

TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC **KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO KẾT THÚC MÔN HỌC**

**Lập trình android nâng cao**

**Xây dựng ứng dụng**

**“Quản lý đơn đặt hàng sản phẩm”**

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

**Sinh viên thực hiện: Nhóm 9**

**1. Trần Bình Văn**

**2. Lê Văn Thắng**

**Khoa: Công nghệ thông tin** **Khóa: 2017**

**Ngày 27/07/2020**

**NHẬT KÝ HOẠT ĐỘNG NHÓM 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | HỌ TÊN | CÔNG VIỆC | TỰ ĐÁNH GIÁ | NHÓM ĐÁNH GIÁ | KÝ TÊN |
| 1 | Trần Bình Văn | * Sản phẩm * Đơn hàng * Splash Screen * Thống kê * Powerpoint | 50% | 50% |  |
| 2 | Lê Văn Thắng | * Khách hàng * Home * Navigation Drawer * About Us * Đổi ngôn ngữ | 50% | 50% |  |

**MỤC LỤC**

[CHƯƠNG 1 – MỞ ĐẦU 4](#_Toc47486226)

[**1.1.** **Giới thiệu môn học và nhóm thực hiện** 4](#_Toc47486227)

[**1.2.** **Mô tả ứng dụng** 4](#_Toc47486228)

[**1.3.** **Ứng dụng trong cuộc sống** 4](#_Toc47486229)

[CHƯƠNG 2 – PHÂN TÍCH CẤU TRÚC HỆ THỐNG 5](#_Toc47486230)

[2.1. Phân tích thiết kế hệ thống 5](#_Toc47486231)

[2.2. Thiết kế hệ thống 12](#_Toc47486232)

[**a)** **Màn hình splash (Trần Bình Văn)** 12](#_Toc47486233)

[**b)** **Màn hình home (Lê Văn Thắng)** 13](#_Toc47486234)

[**c)** **Màn hình danh sách khách hàng ( Lê Văn Thắng)** 16](#_Toc47486235)

[**d)** **Màn hình thêm khách hàng (Lê Văn Thắng)** 19](#_Toc47486236)

[**e)** **Màn hình sửa khách hàng (Lê Văn Thắng)** 20](#_Toc47486237)

[**f)** **Màn hình danh sách sản phẩm (Trần Bình Văn)** 21](#_Toc47486238)

[**g)** **Màn hình thêm sản phẩm (Trần Bình Văn)** 23](#_Toc47486239)

[**h)** **Màn hình sửa sản phẩm (Trần Bình Văn)** 25](#_Toc47486240)

[**i)** **Màn hình danh sách đơn hàng (Trần Bình Văn)** 26](#_Toc47486241)

[**j)** **Màn hình thêm đơn hàng (Trần Bình Văn)** 28](#_Toc47486242)

[**k)** **Màn hình sửa đơn hàng (Trần Bình Văn)** 29](#_Toc47486243)

[**l)** **Màn hình thống kê đơn hàng (Trần Bình Văn)** 31](#_Toc47486244)

[**m)** **Màn hình đổi ngôn ngữ (Lê Văn Thắng)** 32](#_Toc47486245)

[**n)** **Màn hình About Us (Lê Văn Thắng)** 32](#_Toc47486246)

[CHƯƠNG 3 – CÀI ĐẶT 34](#_Toc47486247)

[3.1. Màn hình home 34](#_Toc47486248)

[3.2. Màn hình khách hàng 36](#_Toc47486249)

[3.3. Màn hình sản phẩm 40](#_Toc47486250)

[3.4. Màn hình đơn hàng 43](#_Toc47486251)

[3.5. Màn hình thống kê 47](#_Toc47486252)

[3.6. Màn hình đổi ngôn ngữ 49](#_Toc47486253)

[3.7. Màn hình About Us (Custom View) 52](#_Toc47486254)

[CHƯƠNG 4 – KẾT QUẢ ĐẠT ĐƯỢC 54](#_Toc47486255)

[4.1. Kết quả đạt được 54](#_Toc47486256)

[4.2. Kết luận 54](#_Toc47486257)

# CHƯƠNG 1 – MỞ ĐẦU

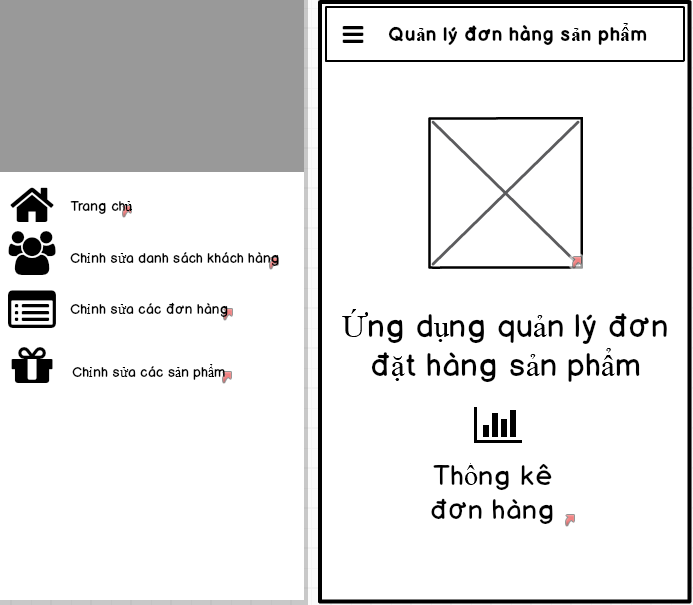
* 1. **Giới thiệu môn học và nhóm thực hiện**
* Môn di động 2 là môn nâng cao của di động 1, áp dụng các kiến thức từ di động 1 (android) sang và học hỏi thêm các kiến thức mới để áp dụng vào cho các dự án thực tế.
* Kiến thức mới: Splash Screen, Navigation Drawer, Fragment, Text Input Layout, Animation, Đa ngôn ngữ, Custom Drawing, Bar Chart,…
* Thành viên thực hiện:
  1. Trần Bình Văn (Leader)
  2. Lê Văn Thắng
  3. **Mô tả ứng dụng**
* Tên ứng dụng: Quản lý đơn đặt hàng
* Không cần kết nối mạng
* Android 4.0+
* Thiết lập và quản lý dữ liệu của: Khách hàng, Sản phẩm, Đơn hàng. Dựa vào Bar Chart có thể quản lý tốt hơn các đơn hàng.
* Hỗ trợ 2 ngôn ngữ: Tiếng Việt và Tiếng Anh
  1. **Ứng dụng trong cuộc sống**
* Với việc nhu cầu mua sắm online diễn ra rất nhiều trong thời gian vài năm trở lại đây, ứng dụng được đưa ra để chủ shop có thể quản lý các đơn hàng mà mình đã thực hiện trong một quãng thời gian dài ra sao, tỉ lệ các đơn hàng hàng bị hủy, hoàn thành, đang giao như thế nào. Qua đó có thể tìm giải pháp thích hợp để phát triển việc kinh doanh online hơn.

# CHƯƠNG 2 – PHÂN TÍCH CẤU TRÚC HỆ THỐNG

## 2.1. Phân tích thiết kế hệ thống

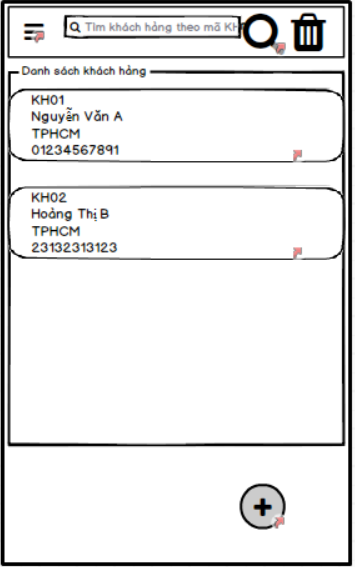
**-** Tổng quan: Đây là ứng dụng dùng để theo dõi, quản lý danh sách sản phẩm, khách hàng, các đơn hàng một cách thuận tiện, nhanh chóng, không cần có internet.

**Yêu cầu chức năng (Home):**



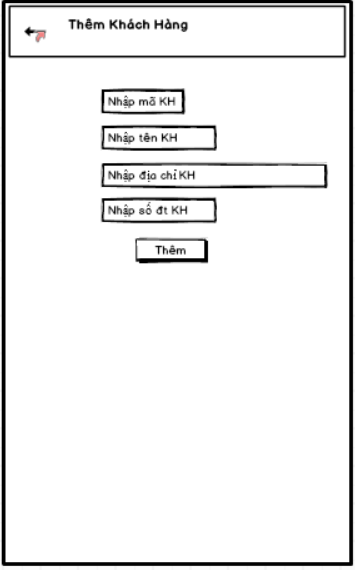
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình home | Màn hình chính có: - 1 button - 1 TextView - 1 background | Thiết kế | Hoàn thành |
| Button “Thống kê đơn hàng” | Chuyển sang giao diện “Thống kê đơn hàng” | Click | Hoàn thành |
| Navigation drawer menu “Trang chủ” | Chuyển sang giao diện “Trang chủ” | Click |  |
| Navigation drawer menu “Chỉnh sửa ds khách hàng” | Chuyển sang giao diện “Danh sách khách hàng” | Click |  |
| Navigation drawer menu “Chỉnh sửa các đơn hàng” | Chuyển sang giao diện “Danh sách đơn hàng” | Click |  |
| Navigation drawer menu “Chỉnh sửa các sản phẩm” | Chuyển sang giao diện “Danh sách sản phẩm” | Click |  |

**Yêu cầu chức năng (Trang danh sách khách hàng):**



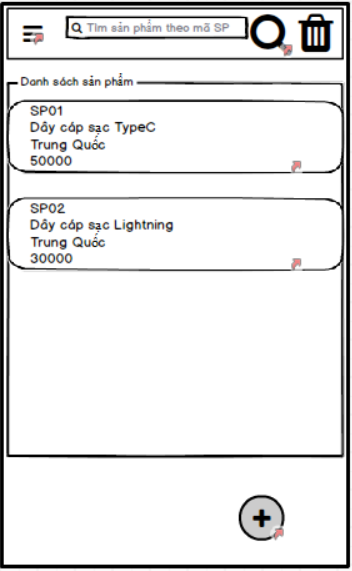
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình khách hàng | Màn hình chính có: - 1 TextView - 3 Button  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Khách Hàng” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả khách hàng khỏi bảng danh sách | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị khách hàng ở RecyclerView theo điều kiện tìm kiếm | Click | Hoàn thành |
| RecyclerView | Hiển thị danh sách khách hàng và khi click vào item thì chuyển sang trang chi tiết khách hàng để SỬA theo từng item được click | Click | Hoàn thành |

**Yêu cầu chức năng (Trang thêm khách hàng)**



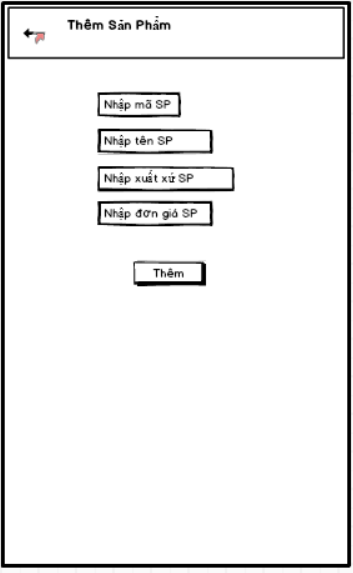
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Khách Hàng | Màn hình chính có: - 4 Edit Text - 2 Button | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách Khách Hàng | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Khách Hàng | Click | Hoàn thành |

**Yêu cầu chức năng (Trang danh sách sản phẩm):**



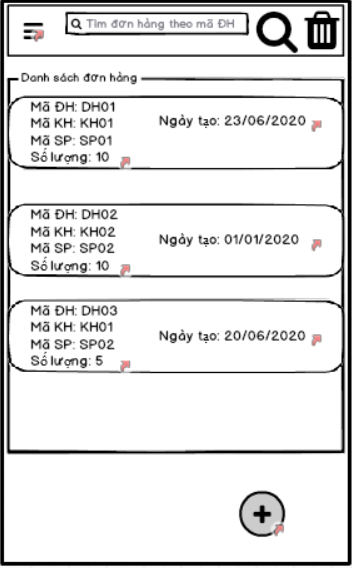
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình sản phẩm | Màn hình chính có: - 1 TextView - 3 Button  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Sản Phẩm” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả sản phẩm khỏi bảng danh sách sản phẩm | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách sản phẩm ở RecyclerView theo điều kiện tìm kiếm | Click | Hoàn thành |
| RecyclerView | Hiển thị danh sách sản phẩm và khi click vào item thì chuyển sang trang chi tiết sản phẩm để SỬA theo từng item được click | Click | Hoàn thành |

