

BÁO CÁO THỰC HÀNH

Môn học: Lập trình ứng dụng trên thiết bị di động

Buổi báo cáo: Lab 03

Tên chủ đề: SQL Lite

GVHD: Phan Xuân Thiện

Ngày thực hiện: 25/10/2024

THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT118

STT	Họ và tên	MSSV	Email
1	Lê Gia Hoàng Thiện	22521387	22521387@gm.uit.edu.vn

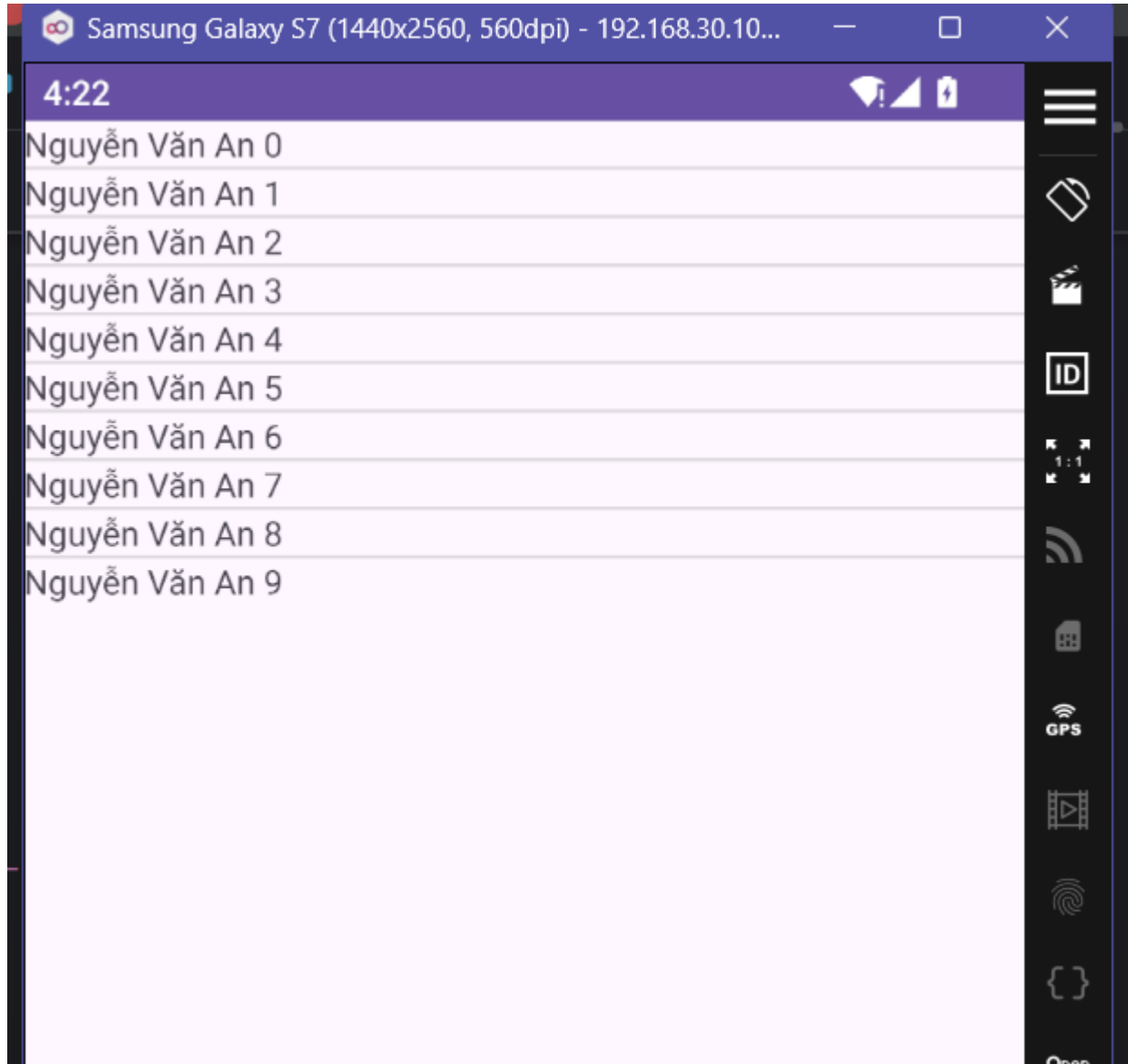
1. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	8 ngày
Link Video thực hiện (nếu có)	
Ý kiến (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá	10/10

