**Trường cao đẳng công nghệ thủ đức**

**CÔNG NGHỆ THÔNG TIN**

****

**LẬP TRÌNH DI ĐỘNG**

***Chủ Đề: Quản lý bách hóa xanh***

**Giáo viên hướng dẫn:** Trương Bá Thái

**TP.HỒ CHÍ MINH 2023- 2024**

**MỤC LỤC**

1. [Giới thiệu 2](file:///C:\Users\Admin\Downloads\Mau-Bia-DAMH.docx#_Toc45459181)

[2. Thành viên 3](file:///C:\Users\Admin\Downloads\Mau-Bia-DAMH.docx#_Toc45459184)

[3. Phân tích thiết kế hệ thống 4](file:///C:\Users\Admin\Downloads\Mau-Bia-DAMH.docx#_Toc45459184)

[4.Cài đặt hệ thống 14](file:///C:\Users\Admin\Downloads\Mau-Bia-DAMH.docx#_Toc45459184)

[5. Kết luận và kiến nghị 4](file:///C:\Users\Admin\Downloads\Mau-Bia-DAMH.docx#_Toc45459184)9

[6. Tài liệu tham khảo 50](file:///C:\Users\Admin\Downloads\Mau-Bia-DAMH.docx#_Toc45459184)

# Giới thiệu

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất đến Thầy Trương Bá Thái . Trong quá trình học tập và tìm hiểu về học phần “Lập trình di động”, chúng em đã nhận được sự quan tâm giúp đỡ, hướng dẫn rất tận tình, tâm huyết của cô. Thầy Thái đã giúp chúng em tích lũy thêm nhiều kiến thức. Từ những kiến thức và sự hướng dẫn của thầy, chúng em đã hoàn thành đồ án.

Chúng em xin chân thành cảm ơn!

TP.HCM, ngày 27 tháng 10 năm 2023.

Sinh viên đồng thực hiện.

# Thành viên

## Thông tin thành viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Vai trò** | **Email** | **SĐT** |
| 1 | Sok Kim Thanh | Nhóm trưởng | 22211tt0063@mail.tdc.edu.vn | 0924964226 |
| 2 | Phạm Liêm | Thành viên | 22211tt0063@mail.tdc.edu.vn | 0797110714 |

## Đánh giá mức độ tham gia đồ án(đánh giá trên thang điểm 5)

|  |  |  |
| --- | --- | --- |
|  | Sok Kim Thanh | Phạm Liêm |
| Đóng góp ý kiến | 5 | 5 |
| Tinh thần làm việc | 5 | 5 |
| Khối lượng công việc hoàn thành | 5 | 5 |
| Chất lượng công việc | 5 | 5 |
| Tuân thủ thời gian làm việc | 5 | 5 |
| **Tổng điểm** | 5 | 5 |
| **Mức độ (%)** | 100% | 100% |

## Quá trình đóng góp công việc chi tiết (Tổng công việc là 100%)

|  |  |  |  |
| --- | --- | --- | --- |
| Họ tên | STT | Công việc đóng góp | Phần trăm đóng góp |
| Sok Kim Thanh | 1 | CRUD Loại sản phẩm | 15% |
| 2 | CRUD Sản phẩm | 15% |
| 3 | Giỏ hàng | 5% |
| 4 | Tìm sản phẩm theo loại | 5% |
| 5 | Thêm sản phẩm vào giỏ hàng. | 5% |
| 6 | Xây dựng main activity | 2.5% |
| 7 | Xây dựng navigation bar | 2.5% |
|  |  | Tổng phần trăm | 50% |
| Phạm Liêm | 1 | Chức năng thanh toán giỏ hàng | 5% |
| 2 | Tăng giảm số lượng giỏ hàng | 5% |
| 3 | Tăng giảm số lượng tồn kho | 5% |
| 4 | CRUD Hóa đơn | 15% |
| 5 | Xem lịch sử hóa đơn | 10% |
| 6 | Thống kê hóa đơn | 10% |
|  |  | Tổng phần trăm | 50% |

1. Phân tích thiết kế hệ thống

Trong phần này, chúng em sẽ trình bày chi tiết về việc phân tích và thiết kế hệ thống cho ứng dụng của chúng em. Chúng em đã xác định một loạt các chức năng cần thiết để đảm bảo rằng ứng dụng hoạt động một cách hiệu quả và đáp ứng nhu cầu của người dùng. Các chức năng này bao gồm các màn hình đi kèm mô tả như sau:

**Chức năng CRUD loại sản phẩm**: Cho phép người dùng tạo, đọc, cập nhật và xóa các loại sản phẩm.

**Chức năng CRUD sản phẩm**: Cho phép người dùng tạo, đọc, cập nhật và xóa các sản phẩm.

**Chức năng tìm sản phẩm theo loại**: Cho phép người dùng tìm kiếm sản phẩm dựa trên loại sản phẩm.

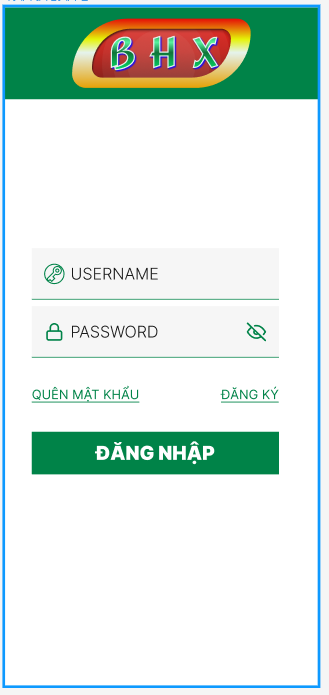
**Chức năng giỏ hàng**: Cho phép người dùng thêm sản phẩm vào giỏ hàng và theo dõi số lượng sản phẩm đang mua sắm.

**Chức năng CRUD Đơn hàng**: Cho phép người dùng tạo, đọc, cập nhật và xóa các đơn hàng.

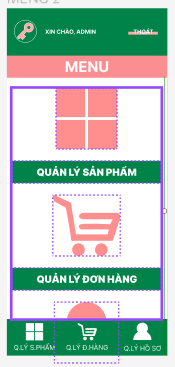
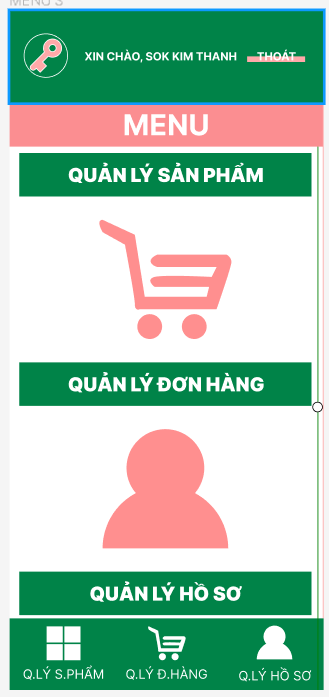
**Chức năng thống kê sản phẩm**: Cho phép người dùng xem lịch sử ngày nhập sản phẩm trong một khoảng thời gian cụ thể.