**Yêu cầu chức năng (Trang thêm sản phẩm)**



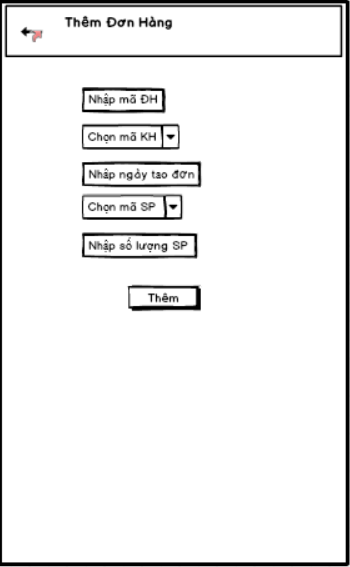
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Sản Phẩm | Màn hình chính có: - 4 Edit Text - 2 Button | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách Sản Phẩm | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Sản Phẩm | Click | Hoàn thành |

**Yêu cầu chức năng (Trang danh sách đơn hàng):**



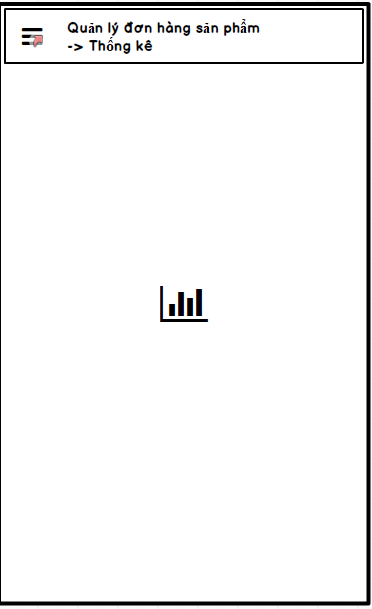
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình đơn hàng | Màn hình chính có: - 1 TextView - 3 Button  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Đơn Hàng” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả ĐH trong RecyclerView List | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách đơn hàng ở RecyclerView theo điều kiện tìm kiếm | Click | Hoàn thành |
| RecyclerView | Hiển thị danh sách đơn hàng và khi click vào item thì chuyển sang trang chi tiết đơn hàng để SỬA theo từng item được click | Click | Hoàn thành |

**Yêu cầu chức năng (Trang thêm đơn hàng)**



|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Đơn Hàng | Màn hình chính có: - 4 Edit Text - 2 Button | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách Đơn Hàng | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Đơn Hàng | Click | Hoàn thành |

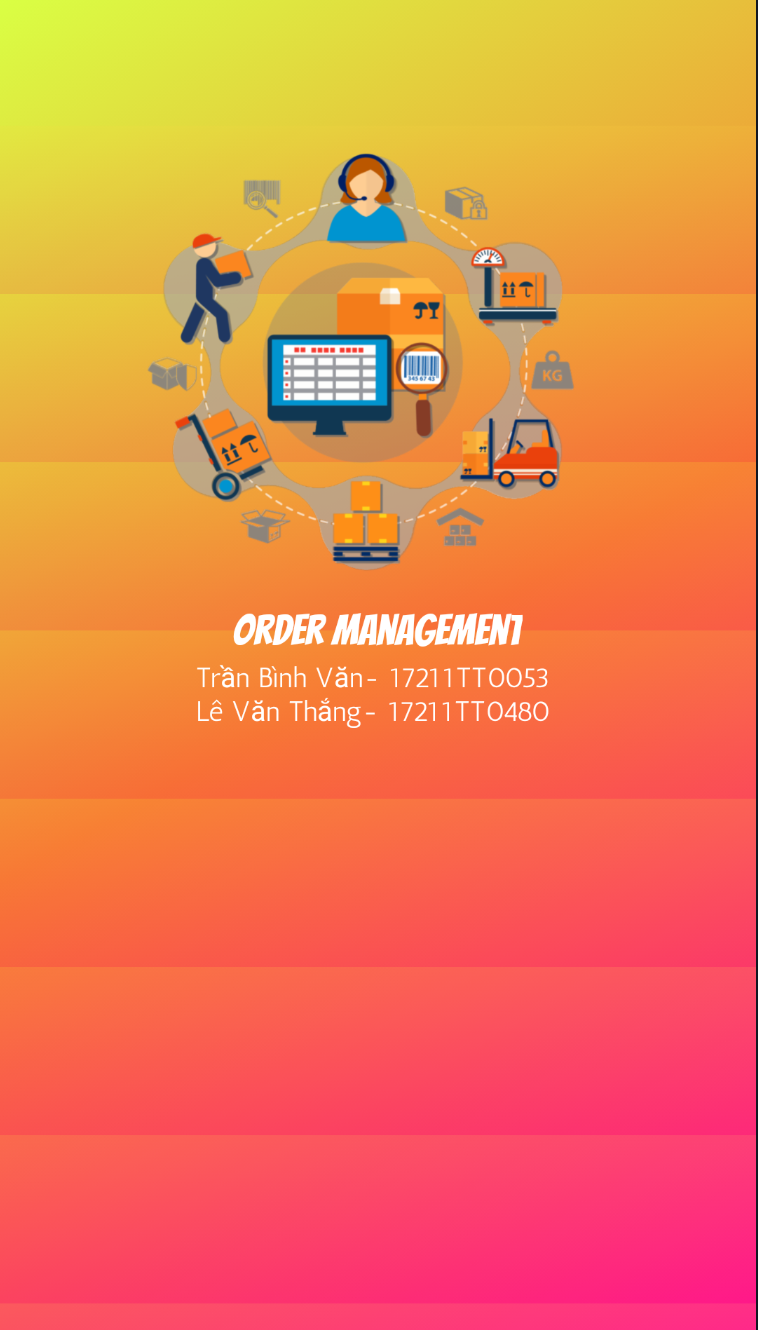
**Yêu cầu chức năng (Trang thống kê đơn hàng):**



|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình sản phẩm | - 1 Chart | Thiết kế | Hoàn thành |
| Chart | - Thống kê theo đơn hàng | Click | Hoàn thành |

## 2.2. Thiết kế hệ thống

* **Tổng quan:** Tài liệu mockup do nhóm trưởng vẽ, phân công công việc theo từng form, từng chức năng giúp dễ dàng bố trí việc làm và phân bố công việc hợp lý, phát triển thành sản phẩm hoàn chỉnh theo thời gian. Mô tả chi tiết công việc cho từng màn hình
* **Yêu cầu chức năng:** 
  1. **Màn hình splash (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.a. Màn hình Splash Screen*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình splash | - Hiển thị logo, text view, animation | Sau 5s chuyển qua màn hình chính | Hoàn thành |

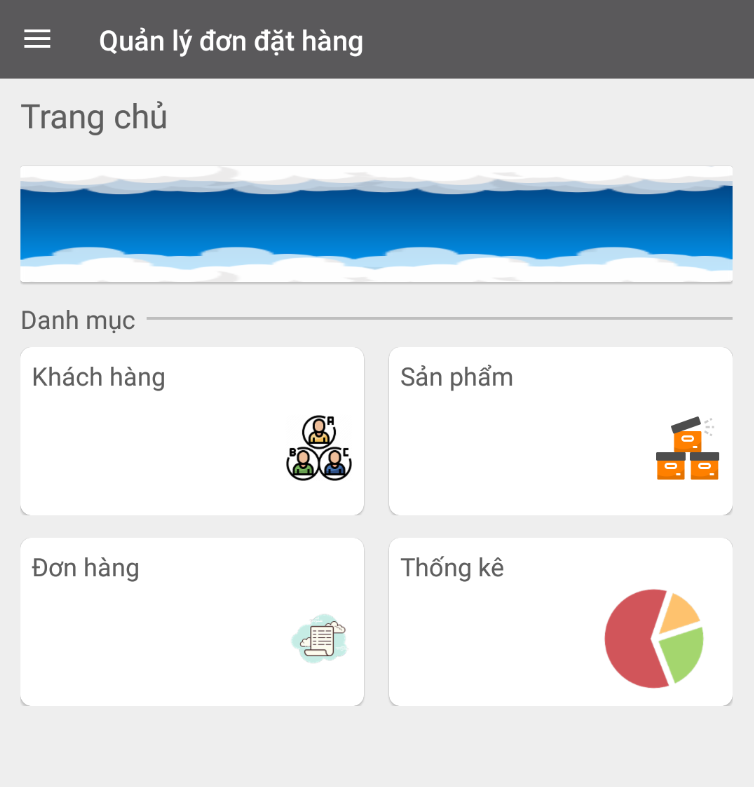
* + - **Làm thế nào**

+ Chạy ứng dụng

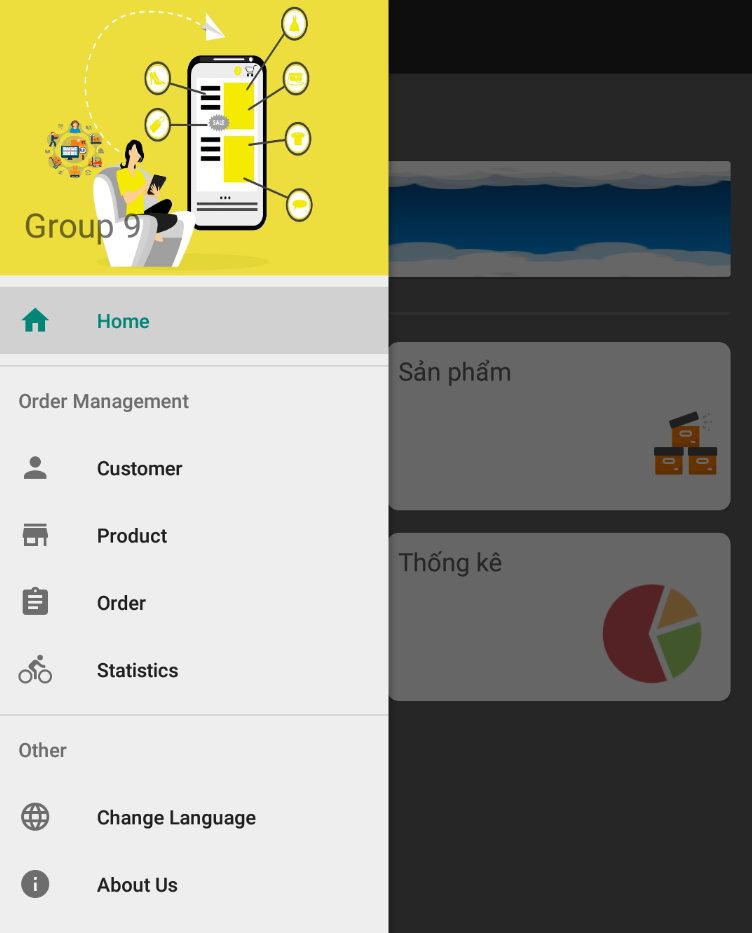
+ Chương trình sẽ chạy màn hình Splash

+ Sau 5s tự động chuyển qua màn hình home

* 1. **Màn hình home (Lê Văn Thắng)**
     + **Giao diện người dùng**



*Hình 2.2.b. Màn hình Home*



*Hình 2.2.b. Navigation Drawer*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình home | Màn hình chính có: - 4 Card View - 4 Text View - 1 background  -Navigation Drawer | Thiết kế | Hoàn thành |
| CardView “Thống kê đơn hàng” | Chuyển sang giao diện “Thống kê đơn hàng” | Click | Hoàn thành |
| CardView “Khách hàng” | Chuyển sang giao diện “Danh sách khách hàng” | Click | Hoàn thành |
| CardView “Sản phẩm” | Chuyển sang giao diện “Danh sách sản phẩm” | Click | Hoàn thành |
| CardView “Đơn hàng” | Chuyển sang giao diện “Danh sách đơn hàng” | Click | Hoàn thành |
| Navigation drawer menu “Trang chủ” | Chuyển sang giao diện “Trang chủ” | Click | Hoàn thành |
| Navigation drawer menu “DS khách hàng” | Chuyển sang giao diện “Danh sách khách hàng” | Click | Hoàn thành |
| Navigation drawer menu “DS đơn hàng” | Chuyển sang giao diện “Danh sách đơn hàng” | Click | Hoàn thành |
| Navigation drawer menu “DS sản phẩm” | Chuyển sang giao diện “Danh sách sản phẩm” | Click | Hoàn thành |
| Navigation drawer menu “Thống kê” | Chuyển sang giao diện “Thống kê bằng chart” | Click | Hoàn thành |
| Navigation drawer menu “Change Language” | Chuyển sang giao diện “Đổi ngôn ngữ” | Click | Hoàn thành |
| Navigation drawer menu “About Us” | Chuyển sang giao diện “Custom View” | Click | Hoàn thành |