BÁO CÁO CHI TIẾT

I. [Tham khảo] tạo 1 CSDL SQLite, thêm vào CSDL danh sách các user, sau đó lấy dữ liệu từ CSDL hiện lên listview:

- Sau khi tham khảo code và chỉnh sửa sao cho phù hợp với phiên bản em chạy ra được đáp form sau:



II. [Thực hành] Quản lý Contact

- Trước hết em tạo lớp `contactsManager` như sau:

```
package com.example.lab3_2;

public class Contact { 20 usages
    private int id; 4 usages
    private String name; 5 usages
    private String phoneNumber; 5 usages

    public Contact() { 1 usage
        this.id = -1;
        this.name = null;
        this.phoneNumber = null;
    }

    public Contact(String name, String phoneNumber) { 4 usages
        this.name = name;
        this.phoneNumber = phoneNumber;
    }

    public Contact(int id, String name, String phoneNumber) { 1 usage
        this.id = id;
        this.name = name;
        this.phoneNumber = phoneNumber;
    }

    public int getId() { 2 usages
        return id;
    }

    public void setId(int id) { 1 usage
        this.id = id;
    }
}
```

- Sau đó em Tham khảo lớp DatabaseHandler của thầy bổ sung thêm các hàm bị thiếu như thêm, lấy giá trị 1 contact, lấy toàn bộ contact, xóa 1 contact.

- Thêm và lấy giá trị theo ID

```
// Thêm một contact
public void addContact(Contact contact) { 4 usages
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, contact.getName()); // Tên liên hệ
    values.put(KEY_PH_NO, contact.getPhoneNumber()); // Số điện thoại liên hệ

    db.insert(TABLE_CONTACTS, nullColumnHack: null, values);
    db.close();
}

// Lấy một contact theo ID
public Contact getContact(int id) { no usages
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db.query(TABLE_CONTACTS, new String[]{KEY_ID, KEY_NAME, KEY_PH_NO},
        selection: KEY_ID + "=?", new String[]{String.valueOf(id)},
        groupBy: null, having: null, orderBy: null, limit: null);

    if (cursor != null) {
        cursor.moveToFirst();
    }

    Contact contact = new Contact(Integer.parseInt(cursor.getString(i: 0)),
        cursor.getString(i: 1), cursor.getString(i: 2));
    cursor.close(); // Đóng con trỏ
    return contact;
}
```

- Lấy tất cả các record trong database

```
// Lấy tất cả contact
public List<Contact> getAllContacts() { 1 usage
    List<Contact> contactList = new ArrayList<>();
    // Lệnh truy vấn để lấy tất cả các liên hệ
    String selectQuery = "SELECT * FROM " + TABLE_CONTACTS;

    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, selectionArgs: null);

    // Duyệt qua tất cả các dòng và thêm vào danh sách
    if (cursor.moveToFirst()) {
        do {
            Contact contact = new Contact();
            contact.setId(Integer.parseInt(cursor.getString(i: 0)));
            contact.setName(cursor.getString(i: 1));
            contact.setPhoneNumber(cursor.getString(i: 2));

            contactList.add(contact);
        } while (cursor.moveToNext());
    }
    cursor.close(); // Đóng con trỏ
    return contactList;
}
```

