

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 Chấm Thi”**

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

**Sinh viên thực hiện: Nhóm 20**

**1. Nguyễn Hoàng Long**

**2. Cao Đại Đạt**

**Khoa: Công nghệ thông tin** **Khóa: 2018**

**Ngày 01/08/2020**

**NHẬT KÝ HOẠT ĐỘNG NHÓM 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | HỌ TÊN | CÔNG VIỆC | TỰ ĐÁNH GIÁ | NHÓM ĐÁNH GIÁ | KÝ TÊN |
| 1 | Nguyễn Hoàng Long | * Môn Học * Thông Tin Bài Thi * Splash Screen * Thống kê * Home | 50% | 50% |  |
| 2 | Cao Đại Đạt | * Giáo Viên * Navigation Drawer * About Us * Đổi ngôn ngữ * Powerpoint | 50% | 50% |  |

**MỤC LỤC**

[CHƯƠNG 1 – MỞ ĐẦU 4](#_Toc47985054)

[**1.1.** **Giới thiệu môn học và nhóm thực hiện** 4](#_Toc47985055)

[**1.2.** **Mô tả ứng dụng** 4](#_Toc47985056)

[**1.3.** **Ứng dụng trong cuộc sống** 4](#_Toc47985057)

[CHƯƠNG 2 – PHÂN TÍCH CẤU TRÚC HỆ THỐNG 5](#_Toc47985058)

[2.1. Phân tích thiết kế hệ thống 5](#_Toc47985059)

[2.2. Thiết kế hệ thống 13](#_Toc47985060)

[**a)** **Màn hình splash** 13](#_Toc47985061)

[**b)** **Màn hình home** 15](#_Toc47985062)

[**c)** **Màn hình danh sách giáo viên** 18](#_Toc47985063)

[**d)** **Màn hình thêm giáo viên** 21](#_Toc47985064)

[**e)** **Màn hình sửa giáo viên** 23](#_Toc47985065)

[**f)** **Màn hình danh sách môn học** 25](#_Toc47985066)

[**g)** **Màn hình thêm môn học**  28](#_Toc47985067)

[**h)** **Màn hình sửa môn học** 30](#_Toc47985068)

[**i)** **Màn hình danh sách Thông tin bài thi** 32](#_Toc47985069)

[**j)** **Màn hình thêm Thông tin bài thi** 35](#_Toc47985070)

[**k)** **Màn hình sửa Thông tin bài thi** 37](#_Toc47985071)

[**l)** **Màn hình thống kê Thông tin bài thi** 39](#_Toc47985072)

[**m)** **Màn hình đổi ngôn ngữ** 41](#_Toc47985073)

[**n)** **Màn hình About Us** 42](#_Toc47985074)

[CHƯƠNG 3 – CÀI ĐẶT 45](#_Toc47985075)

[3.1. Màn hình home 45](#_Toc47985076)

[3.2. Màn hình giáo viên 47](#_Toc47985077)

[3.3. Màn hình môn học 51](#_Toc47985078)

[3.4. Màn hình Thông tin bài thi 54](#_Toc47985079)

[3.5. Màn hình thống kê 58](#_Toc47985080)

[3.6. Màn hình đổi ngôn ngữ 60](#_Toc47985081)

[3.7. Màn hình About Us (Custom View) 63](#_Toc47985082)

[CHƯƠNG 4 – KẾT QUẢ ĐẠT ĐƯỢC 65](#_Toc47985083)

[4.1. Kết quả đạt được 65](#_Toc47985084)

[4.2. Kết luận 65](#_Toc47985085)

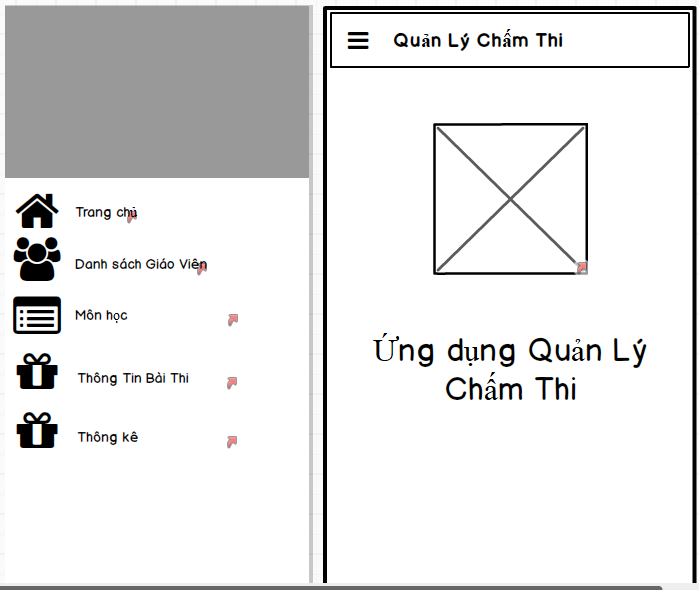
# CHƯƠNG 1 – MỞ ĐẦU

* 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,Style,…
* Thành viên thực hiện:
  1. Nguyễn Hoàng Long (Leader)
  2. Cao Đại Đạt
  3. **Mô tả ứng dụng**
* Tên ứng dụng: Quản lý Chấm Thi
* Không cần kết nối mạng
* Android 4.0+
* Thiết lập và quản lý dữ liệu của: Giáo viên, Môn học,Thông tin chấm bài. Dựa vào Bar Chart có thể quản lý tốt hơn các bài thi.
* Hỗ trợ 2 ngôn ngữ: Tiếng Việt và Tiếng Anh
  1. **Ứng dụng trong cuộc sống**
* Với việc nhu cầu chấm thi nhanh chóng và một cách tiện lợi, và quản lý được việc chấm thi của mỗi giáo viên. Ứng dụng “Quản Lý Chấm Thi” là một sự lựa chọn không thể tốt hơn cho công việc này. Nó giúp người quản lý có thể quản lý các bài thi được chấm bởi giáo viên nào, tình trạng bài thi như thế nào,… và nhiều chức năng khác, hãy cài đặt và trải nghiệm để xem sự tuyệt vời mà nó mang lại.