* + - **Làm thế nào**

+ CardView Khách Hàng:

-Click cardview khách hàng

-Ứng dụng chuyển qua giao diện danh sách khách hàng

+ CardView Thống kê:

-Click cardview thống kê

-Ứng dụng chuyển qua giao diện thống kê đơn hàng

+ CardView Sản Phẩm:

-Click cardview sản phẩm

-Ứng dụng chuyển qua giao diện danh sách sản phẩm

+ CardView Đơn hàng:

-Click cardview đơn hàng

-Ứng dụng chuyển qua giao diện danh sách đơn hàng

+Navigation Drawer menu – CLICK

- Navigation drawer menu “Trang chủ”

- Navigation drawer menu “Khách hàng”

- Navigation drawer menu “Sản phẩm”

- Navigation drawer menu “Đơn hàng”

- Navigation drawer menu “Thống kê”

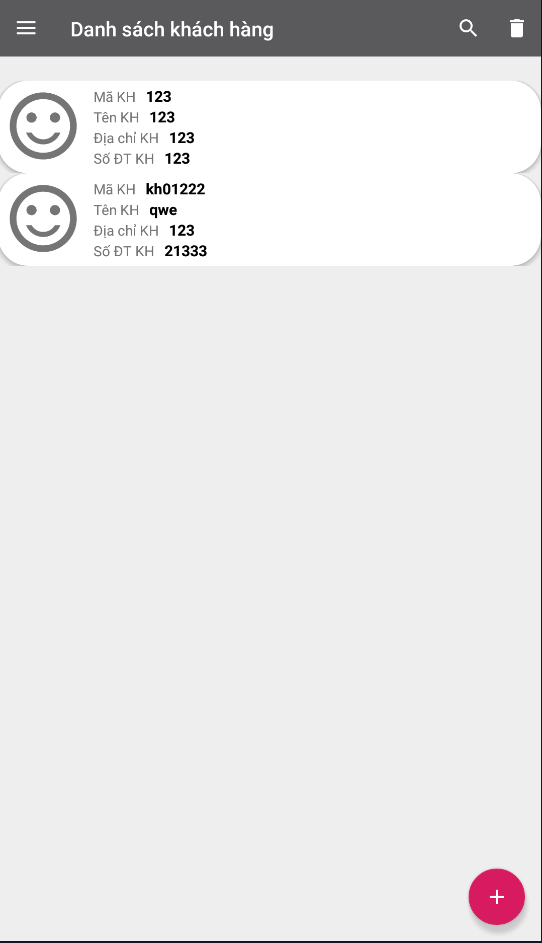
- Navigation drawer menu “Đổi ngôn ngữ”

- Navigation drawer menu “About Us”

* 1. **Màn hình danh sách khách hàng ( Lê Văn Thắng)**
     + **Giao diện người dùng**



*Hình 2.2.c. Danh sách khách hàng khi không có dữ liệu*



*Hình 2.2.c. Danh sách khách hàng khi có dữ liệu*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình sản phẩm | Màn hình chính có: - 9 TextViews - 4 Buttons  - 2 Images  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Khách Hàng” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả sản phẩm khỏi bảng danh sách khách hàng | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách khách hàng ở RecyclerView theo điều kiện tìm kiếm | Click | Hoàn thành |
| Button Swipe | Kéo từ đầu danh sách xuống để refresh Database | Swipe | Hoàn thành |
| RecyclerView | Hiển thị danh sách khách hàng và khi click vào item thì chuyển sang trang chi tiết khách hàng để sửa theo từng item được click | Click | Hoàn thành |

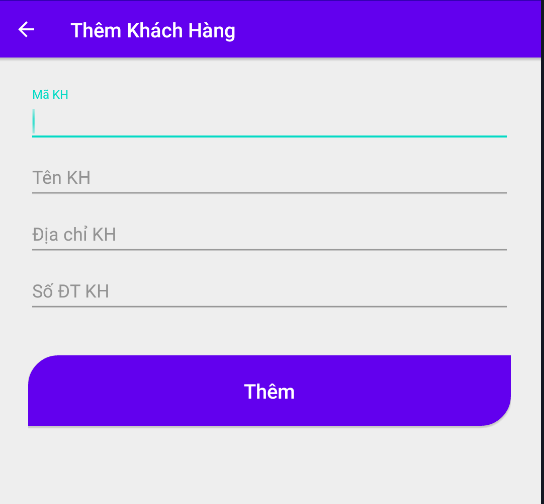
* + - **Làm thế nào**

- Click floating button “+” để vào màn hình “Thêm khách hàng”

- Click button Tìm kiếm trên thanh action bar để tìm khách hàng theo mã khách hàng

- Click button thùng rác trên thanh action bar để xóa tất cả khách hàng

* 1. **Màn hình thêm khách hàng (Lê Văn Thắng)**
     + **Giao diện người dùng**



*Hình 2.2.d. Thêm khách hàng*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm khách hàng | Màn hình chính có: - 4 Edit Text (Text Input Layout) - 2 Button | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách khách hàng | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách khách hàng | Click | Hoàn thành |
| 4 Edit Text (Text Input Layout) | Nhập thông tin cần có cho khách hàng, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |

* + - **Làm thế nào**

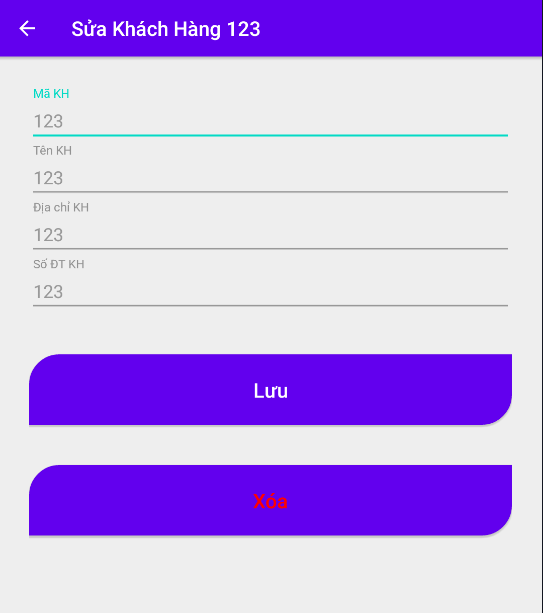
- Nhấn floating button + ở trang danh sách khách hàng để vào trang thêm khách hàng

- Nhập đầy đủ thông tin 4 ô edit text

- Nhấn button Thêm để thêm khách hàng vào

- Nhấn nút 🡨 để quay lại trang danh sách khách hàng

* 1. **Màn hình sửa khách hàng (Lê Văn Thắng)**
     + **Giao diện người dùng**



*Hình 2.2.e. Sửa khách hàng*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Sửa khách hàng | Màn hình chính có: - 4 Edit Text (Text Input Layout) - 3 Button | Thiết kế | Hoàn thành |
| Button “Lưu” | Lưu dữ liệu đã sửa vào RecyclerView ở trang Danh Sách khách hàng | Click | Hoàn thành |
| Button “Xóa” | Xóa dữ liệu đang chọn từ RecyclerView ở trang Danh Sách khách hàng | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách khách hàng | Click | Hoàn thành |
| 4 Edit Text (Text Input Layout) | Nhập thông tin cần có cho khách hàng, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |

* + - **Làm thế nào**

**-** Ở trang danh sách sản phẩm, click vào 1 khách hàng cần sửa để vào màn hình Sửa khách hàng

- Điền các thông tin cần sửa

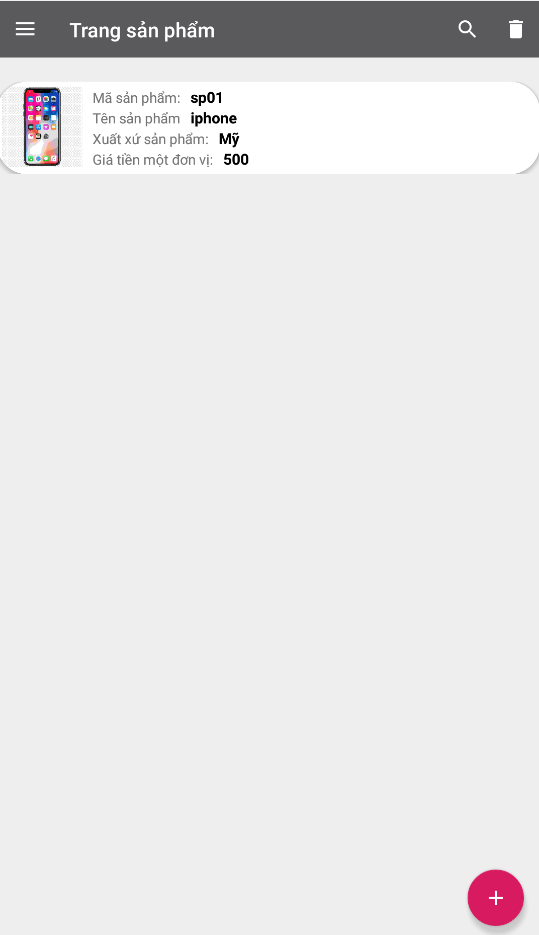
- Nhấn nút Lưu để lưu dữ liệu vừa sửa, hoặc nhấn nút Xóa để xóa đi khách hàng đang chọn

- Nhấn nút 🡨 để quay lại trang danh sách khách hàng

* 1. **Màn hình danh sách sản phẩm (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.f Trang danh sách sản phẩm, khi không có dữ liệu*



*Hình 2.2.f Trang danh sách sản phẩm, khi có dữ liệu*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình sản phẩm | Màn hình chính có: - 9 TextViews - 4 Buttons  - 2 Images  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Sản Phẩm” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả sản phẩm khỏi bảng danh sách sản phẩm | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách sản phẩm ở RecyclerView theo điều kiện tìm kiếm | Click | Hoàn thành |
| Button Swipe | Kéo từ đầu danh sách xuống để refresh Database | Swipe | Hoàn thành |
| RecyclerView | Hiển thị danh sách sản phẩm và khi click vào item thì chuyển sang trang chi tiết sản phẩm để sửa theo từng item được click | Click | Hoàn thành |

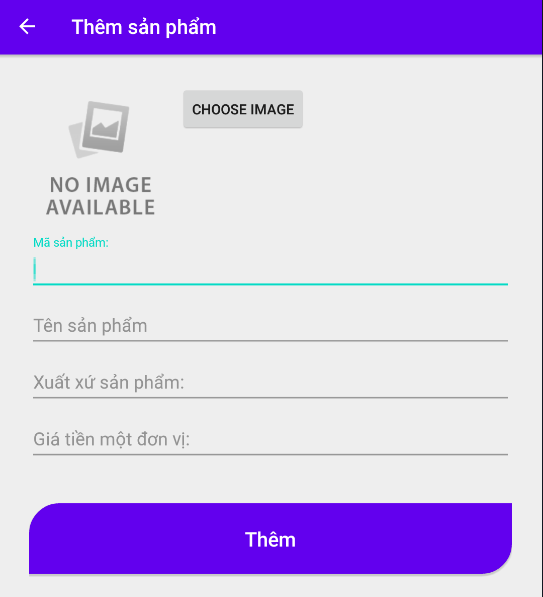
* + - **Làm thế nào**

- Click floating button “+” để vào màn hình “Thêm sản phẩm”