- Cập nhật và xóa contact

```
// Cập nhật một contact
public int updateContact(Contact contact) { no usages
    SQLiteDatabase db = this.getWritableDatabase();

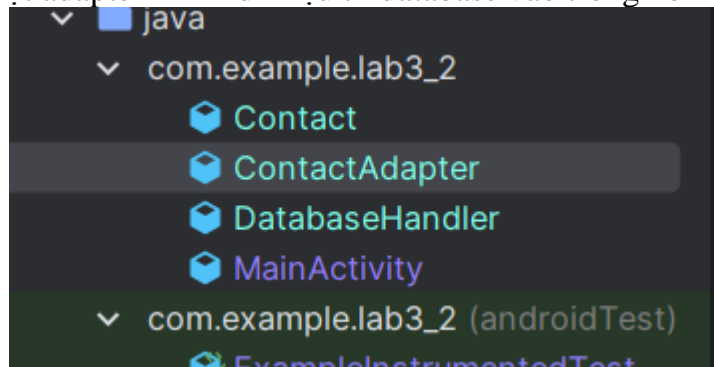
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, contact.getName());
    values.put(KEY_PH_NO, contact.getPhoneNumber());

    // Cập nhật bản ghi dựa trên ID
    return db.update(TABLE_CONTACTS, values, whereClause: KEY_ID + " = ?",
        new String[]{String.valueOf(contact.getId())});
}

// Xóa một 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();
}

// Xóa tất cả contact khi khởi tạo lại app
public boolean deleteAllContacts() {
    SQLiteDatabase db = this.getWritableDatabase();
    int rowsDeleted = db.delete(TABLE_CONTACTS, whereClause: null, whereArgs: null);
    db.close();
    return rowsDeleted > 0;
}
```

Sau đó em tạo một adapter để đổ dữ liệu từ database vào trong đó



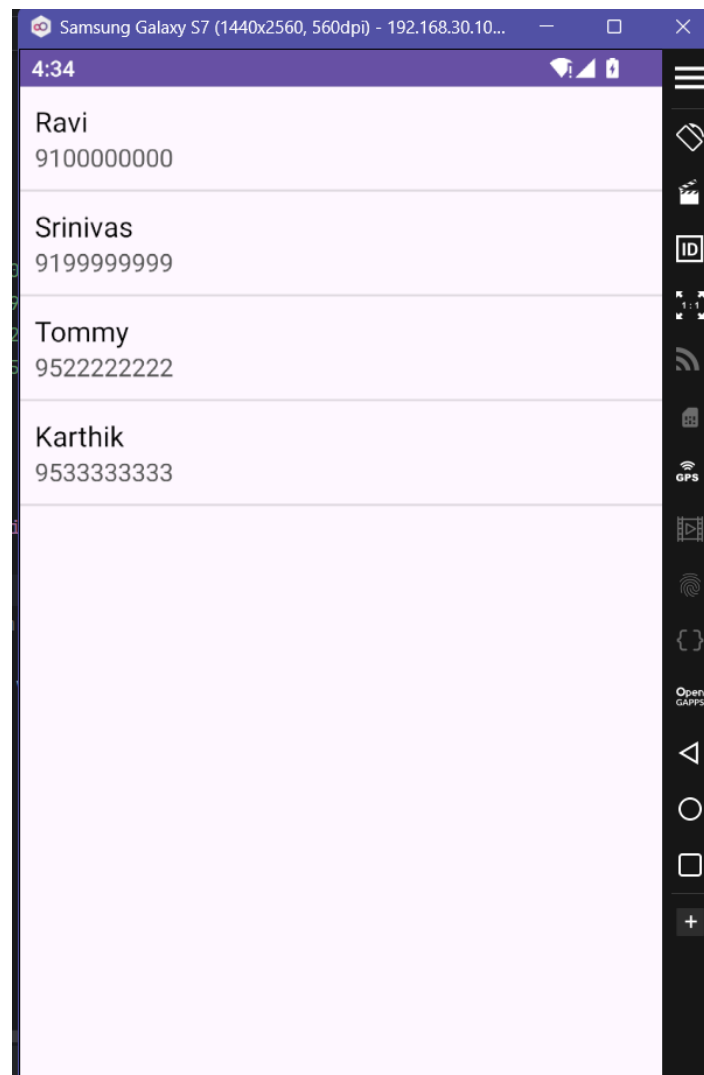
- Sau đó em xử lý hàm main sao cho đổ được dữ liệu vào listview và thực hiện nhấn giữ để xóa 1 contact ra khỏi database