# CHƯƠNG 2 – PHÂN TÍCH CẤU TRÚC HỆ THỐNG

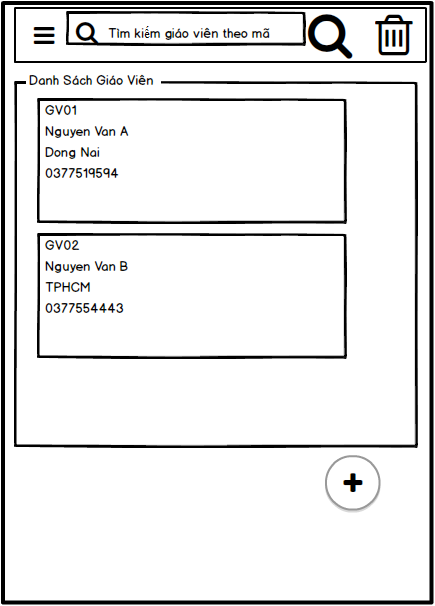
## 2.1. Phân tích thiết kế hệ thống

**-** Tổng quan: Đây là ứng dụng dùng để theo dõi, quản lý danh sách môn, giao viên, các bài thi một cách thuận tiện, nhanh chóng, không cần có internet.

**Yêu cầu chức năng (Home):** 

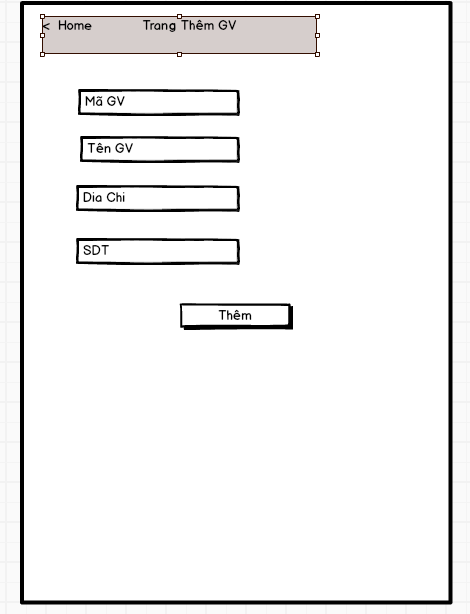
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình home | Màn hình chính có: - 1 TextView - 1 background | Thiết kế | Hoàn thành |
| Navigation drawer menu “Trang chủ” | Chuyển sang giao diện “Trang chủ” | Click |  |
| Navigation drawer menu “Chỉnh sửa ds giáo viên” | Chuyển sang giao diện “Danh sách giáo viên” | Click |  |
| Navigation drawer menu “Chỉnh sửa các Thông tin bài thi” | Chuyển sang giao diện “Danh sách Thông tin bài thi” | Click |  |
| Navigation drawer menu “Chỉnh sửa các môn học” | Chuyển sang giao diện “Danh sách môn học” | Click |  |
| Navigation drawer menu “Thông Kê” | Chuyển sang giao diện “Thống kê bài thi” | Click | Hoàn thành |

**Yêu cầu chức năng (Trang danh sách giáo viên):**



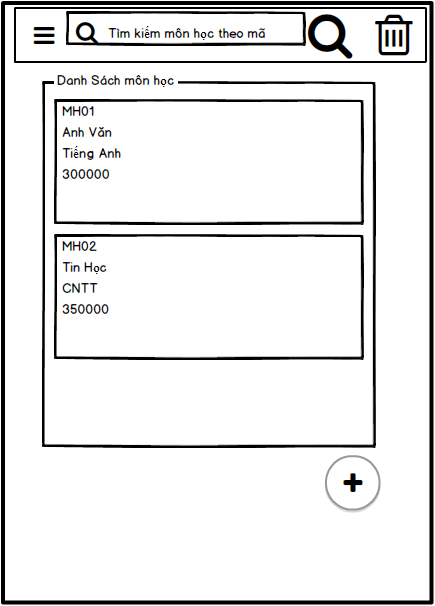
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình giáo viên | Màn hình chính có: - 1 TextView - 3 Button  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Giáo viên” | Click | Hoàn thành |
| Button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

**Yêu cầu chức năng (Trang thêm giáo viên)**



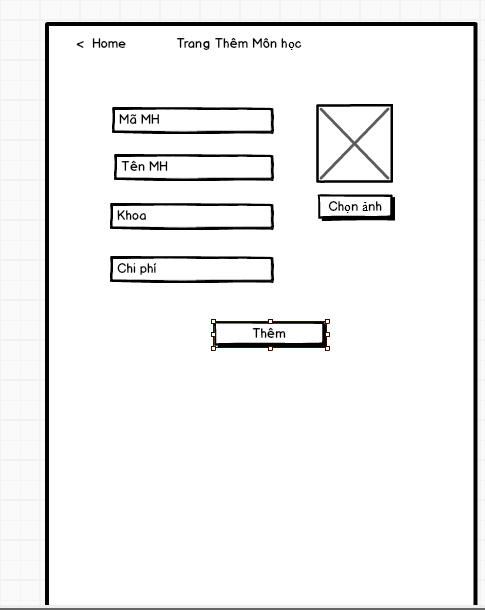
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Giáo viên | Màn hình chính có: - 4 Edit Text - 2 Button | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách Giáo viên | Click | Hoàn thành |
| Button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