- Click button Tìm kiếm trên thanh action bar để tìm sản phẩm theo mã sản phẩm

- Click button thùng rác trên thanh action bar để xóa tất cả sản phẩm

* 1. **Màn hình thêm sản phẩm (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.g. Màn hình trang thêm sản phẩm*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Sản Phẩm | Màn hình chính có: - 4 Edit Text (Text Input Layout) - 3 Button  -1 ImageView | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách Sản Phẩm | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Sản Phẩm | Click | Hoàn thành |
| Button “Choose Image” | Chọn hình ảnh sản phẩm để upload lên DB, hiển thị ở trang danh sách sản phẩm | Click | Hoàn thành |
| 4 Edit Text (Text Input Layout) | Nhập thông tin cần có cho sản phẩm, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |

* + - **Làm thế nào**

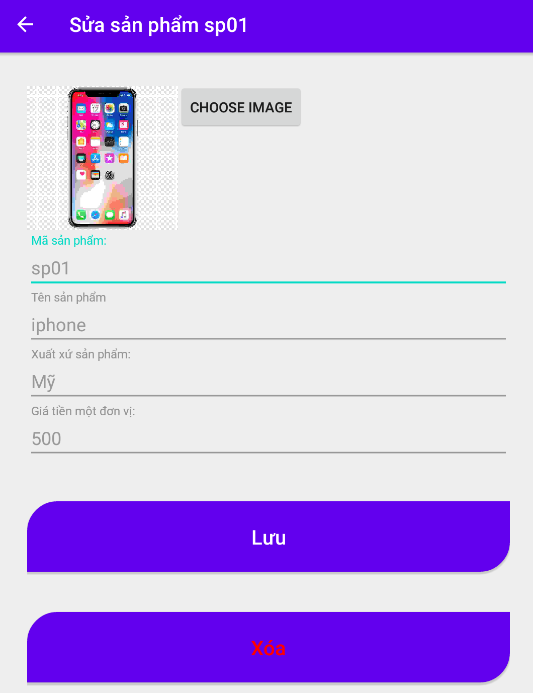
- Nhấn floating button + ở trang danh sách sản phẩm để vào trang thêm sản phẩm

- Nhập đầy đủ thông tin 4 ô edit text, có thể up hình hoặc không

- Nhấn button Thêm để thêm sản phẩm vào

- Nhấn nút 🡨 để quay lại trang danh sách sản phẩm

* 1. **Màn hình sửa sản phẩm (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.h. Màn hình trang sửa sản phẩm*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Sửa Sản Phẩm | Màn hình chính có: - 4 Edit Text (Text Input Layout) - 4 Button  -1 ImageView | Thiết kế | Hoàn thành |
| Button “Lưu” | Lưu dữ liệu đã sửa vào RecyclerView ở trang Danh Sách Sản Phẩm | Click | Hoàn thành |
| Button “Xóa” | Xóa dữ liệu đang chọn từ RecyclerView ở trang Danh Sách Sản Phẩm | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Sản Phẩm | Click | Hoàn thành |
| Button “Choose Image” | Chọn hình ảnh sản phẩm để upload lên DB, hiển thị ở trang danh sách sản phẩm | Click | Hoàn thành |
| 4 Edit Text (Text Input Layout) | Nhập thông tin cần có cho sản phẩm, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |

* + - **Làm thế nào**

**-** Ở trang danh sách sản phẩm, click vào 1 sản phẩm cần sửa để vào màn hình Sửa sản phẩm

- Điền các thông tin cần sửa

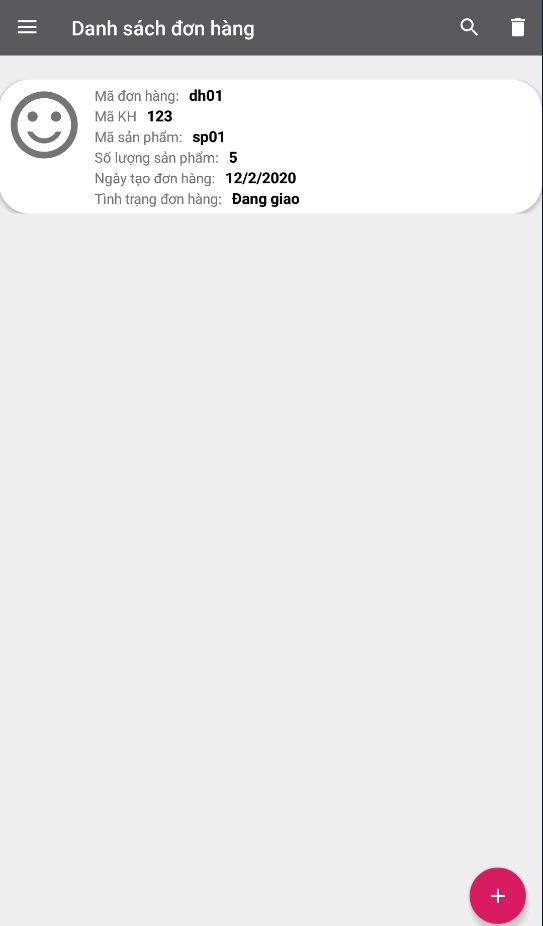
- Nhấn nút Lưu để lưu dữ liệu vừa sửa, hoặc nhấn nút Xóa để xóa đi sản phẩm đang chọn

- Nhấn nút 🡨 để quay lại trang danh sách sản phẩm

* 1. **Màn hình danh sách đơn hàng (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.i. Màn hình trang danh sách đơn hàng khi không có đơn hàng*



*Hình 2.2.i. Màn hình trang danh sách đơn hàng khi có đơn hàng*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình đơn hàng | Màn hình chính có: - 13 TextViews - 4 Buttons  - 2 Images  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm đơn hàng” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả đơn hàng khỏi bảng danh sách sản phẩm | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách đơn hàng ở RecyclerView theo điều kiện tìm kiếm | Click | Hoàn thành |
| Button Swipe | Kéo từ đầu danh sách xuống để refresh Database | Swipe | Hoàn thành |
| RecyclerView | Hiển thị danh sách đơn hàng và khi click vào item thì chuyển sang trang chi tiết đơn hàng sửa theo từng item được click | Click | Hoàn thành |

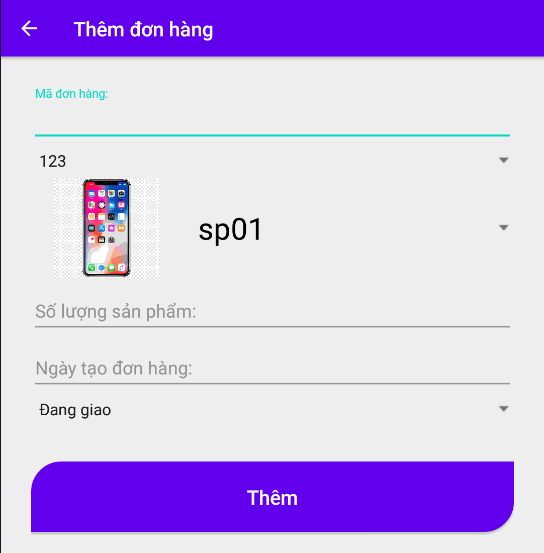
* + - **Làm thế nào**

- Click floating button “+” để vào màn hình “Thêm đơn hàng”

- Click button Tìm kiếm trên thanh action bar để tìm đơn hàng theo mã đơn hàng

- Click button thùng rác trên thanh action bar để xóa tất cả đơn hàng

* 1. **Màn hình thêm đơn hàng (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.j. Màn hình trang thêm đơn hàng*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Đơn Hàng | Màn hình chính có: - 3 Edit Text (Text Input Layout)  - 3 Spinner - 3 Button  -1 ImageView | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách Đơn Hàng | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Đơn Hàng | Click | Hoàn thành |
| Button “Choose Image” | Chọn hình ảnh sản phẩm để upload lên DB, hiển thị ở trang danh sách đơn hàng | Click | Hoàn thành |
| 3 Edit Text (Text Input Layout) | Nhập thông tin cần có cho đơn hàng, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |
| Spinner Mã khách hàng | Hiển thị danh sách mã khách hàng đã tạo từ bảng khách hàng | Select | Hoàn thành |
| Spinner Mã sản phẩm | Hiển thị danh sách mã sản phẩm đã tạo từ bảng sản phẩm | Select | Hoàn thành |
| Spinner Tình trạng đơn hàng | Hiển thị các danh sách các trạng thái đơn hàng đã được tạo sẵn | Select | Hoàn thành |

* + - **Làm thế nào**

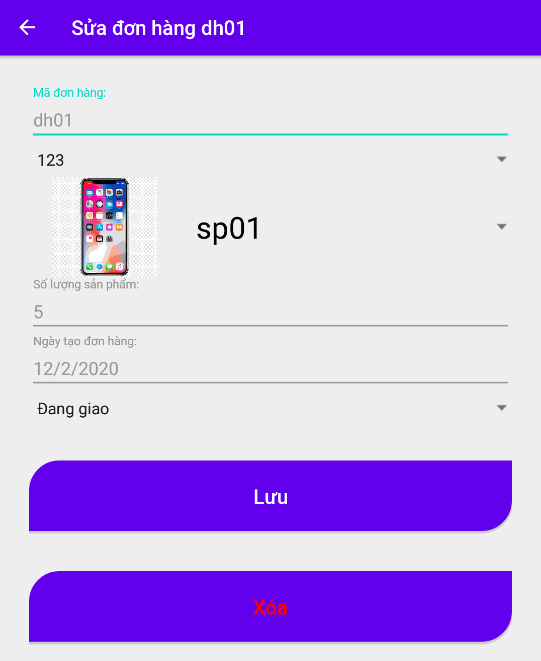
- Nhấn floating button + ở trang danh sách đơn hàng để vào trang thêm đơn hàng

- Nhập đầy đủ thông tin 3 ô edit text, chọn 3 spinner tương ứng theo mong muốn

- Nhấn button Thêm để thêm đơn hàng vào

- Nhấn nút 🡨 để quay lại trang danh sách đơn hàng

* 1. **Màn hình sửa đơn hàng (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.k. Màn hình trang sửa đơn hàng*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Sửa Đơn hàng | Màn hình chính có: - 3 Edit Text (Text Input Layout)  - 3 Spinner - 3 Button | Thiết kế | Hoàn thành |
| Button “Lưu” | Lưu dữ liệu đã sửa vào RecyclerView ở trang Danh Sách Đơn Hàng | Click | Hoàn thành |
| Button “Xóa” | Xóa dữ liệu đang chọn từ RecyclerView ở trang Danh Sách Đơn Hàng | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Sản Phẩm | Click | Hoàn thành |
| 3 Edit Text (Text Input Layout) | Nhập thông tin cần có cho đơn hàng, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |
| 3 Spinner | Chọn trạng thái muốn sửa | Select | Hoàn thành |

* + - **Làm thế nào**

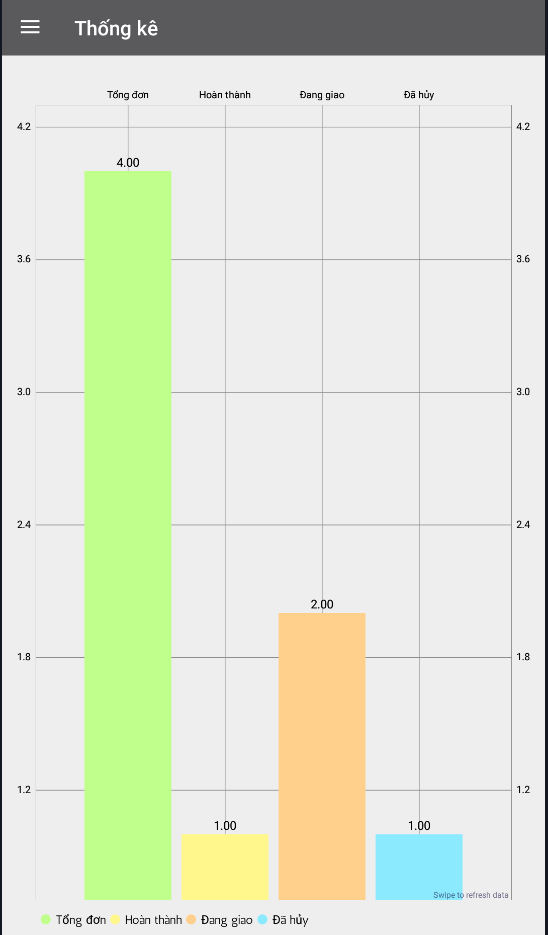
**-** Ở trang danh sách đơn hàng, click vào 1 đơn hàng cần sửa để vào màn hình Sửa đơn hàng