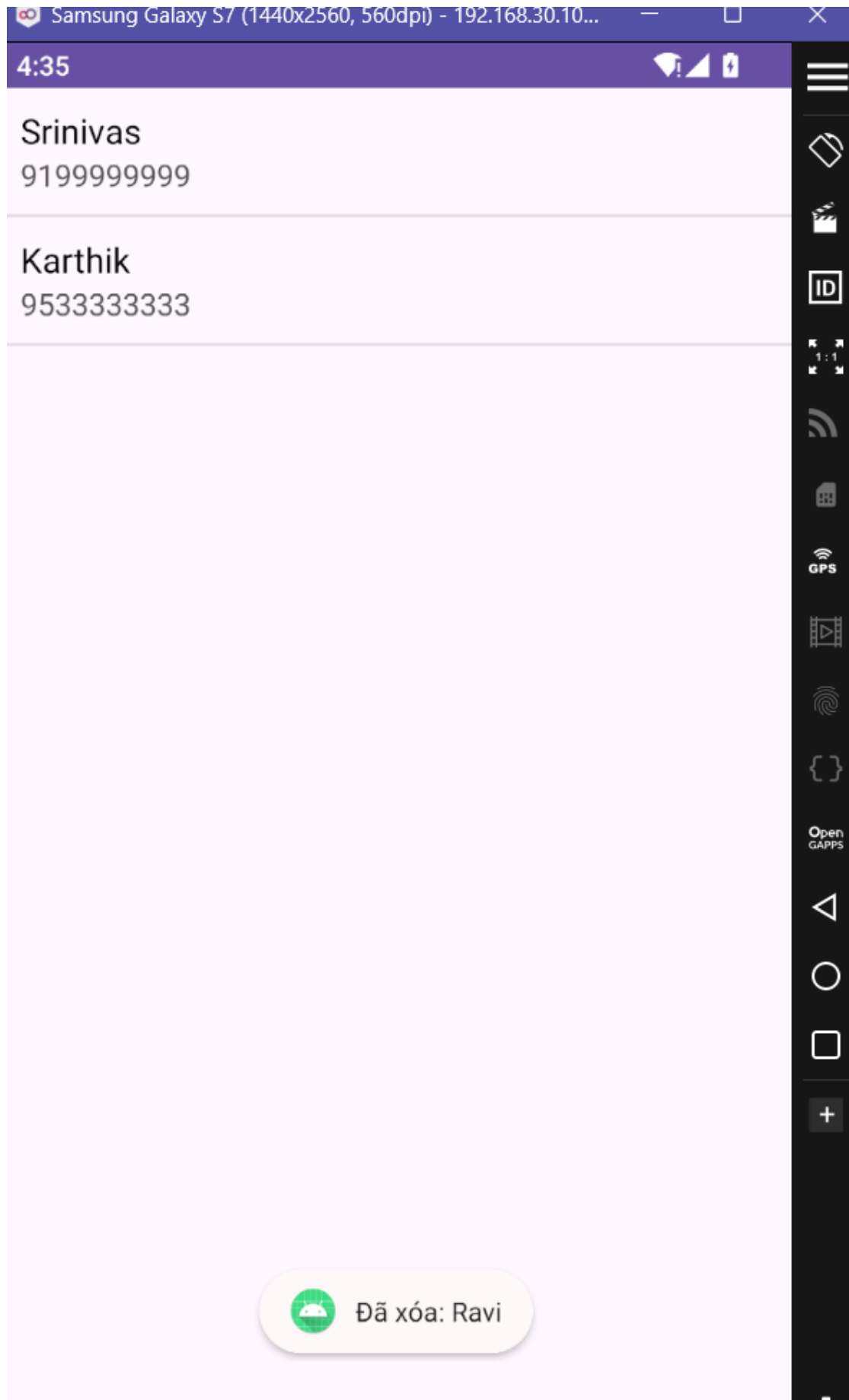
```
db = new DatabaseHandler(context: this);
db.deleteAllContacts();
db.addContact(new Contact(name: "Ravi", phoneNumber: "9100000000"));
db.addContact(new Contact(name: "Srinivas", phoneNumber: "9199999999"));
db.addContact(new Contact(name: "Tommy", phoneNumber: "9522222222"));
db.addContact(new Contact(name: "Karthik", phoneNumber: "9533333333"));

contactList = db.getAllContacts();
ListView = findViewById(R.id.listViewContacts);
contactAdapter = new ContactAdapter(context: this, contactList);
ListView.setAdapter(contactAdapter);

ListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override no usages
    public boolean onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Contact contact = contactList.get(position);
        db.deleteContact(contact);
        contactList.remove(position);
        contactAdapter.notifyDataSetChanged();
        Toast.makeText(context: MainActivity.this, text: "Đã xóa: " + contact.getName(), Toast.LENGTH_SHORT).show();
        return true;
    }
});
```