**Yêu cầu chức năng (Trang danh sách môn học):**



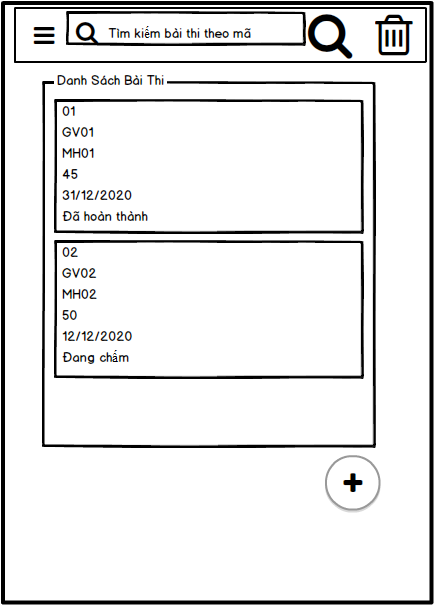
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình môn học | Màn hình chính có: - 1 TextView - 3 Button  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Môn học” | Click | Hoàn thành |
| Button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

**Yêu cầu chức năng (Trang thêm môn học)**



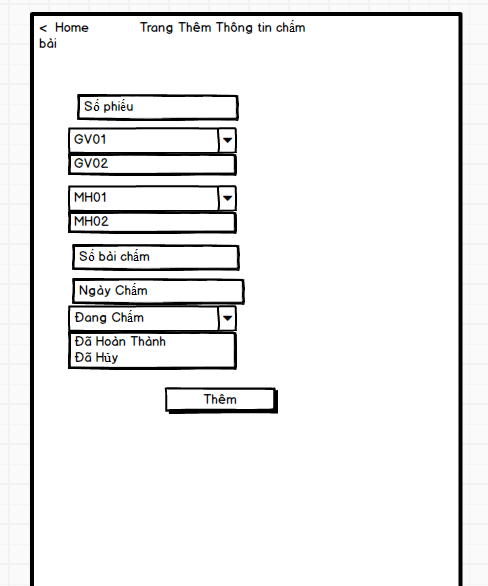
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Môn học | Màn hình chính có: - 4 Edit Text - 2 Button | Thiết kế | Hoàn thành |
| Button “Thêm” | Thêm dữ liệu vào RecyclerView ở trang Danh Sách Môn học | Click | Hoàn thành |
| Button 🡨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

**Yêu cầu chức năng (Trang danh sách Thông tin bài thi):**



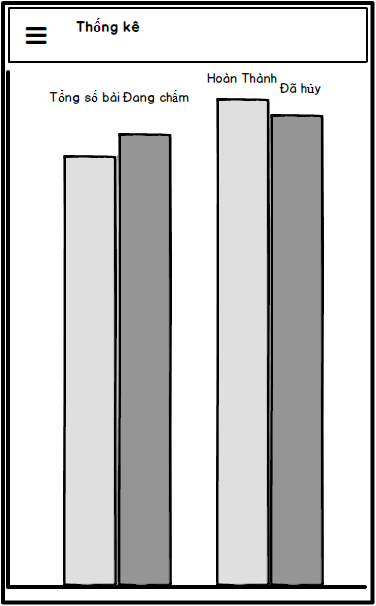
|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thông tin bài thi | Màn hình chính có: - 1 TextView - 3 Button  -1 RecyclerView + Card View Item | Thiết kế | Hoàn thành |
| Button “+” | Dẫn tới màn hình “Thêm Thông tin bài thi” | Click | Hoàn 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 Thông tin bài thi ở RecyclerView theo điều kiện tìm kiếm | Click | Hoàn thành |
| RecyclerView | Hiển thị danh sách Thông tin bài thi và khi click vào item thì chuyển sang trang chi tiết Thông tin bài thi để SỬA theo từng item được click | Click | Hoàn thành |

**Yêu cầu chức năng (Trang thêm Thông tin bài thi)**



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

**Yêu cầu chức năng (Trang thống kê bài thi):**



|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình môn học | - 1 Chart | Thiết kế | Hoàn thành |
| Chart | - Thống kê theo bài thi | 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 môn học hoàn chỉnh theo thời gian. Mô tả chi tiết công việc cho từng màn hình
* **Yêu cầu chức năng:** 
  1. **Màn hình splash**
     + **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 |