- Điền các thông tin cần sửa

- Nhấn nút Lưu để lưu dữ liệu vừa sửa, hoặc nhấn nút Xóa để xóa đi đơn hàng đang chọn

- Nhấn nút 🡨 để quay lại trang danh sách đơn hàng

* 1. **Màn hình thống kê đơn hàng (Trần Bình Văn)**
     + **Giao diện người dùng**



*Hình 2.2.l. Màn hình trang sửa đơn hàng*

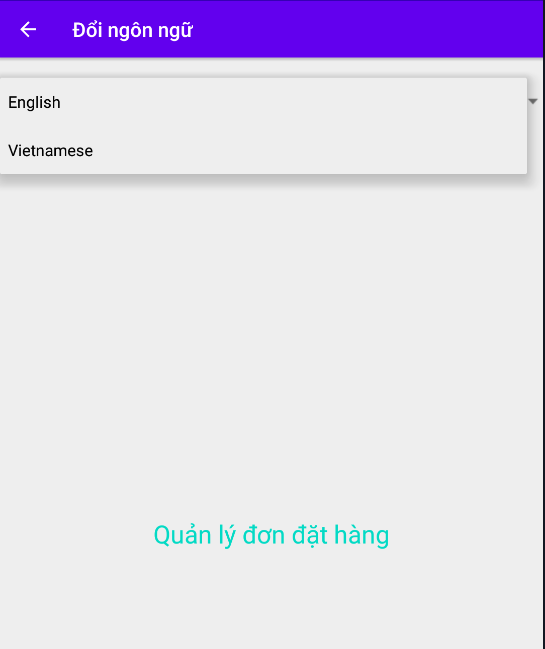
* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thống kê đơn hàng | Màn hình chính có: - 1 Bar Chart | Thiết kế | Hoàn thành |
| Swipe Button | -Kéo từ trên xuống để refresh bảng thống kê khi vừa thay đổi dữ liệu ở bảng đơn hàng | Swipe | Hoàn thành |

* + - **Làm thế nào**

- Nhận dữ liệu đơn hàng và tự cập nhật theo thời gian thực mà người dùng sử dụng

* 1. **Màn hình đổi ngôn ngữ (Lê Văn Thắng)**
     + **Giao diện người dùng**



* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình đổi ngôn ngữ | Màn hình chính có: - 1 Spinner  -1 TextView  -1 Button | Thiết kế | Hoàn thành |
| Button “🡨” | -Quay lại trang chủ | Click | Hoàn thành |
| Spinner | Hiển thị danh sách ngôn ngữ được tạo | Select | Hoàn thành |

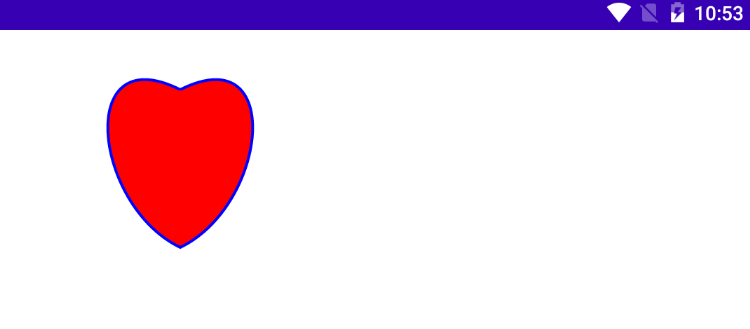
* + - **Làm thế nào**

-Từ NavigationDrawer, click chọn đổi ngôn ngữ

- Chọn ngôn ngữ sau nó bấm 🡨 quay về trang chủ

- Từ NavigationDrawer, click chọn trang chủ 1 lần nữa để thấy giao diện đổi sang ngôn ngữ đã chọn

* 1. **Màn hình About Us (Lê Văn Thắng)**
     + **Giao diện người dùng**



* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình About Us | - Hiển thị custom view (custom drawing) | View | Hoàn thành |

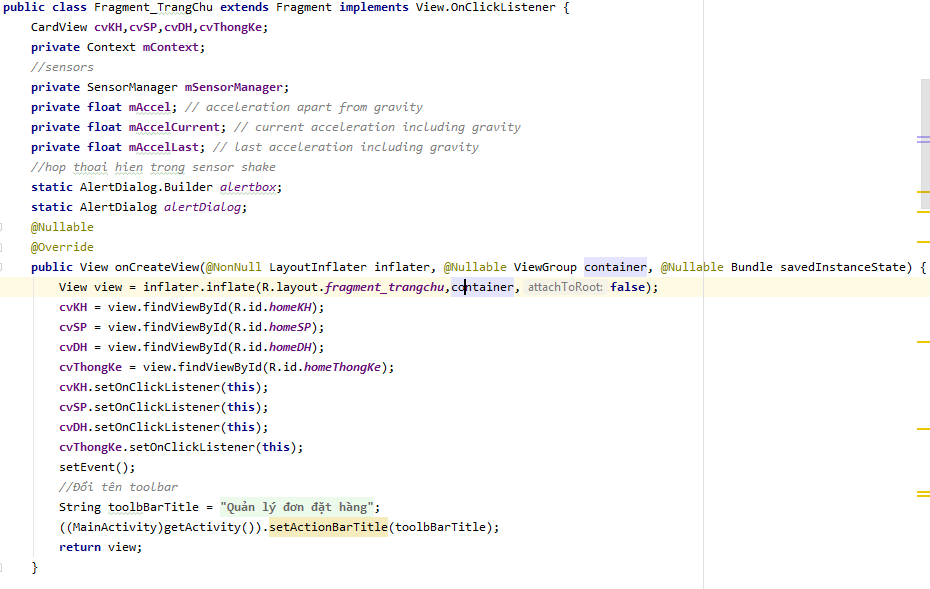
* + - **Làm thế nào**

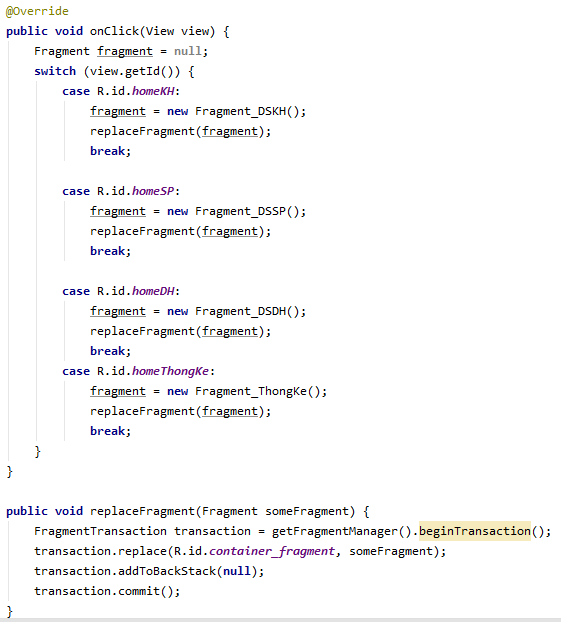
- Từ trang chính click navigation drawer chọn About Us để vào xem được Custom view

# CHƯƠNG 3 – CÀI ĐẶT

## 3.1. Màn hình home

- Khai báo fragment Home để có thể click nhảy vào các fragment khác





- Navigation Drawer xử lý ở Main Activity

**public class** MainActivity **extends** AppCompatActivity **implements** NavigationView.OnNavigationItemSelectedListener {  
 DrawerLayout **drawerLayout**;  
 ActionBarDrawerToggle **actionBarDrawerToggle**;  
 Toolbar **toolbar**;  
 TextView **toolbarTitle**;  
 NavigationView **navigationView**;  
 FragmentManager **fragmentManager**;  
 FragmentTransaction **fragmentTransaction**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 *//Đổi tên toolbar* **toolbar** = findViewById(R.id.***toolBar***);  
 **toolbarTitle** = **toolbar**.findViewById(R.id.***toolBar\_title***);  
 setSupportActionBar(**toolbar**);  
 **drawerLayout** = findViewById(R.id.***drawerLayoutKH***);  
 **navigationView** = findViewById(R.id.***nav\_view***);  
 **navigationView**.setNavigationItemSelectedListener(**this**);  
 **actionBarDrawerToggle** = **new** ActionBarDrawerToggle(**this**,**drawerLayout**,**toolbar**,R.string.***openBar***,R.string.***closeBar***);  
 **drawerLayout**.addDrawerListener(**actionBarDrawerToggle**);  
 **actionBarDrawerToggle**.setDrawerIndicatorEnabled(**true**);  
 **actionBarDrawerToggle**.syncState();  
 *//Load default fragment* **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.add(R.id.***container\_fragment***,**new** Fragment\_TrangChu());  
 **fragmentTransaction**.commit();  
 }  
  
 @Override  
 **public boolean** onNavigationItemSelected(@NonNull MenuItem menuItem) {  
 *//Click menuItem của Navigation thì đóng Nav lại* **drawerLayout**.closeDrawer(GravityCompat.***START***);  
 **if**(menuItem.getItemId() == R.id.***nav\_home***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_TrangChu());  
 **fragmentTransaction**.commit();  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_kh***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_DSKH());  
 **fragmentTransaction**.commit();  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_sp***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_DSSP());  
 **fragmentTransaction**.commit();  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_dh***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_DSDH());  
 **fragmentTransaction**.commit();  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_thongke***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_ThongKe());  
 **fragmentTransaction**.commit();  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_doingonngu***) {  
 Intent myIntent = **new** Intent(MainActivity.**this**, ActivityDoiNgonNgu.**class**);  
 MainActivity.**this**.startActivity(myIntent);  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_aboutus***) {  
 Intent myIntent = **new** Intent(MainActivity.**this**, ActivityAboutUs.**class**);  
 MainActivity.**this**.startActivity(myIntent);  
 }  
 **return true**;  
 }  
  
 *//Đổi tên toolbar* **public void** setActionBarTitle(String title) {  
 **toolbarTitle**.setText(title);  
 }  
}

## 3.2. Màn hình khách hàng

- Sử dụng fragment danh sách khách hàng

**public class** Fragment\_DSKH **extends** Fragment {  
 RecyclerView **recyclerView**;  
 FloatingActionButton **add\_button**;  
 ImageView **img\_empty**;  
 TextView **khongDuLieu**;  
 ArrayList<KhachHang> **data** = **new** ArrayList<>();  
 DBDonDatHang **myDB**;  
 CustomAdapterKH **customAdapterKH**;  
 SwipeRefreshLayout **swipeRefreshLayout**;  
 CoordinatorLayout **coLayoutKH**;  
 NavigationView **navigationView**;  
 **private** Context **mContext**;  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setHasOptionsMenu(**true**);*//Hiện option menu (Delete all + Search)* }  
  
 @Override  
 **public void** onAttach(Context context) {  
 **super**.onAttach(context);  
 **mContext**=context;  
 }  
 @Nullable  
 @Override  
 **public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***fragment\_dskh***,container,**false**);  
 **recyclerView** = view.findViewById(R.id.***recyclerViewKH***);  
 **add\_button** = view.findViewById(R.id.***add\_button\_kh***);  
 **img\_empty** = view.findViewById(R.id.***image\_empty***);  
 **khongDuLieu** = view.findViewById(R.id.***no\_data***);  
 **coLayoutKH** = view.findViewById(R.id.***coordinatorLayoutKH***);  
 **swipeRefreshLayout** = view.findViewById(R.id.***swipeToRefreshKH***);  
 **navigationView** = view.findViewById(R.id.***nav\_view***);  
 *//Đổi tên toolbar* String toolbBarTitle = **mContext**.getResources().getString(R.string.***trangKH***);  
 ((MainActivity)getActivity()).setActionBarTitle(toolbBarTitle);  
 setEvent();  
 hienThiDatabase();  
 **return** view;  
 }  
  
 **private void** setEvent() {  
 **add\_button**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(getActivity(), ActivityThemKH.**class**);  
 startActivity(intent);  
 }  
 });  
  
 *//Kéo trái, phải item để xóa + snackbar undo* **new** ItemTouchHelper((**new** ItemTouchHelper.SimpleCallback(0,  
 ItemTouchHelper.***LEFT*** | ItemTouchHelper.***RIGHT***) {  
 @Override  
 **public boolean** onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {  
 **return false**;  
 }  
  