Kết quả:





III. Bài thực hành:

- Đầu tiên đây là giao diện chính cho phép em nhập thông tin học viên

The screenshot shows a mobile application interface for managing students. At the top, there is a status bar with the time 4:40 and various icons. Below the status bar is a green header with the text "Quản lý học viên". The main form area has a light purple background and contains four input fields with labels: "MSSV:" with the placeholder "Mã Số Sinh Viên", "Tên SV:" with the placeholder "Tên Sinh Viên", "Địa chỉ:" with the placeholder "Địa chỉ thường trú", and "SĐT:" with the placeholder "Số Điện thoại". Below these fields is a purple button labeled "Nhập". At the bottom of the form is a light blue header with the text "Hiển thị tên".

- Khi em nhập thiếu trường nào thì sẽ hiển thị vui lòng nhập đủ thông tin



Samsung Galaxy S7 (1440x2560, 560dpi) - 192.168.30.10...

4:40

Quản lý học viên

MSSV: 22521387

Tên SV: Tên Sinh Viên

Địa chỉ: Địa chỉ thường trú

SĐT: Số Điện thoại

Nhập

Vui lòng nhập đầy đủ thông tin sinh viên.

- Khi nhập xong thì sẽ hiển thị nhập xong và hiển thị dưới dạng Recyclerview

Quản lý học viên

MSSV:

Tên SV:

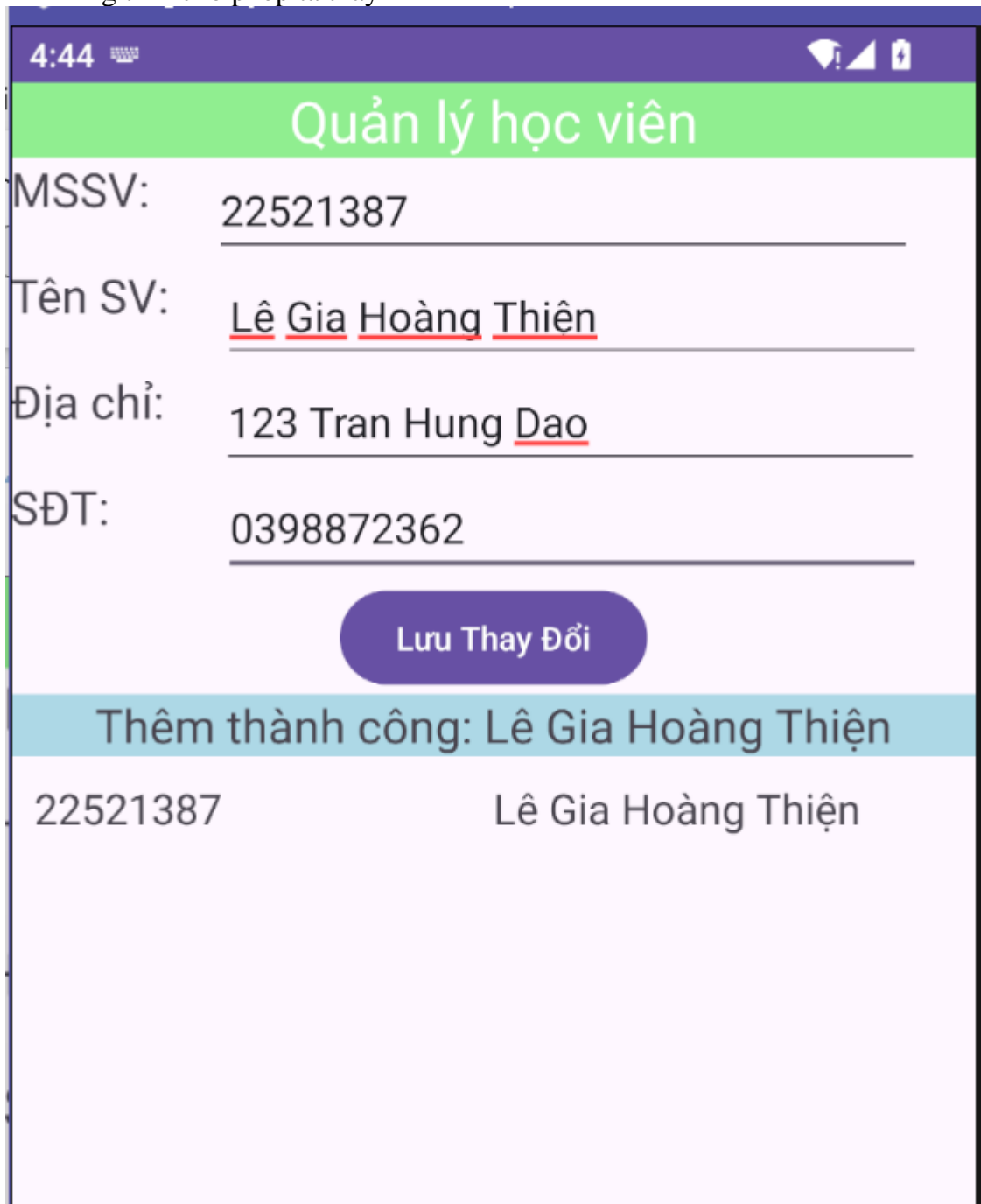
Địa chỉ:

SĐT:

Thêm thành công: Lê Gia Hoàng Thiện

22521387	Lê Gia Hoàng Thiện
----------	--------------------

- Khi nhấn vào node trong Recyclerview thì mọi thông tin sẽ được hiển thị trên màn hình đồng thời cho phép ta thay đổi



4:44

Quản lý học viên

MSSV: 22521387

Tên SV: Lê Gia Hoàng Thiên

Địa chỉ: 123 Tran Hung Dao

SĐT: 0398872362

Lưu Thay Đổi

Thêm thành công: Lê Gia Hoàng Thiên

22521387	Lê Gia Hoàng Thiên
----------	--------------------

- Đây là tên em muốn thay đổi

Samsung Galaxy S7 (1440x2560, 560dpi) - 192.168.30.10...