**Chức năng thống kê đơn hàng**: Cho phép người dùng xem lịch sử đơn hàng trong một khoảng thời gian cụ thể.

Các chức năng này đã được thiết kế để đảm bảo rằng ứng dụng của chúng em cung cấp một trải nghiệm người dùng mạnh mẽ và dễ sử dụng. Trong các phần tiếp theo, chúng em sẽ đi sâu vào việc phân tích và thiết kế từng chức năng này theo màn hình và code xử lý.

****

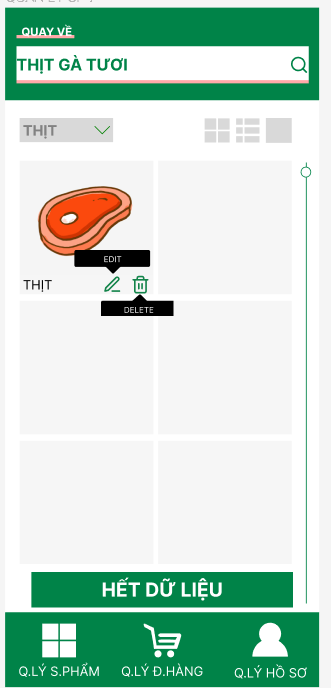
Giao diện màn hình đăng nhập



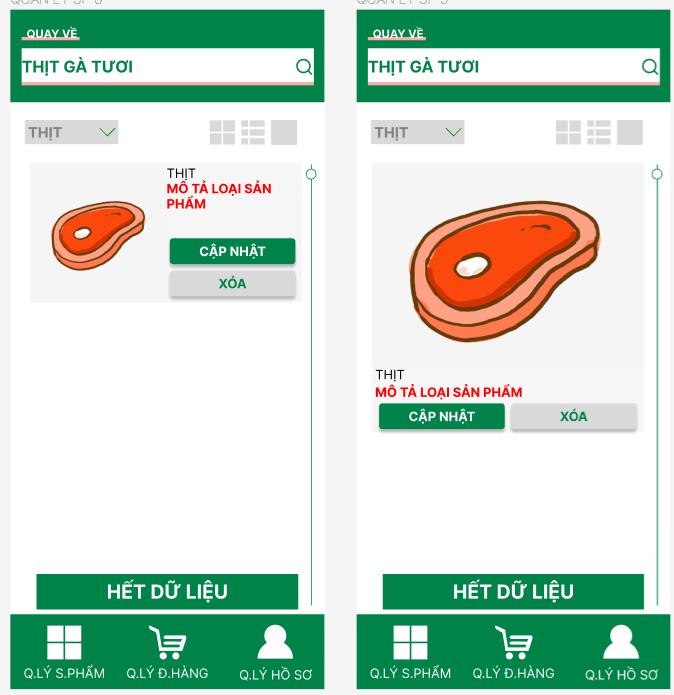
Giao diện trang chính khi vừa vào app quản lý bách hóa xanh



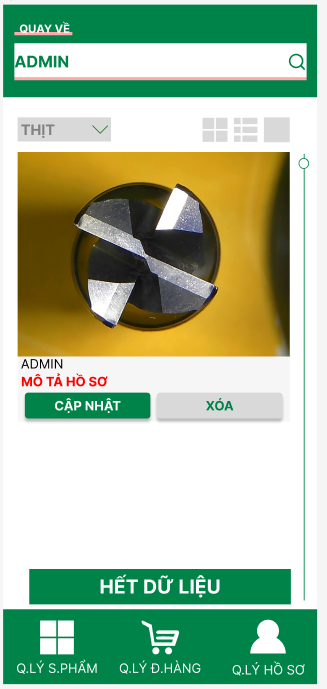
Màn hình danh mục các loại sản phẩm



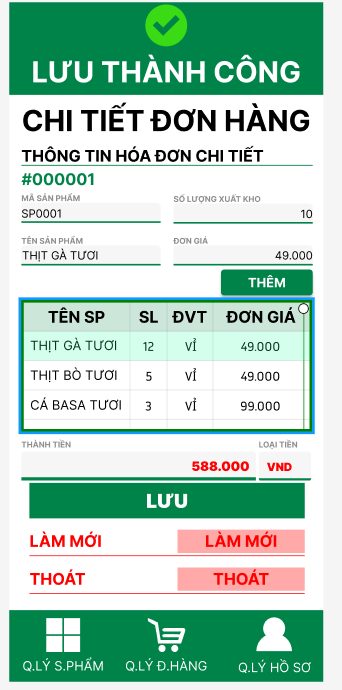
Màn hình thông tin các loại sản phẩm



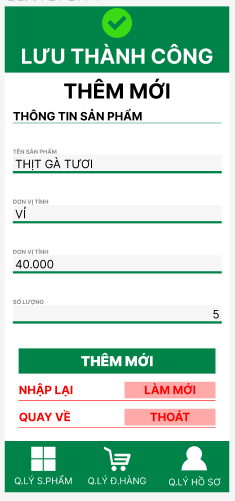
Màn hình dùng để lựa chọn chỉnh sửa danh sách sản phẩm