 @Override  
 **public void** onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, **int** direction) {  
 **final int** position = viewHolder.getAdapterPosition();  
 **final** KhachHang recoverItem = **data**.get(position);  
 *//Lưu giữ id position  
 //final int checkingId = data.get(position).getSttKH();* **data**.remove(position);  
 **customAdapterKH**.notifyItemRemoved(position);  
 *//long result = myDB.XoaItemTheoIDKH(checkingId);* **final** Snackbar snackbar = Snackbar.*make*(**coLayoutKH**, **"Đã xóa khách hàng"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Hoàn Tác"**, **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **data**.add(position, recoverItem);  
 **customAdapterKH**.notifyItemInserted(position);  
 **customAdapterKH**.notifyDataSetChanged();  
 **myDB**.ThemKH(recoverItem);  
 }  
  
 }).addCallback(**new** Snackbar.Callback() {  
 @Override  
 **public void** onDismissed(Snackbar snackbar, **int** dismissType) {  
 **super**.onDismissed(snackbar, dismissType);  
 **if**(dismissType == ***DISMISS\_EVENT\_TIMEOUT*** || dismissType == ***DISMISS\_EVENT\_SWIPE*** || dismissType == ***DISMISS\_EVENT\_CONSECUTIVE*** || dismissType == ***DISMISS\_EVENT\_MANUAL*** || dismissType == ***DISMISS\_EVENT\_ACTION***) {  
 **myDB**.XoaMotItemKH(recoverItem);  
 }  
 }  
 });  
 snackbar.show();  
 }  
 })) .attachToRecyclerView(**recyclerView**);  
  
 *//Kéo refresh list* **swipeRefreshLayout**.setOnRefreshListener(**new** SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 **public void** onRefresh() {  
 hienThiDatabase();  
 **swipeRefreshLayout**.setRefreshing(**false**);  
 }  
 });  
 }  
 **public void** hienThiDatabase(){  
 **myDB** = **new** DBDonDatHang(**mContext**);  
 **data** = **myDB**.listKhachHangs();  
 **customAdapterKH** = **new** CustomAdapterKH(getActivity(),**mContext**, **data**);  
 **recyclerView**.setAdapter(**customAdapterKH**);  
 **recyclerView**.setLayoutManager(**new** LinearLayoutManager(**mContext**));  
 **if**(**data**.size() > 0){  
 **recyclerView**.setVisibility(View.***VISIBLE***);  
 **img\_empty**.setVisibility(View.***GONE***);  
 **khongDuLieu**.setVisibility(View.***GONE***);  
 }**else** {  
 **recyclerView**.setVisibility(View.***GONE***);  
 **img\_empty**.setVisibility(View.***VISIBLE***);  
 **khongDuLieu**.setVisibility(View.***VISIBLE***);  
 String toastMessage = **mContext**.getResources().getString(R.string.***noData***);  
 Toast.*makeText*(**mContext**,toastMessage, Toast.***LENGTH\_SHORT***).show();  
 }  
 **customAdapterKH**.notifyDataSetChanged();  
 }  
  
 *//Tạo menu* @Override  
 **public void** onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 inflater.inflate(R.menu.***chucnang\_menu***, menu) ;  
  
 MenuItem searchItem = menu.findItem(R.id.***search***);  
 SearchView searchView = (SearchView) searchItem.getActionView();  
 search(searchView);  
 **super**.onCreateOptionsMenu(menu,inflater);  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(@NonNull MenuItem item) {  
 **if**(item.getItemId() == R.id.***delete\_all***){  
 confirmXoaTatCa();  
 }  
  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 **private void** search(SearchView searchView) {  
 searchView.setOnQueryTextListener(**new** SearchView.OnQueryTextListener() {  
 @Override  
 **public boolean** onQueryTextSubmit(String query) {  
 **return false**;  
 }  
  
 @Override  
 **public boolean** onQueryTextChange(String newText) {  
 **if** (**customAdapterKH**!=**null**)  
 **customAdapterKH**.getFilter().filter(newText);  
 **return true**;  
 }  
 });  
 }  
  
 **void** confirmXoaTatCa () {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**mContext**);  
 String toastTittle = **mContext**.getResources().getString(R.string.***confirmDeleteAllTittle***);  
 String toastMessage = **mContext**.getResources().getString(R.string.***confirmDeleteAllMessages***);  
 String toastYes = **mContext**.getResources().getString(R.string.***confirmYes***);  
 String toastNo =**mContext**.getResources().getString(R.string.***confirmNo***);  
 builder.setTitle(toastTittle);  
 builder.setMessage(toastMessage);  
 builder.setPositiveButton(toastYes, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 DBDonDatHang myDB = **new** DBDonDatHang(**mContext**);  
 myDB.XoaTatCaKH();  
 hienThiDatabase();  
 }  
 });  
 builder.setNegativeButton(toastNo, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 }  
 });  
 builder.create().show();  
 }  
  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 hienThiDatabase();  
 }  
  
 @Override  
 **public void** onPause() {  
 **super**.onPause();  
 }  
  
 **public void** onStart(){  
 **super**.onStart();  
 }  
}

## 3.3. Màn hình sản phẩm

- Fragment ds sản phẩm:

**public class** Fragment\_DSSP **extends** Fragment {  
 RecyclerView **recyclerView**;  
 FloatingActionButton **add\_button**;  
 ImageView **img\_empty**, **imgSP**;  
 TextView **khongDuLieu**;  
 ArrayList<SanPham> **data** = **new** ArrayList<>();  
 DBDonDatHang **myDB**;  
 CustomAdapterSP **customAdapterSP**;  
 SwipeRefreshLayout **swipeRefreshLayout**;  
 CoordinatorLayout **coLayoutSP**;  
 NavigationView **navigationView**;  
 **private** Context **mContext**;  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setHasOptionsMenu(**true**);*//Hiện option menu (Delete all + Search)* }  
  
 @Override  
 **public void** onAttach(Context context) {  
 **super**.onAttach(context);  
 **mContext**=context;  
 }  
  
 @Nullable  
 @Override  
 **public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***fragment\_dssp***,container,**false**);  
 **recyclerView** = view.findViewById(R.id.***recyclerViewSP***);  
 **add\_button** = view.findViewById(R.id.***add\_button\_sp***);  
 **img\_empty** = view.findViewById(R.id.***image\_empty***);  
 **khongDuLieu** = view.findViewById(R.id.***no\_data***);  
 **coLayoutSP** = view.findViewById(R.id.***coordinatorLayoutSP***);  
 **swipeRefreshLayout** = view.findViewById(R.id.***swipeToRefreshSP***);  
 **navigationView** = view.findViewById(R.id.***nav\_view***);  
  
 **imgSP** = view.findViewById(R.id.***imgRecyclerViewIDSP***);  
 *//Đổi tên toolbar* String toolbBarTitle = **mContext**.getResources().getString(R.string.***trangSP***);  
 ((MainActivity)getActivity()).setActionBarTitle(toolbBarTitle);  
 setEvent();  
 hienThiDatabase();  
 **return** view;  
 }  
 **private void** setEvent() {  
 **add\_button**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(getActivity(), ActivityThemSP.**class**);  
 startActivity(intent);  
 }  
 });  
 *//Kéo trái, phải item để xóa + snackbar undo* **new** ItemTouchHelper((**new** ItemTouchHelper.SimpleCallback(0,  
 ItemTouchHelper.***LEFT*** | ItemTouchHelper.***RIGHT***) {  
 @Override  
 **public boolean** onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {  
 **return false**;  
 }  
 @Override  
 **public void** onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, **int** direction) {  
 **final int** position = viewHolder.getAdapterPosition();  
 **final** SanPham recoverItem = **data**.get(position);  
 *//Lưu giữ id position  
 //final int checkingId = data.get(position).getSttKH();* **data**.remove(position);  
 **customAdapterSP**.notifyItemRemoved(position);  
 *//long result = myDB.XoaItemTheoIDKH(checkingId);* **final** Snackbar snackbar = Snackbar.*make*(**coLayoutSP**, **"Đã xóa sản phẩm"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Hoàn Tác"**, **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **data**.add(position, recoverItem);  
 **customAdapterSP**.notifyItemInserted(position);  
 **customAdapterSP**.notifyDataSetChanged();  
 **myDB**.ThemSP(recoverItem);  
 }  
 }).addCallback(**new** Snackbar.Callback() {  
 @Override  
 **public void** onDismissed(Snackbar snackbar, **int** dismissType) {  
 **super**.onDismissed(snackbar, dismissType);  
 **if**(dismissType == ***DISMISS\_EVENT\_TIMEOUT*** || dismissType == ***DISMISS\_EVENT\_SWIPE*** || dismissType == ***DISMISS\_EVENT\_CONSECUTIVE*** || dismissType == ***DISMISS\_EVENT\_MANUAL*** || dismissType == ***DISMISS\_EVENT\_ACTION***) {  
 **myDB**.XoaMotItemSP(recoverItem);  
 }  
 }  
 });  
 snackbar.show();  
 }  
 })) .attachToRecyclerView(**recyclerView**);  
  
 *//Kéo refresh list* **swipeRefreshLayout**.setOnRefreshListener(**new** SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 **public void** onRefresh() {  
 hienThiDatabase();  
 **swipeRefreshLayout**.setRefreshing(**false**);  
 }  
 });  
 }  
 **public void** hienThiDatabase(){  
 **myDB** = **new** DBDonDatHang(**mContext**);  
 **data** = **myDB**.listSanPhams();  
 **customAdapterSP** = **new** CustomAdapterSP(getActivity(),**mContext**, **data**);  
 **recyclerView**.setAdapter(**customAdapterSP**);  
 **recyclerView**.setLayoutManager(**new** LinearLayoutManager(**mContext**));  
 **if**(**data**.size() > 0){  
 **recyclerView**.setVisibility(View.***VISIBLE***);  
 **img\_empty**.setVisibility(View.***GONE***);  
 **khongDuLieu**.setVisibility(View.***GONE***);  
 }**else** {  
 **recyclerView**.setVisibility(View.***GONE***);  
 **img\_empty**.setVisibility(View.***VISIBLE***);  
 **khongDuLieu**.setVisibility(View.***VISIBLE***);  
 String toastMessage = **mContext**.getResources().getString(R.string.***noData***);  
 Toast.*makeText*(**mContext**,toastMessage, Toast.***LENGTH\_SHORT***).show();  
 }  
 **customAdapterSP**.notifyDataSetChanged();  
 }  
 *//Tạo menu* @Override  
 **public void** onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 inflater.inflate(R.menu.***chucnang\_menu***, menu) ;  
  
 MenuItem searchItem = menu.findItem(R.id.***search***);  
 SearchView searchView = (SearchView) searchItem.getActionView();  
 search(searchView);  
 **super**.onCreateOptionsMenu(menu,inflater);  
 }  
 @Override  
 **public boolean** onOptionsItemSelected(@NonNull MenuItem item) {  
 **if**(item.getItemId() == R.id.***delete\_all***){  
 confirmXoaTatCa();  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
 **private void** search(SearchView searchView) {  
 searchView.setOnQueryTextListener(**new** SearchView.OnQueryTextListener() {  
 @Override  
 **public boolean** onQueryTextSubmit(String query) {  
 **return false**;  
 }  
 @Override  
 **public boolean** onQueryTextChange(String newText) {  
 **if** (**customAdapterSP**!=**null**)  
 **customAdapterSP**.getFilter().filter(newText);  
 **return true**;  
 }  
 });  
 }  
 **void** confirmXoaTatCa () {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**mContext**);  
 String toastTittle = **mContext**.getResources().getString(R.string.***confirmDeleteAllTittle***);  
 String toastMessage = **mContext**.getResources().getString(R.string.***confirmDeleteAllMessages***);  
 String toastYes = **mContext**.getResources().getString(R.string.***confirmYes***);  
 String toastNo =**mContext**.getResources().getString(R.string.***confirmNo***);  
 builder.setTitle(toastTittle);  
 builder.setMessage(toastMessage);  
 builder.setPositiveButton(toastYes, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 DBDonDatHang myDB = **new** DBDonDatHang(**mContext**);  
 myDB.XoaTatCaSP();  
 hienThiDatabase();  
 }  
 });  
 builder.setNegativeButton(toastNo, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 }  
 });  
 builder.create().show();  
 }  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 hienThiDatabase();  
 }  
 @Override  
 **public void** onPause() {  
 **super**.onPause();  
 }  
 **public void** onStart(){  
 **super**.onStart();  
 }  
}

