสถานะ:** เมธอด initState เพื่อดึงข้อมูลเมื่อมีการเริ่มต้นวิดเจ็ตเป็นครั้งแรก และอัปเดตอินเทอร์เฟซเมื่อมีข้อมูล user เปลี่ยนแปลง

**การแสดงข้อมูล** : แสดง username และ email โดยใช้ วิดเจ็ต **ListTile** ภายใน **ListView.builder**





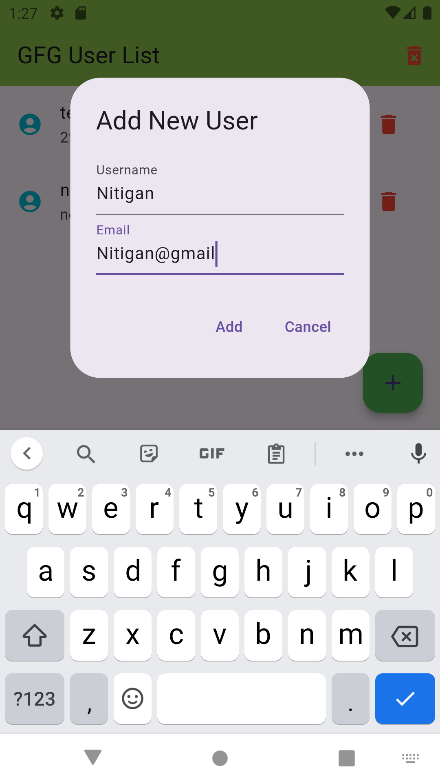
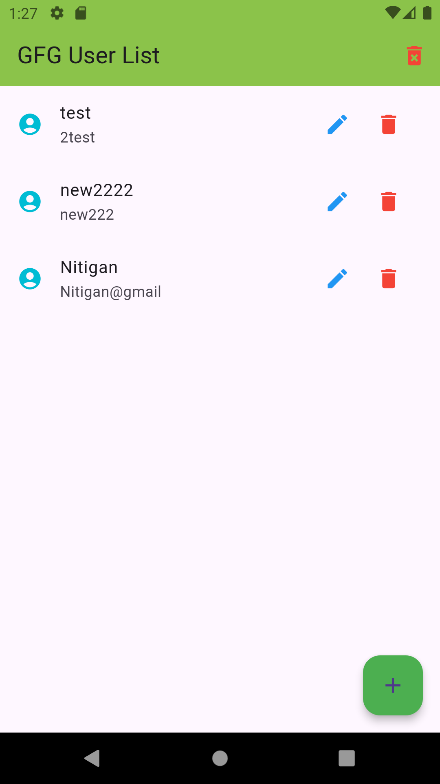




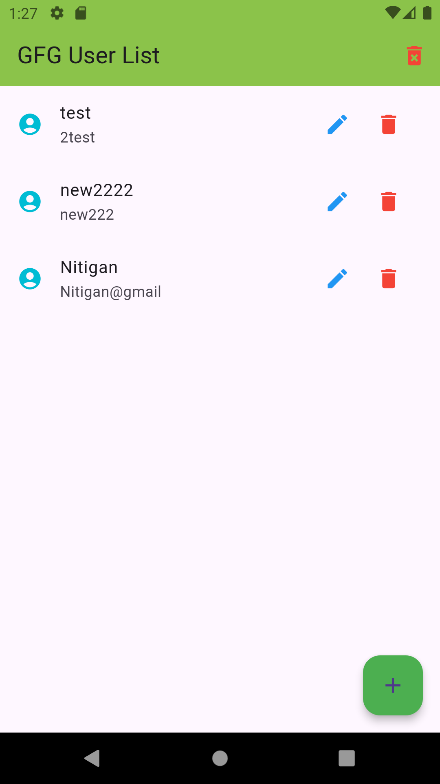
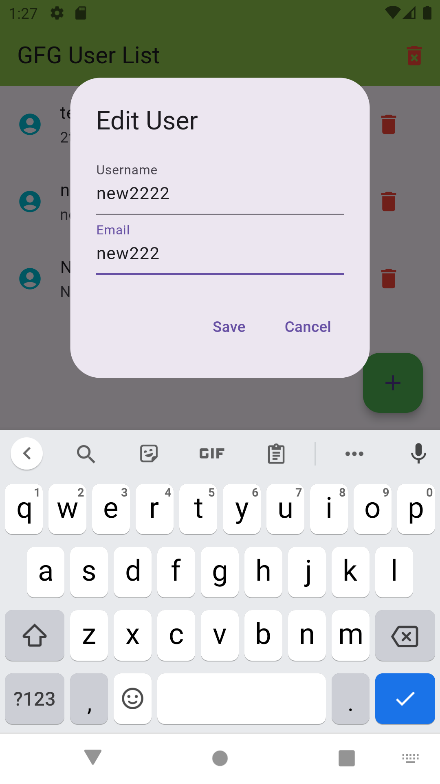
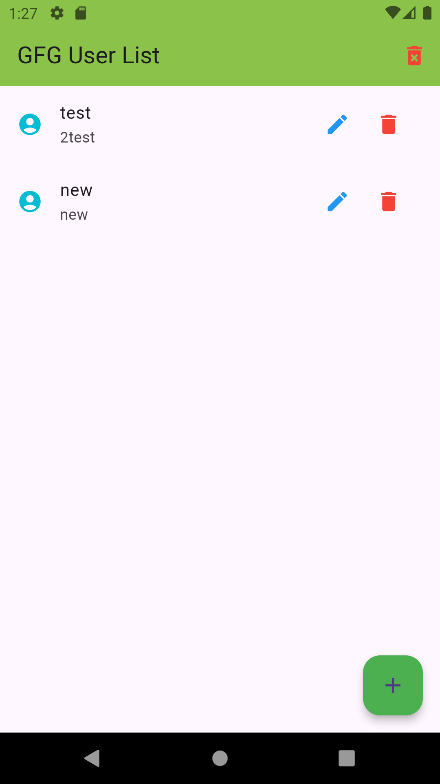


