

Giới thiệu SQLite



Nội dung bài học

- Giới thiệu SQLITE
- Cách cài plug in SqliteManager trong firefox/Google Chrome

Giới thiệu SQLite

- ❑ SQLite là **cơ sở dữ liệu mở** được nhúng vào Android, hỗ trợ các đặc điểm về quan hệ chuẩn của cơ sở dữ liệu như cú pháp, transaction, các câu lệnh.
- ❑ Sử dụng SQLite không yêu cầu về thiết lập bất cứ cơ sở dữ liệu hoặc đòi hỏi quyền admin.
- ❑ Hỗ trợ các kiểu dữ liệu: **TEXT, INTEGER, REAL, BLOB.**

Giới thiệu SQLite

- ❑ Đường dẫn của cơ sở dữ liệu:

DATA/data/APP_NAME/databases/FILENAME

- ❑ Cơ sở dữ liệu SQLite cung cấp riêng cho ứng dụng tạo nó.

Nếu cần cung cấp dữ liệu cho các ứng dụng khác, nên sử dụng ***Content Provider***.

SQLiteOpenHelper

- ❑ SQLiteOpenHelper là một lớp tích hợp sẵn của gói `android.database.sqlite.SQLiteDatabase`.
- ❑ Là một lớp trợ giúp để quản lý việc tạo cơ sở dữ liệu SQLite và quản lý phiên bản.
- ❑ Trong lớp này, chúng ta sẽ thực hiện hai phương thức ghi đè `onCreate ()` và `onUpgrade ()`

Phương thức onCreate ()

- ❑ onCreate() được gọi khi cơ sở dữ liệu được tạo lần đầu tiên.
- ❑ Nó chỉ được gọi một lần trong toàn bộ vòng đời ứng dụng.
- ❑ Nó sẽ được gọi bất cứ khi nào có lệnh gọi đầu tiên đến hàm `getReadableDatabase()` hoặc `getWritableDatabase ()`. Các hàm này có sẵn trong lớp siêu `SQLiteOpenHelper`.

Phương thức onCreate (), onUpgrade

```
@Override
public void onCreate(SQLiteDatabase db) {
    // Phương thức này tự động gọi nếu storage chưa có DATABASE_NAME
    // Bạn viết code để tạo DB
}
```

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    //Phương thức này tự động gọi khi đã có DB trên Storage, nhưng phiên bản khác
    //với DATABASE_VERSION
    //Bạn cần thực hiện các thao tác để nâng cấp phiên bản
}
```

Các loại truy vấn

```
//Khi cần đọc dữ liệu (SELECT)
```

```
SQLiteDatabase db = getReadableDatabase();
```

```
//Khi cần cập nhật dữ liệu (CREATE, DELETE, UPDATE, INSERT)
```

```
SQLiteDatabase db = getWritableDatabase();
```

Ví dụ 1

Một công việc gồm: Mã Công việc, Tên công việc, mô tả. Tạo 1 chương trình sử dụng SQL để tạo 1 CSDL với bảng công việc trên, thêm 5 mẫu tin và xuất 5 mẫu tin ra màn hình



Các bước thực hiện

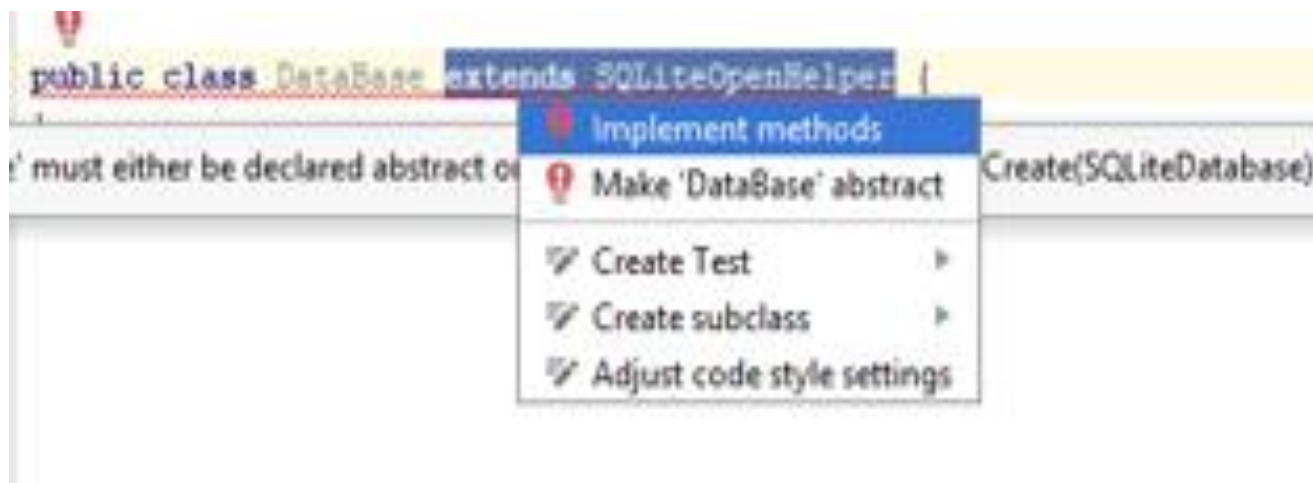
Bước 1:

Tạo một class – DataBase

Gõ **extends** SQLiteOpenHelper

Bước 2: Tạo 2 phương thức onCreate() và onUpgrade()

Đứng Extends nhấn ALT+Enter(chọn tất cả)



Các bước thực hiện

Bước 3:

Tạo Contructor cho CSDL(Chọn
ALT+Enter, chọn đầu tiên)

**Bước 4: Viết thêm 2 phương thức: QueryData(), GetDada()
trong Class DataBase**