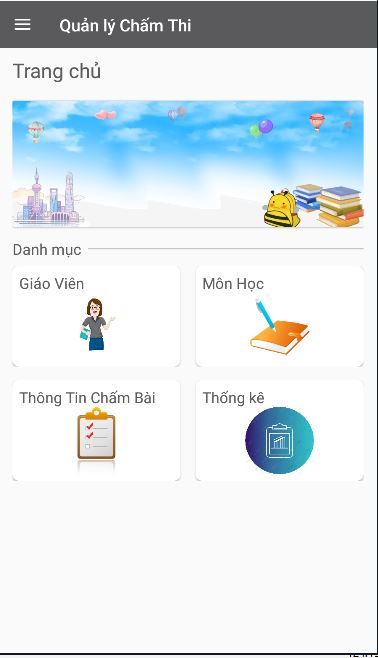
* + - **Cách vận hành**

+ Chạy ứng dụng

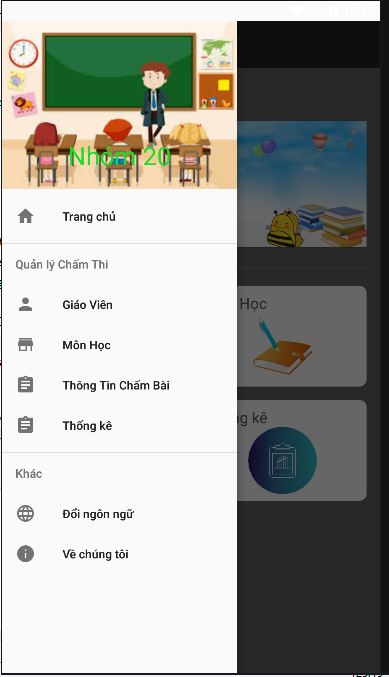
+ Chương trình sẽ chạy màn hình Splash

+ Sau 5s tự động chuyển qua màn hình home

* 1. **Màn hình home** 
     + **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 Thông tin bài thi” | Chuyển sang giao diện “Thống kê bài thi” | Click | Hoàn thành |
| CardView “Giáo viên” | Chuyển sang giao diện “Danh sách giáo viên” | Click | Hoàn thành |
| CardView “Môn học” | Chuyển sang giao diện “Danh sách môn học” | Click | Hoàn thành |
| CardView “Thông tin bài thi” | Chuyển sang giao diện “Danh sách Thông tin bài thi” | Click | Hoàn thành |
| Navigation drawer menu “Trang chủ” | Chuyển sang giao diện “Trang chủ” | Click | Hoàn thành |
| Navigation drawer menu “DS giáo viên” | Chuyển sang giao diện “Danh sách giáo viên” | Click | Hoàn thành |
| Navigation drawer menu “DS Thông tin bài thi” | Chuyển sang giao diện “Danh sách Thông tin bài thi” | Click | Hoàn thành |
| Navigation drawer menu “DS môn học” | Chuyển sang giao diện “Danh sách môn học” | 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 |

* + - **Cách vận hành**

+ CardView Giáo Viên:

-Click cardview giáo viên

-Ứng dụng chuyển qua giao diện danh sách giáo viên.

+ CardView Thống kê:

-Click cardview thống kê

-Ứng dụng chuyển qua giao diện thống kê bài thi

+ CardView Môn học:

-Click cardview môn học

-Ứng dụng chuyển qua giao diện danh sách môn học

+ CardView Thông tin bài thi:

-Click cardview thông tin bài thi

-Ứng dụng chuyển qua giao diện danh sách thông tin bài thi.

+Navigation Drawer menu – CLICK

- Navigation drawer menu “Trang chủ”

- Navigation drawer menu “Giáo Viên”

- Navigation drawer menu “Môn học”

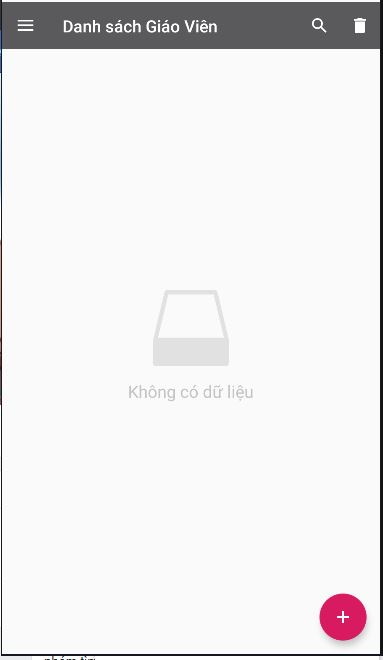
- Navigation drawer menu “Thông tin môn học”

- Navigation drawer menu “Thống kê”

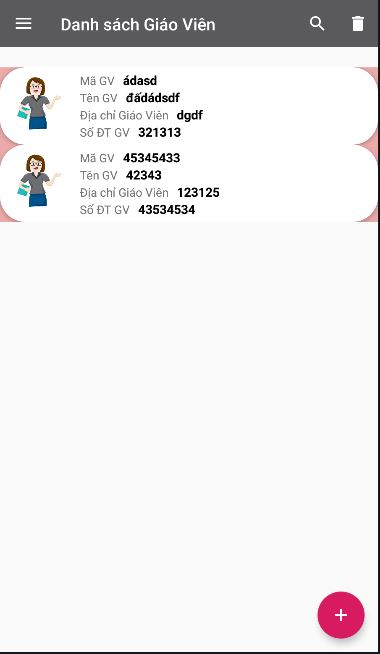
- Navigation drawer menu “Đổi ngôn ngữ”

- Navigation drawer menu “About Us”

* 1. **Màn hình danh sách giáo viên**
     + **Giao diện người dùng**

****

*Hình 2.2.c. Danh sách giáo viên khi không có dữ liệu*

****

*Hình 2.2.c. Danh sách giáo viên khi có dữ liệu*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình giáo viên | 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 Giáo Viên” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả môn học khỏi bảng danh sách giáo viên | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách giáo viên 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 giáo viên và khi click vào item thì chuyển sang trang chi tiết giáo viên để sửa theo từng item được click | Click | Hoàn thành |

* + - **Cách vận hành**

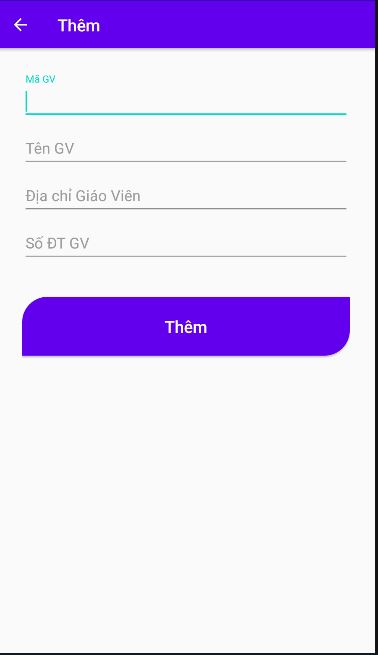
- Click floating button “+” để vào màn hình “Thêm giáo viên”