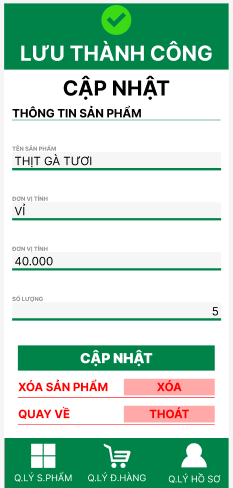
Màn hình chi tiết sản phẩm



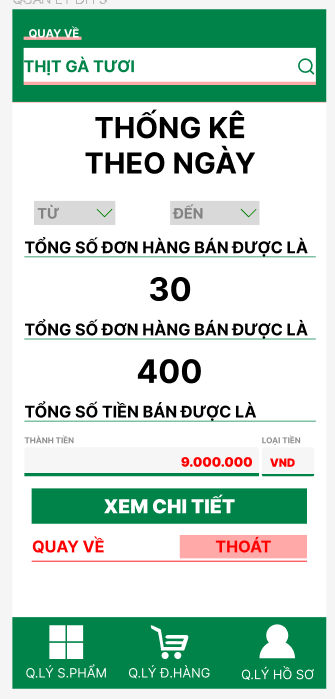
Màn hình chi tiết đơn hàng



Màn hình thêm mới sản phẩm

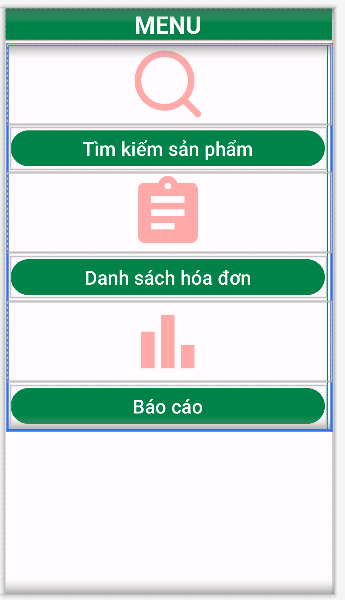


Màn hình cập nhật sản phẩm



Màn hình thống kê theo ngày

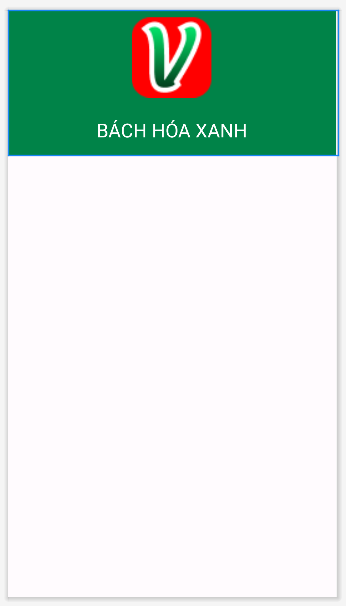
1. Cài đặt hệ thống
2. **Sản phẩm**

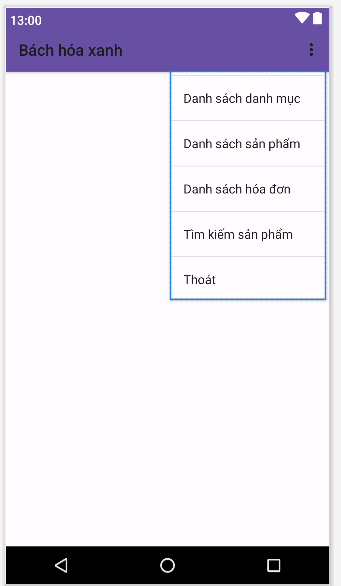


Màn hình layout chính khi vừa khởi động ứng dụng



Sự kiện khi nhấn vào các nút Báo cáo, Danh sách hóa đơn, Tìm kiếm sản phẩm

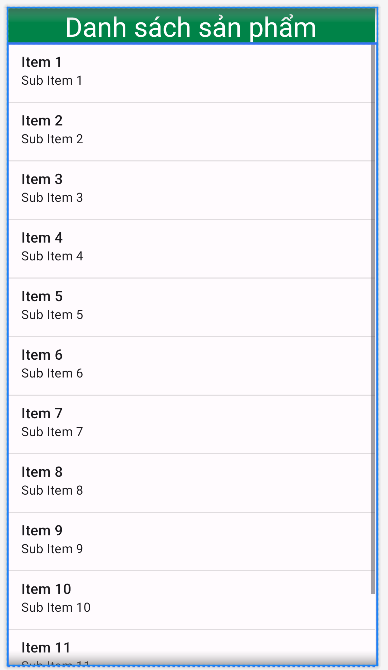
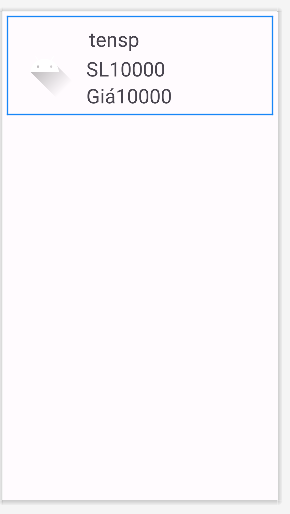




Màn hình navigation\_header và màn hình menu\_items

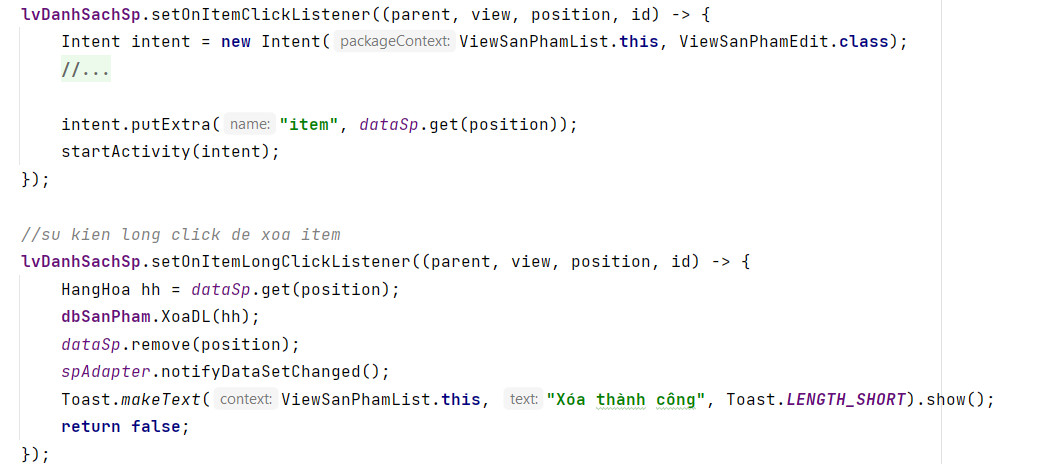


Code cho sự kiện chuyển trang khi nhấn vào các item của navigation

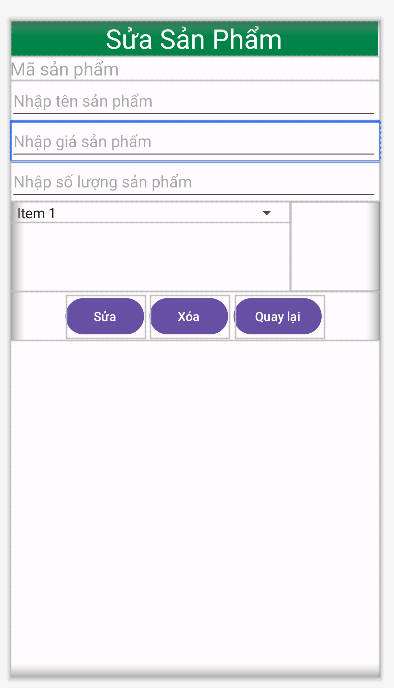


Layout màn hình hiện thị danh sách sản phẩm và layout màn hình định dạng kiểu xuất trong từng phần tử của danh sách đó



