

# BÁO CÁO THỰC HÀNH

Môn học: Phát triển ứng dụng trên thiết bị di động

Buổi báo cáo: Lab 03

Tên chủ đề: SQLITE

GVHD: Phan Xuân Thiện

Ngày thực hiện: 27/10/2024

## THÔNG TIN CHUNG:

Lớp: NT118.P13

STT	Họ và tên	MSSV	Email
1	Lê Hoàng Vũ	22521691	22521691@gm.uit.edu.vn

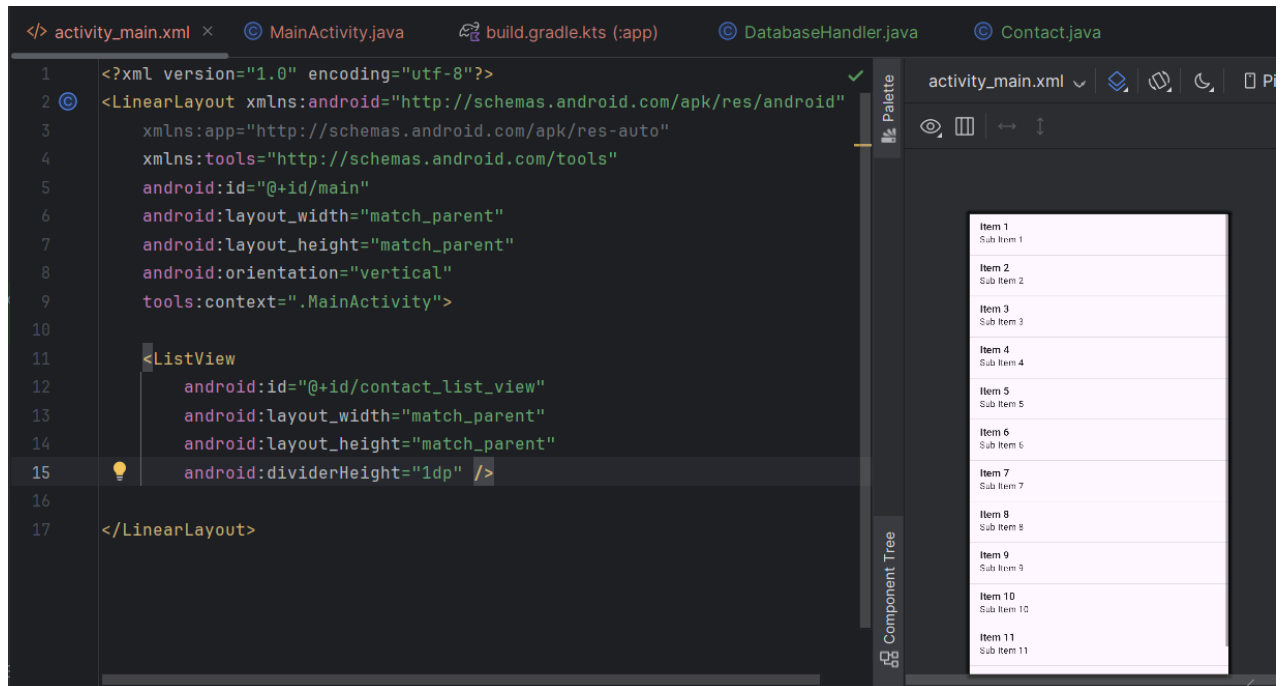
## 1. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	7 ngày
Link Video thực hiện (nếu có)	
Mã nguồn	<a href="https://github.com/kOpHAlvU/LAB_NT118">https://github.com/kOpHAlvU/LAB_NT118</a>
Điểm tự đánh giá	10/10

# BẢO CÁO CHI TIẾT

## II.

- Thiết kế một giao diện sử dụng ListView đơn giản như hình:



- Bổ sung Constructor, Getter, Setter cho class Contact để format kiểu dữ liệu:

```
1 package com.example.lab03;
2
3 public class Contact { 17 usages
4     private int id; 2 usages
5     private String name; 2 usages
6     private String phoneNumber; 2 usages
7
8     public Contact(String name, String phoneNumber) { 6 usages
9         this.name = name;
10        this.phoneNumber = phoneNumber;
11    }
12
13    public void setId(int id) { 2 usages
14        this.id = id;
15    }
16
17    public int getId() { 3 usages
18        return id;
19    }
20
21    public String getName() { 5 usages
22        return name;
23    }
24
25    public String getPhoneNumber() { 4 usages
26        return phoneNumber;
27    }
28 }
```

- Bổ sung các hàm trong class DatabaseHandler dựa trên format cho sẵn:



```
// Updating single contact
public int updateContact(Contact contact) { no usages
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_NAME, contact.getName());
    values.put(KEY_PH_NO, contact.getPhoneNumber());

    // Updating row
    return db.update(TABLE_CONTACTS, values, whereClause: KEY_ID + " = ?",
        new String[] { String.valueOf(contact.getId()) });
}

// Deleting single contact
public void deleteContact(Contact contact) { 1 usage
    SQLiteDatabase db = this.getWritableDatabase();
    db.delete(TABLE_CONTACTS, whereClause: KEY_ID + " = ?",
        new String[] { String.valueOf(contact.getId()) });
    db.close();
}
```

- Ở MainActivity bổ sung hàm sau để load các Contact và log nó ra:

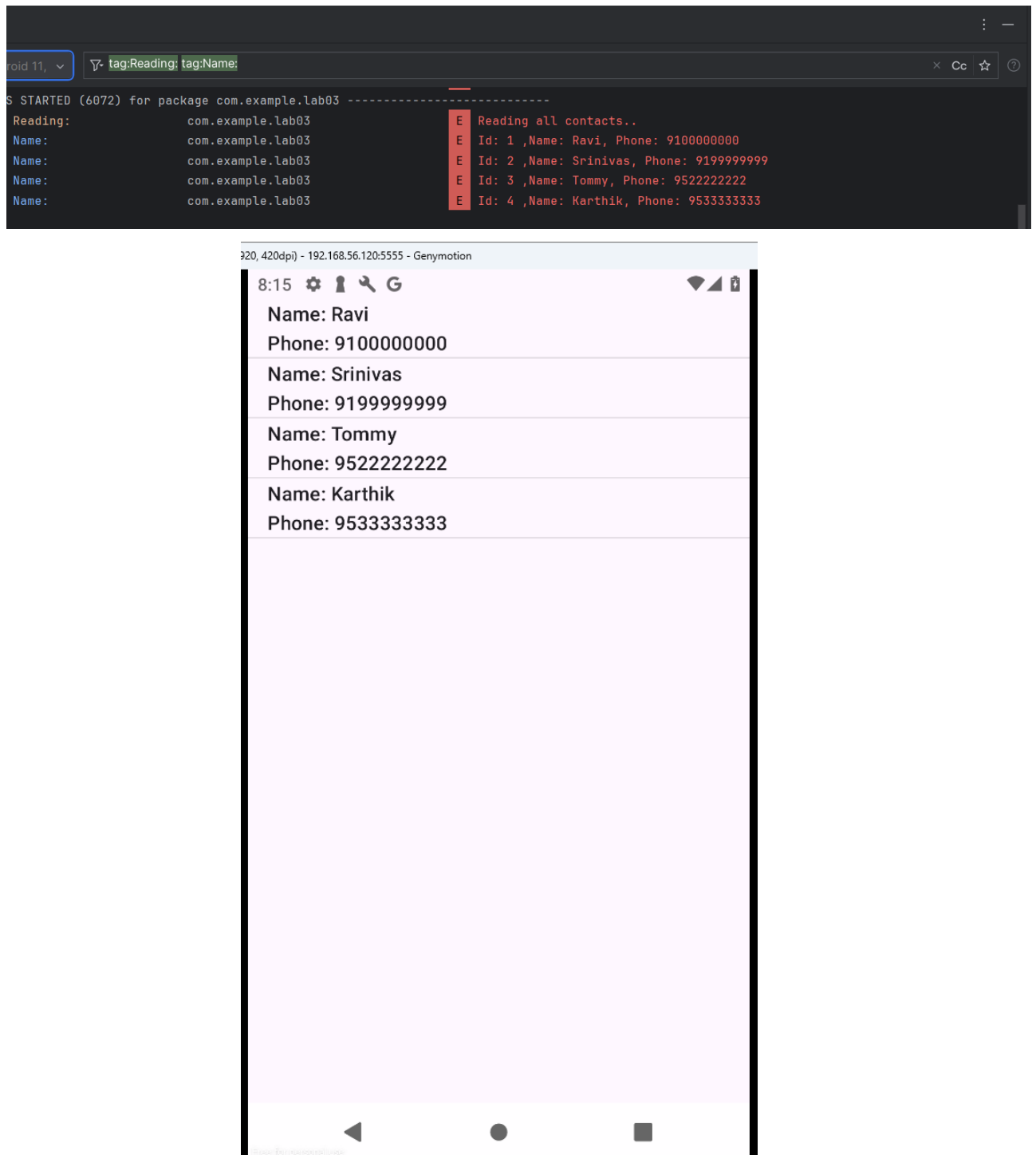
```
private void loadContacts() { 2 usages
    // Lấy tất cả các liên hệ từ cơ sở dữ liệu
    contactList = db.getAllContacts();
    contactDisplayList = new ArrayList<>();

    Log.e( tag: "Reading: ", msg: "Reading all contacts..");
    // Chuẩn bị dữ liệu để hiển thị trong ListView
    for (Contact cn : contactList) {
        String contactInfo = "Name: " + cn.getName() + "\nPhone: " + cn.getPhoneNumber();
        contactDisplayList.add(contactInfo);

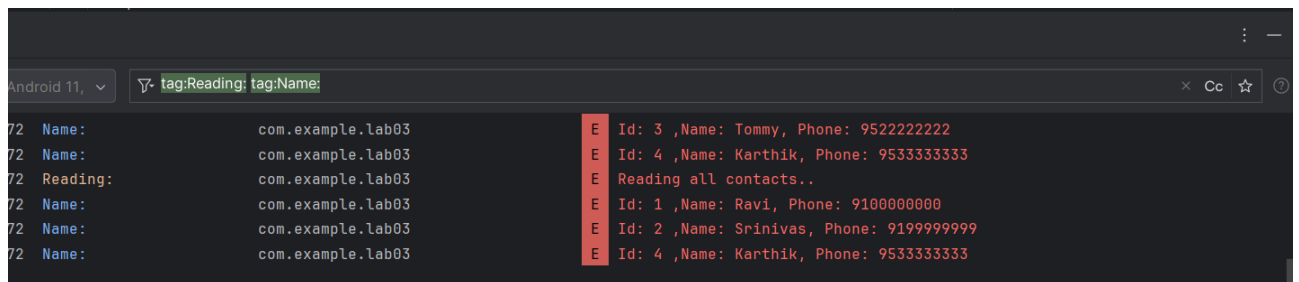
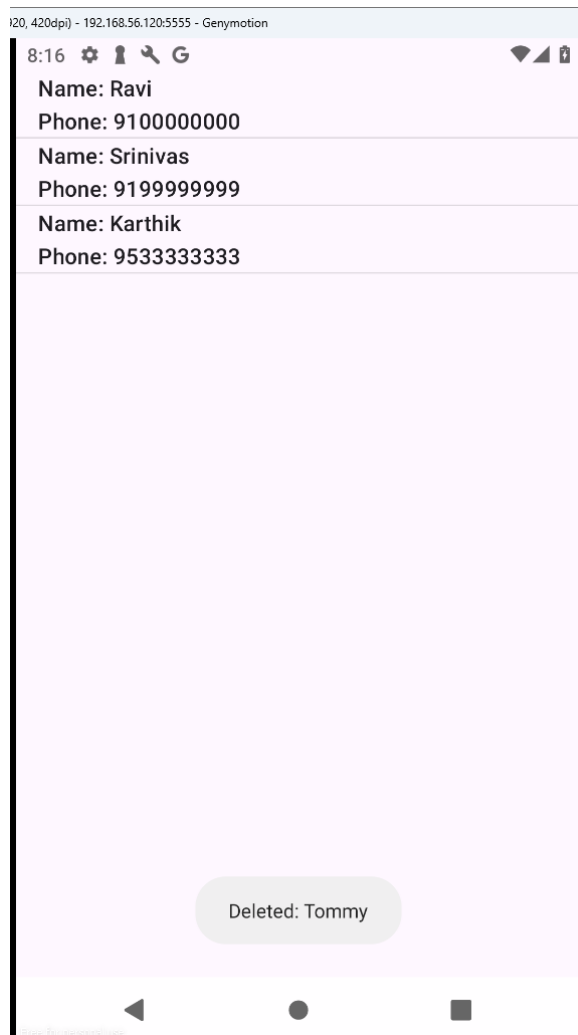
        // Ghi từng liên hệ vào log
        String log = "Id: " + cn.getId() + ", Name: " + cn.getName() + ", Phone: " + cn.getPhoneNumber();
        Log.e( tag: "Name: ", log);
    }

    // Thiết lập adapter và gán vào ListView
    adapter = new ArrayAdapter<>( context: this, android.R.layout.simple_list_item_1, contactDisplayList);
    contactListView.setAdapter(adapter);
}
```