- Click button Tìm kiếm trên thanh action bar để tìm giáo viên theo mã giáo viên

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

giáo viên

* 1. **Màn hình thêm giáo viên**
     + **Giao diện người dùng**

****

*Hình 2.2.d. Thêm giáo viên*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm giáo viên | 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 giáo viên | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách giáo viên | Click | Hoàn thành |
| 4 Edit Text (Text Input Layout) | Nhập thông tin cần có cho giáo viên, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |

* + - **Cách vận hành**

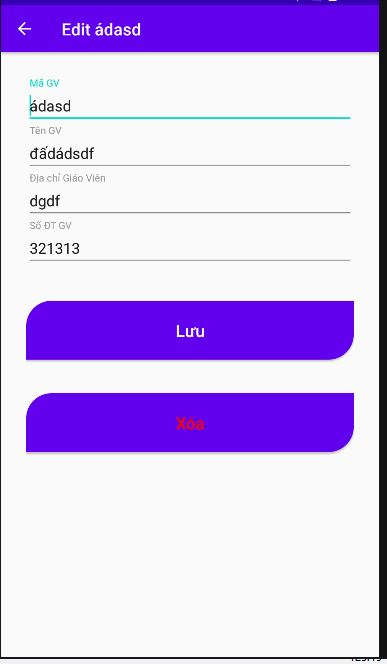
- Nhấn floating button + ở trang danh sách giáo viên để vào trang thêm giáo viên

- Nhập đầy đủ thông tin 4 ô edit text

- Nhấn button Thêm để thêm giáo viên vào

- Nhấn nút 🡨 để quay lại trang danh sách giáo viên

* 1. **Màn hình sửa giáo viên** 
     + **Giao diện người dùng**

****

*Hình 2.2.e. Sửa giáo viên*

* + - **Yêu cầu chức năng**

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

* + - **Cách vận hành**

**-** Ở trang danh sách môn học, click vào 1 giáo viên cần sửa để vào màn hình Sửa giáo viên

- Điền các thông tin cần sửa

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

- Nhấn nút 🡨 để quay lại trang danh sách giáo viên

* 1. **Màn hình danh sách môn học**
     + **Giao diện người dùng**

****

*Hình 2.2. Trang danh sách môn học, khi không có dữ liệu*

****

*Hình 2.2.f Trang danh sách môn học, khi có dữ liệu*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình môn học | 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 Môn học” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả môn học khỏi bảng danh sách môn học | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách môn học 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 môn học và khi click vào item thì chuyển sang trang chi tiết môn học để sửa theo từng item được click | Click | Hoàn thành |

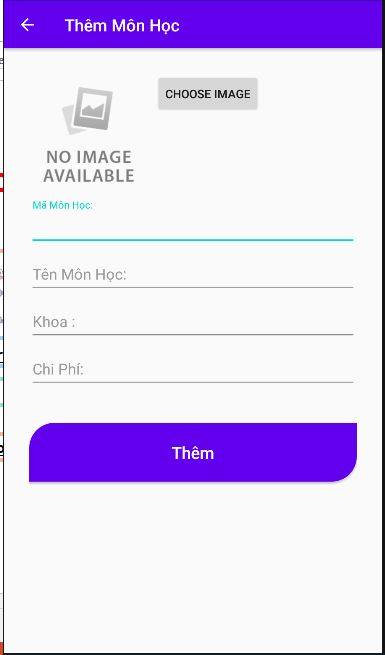
* + - **Cách vận hành**

- Click floating button “+” để vào màn hình “Thêm môn học”

- Click button Tìm kiếm trên thanh action bar để tìm môn học theo mã môn học

- Click button thùng rác trên thanh action bar để xóa tất cả môn học

* 1. **Màn hình thêm môn học**
     + **Giao diện người dùng**

****

*Hình 2.2.g. Màn hình trang thêm môn học*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Môn học | 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 Môn học | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Môn học | Click | Hoàn thành |
| Button “Choose Image” | Chọn hình ảnh môn học để upload lên DB, hiển thị ở trang danh sách môn học | Click | Hoàn thành |
| 4 Edit Text (Text Input Layout) | Nhập thông tin cần có cho môn học, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |

* + - **Cách vận hành**

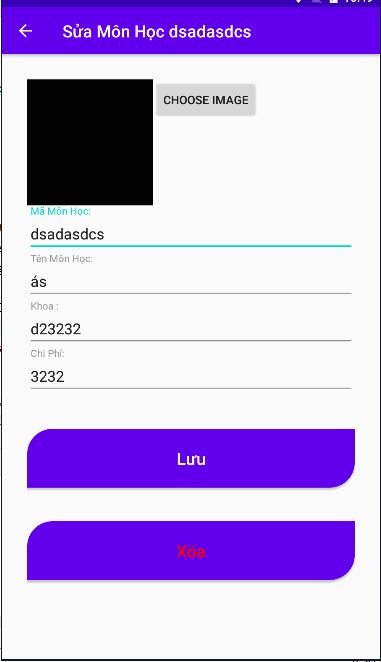
- Nhấn floating button + ở trang danh sách môn học để vào trang thêm môn học