```
public void QueryData(String sql)
{
    SQLiteDatabase database = getWritableDatabase();
    database.execSQL(sql);
}
```

Các bước thực hiện

```
|  
//TRUY VẤN CÓ KẾT QUẢ TRẢ VỀ SELECT  
public Cursor GetData(String sql)  
{  
    SQLiteDatabase database = getReadableDatabase();  
    return database.rawQuery(sql, null);  
}
```

**Bước 5: Vào MainActivity Khai báo DataBase database;
và viết lần lượt các phương thức**

```
private void onCreateCSDL() {  
    dataBase = new DataBase(context: MainActivity.this, name: "congviiec.sqlite", factory: null, version: 1);  
}
```

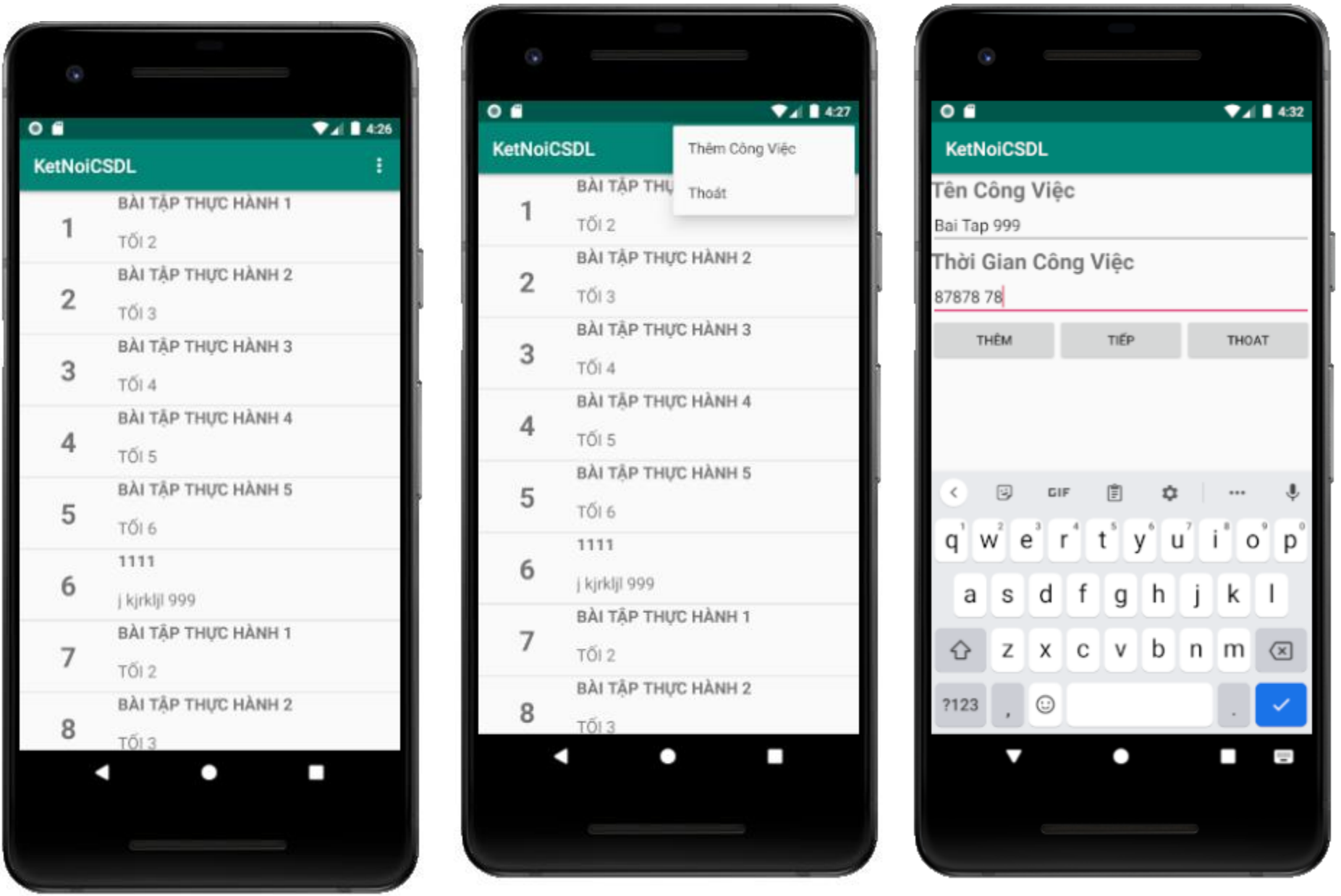
Các bước thực hiện

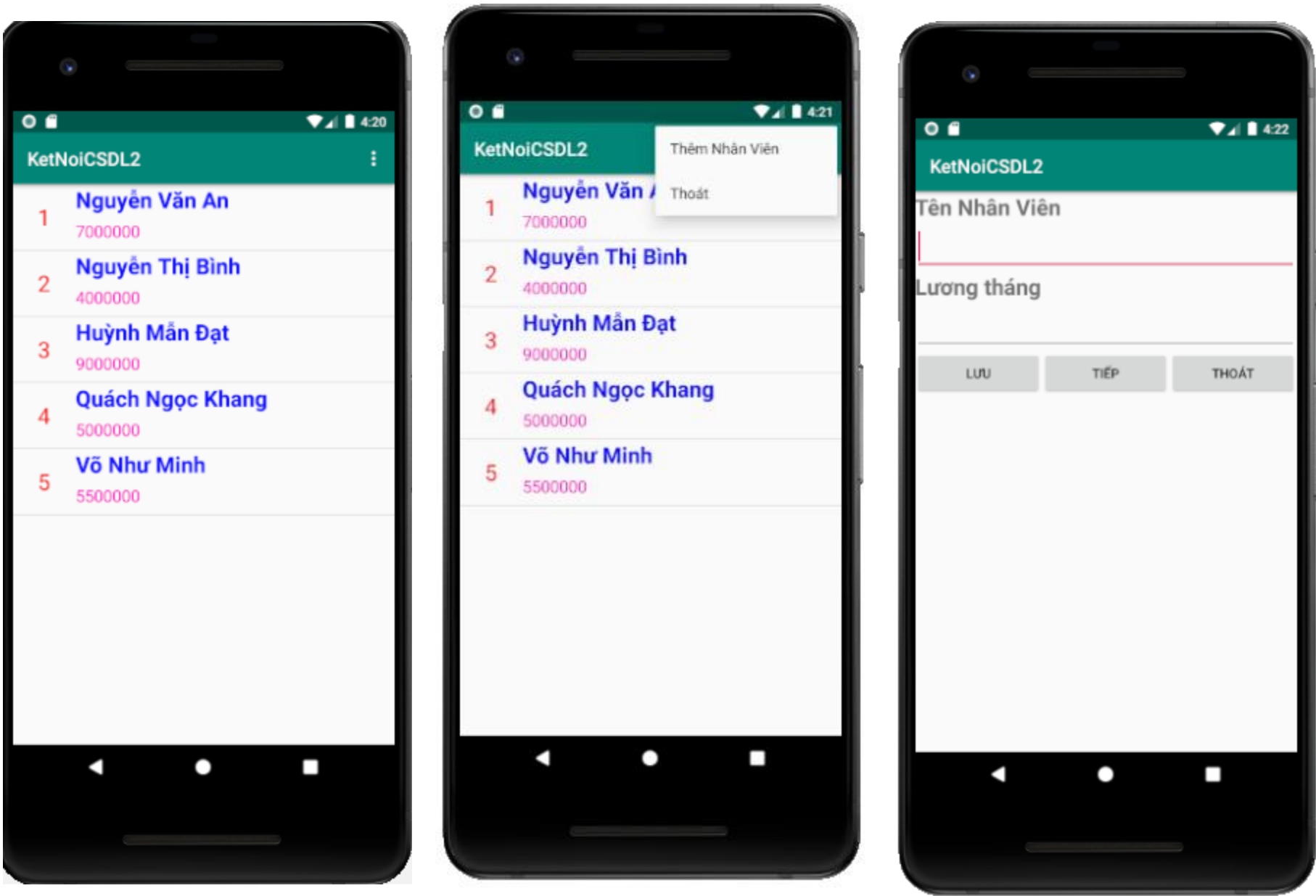
```
private void OnCreateTable() {  
    String sql = "CREATE TABLE IF NOT EXISTS CONGVIEC(MACV INTEGER PRIMARY KEY AUTOINCREMENT, " +  
        "TENCV VARCHAR(100), MOTA VARCHAR(100))";  
    dataBase.QueryData(sql);  
}
```