- Từ đó ta ra được kết quả như sau:

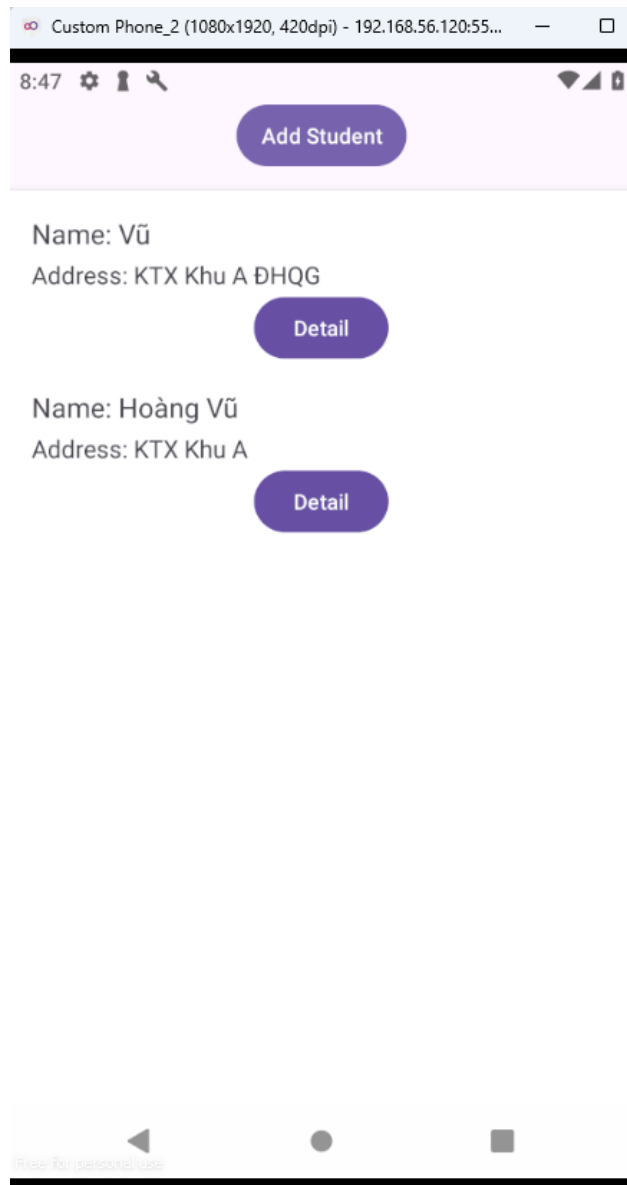


- Khi nhấn giữ ở 1 item thì nó sẽ bị xóa kèm thêm Toast thông báo đã xóa Contact nào đồng thời sẽ thực hiện lại hàm loadContacts rồi in ra danh sách các Contact mới:



### III.

- Giao diện chính (MainActivity): ở giao diện này sẽ có list Sinh viên đã được thêm trước đó kèm theo nút để thêm sinh viên:



- Thêm sinh viên (AddEditStudentActivity): dùng để nhập thông tin sinh viên mới, sau khi nhập đủ các trường thì ấn nút Save:



Custom Phone\_2 (1080x1920, 420dpi) - 192.168.56.120:55...