## 3.4. Màn hình đơn hàng

- Fragment đơn hàng:

**public class** Fragment\_DSDH **extends** Fragment {  
 RecyclerView **recyclerViewDH**;  
 FloatingActionButton **add\_button\_dh**;  
 ImageView **img\_empty**;  
 TextView **khongDuLieu**;  
 ArrayList<DonHang> **data** = **new** ArrayList<>();  
 DBDonDatHang **myDB**;  
 CustomAdapterDH **customAdapterDH**;  
 SwipeRefreshLayout **swipeRefreshLayoutDH**;  
 CoordinatorLayout **coLayoutDH**;  
 NavigationView **navigationView**;  
 Toolbar **toolbar**;  
 **private** Context **mContext**;  
  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setHasOptionsMenu(**true**);*//Hiện option menu (Delete all + Search)* }  
 @Override  
 **public void** onAttach(Context context) {  
 **super**.onAttach(context);  
 **mContext**=context;  
 }  
 @Nullable  
 @Override  
 **public** View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 View view = inflater.inflate(R.layout.***fragment\_dsdh***,container,**false**);  
 **recyclerViewDH** = view.findViewById(R.id.***recyclerViewDH***);  
 **add\_button\_dh** = view.findViewById(R.id.***add\_button\_dh***);  
 **img\_empty** = view.findViewById(R.id.***image\_empty***);  
 **khongDuLieu** = view.findViewById(R.id.***no\_data***);  
 **coLayoutDH** = view.findViewById(R.id.***coordinatorLayoutDH***);  
 **swipeRefreshLayoutDH** = view.findViewById(R.id.***swipeToRefreshDH***);  
 **navigationView** = view.findViewById(R.id.***nav\_view***);  
 **toolbar** = view.findViewById(R.id.***toolBar***);  
 setEvent();  
 hienThiDatabase();  
 *//Đổi tên toolbar* String toolbBarTitle = **mContext**.getResources().getString(R.string.***trangDH***);  
 ((MainActivity)getActivity()).setActionBarTitle(toolbBarTitle);  
 **return** view;  
 }  
 **private void** setEvent() {  
 **add\_button\_dh**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(getActivity(), ActivityThemDH.**class**);  
 startActivity(intent);  
 }  
 });  
 *//Kéo trái, phải item để xóa + snackbar undo* **new** ItemTouchHelper((**new** ItemTouchHelper.SimpleCallback(0,  
 ItemTouchHelper.***LEFT*** | ItemTouchHelper.***RIGHT***) {  
 @Override  
 **public boolean** onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {  
 **return false**;  
 }  
 @Override  
 **public void** onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, **int** direction) {  
 **final int** position = viewHolder.getAdapterPosition();  
 **final** DonHang recoverItem = **data**.get(position);  
 *//Lưu giữ id position  
 //final int checkingId = data.get(position).getSttKH();* **data**.remove(position);  
 **customAdapterDH**.notifyItemRemoved(position);  
 *//long result = myDB.XoaItemTheoIDKH(checkingId);* **final** Snackbar snackbar = Snackbar.*make*(**coLayoutDH**, **"Đã xóa đơn hàng"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Hoàn Tác"**, **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **data**.add(position, recoverItem);  
 **customAdapterDH**.notifyItemInserted(position);  
 **customAdapterDH**.notifyDataSetChanged();  
 **myDB**.ThemDH(recoverItem);  
 }  
 }).addCallback(**new** Snackbar.Callback() {  
 @Override  
 **public void** onDismissed(Snackbar snackbar, **int** dismissType) {  
 **super**.onDismissed(snackbar, dismissType);  
 **if**(dismissType == ***DISMISS\_EVENT\_TIMEOUT*** || dismissType == ***DISMISS\_EVENT\_SWIPE*** || dismissType == ***DISMISS\_EVENT\_CONSECUTIVE*** || dismissType == ***DISMISS\_EVENT\_MANUAL*** || dismissType == ***DISMISS\_EVENT\_ACTION***) {  
 **myDB**.XoaMotItemDH(recoverItem);  
 }  
 }  
 });  
 snackbar.show();  
 }  
 })) .attachToRecyclerView(**recyclerViewDH**);  
 *//Kéo refresh list* **swipeRefreshLayoutDH**.setOnRefreshListener(**new** SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 **public void** onRefresh() {  
 hienThiDatabase();  
 **swipeRefreshLayoutDH**.setRefreshing(**false**);  
 }  
 });  
 }  
 **public void** hienThiDatabase(){  
 **myDB** = **new** DBDonDatHang(**mContext**);  
 **data** = **myDB**.listDonHangs();  
 **customAdapterDH** = **new** CustomAdapterDH(getActivity(),**mContext**, **data**);  
 **recyclerViewDH**.setAdapter(**customAdapterDH**);  
 **recyclerViewDH**.setLayoutManager(**new** LinearLayoutManager(**mContext**));  
 **if**(**data**.size() > 0){  
 **recyclerViewDH**.setVisibility(View.***VISIBLE***);  
 **img\_empty**.setVisibility(View.***GONE***);  
 **khongDuLieu**.setVisibility(View.***GONE***);  
 }**else** {  
 **recyclerViewDH**.setVisibility(View.***GONE***);  
 **img\_empty**.setVisibility(View.***VISIBLE***);  
 **khongDuLieu**.setVisibility(View.***VISIBLE***);  
 String toastMessage = **mContext**.getResources().getString(R.string.***noData***);  
 Toast.*makeText*(**mContext**,toastMessage, Toast.***LENGTH\_SHORT***).show();  
 }  
 **customAdapterDH**.notifyDataSetChanged();  
 }  
 *//Tạo menu* @Override  
 **public void** onCreateOptionsMenu(Menu menu, MenuInflater inflater) {  
 inflater.inflate(R.menu.***chucnang\_menu***, menu) ;  
 MenuItem searchItem = menu.findItem(R.id.***search***);  
 SearchView searchView = (SearchView) searchItem.getActionView();  
 search(searchView);  
 **super**.onCreateOptionsMenu(menu,inflater);  
 }  
 @Override  
 **public boolean** onOptionsItemSelected(@NonNull MenuItem item) {  
 **if**(item.getItemId() == R.id.***delete\_all***){  
 confirmXoaTatCa();  
 }  
 **return super**.onOptionsItemSelected(item);  
 }  
 **private void** search(SearchView searchView) {  
 searchView.setOnQueryTextListener(**new** SearchView.OnQueryTextListener() {  
 @Override  
 **public boolean** onQueryTextSubmit(String query) {  
 **return false**;  
 }  
 @Override  
 **public boolean** onQueryTextChange(String newText) {  
 **if** (**customAdapterDH**!=**null**)  
 **customAdapterDH**.getFilter().filter(newText);  
 **return true**;  
 }  
 });  
 }  
 **void** confirmXoaTatCa () {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**mContext**);  
 String toastTittle = **mContext**.getResources().getString(R.string.***confirmDeleteAllTittle***);  
 String toastMessage = **mContext**.getResources().getString(R.string.***confirmDeleteAllMessages***);  
 String toastYes = **mContext**.getResources().getString(R.string.***confirmYes***);  
 String toastNo =**mContext**.getResources().getString(R.string.***confirmNo***);  
 builder.setTitle(toastTittle);  
 builder.setMessage(toastMessage);  
 builder.setPositiveButton(toastYes, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 DBDonDatHang myDB = **new** DBDonDatHang(**mContext**);  
 myDB.XoaTatCaDH();  
 hienThiDatabase();  
 }  
 });  
 builder.setNegativeButton(toastNo, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
 }  
 });  
 builder.create().show();  
 }  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 hienThiDatabase();  
 }  
 @Override  
 **public void** onPause() {  
 **super**.onPause();  
 }  
 **public void** onStart(){ **super**.onStart(); }  
}

## 3.5. Màn hình thống kê

**public class** Fragment\_ThongKe **extends** Fragment {  
 **private** BarChart **bar**;  
 **private** BarData **data**;  
 **private** BarDataSet **dataSet**;  
  
 **private** Context **mContext**;  
 DBDonDatHang **myDB**;  
 SwipeRefreshLayout **swipeRefreshLayoutThongKe**;  
  
 **public static** Fragment\_ThongKe newInstance() {  
 Fragment\_ThongKe fragment = **new** Fragment\_ThongKe();  
 **return** fragment;  
 }  
 @Override  
 **public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 }  
  
 **public** Fragment\_ThongKe() {  
 *// Required empty public constructor* }  
 @Override  
 **public void** onAttach(Context context) {  
 **super**.onAttach(context);  
 **mContext**=context;  
 }  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState) {  
 *// Inflate the layout for this fragment  
  
 //Đổi tên toolbar* String toolbBarTitle = **mContext**.getResources().getString(R.string.***trangThongKe***);  
 ((MainActivity)getActivity()).setActionBarTitle(toolbBarTitle);  
  
 View view =inflater.inflate(R.layout.***fragment\_thongke***, container, **false**);  
  
 Typeface font = Typeface.*createFromAsset*(getActivity().getAssets(), **"antic\_regular.ttf"**);  
  
  
 **swipeRefreshLayoutThongKe** = view.findViewById(R.id.***swipeToRefreshThongKe***);  
 **bar** = (BarChart)view.findViewById(R.id.***barChart***);  
  
 **myDB** = **new** DBDonDatHang(**mContext**);  
 **int** donHangHoanThanh = **myDB**.getSLDonHangHoanThanh();  
 **int** donHangDangGiao = **myDB**.getSLDonHangDangGiao();  
 **int** donHangDaHuy = **myDB**.getSLDonHangDaHuy();  
 **int** donHangTong = **myDB**.getSLDonHangTong();  
  
 List<BarEntry> entries = **new** ArrayList<>();  
 entries.add(**new** BarEntry(0f, donHangTong,**"Tổng đơn"**));  
 entries.add(**new** BarEntry(1f, donHangHoanThanh,**"Hoàn thành"**));  
 entries.add(**new** BarEntry(2f, donHangDangGiao,**"Đang giao"**));  
 entries.add(**new** BarEntry(3f, donHangDaHuy,**"Đã hủy"**));  
  
 **dataSet** = **new** BarDataSet(entries, **"Marks"**);  
 **dataSet**.setColors(ColorTemplate.***VORDIPLOM\_COLORS***);  
  
 ArrayList<String> barFactors = **new** ArrayList<>();  
 barFactors.add(**"Tổng đơn"**);  
 barFactors.add(**"Hoàn thành"**);  
 barFactors.add(**"Đang giao"**);  
 barFactors.add(**"Đã hủy"**);  
  
 XAxis xAxis = **bar**.getXAxis();  
 xAxis.setGranularity(1f);  
 xAxis.setGranularityEnabled(**true**);  
  
 **data** = **new** BarData(**dataSet**);  
 **data**.setBarWidth(0.9f); *// set custom bar width* **data**.setValueTextSize(12);  
 **data**.notifyDataChanged();  
  
 Description description = **new** Description();  
 description.setTextColor(R.color.***colorPrimary***);  
 description.setText(**"Swipe to refresh data"**);  
  
 **bar**.setDescription(description);  
 **bar**.setData(**data**);  
 **bar**.setFitBars(**true**); *// make the x-axis fit exactly all bars* **bar**.notifyDataSetChanged();  
 **bar**.invalidate(); *// refresh* **bar**.getXAxis().setValueFormatter(**new** IndexAxisValueFormatter(barFactors));  
  
 Legend l = **bar**.getLegend();  
 l.setFormSize(10f); *// set the size of the legend forms/shapes* l.setForm(Legend.LegendForm.***CIRCLE***); *// set what type of form/shape should be used* l.setTypeface(font);  
 l.setTextSize(12f);  
 l.setTextColor(Color.***BLACK***);  
 List<LegendEntry> lentries = **new** ArrayList<>();  
 **for** (**int** i = 0; i < barFactors.size(); i++) {  
 LegendEntry entry = **new** LegendEntry();  
 entry.**formColor** = ColorTemplate.***VORDIPLOM\_COLORS***[i];  
 entry.**label** = barFactors.get(i);  
 lentries.add(entry);  
 }  
 l.setXEntrySpace(5f); *// set the space between the legend entries on the x-axis* l.setYEntrySpace(5f);  
 l.setCustom(lentries);  
  
 setEvent();  
 **return** view;  
 }  
  
 **private void** setEvent() {  
 *//Kéo refresh list* **swipeRefreshLayoutThongKe**.setOnRefreshListener(**new** SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 **public void** onRefresh() {  
 **swipeRefreshLayoutThongKe**.setRefreshing(**false**);  
 updateData();  
 }  
 });  
 }  
  