```
private void AddData() {  
    String sql1 = "INSERT INTO CONGVIEC VALUES (NULL, 'CONG VIEC 1', 'TOI 2')";  
    String sql2 = "INSERT INTO CONGVIEC VALUES (NULL, 'CONG VIEC 2', 'TOI 3')";  
    String sql3 = "INSERT INTO CONGVIEC VALUES (NULL, 'CONG VIEC 3', 'TOI 4')";  
    String sql4 = "INSERT INTO CONGVIEC VALUES (NULL, 'CONG VIEC 4', 'TOI 5')";  
    String sql5 = "INSERT INTO CONGVIEC VALUES (NULL, 'CONG VIEC 5', 'TOI 6')";  
    dataBase.QueryData(sql1);  
    dataBase.QueryData(sql2);  
    dataBase.QueryData(sql3);  
    dataBase.QueryData(sql4);  
    dataBase.QueryData(sql5);  
}
```

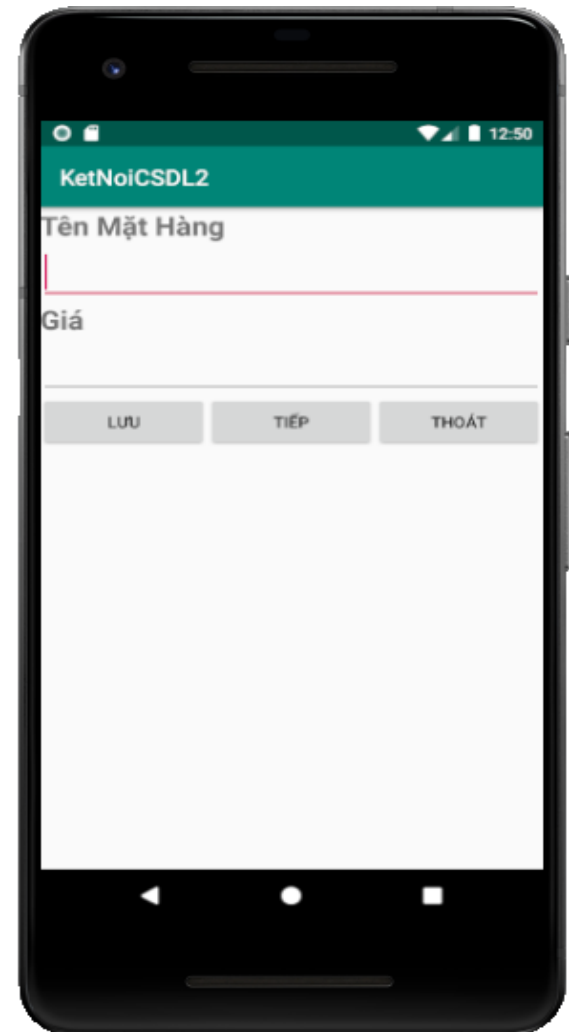
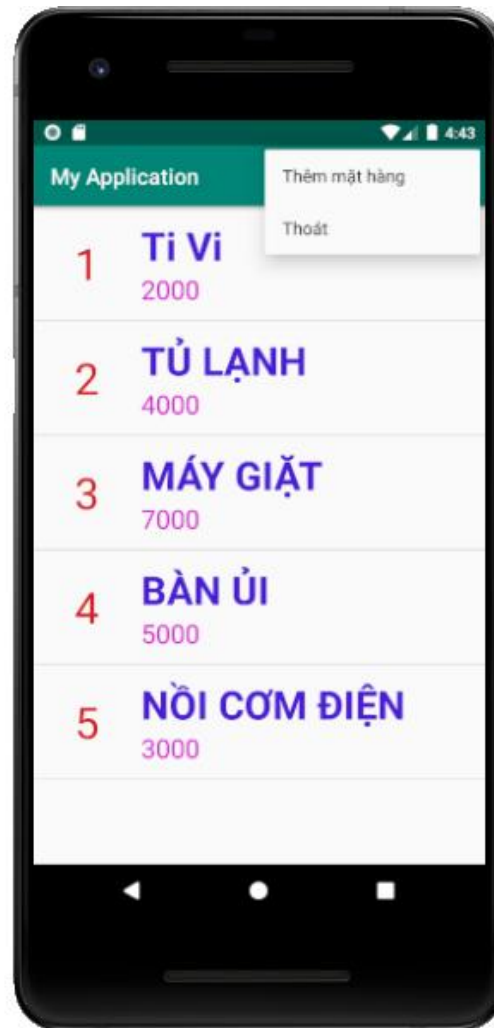
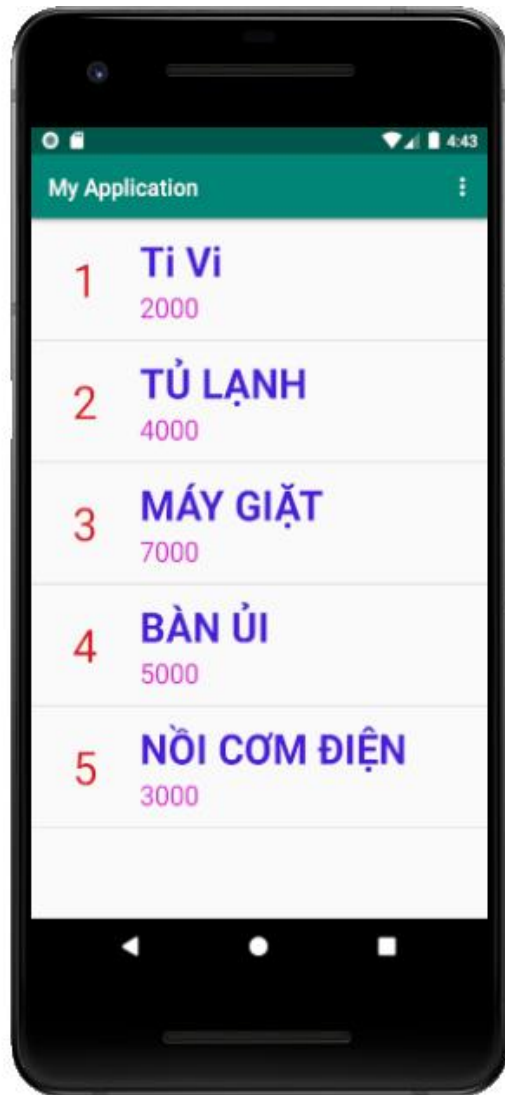
```
private void OutPutData() {  
    String sql = "SELECT * FROM CONGVIEC";  
    Cursor dataCongViec = dataBase.GetData(sql);  
    while (dataCongViec.moveToNext())  
    {  
        int MaCV = dataCongViec.getInt(columnIndex: 0);  
        String TenCV = dataCongViec.getString(columnIndex: 1);  