Code hàm khởi tạo cho danh sách sản phẩm

Code cho sự kiện khi nhấn giữ vào từng phần tử trong lvDanhSachSP



Layout màn hình sửa sản phẩm



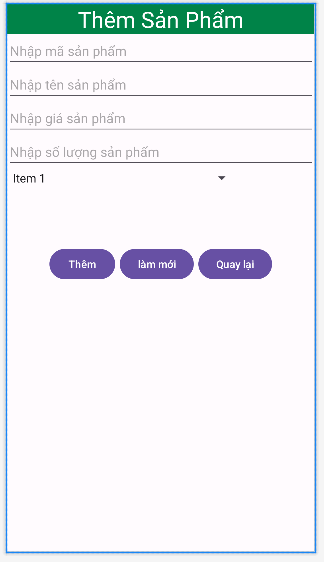
Sự kiện khi nhấn nút sửa sản phẩm



Sự kiện khi chọn nút spinner



Sự kiện khi nhấn nút thoát sẽ xóa dữ liệu trong databases

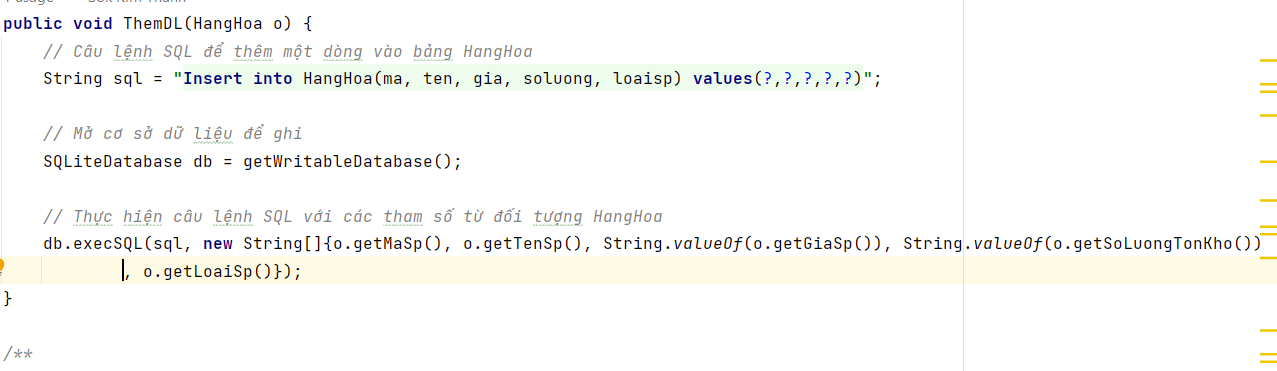


Layout màn hình thêm sản phẩm

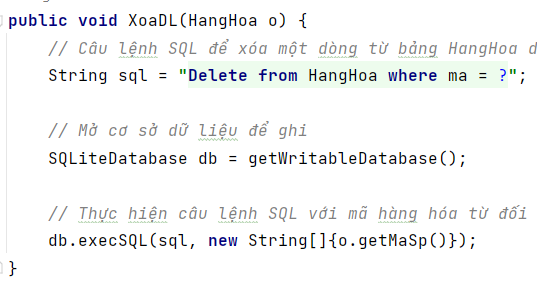




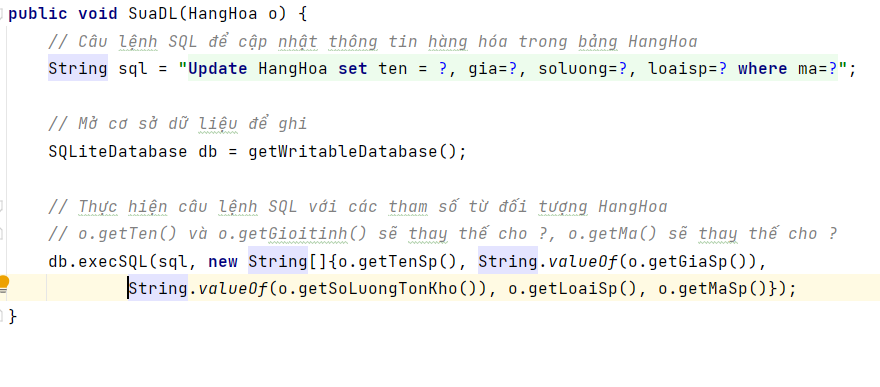
Sự kiện khi nhấn vào nút thêm các dữ liệu sản phẩm sẽ được lưu vào dbHangHoa



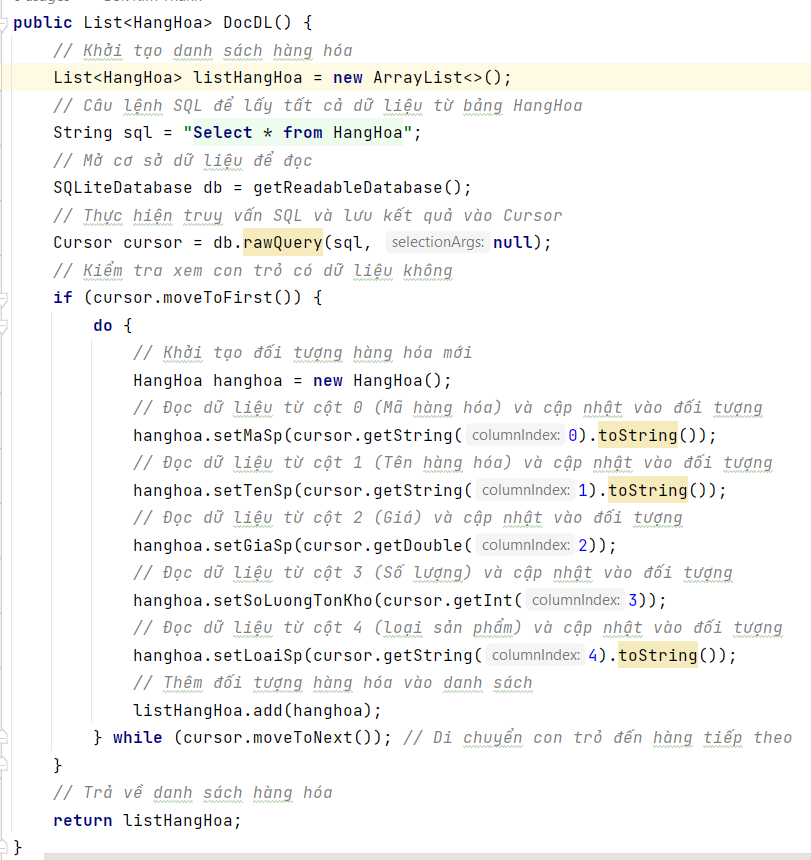
Hàm thêm hàng hóa vào trong DBHangHoa



Hàm xóa hàng hóa trong DBHangHoa



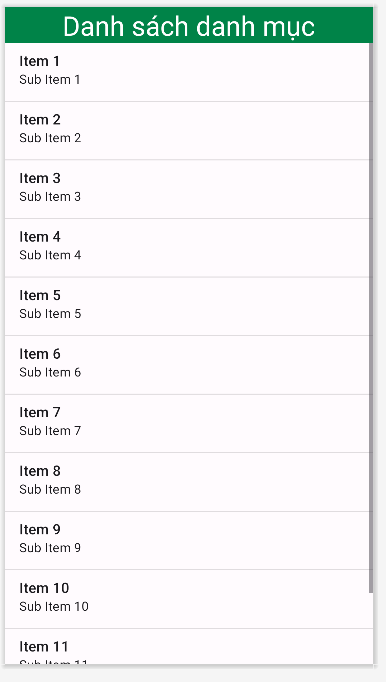
Hàm sửa hàng hóa tròn DBHangHoa



Hàm đọc dữ liệu từ DBHangHoa trả về danh sách HangHoa

1. Danh mục





Layout màn hình hình danh sách danh mục và layout màn hình dùng để định dạng các phần tử trong danh sách danh mục