4:45

Quản lý học viên

MSSV: 22521387

Tên SV: Lê Gia Hoàng

Địa chỉ: 123 Tran Hung Dao

SĐT: 0398872362

Lưu Thay Đổi

Hiển thị tên

22521387 Lê Gia Hoàng Thiện

Sau khi nhấn thay đổi thì em sẽ quay lại màn hình thêm học viên và cập nhật trên Recyclerview

Quản lý học viên

MSSV:

Tên SV:

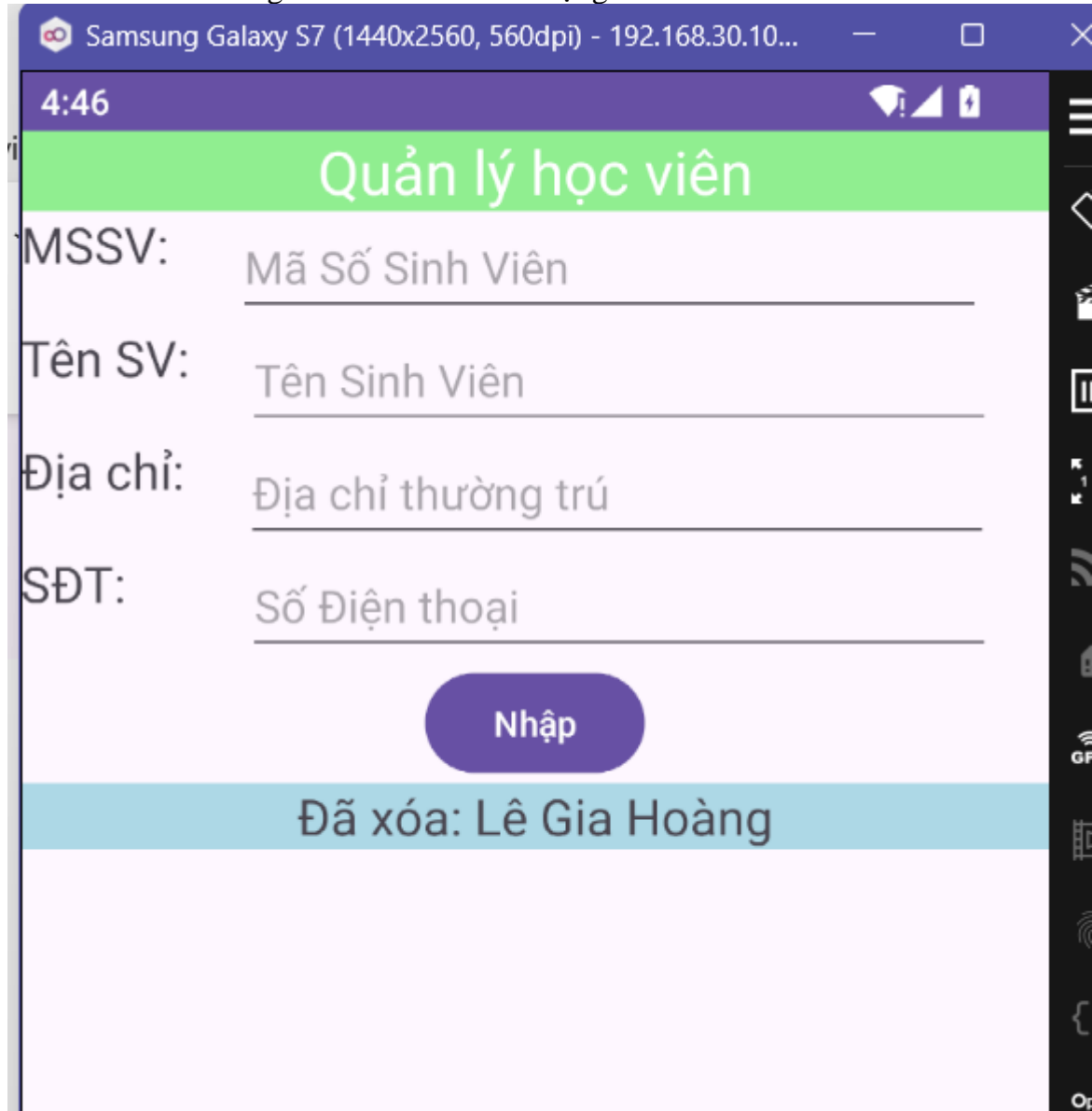
Địa chỉ:

SĐT:

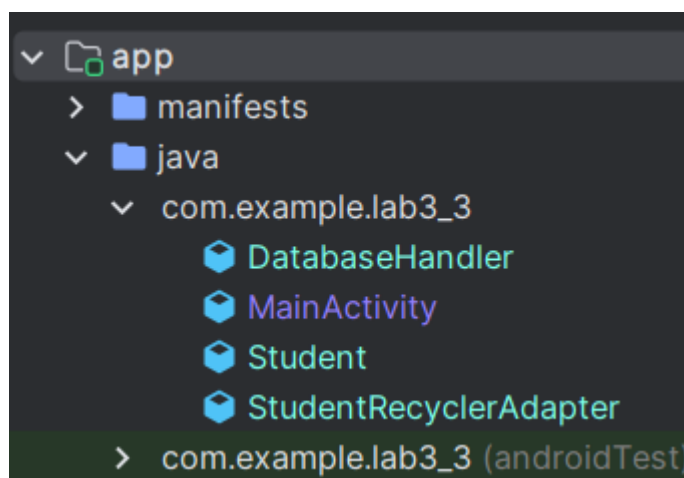
Hiển thị tên

22521387	Lê Gia Hoàng
----------	--------------

- Sau đó em nhấn giữ node để xóa đối tượng



- Các hàm java



- Một vài hàm xử lý trong Main activity

```

private void loadStudents() { 4 usages
    List<Student> students = databaseHandler.getAllStudents();
    adapter = new StudentRecyclerViewAdapter(students, clickListener: this, longClickListener: this);
    rvStudent.setAdapter(adapter);
}

private void addStudent() { 1 usage
    String code = etCode.getText().toString().trim();
    String name = etName.getText().toString().trim();
    String address = etAddress.getText().toString().trim();
    String phone = etPhone.getText().toString().trim();

    if (!code.isEmpty() && !name.isEmpty() && !address.isEmpty() && !phone.isEmpty()) {
        Student student = new Student(code, name, address, phone);
        databaseHandler.addStudent(student);
        loadStudents();
        clearFields();
        tvResult.setText("Thêm thành công: " + name);
    } else {
        tvResult.setText("Vui lòng nhập đầy đủ thông tin sinh viên.");
    }
}

private void loadStudentForEditing(Student student) { 1 usage
    if (student != null) {
        etCode.setText(student.getCode());
        etName.setText(student.getName());
        etAddress.setText(student.getAddress());
        etPhone.setText(student.getPhone());
        selectedStudent = student; // Set selected student for updating

        btnSaveChanges.setVisibility(View.VISIBLE); // Show save button
        btnAddStudent.setVisibility(View.GONE); // Hide add button
    }
}

private void updateStudent() { 1 usage
    if (selectedStudent != null) {
        // Update the selected student with new values
        String code = etCode.getText().toString().trim();
        String name = etName.getText().toString().trim();
        String address = etAddress.getText().toString().trim();
        String phone = etPhone.getText().toString().trim();

        // Check if fields are valid before updating
    }
}

```

```
}

@Override 1usage
public void onItemClick(Student student) {
    if (student != null) {
        databaseHandler.deleteStudent(student);
        loadStudents();
        clearFields(); // Clear fields in case the deleted student was selected
        tvResult.setText("Đã xóa: " + student.getName());
    } else {
        tvResult.setText("Lỗi: Sinh viên không tồn tại.");
    }
}
}
```

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.

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

HẾT