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

- Nhấn button Thêm để thêm môn học vào

- Nhấn nút 🡨 để quay lại trang danh sách môn học

* 1. **Màn hình sửa môn học** 
     + **Giao diện người dùng**

****

*Hình 2.2.h. Màn hình trang sửa môn học*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Sửa Môn học | 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 Môn học | Click | Hoàn thành |
| Button “Xóa” | Xóa dữ liệu đang chọn từ RecyclerView ở trang Danh Sách Môn học | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Môn học | Click | Hoàn thành |
| Button “Choose Image” | Chọn hình ảnh môn học để upload lên DB, hiển thị ở trang danh sách môn học | Click | Hoàn thành |
| 4 Edit Text (Text Input Layout) | Nhập thông tin cần có cho môn học, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |

* + - **Cách vận hành**

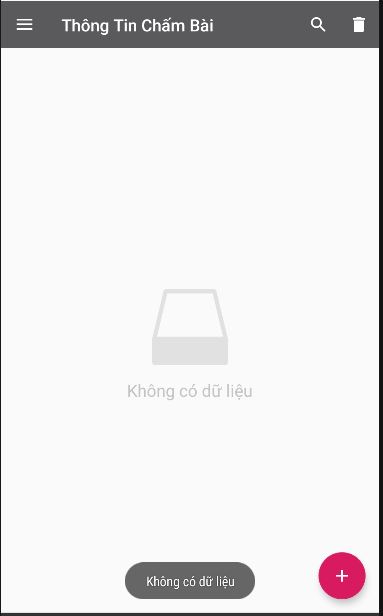
**-** Ở trang danh sách môn học, click vào 1 môn học cần sửa để vào màn hình Sửa môn học

- Điền các thông tin cần sửa

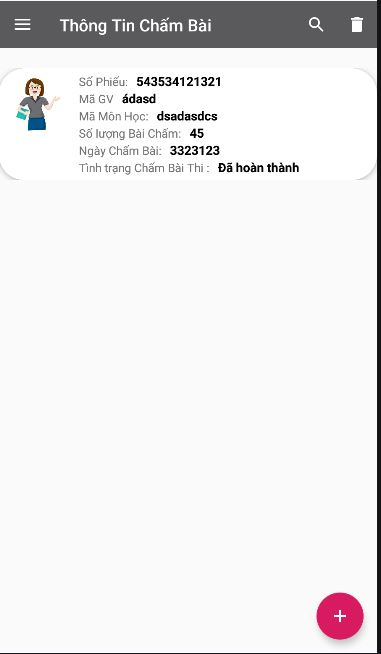
- Nhấn nút Lưu để lưu dữ liệu vừa sửa, hoặc nhấn nút Xóa để xóa đi môn học đang chọn

- Nhấn nút 🡨 để quay lại trang danh sách môn học

* 1. **Màn hình danh sách Thông tin bài thi** 
     + **Giao diện người dùng**

****

*Hình 2.2.i. Màn hình trang danh sách Thông tin bài thi khi không có Thông tin bài thi*

****

*Hình 2.2.i. Màn hình trang danh sách Thông tin bài thi khi có Thông tin bài thi*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thông tin bài thi | 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 Thông tin bài thi” | Click | Hoàn thành |
| Button Thùng rác | Xóa tất cả Thông tin bài thi khỏi bảng danh sách môn học | Click | Hoàn thành |
| Button Tìm kiếm | Hiển thị thanh tìm kiếm bên trái tiêu đề và hiển thị danh sách Thông tin bài thi 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 Thông tin bài thi và khi click vào item thì chuyển sang trang chi tiết Thông tin bài thi sửa theo từng item được click | Click | Hoàn thành |

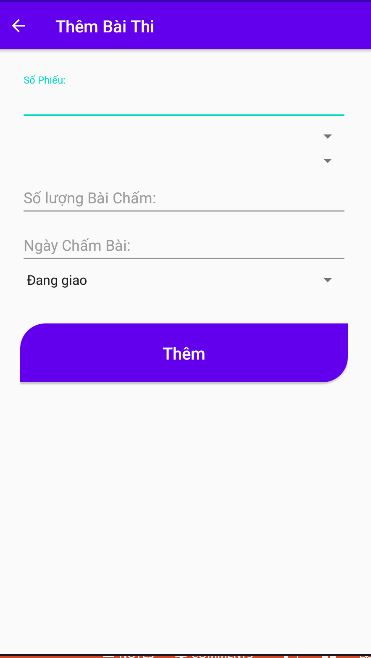
* + - **Cách vận hành**

- Click floating button “+” để vào màn hình “Thêm Thông tin bài thi”

- Click button Tìm kiếm trên thanh action bar để tìm Thông tin bài thi theo mã Thông tin bài thi

- Click button thùng rác trên thanh action bar để xóa tất cả Thông tin bài thi

* 1. **Màn hình thêm Thông tin bài thi** 
     + **Giao diện người dùng**

****

*Hình 2.2.j. Màn hình trang thêm Thông tin bài thi*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thêm Thông tin bài thi | 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 Thông tin bài thi | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Thông tin bài thi | Click | Hoàn thành |
| Button “Choose Image” | Chọn hình ảnh môn học để upload lên DB, hiển thị ở trang danh sách Thông tin bài thi | Click | Hoàn thành |
| 3 Edit Text (Text Input Layout) | Nhập thông tin cần có cho Thông tin bài thi, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |
| Spinner Mã giáo viên | Hiển thị danh sách mã giáo viên đã tạo từ bảng giáo viên | Select | Hoàn thành |
| Spinner Mã môn học | Hiển thị danh sách mã môn học đã tạo từ bảng môn học | Select | Hoàn thành |
| Spinner Tình trạng Thông tin bài thi | Hiển thị các danh sách các trạng thái Thông tin bài thi đã được tạo sẵn | Select | Hoàn thành |