**public class** ViewDanhMucList **extends** AppCompatActivity {  
  
 DBDanhMuc **dbDanhMuc**;  
**static** DanhMucList *danhMucList*;  
 **static** DanhMucListAdapter *adapter*;  
  
 ListView **lvDanhMucList**;  
  
 ImageView **ivHinhDM**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.*layout\_danhmuc\_list*);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);*// hiển thị nút quay lại trang chủ  
  
 // anh xa* setControl();  
 setEvent();  
 }  
  
 *//hien thi danh sach* **private void** setControl() {  
  
 **this**.**lvDanhMucList** = findViewById(R.id.*lvDanhSachDM*);  
 }  
  
 *// gan menu bar* **private void** setEvent() {  
 *// khoi tao san pham* KhoiTao();  
   
 *// su kien click vao item de update* **lvDanhMucList**.setOnItemClickListener((parent, view, position, id) -> {  
 Intent intent = **new** Intent(ViewDanhMucList.**this**, ViewDanhMucEdit.**class**);  
 *// bạn cần phải chắc chắn rằng DanhMuc  
 // có thể được chuyển đổi thành CharSequence,  
 // hoặc bạn cần thay đổi cách bạn chuyển dữ liệu  
 // giữa các hoạt động.  
 // Một cách để làm điều này là để làm cho lớp  
 // DanhMuc triển khai Serializable hoặc Parcelable,  
 // sau đó bạn có thể chuyển toàn bộ đối tượng qua Intent.  
 /\*\*  
 \* public class DanhMuc implements Serializable {  
 \* // các trường và phương thức của bạn ở đây  
 \* }  
 \* \*/* intent.putExtra(**"item"**, *danhMucList*.getDanhMucList().get(position));  
 startActivity(intent);  
 });  
}  
**private void** KhoiTao() {  
  
 *danhMucList* = **new** DanhMucList();**dbDanhMuc** = **new** DBDanhMuc(**this**);  
  
 *danhMucList*.getDanhMucList().clear();  
  
 *// Kiểm tra xem cơ sở dữ liệu có rỗng không* **if** ((**long**) **dbDanhMuc**.DocDL().size() <= 0) {  
 Toast.*makeText*(**this**, **"DB Rỗng không có dữ liệu"**, Toast.***LENGTH\_SHORT***).show();  
 **return**;  
 }  
 *// Thêm dữ liệu từ cơ sở dữ liệu vào danh sách và cập nhật giao diện* **for** (DanhMuc danhMuc: **dbDanhMuc**.DocDL()) {  
 *danhMucList*.Them(danhMuc);  
 }  
  
 *// gan san pham bang menu item layout(gan template item)  
 adapter* = **new** DanhMucListAdapter(**this**, R.layout.*layout\_danhmuc\_item*, *danhMucList*.getDanhMucList());  
 *// hien thi len listview* **lvDanhMucList**.setAdapter(*adapter*);  
  
 *adapter*.notifyDataSetChanged();  
 }  
  
 *// gan menu vao danh sach* @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*menu\_layout*, menu);  
 **return super**.onCreateOptionsMenu(menu);  
 }  
  
 *// gan su kien cho menu* @Override  
 **public boolean** onOptionsItemSelected(@NonNull MenuItem item) {  
 *// TH: click menu them* **if** (item.getItemId() == R.id.*mnThem*) {  
 Intent intent = **new** Intent(**this**, ViewDanhMucAdd.**class**);  
 startActivity(intent);  
 }**if**(item.getItemId() == android.R.id.***home***){  
onBackPressed();  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

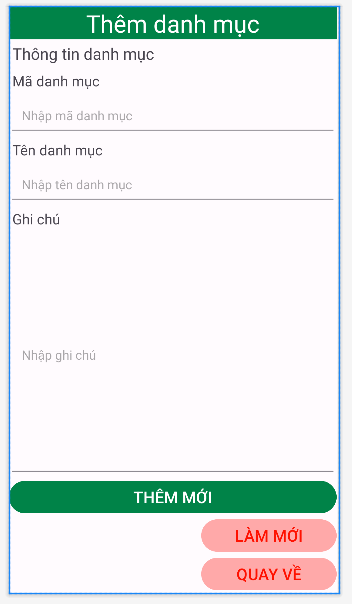
Code sự kiện màn hình viewDanhMucList



Layout màn hình sửa danh mục

**public class** ViewDanhMucEdit **extends** AppCompatActivity {  
 TextView **tvMaDM**;  
 EditText **edtTenDM**, **edtGhiChuDM**;  
 Button **btnSua**, **btnXoa**, **btnQuayVe**;  
  
 DanhMuc **danhmuc**;  
 DBDanhMuc **dbDanhMuc**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.*layout\_danhmuc\_edit*);  
 setControl();  
 setEvent();  
 }  
  
 **private void** setEvent() {  
 KhoiTao();  
 **btnSua**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (**tvMaDM**.getText().length() <= 0) {  
 **tvMaDM**.setError(**"Vui long nhap ma"**);  
 **return**;  
 }  
 **if** (**edtTenDM**.getText().length() <= 0) {  
 **edtTenDM**.setError(**"Vui long nhap ten"**);  
 **return**;  
 }  
 **if** (**edtGhiChuDM**.getText().length() <= 0) {  
 **edtGhiChuDM**.setError(**"Vui long ghi chu"**);  
 **return**;  
 }  
 *//Cập nhật thông tin mới* **danhmuc**.setMa(**tvMaDM**.getText().toString());  
 **danhmuc**.setTen(**edtTenDM**.getText().toString());  
 **danhmuc**.setGhichu(**edtGhiChuDM**.getText().toString());  
  
 *// Lưu* **if** (ViewDanhMucList.*danhMucList*.Sua(**danhmuc**)) {  
 **dbDanhMuc**.SuaDL(**danhmuc**);  
 Toast.*makeText*(ViewDanhMucEdit.**this**, **"Sửa Thành Công!"**, Toast.***LENGTH\_SHORT***).show();  
 ViewDanhMucList.*adapter*.notifyDataSetChanged();*// Cập nhật ListView* }  
  