8:55

Name: Lê Vũ

Address: KTX ĐHQG TP.HCM

Age: 21

Save Save

- Sau khi ấn Save sẽ quay trở về MainActivity và thông tin gồm tên và địa chỉ sinh viên đó sẽ hiện thêm:

Custom Phone\_2 (1080x1920, 420dpi) - 192.168.56.120:55...

8:56

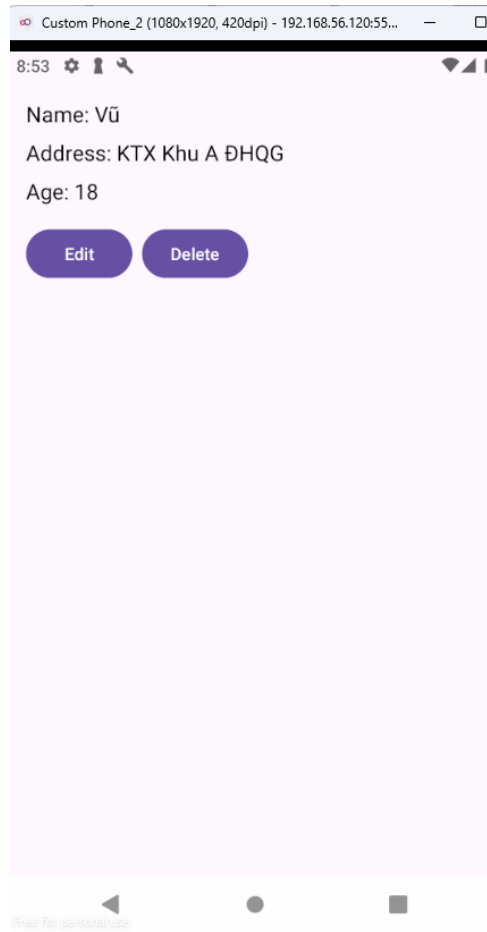
Add Student

Name: Vũ  
Address: KTX Khu A ĐHQG  
Detail

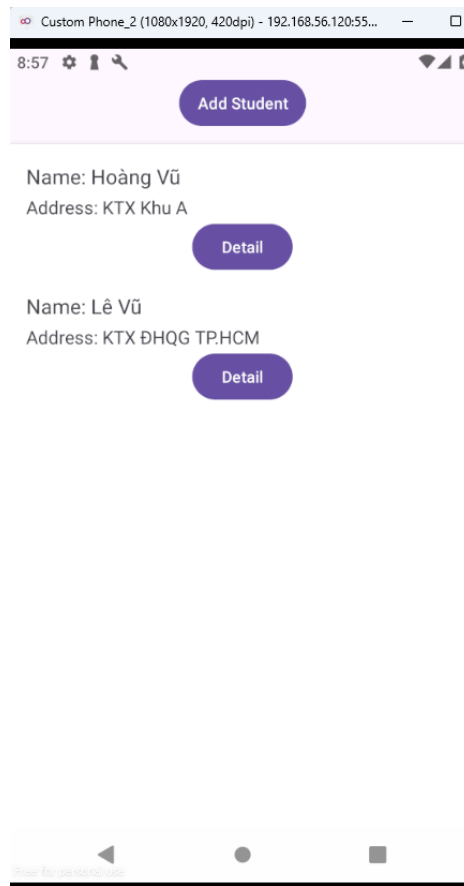
Name: Hoàng Vũ  
Address: KTX Khu A  
Detail