* + - **Cách vận hành**

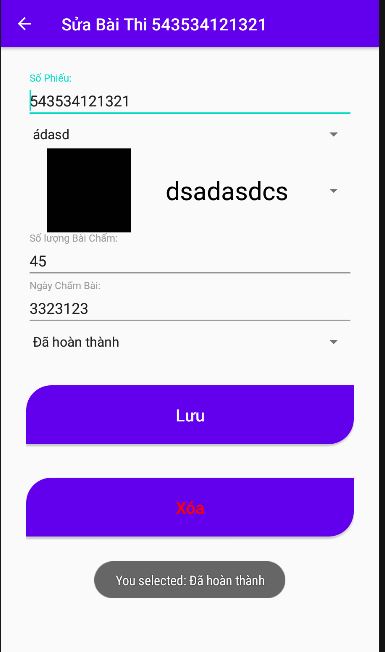
- Nhấn floating button + ở trang danh sách Thông tin bài thi để vào trang thêm Thông tin bài thi

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

- Nhấn button Thêm để thêm Thông tin bài thi vào

- Nhấn nút 🡨 để quay lại trang danh sách Thông tin bài thi

* 1. **Màn hình sửa Thông tin bài thi** 
     + **Giao diện người dùng**

****

*Hình 2.2.k. Màn hình trang sửa Thông tin bài thi*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Sửa Thông tin bài thi | 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 Thông tin bài thi | Click | Hoàn thành |
| Button “Xóa” | Xóa dữ liệu đang chọn từ RecyclerView ở trang Danh Sách Thông tin bài thi | Click | Hoàn thành |
| Button 🡨 | Quay lại trang Danh Sách Môn học | Click | Hoàn thành |
| 3 Edit Text (Text Input Layout) | Nhập thông tin cần có cho Thông tin bài thi, phải dùng Text Input Layout để đẩy tiêu đề lên | Input | Hoàn thành |
| 3 Spinner | Chọn trạng thái muốn sửa | Select | Hoàn thành |

* + - **Cách vận hành**

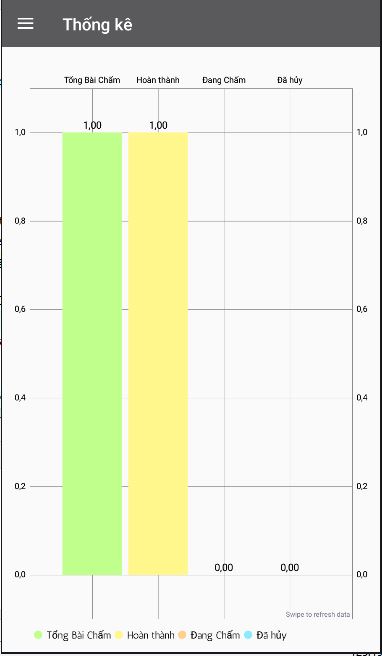
**-** Ở trang danh sách Thông tin bài thi, click vào 1 Thông tin bài thi cần sửa để vào màn hình Sửa Thông tin bài thi

- Điền các thông tin cần sửa

- Nhấn nút Lưu để lưu dữ liệu vừa sửa, hoặc nhấn nút Xóa để xóa đi Thông tin bài thi đang chọn

- Nhấn nút 🡨 để quay lại trang danh sách Thông tin bài thi

* 1. **Màn hình thống kê Thông tin bài thi**
     + **Giao diện người dùng**

****

*Hình 2.2.l. Màn hình trang sửa Thông tin bài thi*

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình Thống kê Thông tin bài thi | 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 Thông tin bài thi | Swipe | Hoàn thành |

* + - **Cách vận hành**

- Nhận dữ liệu Thông tin bài thi và tự cập nhật theo thời gian thực mà người dùng sử dụng

* 1. **Màn hình đổi ngôn ngữ** 
     + **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 |

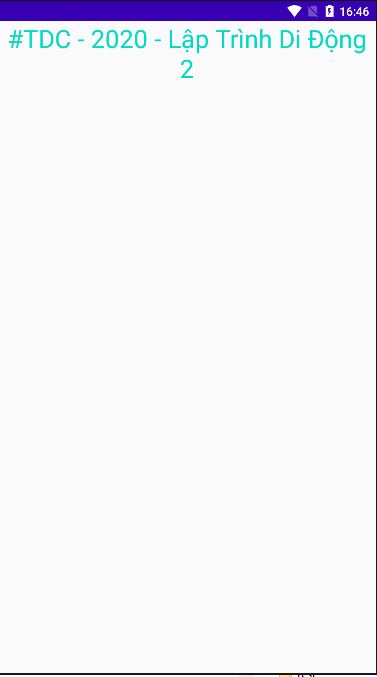
* + - **Cách vận hành**

-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**
     + **Giao diện người dùng**

****

* + - **Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu đề** | **Mô tả** | **Qui trình** | **Kết quả** |
| Màn hình About Us | - Hiển thị text view ( Thông tin môn học) | View | Hoàn thành |