 }  
 });  
 **btnXoa**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **new** AlertDialog.Builder(ViewDanhMucEdit.**this**)  
 .setTitle(**"Xác nhận xóa"**)  
 .setMessage(**"Bạn có chắc chắn muốn xóa không?"**)  
 .setIcon(android.R.drawable.***ic\_dialog\_alert***)  
 .setPositiveButton(android.R.string.***yes***, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** whichButton) {  
 *// Xóa phần tử tại đây* **if** (ViewDanhMucList.*danhMucList*.Xoa(**danhmuc**)) {  
 **dbDanhMuc**.XoaDL(**danhmuc**);  
 Toast.*makeText*(ViewDanhMucEdit.**this**, **"Xóa Thành Công!"**, Toast.***LENGTH\_SHORT***).show();  
 ViewDanhMucList.*adapter*.notifyDataSetChanged();*// Cập nhật ListView* }  
 }  
 })  
 .setNegativeButton(android.R.string.***no***, **null**).show();  
  
 }  
 });  
 **btnQuayVe**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 finish();  
 }  
 });  
 }  
  
 **private void** KhoiTao() {  
 *// khoi tao gia tri từ danh sach danh muc  
 // dbDanhMuc truy cập dữ liệu DB* **dbDanhMuc** = **new** DBDanhMuc(**this**);  
 *// Khởi tạo item để chuyển đối tượng lên màn hình* **danhmuc** = (DanhMuc) getIntent().getSerializableExtra(**"item"**);  
  
 *//hiển thị dữ liệu lên màn hình sửa* **tvMaDM**.setText(**danhmuc**.getMa());*// mã* **edtTenDM**.setText(**danhmuc**.getTen());*// tên* **edtGhiChuDM**.setText(**danhmuc**.getGhichu());*// ghi chu* }  
}

Code màn hình viewDanhMucEdit



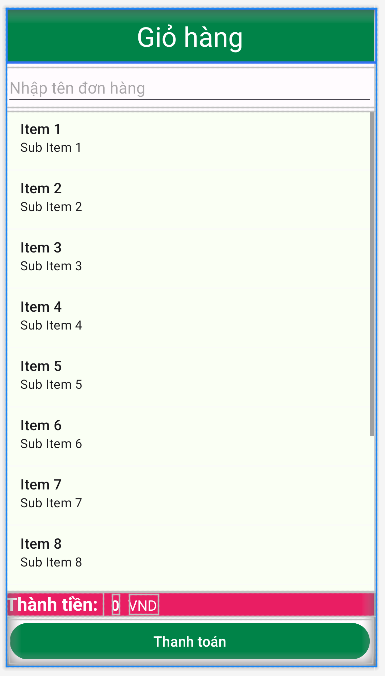
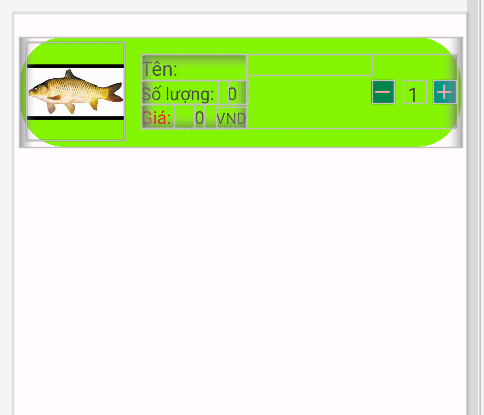
Layout màn hình thêm danh mục

**package** tdc.edu.danhsachdm;  
  
**import** android.os.Bundle;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.EditText;  
**import** android.widget.Toast;  
  
**import** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** tdc.edu.danhsachsp.R;  
  
**public class** ViewDanhMucAdd **extends** AppCompatActivity {  
  
 EditText **edtMaDM**, **edtTenDM**, **edtGhiChuDM**;  
 Button **btnThem**, **btnLamMoi**, **btnQuayVe**;  
  
 DBDanhMuc **dbDanhMuc**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 *// hiển thị nút quay lại trang chủ* getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 setContentView(R.layout.*layout\_danhmuc\_add*);  
 setControl();  
 setEvent();  
 }  
  
 **private void** setEvent() {  
 KhoiTao();  
 **btnThem**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **if** (**edtMaDM**.getText().length() <= 0) {  
 **edtMaDM**.setError(**"Vui long nhap ma"**);  
 **return**;  
 }  
 **if** (**edtTenDM**.getText().length() <= 0) {  
 **edtTenDM**.setError(**"Vui long nhap ten"**);  
 **return**;  
 }  
 **if** (**edtGhiChuDM**.getText().length() <= 0) {  
 **edtGhiChuDM**.setError(**"Vui long ghi chu"**);  
 **return**;  
 }  
 DanhMuc danhMuc = **new** DanhMuc(**edtMaDM**.getText().toString(), **edtTenDM**.getText().toString(), **edtGhiChuDM**.getText().toString());  
  
 **if** (ViewDanhMucList.*danhMucList*.Them(danhMuc)) {  
 *// Thêm đối tượng DanhMuc vào cơ sở dữ liệu* **dbDanhMuc**.ThemDL(danhMuc);  
 Toast.*makeText*(ViewDanhMucAdd.**this**, **"Thêm Thành Công!"**, Toast.***LENGTH\_SHORT***).show();  
 ViewDanhMucList.*adapter*.notifyDataSetChanged();*// Cập nhật ListView* }  
  
 }  
 });  
 **btnLamMoi**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **edtMaDM**.setText(**""**);  
 **edtTenDM**.setText(**""**);  
 **edtGhiChuDM**.setText(**""**);  
 **edtGhiChuDM**.clearFocus();  
 }  
 });  
 **btnQuayVe**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 finish();  
 }  
 });  
 }  
  
 **private void** KhoiTao() {  
 *// khoi tao gia tri danh sach danh muc  
 // dbsinhvien truy cập dữ liệu DB* **dbDanhMuc** = **new** DBDanhMuc(**this**);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(@NonNull MenuItem item) {  
  
 *// hỗ trợ quay lại màn hình chính* **if** (item.getItemId() == android.R.id.***home***) {  
 onBackPressed();  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
}

Code màn hình thêm danh mục  
  
**public class** DBDanhMuc **extends** SQLiteOpenHelper {  
  
 */\*\*  
 \* Khởi tạo DBDanhMuc  
 \*/* **public** DBDanhMuc(@Nullable Context context) {  
 *// Gọi hàm khởi tạo của lớp cha SQLiteOpenHelper* **super**(context, **"dbQuanLyBanHang"**, **null**, 1);  
 }  
  
 */\*\*  
 \* Tạo bảng tbDanhMuc  
 \*  
 \** ***@param db*** *The database.  
 \*/* @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 *// Câu lệnh SQL để tạo bảng tblDanhMuc với các cột ma, ten, ghichu* String sql = **"Create Table If not exists tblDanhMuc(ma Text, ten Text, ghichu text)"**;  
  
 *// Thực hiện câu lệnh SQL để tạo bảng* db.execSQL(sql);  
 }  
  
 */\*\*  
 \* Thêm danh mục  
 \*  
 \** ***@param sv*** *Danh mục  
 \*/* **public void** ThemDL(DanhMuc sv) {  
 *// Câu lệnh SQL để thêm một dòng vào bảng tblDanhMuc* String sql = **"Insert into tblDanhMuc values(?,?,?)"**;  
  