**ทดสอบการทำงาน**

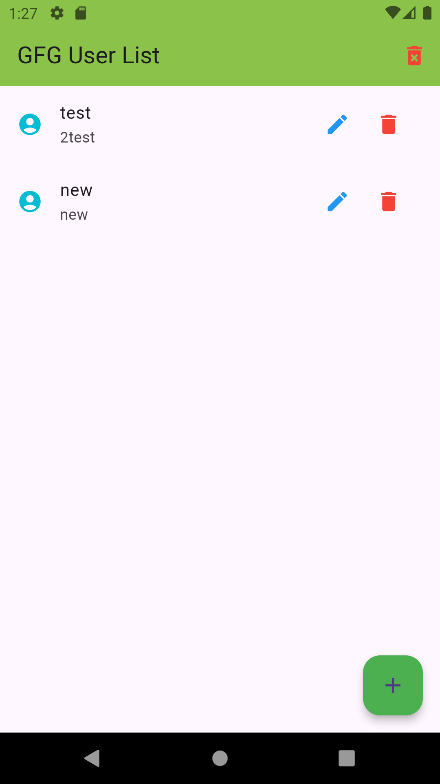
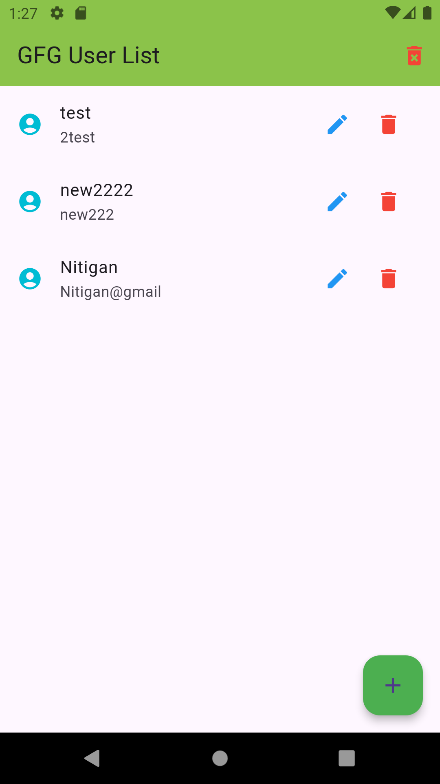
**Add New User**

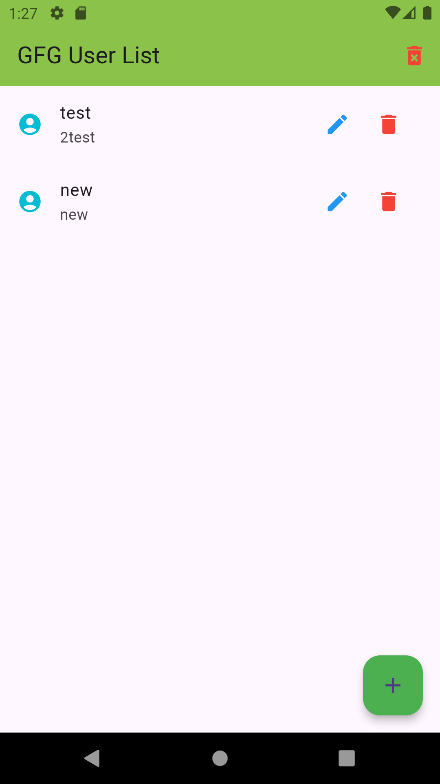
**Edit User**



**Delete user**



**Delete All**

 ****

**\*\* นศ.ทดลอง การเพิ่ม Attribute เพิ่ม คือ password และ createdAt**

**password เพื่อเก็บข้อมูล รหัสผ่าน**

**createdAt เวลาที่มีการบันทึกข้อมูล โดยใช้กำหนดเป็น** TIMESTAMP เพื่อบันทึกเวลาอัตโนมัติโดยฐานข้อมูล