 **public void** updateData() {  
 **myDB** = **new** DBDonDatHang(**mContext**);  
 **int** donHangHoanThanh = **myDB**.getSLDonHangHoanThanh();  
 **int** donHangDangGiao = **myDB**.getSLDonHangDangGiao();  
 **int** donHangDaHuy = **myDB**.getSLDonHangDaHuy();  
 **int** donHangTong = **myDB**.getSLDonHangTong();  
  
 **dataSet**.removeEntry(0);  
 **data**.addEntry(**new** BarEntry(0f, donHangTong,**"Tổng đơn"**),0);  
 **data**.addEntry(**new** BarEntry(1f, donHangHoanThanh,**"Hoàn thành"**),0);  
 **data**.addEntry(**new** BarEntry(2f, donHangDangGiao,**"Đang giao"**),0);  
 **data**.addEntry(**new** BarEntry(3f, donHangDaHuy,**"Đã hủy"**),0);  
  
 **data**.notifyDataChanged(); *// NOTIFIES THE DATA OBJECT* **bar**.notifyDataSetChanged(); *// let the chart know it's data changed* **bar**.invalidate(); *// refresh* }  
}

## 3.6. Màn hình đổi ngôn ngữ

- Tạo class LocaleHelper để đổi ngôn ngữ:

**public class** LocaleHelper {  
 **private static final** String ***SELECTED\_LANGUAGE*** = **"Locale.Helper.Selected.Language"**;  
 **public static** Context onAttach(Context context) {  
 String lang = *getPersistedData*(context, Locale.*getDefault*().getLanguage());  
 **return** *setLocale*(context, lang);  
 }  
 **public static** Context onAttach(Context context, String defaultLanguage) {  
 String lang = *getPersistedData*(context, defaultLanguage);  
 **return** *setLocale*(context, lang);  
 }  
 **public static** String getLanguage(Context context) {  
 **return** *getPersistedData*(context, Locale.*getDefault*().getLanguage());  
 }  
 **public static** Context setLocale(Context context, String language) {  
 *persist*(context, language);  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***N***) {  
 **return** *updateResources*(context, language);  
 }  
 **return** *updateResourcesLegacy*(context, language);  
 }  
 @SuppressWarnings(**"deprecation"**)  
 **private static** String getPersistedData(Context context, String defaultLanguage) {  
 SharedPreferences preferences = PreferenceManager.*getDefaultSharedPreferences*(context);  
 **return** preferences.getString(***SELECTED\_LANGUAGE***, defaultLanguage);  
 }  
  
 @SuppressWarnings(**"deprecation"**)  
 **private static void** persist(Context context, String language) {  
 SharedPreferences preferences = PreferenceManager.*getDefaultSharedPreferences*(context);  
 SharedPreferences.Editor editor = preferences.edit();  
 editor.putString(***SELECTED\_LANGUAGE***, language);  
 editor.apply();  
 }  
 @SuppressWarnings(**"deprecation"**)  
 @TargetApi(Build.VERSION\_CODES.***N***)  
 **private static** Context updateResources(Context context, String language) {  
 Locale locale = **new** Locale(language);  
 Locale.*setDefault*(locale);  
 Configuration configuration = context.getResources().getConfiguration();  
 configuration.setLocale(locale);  
 configuration.setLayoutDirection(locale);  
 **return** context.createConfigurationContext(configuration);  
 }  
 @SuppressWarnings(**"deprecation"**)  
 **private static** Context updateResourcesLegacy(Context context, String language) {  
 Locale locale = **new** Locale(language);  
 Locale.*setDefault*(locale);  
 Resources resources = context.getResources();  
 Configuration configuration = resources.getConfiguration();  
 configuration.**locale** = locale;  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***JELLY\_BEAN\_MR1***) {  
 configuration.setLayoutDirection(locale);  
 }  
 resources.updateConfiguration(configuration, resources.getDisplayMetrics());  
 **return** context;  
 }  
}

-Tạo activity để dùng

**public class** ActivityDoiNgonNgu **extends** AppCompatActivity {  
 TextView **mTextView**;  
 Spinner **mLanguage**;  
 ArrayAdapter<String> **mAdapter**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_change\_language***);  
 ActionBar actionBar = getSupportActionBar();  
 actionBar.setTitle(getResources().getString(R.string.***trangDoiNgonNgu***));  
 actionBar.setDisplayHomeAsUpEnabled(**true**);  
 **mLanguage** = (Spinner) findViewById(R.id.***spLanguage***);  
 **mTextView** = (TextView) findViewById(R.id.***textView***);  
 **mAdapter** = **new** ArrayAdapter<String>(ActivityDoiNgonNgu.**this**,  
 android.R.layout.***simple\_spinner\_dropdown\_item***,  
 getResources().getStringArray(R.array.***language\_option***));  
 **mLanguage**.setAdapter(**mAdapter**);  
 **if** (com.example.testrecycler.Adapter.LocaleHelper.*getLanguage*(ActivityDoiNgonNgu.**this**).equalsIgnoreCase(**"en"**)) {  
 **mLanguage**.setSelection(**mAdapter**.getPosition(**"English"**));  
 } **else if** (com.example.testrecycler.Adapter.LocaleHelper.*getLanguage*(ActivityDoiNgonNgu.**this**).equalsIgnoreCase(**"vi-rVN"**)) {  
 **mLanguage**.setSelection(**mAdapter**.getPosition(**"Vietnamese"**));  
 }  
 **mLanguage**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  
  
 @Override  
 **public void** onItemSelected(AdapterView<?> adapterView, View view, **int** i, **long** l) {  
 Context context;  
 Resources resources;  
 **switch** (i) {  
 **case** 0:  
 context = com.example.testrecycler.Adapter.LocaleHelper.*setLocale*(ActivityDoiNgonNgu.**this**, **"en"**);  
 resources = context.getResources();  
 **mTextView**.setText(resources.getString(R.string.***app\_name***));  
 **break**;  
 **case** 1:  
 context = com.example.testrecycler.Adapter.LocaleHelper.*setLocale*(ActivityDoiNgonNgu.**this**, **"vi-rVN"**);  
 resources = context.getResources();  
 **mTextView**.setText(resources.getString(R.string.***app\_name***));  
 **break**;  
 }  
 }  
 @Override  
 **public void** onNothingSelected(AdapterView<?> adapterView) {  
  
 }  
 });  
 }  
  
 @Override  
 **protected void** attachBaseContext(Context newBase) {  
 **super**.attachBaseContext(com.example.testrecycler.Adapter.LocaleHelper.*onAttach*(newBase));  
 }  
  
 @Override  
 **public void** onBackPressed() {  
 **if** (getSupportFragmentManager().getBackStackEntryCount() > 0) {  
 getSupportFragmentManager().popBackStack();  
 } **else** {  
 **super**.onBackPressed();  
 }  
 }  
  
 *//Bấm nút quay về trang chủ (ko dùng AndroidManifest được vì là từ Fragment về Activity)* @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 **super**.onOptionsItemSelected(item);  
 **switch** (item.getItemId()) {  
 **case** android.R.id.***home***:  
 finish();  
 **break**;  
 }  
 **return true**;  
 }  
}

## 3.7. Màn hình About Us (Custom View)

- Khai báo 1 class MyHeartShape để code vẽ

**public class** MyHeartShape **extends** View{  
  
 **private static int** *WIDTH* = 500;  
 **private static int** *HEIGHT* = 300;  
  
 **private** Path **path**;  
 **private** Paint **paint**;  
  
 **private int top**;  
 **private int left**;  
  
  
 **public** MyHeartShape(Context context) {  
 **super**(context);  
  
 **path** = **new** Path();  
 **paint** = **new** Paint(Paint.***ANTI\_ALIAS\_FLAG***);  
  
 **top**=10;  
 **left**=10;  
 }  
  
 @Override  
 **protected void** onDraw(Canvas canvas) {  
 **super**.onDraw(canvas);  
 *// Fill the canvas with background color* canvas.drawColor(Color.***WHITE***);  
 **paint**.setShader(**null**);  
  
 *// Defining of the heart path starts* **path**.moveTo(**left**+*WIDTH*/2, **top**+*HEIGHT*/4); *// Starting point  
 // Create a cubic Bezier cubic left path* **path**.cubicTo(**left**+*WIDTH*/5,**top**,  
 **left**+*WIDTH*/4,**top**+4\**HEIGHT*/5,  
 **left**+*WIDTH*/2, **top**+*HEIGHT*);  
 *// This is right Bezier cubic path* **path**.cubicTo(**left**+3\**WIDTH*/4,**top**+4\**HEIGHT*/5,  
 **left**+4\**WIDTH*/5,**top**,  
 **left**+*WIDTH*/2, **top**+*HEIGHT*/4);  
  
 **paint**.setColor(Color.***RED***); *// Set with heart color  
 //paint.setShader(shader);* **paint**.setStyle(Style.***FILL***); *// Fill with heart color* canvas.drawPath(**path**, **paint**); *// Actual drawing happens here  
  
 // Draw Blue Boundary* **paint**.setShader(**null**);  
 **paint**.setColor(Color.***BLUE***); *// Change the boundary color* **paint**.setStrokeWidth(4);  
 **paint**.setStyle(Style.***STROKE***);  
 canvas.drawPath(**path**, **paint**);  
 }  
}

- Sử dụng class vừa tạo ở activity mới

**public class** ActivityAboutUs **extends** Activity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 *//setContentView(R.layout.activity\_main);* MyHeartShape heart = **new** MyHeartShape(**this**);  
 setContentView(heart);  
 }  
}

# CHƯƠNG 4 – KẾT QUẢ ĐẠT ĐƯỢC

## 4.1. Kết quả đạt được

* Phần mềm có thể thêm, xóa, sửa, tìm kiếm ở các bảng Khách hàng, Sản phẩm, Đơn hàng
* Sử dụng Recyclerview + Cardview để hiển thị dữ liệu
* Splash Screen có animation, drawable
* Custom View (Custom Drawing) ở About Us
* Custom Button
* Dùng SharedPreferences để hỗ trợ đa ngôn ngữ (Tiếng Việt và Tiếng Anh)
* Có cảm biến shake sensor để hiển thị hộp thoại.
* Swipe touch để làm mới các dữ liệu được thêm vào
* Swipe touch để xóa + Undo dữ liệu (dùng Snackbar)
* Đã có Navigation Drawer
* Đã có Text Input Layout cho các ô input
* Thêm ảnh cho sản phẩm
* Custom Spinner Sản phẩm bao gồm ảnh
* Thêm trạng thái đơn hàng để dễ quản lý
* Áp dụng dữ liệu trạng thái đơn hàng để tạo ra bar chart

## 4.2. Kết luận

* Những việc đã làm được
  + Liên kết dữ liệu ở 3 bảng chính Khách hàng, sản phẩm, đơn hàng
  + Ứng dụng được Recyclerview + cardview
  + Ứng dụng được animation, drawable, custom view
  + Ứng dụng được cảm biến sensor
  + Ứng dụng được Bar Chart
  + Ứng dụng được SharedPreferences cho đa ngôn ngữ
  + Ứng dụng được Navigation Drawer
  + Ứng dụng được custom spinner
  + Ứng dụng được Snackbar và swipe touch để refresh, delete dữ liệu
  + Ứng dụng được fragment cho các màn hình Home, Danh sách khách hàng, Danh sách sản phẩm, Danh sách đơn hàng, Thống kê
* Những việc chưa làm được
  + Chưa thể nhận được ảnh sản phẩm chất lượng tốt (Phải nén ảnh xuống chất lượng thấp nhất).
  + Chưa lưu dữ liệu ngày/tháng/năm theo đúng định dạng thời gian trong Database.
  + Chưa xử lý lưu trùng mã.
  + Sau khi Undo thì chưa xử lý cho đối tượng được hoàn trả về đúng vị trí nó bị xóa, bị nhảy xuống vị trí cuối cùng.