 *// Mở cơ sở dữ liệu để ghi* SQLiteDatabase db = getWritableDatabase();  
  
 *// Thực hiện câu lệnh SQL với các tham số từ đối tượng DanhMuc* db.execSQL(sql, **new** String[]{sv.getMa(), sv.getTen(), sv.getGhichu()});  
 }  
*//xxxxxx  
 /\*\*  
 \* Xóa danh mục  
 \*  
 \** ***@param sv*** *danhMuc  
 \*/* **public void** XoaDL(DanhMuc sv) {  
 *// Câu lệnh SQL để xóa một dòng từ bảng tblDanhMuc dựa trên mã danh mục* String sql = **"Delete from tblDanhMuc where ma = ?"**;  
  
 *// Mở cơ sở dữ liệu để ghi* SQLiteDatabase db = getWritableDatabase();  
  
 *// Thực hiện câu lệnh SQL với mã danh mục từ đối tượng DanhMuc* db.execSQL(sql, **new** String[]{sv.getMa()});  
 }  
  
  
 */\*\*  
 \* Sửa thông tin danh mục  
 \*  
 \** ***@param sv*** *The danhMuc.  
 \*/* **public void** SuaDL(DanhMuc sv) {  
 *// Câu lệnh SQL để cập nhật thông tin danh mục trong bảng tblDanhMuc* String sql = **"Update tblDanhMuc set ten = ?, ghichu=? where ma=?"**;  
  
 *// Mở cơ sở dữ liệu để ghi* SQLiteDatabase db = getWritableDatabase();  
  
 *// Thực hiện câu lệnh SQL với các tham số từ đối tượng DanhMuc  
 // sv.getTen() và sv.getGhichu() sẽ thay thế cho ?, sv.getMa() sẽ thay thế cho ?* db.execSQL(sql, **new** String[]{sv.getTen(), sv.getGhichu(), sv.getMa()});  
 }  
  
  
 */\*\*  
 \* Đọc dữ liệu từ db  
 \*  
 \** ***@return*** *danh sách đối tượng danh mục  
 \*/* **public** List<DanhMuc> DocDL() {  
 *// Khởi tạo danh sách danh mục* List<DanhMuc> listDanhMuc = **new** ArrayList<>();  
  
 *// Câu lệnh SQL để lấy tất cả dữ liệu từ bảng tblDanhMuc* String sql = **"Select \* from tblDanhMuc"**;  
  
 *// Mở cơ sở dữ liệu để đọc* SQLiteDatabase db = getReadableDatabase();  
  
 *// Thực hiện truy vấn SQL và lưu kết quả vào Cursor* Cursor cursor = db.rawQuery(sql, **null**);  
  
 *// Kiểm tra xem con trỏ có dữ liệu không* **if** (cursor.moveToFirst()) {  
 **do** {  
 *// Khởi tạo đối tượng danh mục mới* DanhMuc danhMuc = **new** DanhMuc();  
  
 *// Đọc dữ liệu từ cột 0 (Mã danh mục) và cập nhật vào đối tượng* danhMuc.setMa(cursor.getString(0).toString());  
  
 *// Đọc dữ liệu từ cột 1 (Tên danh mục) và cập nhật vào đối tượng* danhMuc.setTen(cursor.getString(1).toString());  
  
 *// Đọc dữ liệu từ cột 2 (Giới tính) và cập nhật vào đối tượng* danhMuc.setGhichu(cursor.getString(2).toString());  
  
 *// Thêm đối tượng danh mục vào danh sách* listDanhMuc.add(danhMuc);  
 } **while** (cursor.moveToNext()); *// Di chuyển con trỏ đến hàng tiếp theo* }  
  
 *// Trả về danh sách danh mục* **return** listDanhMuc;  
 }  
  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
  
 }  
}

Code dbDanhMuc dùng để thêm sửa xóa dữ liệu

1. Giỏ hàng



**public class** ViewGioHangList **extends** AppCompatActivity {  
 *// Khởi tạo adapter với danh sách GioHang* GioHangAdapter **adapter**;  
  
  
 *// Lấy danh sách hàng hóa từ giỏ hàng* List<HangHoa> **hangHoaList**;  
  
 *// Cài đặt adapter cho ListView của bạn* ListView **listView**;  
 **static** TextView *tvTongThanhTien*;  
 Button **btnThanhToan**;  
 EditText **edttenDonHang**;  
 ImageView **ivGioHangNull**;  
 DBGioHang **dbGioHang** = **new** DBGioHang(ViewGioHangList.**this**);  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.*layout\_giohang\_list*);  
 Objects.*requireNonNull*(getSupportActionBar()).setDisplayHomeAsUpEnabled(**true**);*// hiển thị nút quay lại trang chủ* setControl();  
 setEvent();  
 }  
  
 **private void** setEvent() {  
 KhoiTao();  
 Calendar calendar = Calendar.*getInstance*();  
 **int** year = calendar.get(Calendar.***YEAR***);  
 **int** month = calendar.get(Calendar.***MONTH***) + 1;  
 **int** day = calendar.get(Calendar.***DAY\_OF\_MONTH***);  
 *// xử lý cộng trừ gio hàng* **if**(**hangHoaList**.size() == 0){  
 Toast.*makeText*(**this**, **"Giỏ hàng rỗng"**, Toast.***LENGTH\_SHORT***).show();  
 *// khóa thao tác thanh toán* **btnThanhToan**.setEnabled(**false**);  
 }**else** {  
 *// tắt thông báo  
 // mở khóa thao tác thanh toán* **btnThanhToan**.setEnabled(**true**);  
 }  
 **btnThanhToan**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 DBHangHoa dbHangHoa = **new** DBHangHoa(ViewGioHangList.**this**);  
  
 **if**(**edttenDonHang**.getText()!=**null**){  
 String msg = **" "**+day+**"/"**+month+**"/"**+year+**"\n"**;  
 msg +=**"\n "**+ **edttenDonHang**.getText().toString()+**"\n\n"**;  
 **for** (**int** i = 0;i<**hangHoaList**.size();i++){  
 dbHangHoa.SuaDL(**hangHoaList**.get(i));  
 msg+=**"Tên sản phẩm: "**+**hangHoaList**.get(i).getTenSp()+**"\n"** +**"Giá: "**+**hangHoaList**.get(i).getGiaSp()+**"VND sl: "**+**hangHoaList**.get(i).getSlban()+**"\n"**;  
 }  
 msg+=**"\nTổng tiền: "**+*tvTongThanhTien*.getText().toString()+**" VND"**;  
 **dbGioHang**.ThemDL(**edttenDonHang**.getText().toString(),msg,*tvTongThanhTien*.toString());  
 Toast.*makeText*(ViewGioHangList.**this**, **"Thêm thành công"**, Toast.***LENGTH\_SHORT***).show();  
 }  
 }  
 });  
 }  
  