        String thoigian = dataCongViec.getString(columnIndex: 2);  
        Toast.makeText(context: MainActivity.this, text: TenCV+" "+thoigian, Toast.LENGTH_LONG).show();  
    }  
}  
  
private void DeleteTable() {  
    String sql = "DROP TABLE CONGVIEC";  
    dataBase.QueryData(sql);  
}
```

```
DataBase database;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    //Tạo CSDL  
    onCreateCSDL();  
    //Tạo bảng công việc  
    onCreateTable();  
    // Thêm dữ liệu  
    AddData();  
    // xuất dữ liệu lên  
    OutPutData();  
    // Xóa Table  
    DeleteTable();  
}
```





Thiết kế giao diện hiển thị dữ liệu từ 1 DataBase với thành phần như sau:



Bước 1: Tạo lớp CSDL

Bước 2: Vào MainActivity → viết tạo CSDL → Xóa

Bước 3: Tạo lớp cần quản lý: Sinh Viên, Mặt hàng

- Tạo các thuộc tính trong lớp
- Tạo các Getter và Setter()
- Tạo các Constructor (có đối số, không đối số)

Bước 4: Thiết kế hiển thị 1 dữ liệu: item(Thêm 1 layout)

Bước 5: Tạo lớp Mặt hàng Adapter

END