Name: Lê Vũ  
Address: KTX ĐHQG TP.HCM  
Detail

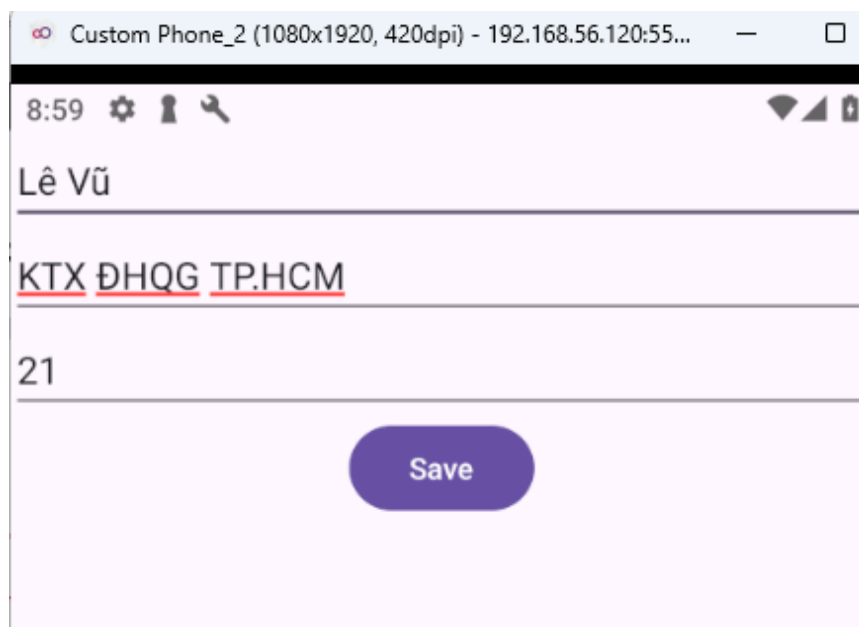
- Thông tin sinh viên (StudentDetailActivity), được mở ra khi ấn nút Detail ở mỗi sinh viên, giao diện sẽ hiển thị tên, địa chỉ, tuổi của sinh viên đó:



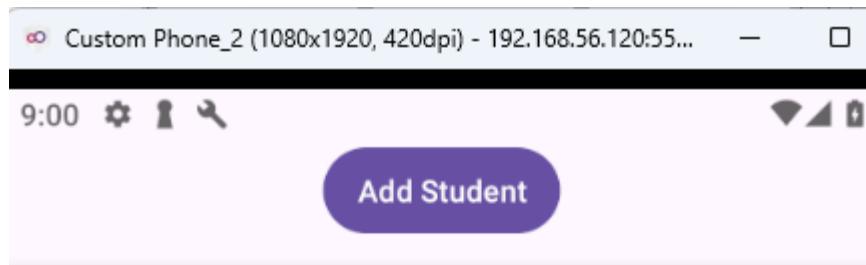
- Ấn Delete thì ngay lập tức quay trở về MainActivity và thông tin sinh viên đó sẽ biến mất:



- Còn khi ấn Edit thì sẽ tới AddEditStudentActivity, các trường dữ liệu sẽ được ghi sẵn như dưới:



- Và sau khi sửa đổi thông tin (ở đây là thêm chữ "Khu A" vào mục địa chỉ) rồi ấn Save thì sẽ trở về MainActivity với thông tin đã được cập nhật:



Name: Hoàng Vũ

Address: KTX Khu A

Detail

Name: Lê Vũ

Address: KTX Khu A ĐHQG TP.HCM

Detail

# YÊU CẦU CHUNG

## 1) Đánh giá

- Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
- Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
- Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

## 2) Báo cáo

- File .PDF hoặc .docx. Tập trung vào nội dung, giải thích.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Avo)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: LabX\_MSSV1. (trong đó X là Thứ tự buổi Thực hành).

Ví dụ: Lab01\_21520001

- Nộp file báo cáo trên theo thời gian đã thống nhất tại [courses.uit.edu.vn](https://courses.uit.edu.vn).

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

# HẾT