 **private void** KhoiTao() {  
 *// Lấy danh sách hàng hóa từ giỏ hàng* **hangHoaList** = ViewProtypeProductSearch.*gioHang*.getHangHoaList();  
  
 *// Tính tổng số tiền và cập nhật tvTongThanhTien* **double** tongThanhTien = ViewProtypeProductSearch.*gioHang*.getTongThanhTien();  
 *tvTongThanhTien*.setText(String.*valueOf*(tongThanhTien));  
  
 *// Khởi tạo adapter với danh sách GioHang* **adapter** = **new** GioHangAdapter(**this**, **hangHoaList**);  
 **listView**.setAdapter(**adapter**);  
 }  
  
 **private void** setControl() {  
 *// Cài đặt adapter cho ListView* **listView** = findViewById(R.id.*cartListView*);  
 *// cài đặt textview tổng thành tiền  
 tvTongThanhTien* = findViewById(R.id.*tvTongThanhTien*);  
 *// Cài đặt button thanh toán* **btnThanhToan** = findViewById(R.id.*btnThanhToan*);  
 **edttenDonHang** = findViewById(R.id.*edtTenDonHang*);  
  
 }  
 */\*\*  
 \* Xử lý menu event  
 \*  
 \** ***@param item*** *The menu item that was selected.  
 \*/* @Override  
 **public boolean** onOptionsItemSelected(@NonNull MenuItem item) {  
  
 *// hỗ trợ quay lại màn hình chính* **if** (item.getItemId() == android.R.id.***home***) {  
 onBackPressed();  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
  
}

Code trang ViewGioHangList dùng để hiện thị danh sách các mặt hàng

**package** tdc.edu.giohang;  
  
**import** android.annotation.SuppressLint;  
**import** android.content.Context;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.database.sqlite.SQLiteOpenHelper;  
  
**import** androidx.annotation.Nullable;  
  
**import** java.util.ArrayList;  
**import** java.util.List;  
  
**public class** DBGioHang **extends** SQLiteOpenHelper {  
  
 **public** DBGioHang(@Nullable Context context) {  
 **super**(context, **"QLBach\_HoaXanh"**, **null**, 1);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
 String sql = **"create table Hoa\_Don(maDH text,thongtin text,ngay text)"**;  
 db.execSQL(sql);  
 }  
 **public void** ThemDL(String maHD, String chuoi,String ngay) {  
 SQLiteDatabase db = getWritableDatabase();  
 String sql = **"INSERT INTO Hoa\_Don VALUES(?,?,?)"**;  
 db.execSQL(sql, **new** String[]{maHD,chuoi,ngay});  
 db.close();  
 }  
 **public void** XoaDL(String ma) {  
 String sql = **"Delete from Hoa\_Don where maDH = ?"**;  
 SQLiteDatabase db = getWritableDatabase();  
 db.execSQL(sql, **new** String[]{ma});  
 }  
 **public** List<String> TraCuu() {  
 List<String> listHoaDon = **new** ArrayList<>();  
 String sql = **"Select \* from Hoa\_Don"**;  
 SQLiteDatabase db = getReadableDatabase();  
 Cursor cursor = db.rawQuery(sql, **null**);  
 **if** (cursor.moveToFirst()) {  
 **do** {  
 listHoaDon.add((String) cursor.getString(0).toString());  
 } **while** (cursor.moveToNext());  
 }  
 **return** listHoaDon;  
 }  
 **public** List<String> TraCuu2() {  
 List<String> listHoaDon = **new** ArrayList<>();  
 String sql = **"Select \* from Hoa\_Don"**;  
 SQLiteDatabase db = getReadableDatabase();  
 Cursor cursor = db.rawQuery(sql, **null**);  
 **if** (cursor.moveToFirst()) {  
 **do** {  
 listHoaDon.add((String) cursor.getString(2).toString());  
 } **while** (cursor.moveToNext());  
 }  
 **return** listHoaDon;  
 }  
 **public** List<String> DocDL() {  
 List<String> listHoaDon = **new** ArrayList<>();  
 String sql = **"Select \* from Hoa\_Don"**;  
 SQLiteDatabase db = getReadableDatabase();  
 Cursor cursor = db.rawQuery(sql, **null**);  
 **if** (cursor.moveToFirst()) {  
 **do** {  
 *// listHoaDon.add((String) cursor.getString(0).toString());* listHoaDon.add((String) cursor.getString(1).toString());  
 *// listHoaDon.add((String) cursor.getString(2).toString());* } **while** (cursor.moveToNext());  
 }  
 **return** listHoaDon;  
 }  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
  
 }  
}

Code DBGioHang dùng để thêmn xóa các sản phẩm ở trong giỏ hàng

1. Kết luận và kiến nghị

## Kết luận

Trong quá trình thực hiện dự án, chúng em đã thành công trong việc xây dựng và triển khai một hệ thống quản lý sản phẩm hiệu quả. Chúng em đã cài đặt các chức năng CRUD cho loại sản phẩm và sản phẩm, tạo ra một giỏ hàng hoạt động, và cung cấp khả năng tìm kiếm sản phẩm theo loại. Chúng em cũng đã xây dựng thành công main activity và navigation bar, cung cấp chức năng thanh toán giỏ hàng, và cho phép người dùng tăng hoặc giảm số lượng sản phẩm trong giỏ hàng và kho. Cuối cùng, chúng em đã cài đặt chức năng CRUD cho hóa đơn và cho phép người dùng xem lịch sử hóa đơn.

## Kiến nghị

Dựa trên kết quả của dự án, chúng em đề xuất một số cải tiến cho tương lai. Đầu tiên, chúng em đề xuất thêm chức năng lọc sản phẩm dựa trên giá cả, đánh giá, hoặc thương hiệu. Thứ hai, chúng em đề xuất cải tiến chức năng giỏ hàng bằng cách cho phép người dùng lưu giỏ hàng để mua sắm sau. Cuối cùng, chúng em đề xuất thêm chức năng gửi thông báo cho người dùng khi có sản phẩm mới hoặc khi có khuyến mãi đặc biệt.

1. Tài liệu tham khảo

[1] Android Developers. (2023). Create a List with RecyclerView. https://developer.android.com/guide/topics/ui/layout/recyclerview (Truy cập ngày 20 tháng 11 năm 2023)

[2] Android Developers. (2023). Save data using SQLite. https://developer.android.com/training/data-storage/sqlite (Truy cập ngày 21 tháng 11 năm 2023)

[3] Android Developers. (2023). Build a Responsive UI with ConstraintLayout. https://developer.android.com/training/constraint-layout

[4] Android Developers. (2023). Create a Navigation Drawer. https://developer.android.com/training/implementing-navigation/nav-drawer (Truy cập ngày 23 tháng 11 năm 2023)

[5] Android Developers. (2023). Create a Card-Based Layout. https://developer.android.com/guide/topics/ui/layout/cardview (Truy cập ngày 24 tháng 11 năm 2023)