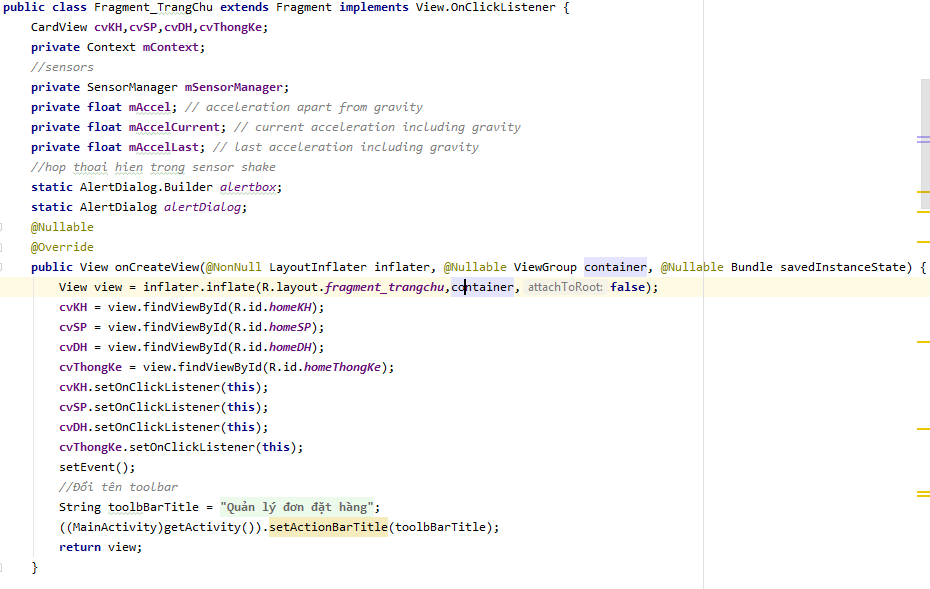
* + - **Cách vận hành**

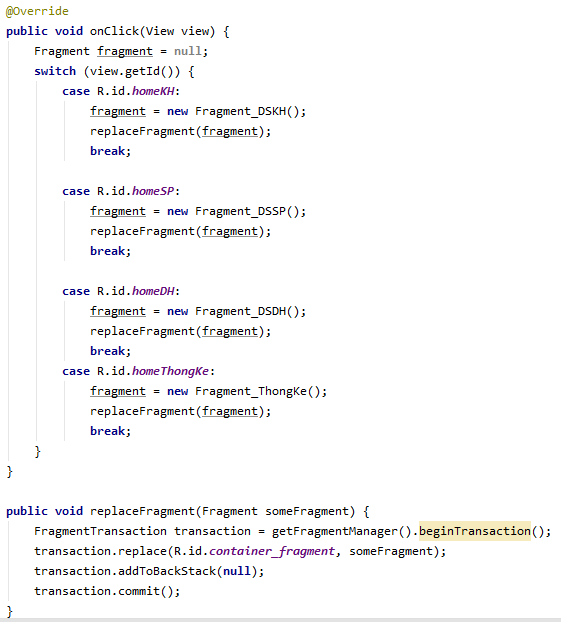
- Từ trang chính click navigation drawer chọn About Us để vào xem được Custom view

# 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.***drawerLayoutGV***);  
 **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\_gv***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_DSGV());  
 **fragmentTransaction**.commit();  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_mh***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_DSMH());  
 **fragmentTransaction**.commit();  
 }  
 **if**(menuItem.getItemId() == R.id.***nav\_ttct***) {  
 **fragmentManager** = getSupportFragmentManager();  
 **fragmentTransaction** = **fragmentManager**.beginTransaction();  
 **fragmentTransaction**.replace(R.id.***container\_fragment***,**new** Fragment\_DSTTCB());  
 **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 giáo viên

- Sử dụng fragment danh sách giáo viên

**public class** Fragment\_DSGV **extends** Fragment {  
 RecyclerView **recyclerView**;  
 FloatingActionButton **add\_button**;  
 ImageView **img\_empty**;  
 TextView **khongDuLieu**;  
 ArrayList<GiaoVien> **data** = **new** ArrayList<>();  
 DBTTChamBai **myDB**;  
 CustomAdapterKH **customAdapterGV**;  
 SwipeRefreshLayout **swipeRefreshLayout**;  
 CoordinatorLayout **coLayoutGV**;  
 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\_dsgv***,container,**false**);  
 **recyclerView** = view.findViewById(R.id.***recyclerViewGV***);  
 **add\_button** = view.findViewById(R.id.***add\_button\_gv***);  
 **img\_empty** = view.findViewById(R.id.***image\_empty***);  
 **khongDuLieu** = view.findViewById(R.id.***no\_data***);  
 **coLayoutGV** = view.findViewById(R.id.***coordinatorLayoutGV***);  
 **swipeRefreshLayout** = view.findViewById(R.id.***swipeToRefreshGV***);  
 **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** GiaoVien recoverItem = **data**.get(position);  
 *//Lưu giữ id position  
 //final int checkingId = data.get(position).getSttKH();* **data**.remove(position);  
 **customAdapterGV**.notifyItemRemoved(position);  
 *//long result = myDB.XoaItemTheoIDKH(checkingId);* **final** Snackbar snackbar = Snackbar.*make*(**coLayoutKH**, **"Đã xóa giáo viên"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Hoàn Tác"**, **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **data**.add(position, recoverItem);  
 **customAdapterGV**.notifyItemInserted(position);  
 **customAdapterGV**.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();  
 **customAdapterGV** = **new** CustomAdapterGV(getActivity(),**mContext**, **data**);  
 **recyclerView**.setAdapter(**customAdapterGV**);  
 **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();  
 }  
 **customAdapterGV**.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** (**customAdapterGV**!=**null**)  
 **customAdapterGV**.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 môn học

- Fragment ds môn học:

**public class** Fragment\_DSMH **extends** Fragment {  
 RecyclerView **recyclerView**;  
 FloatingActionButton **add\_button**;  
 ImageView **img\_empty**, **imgMH**;  
 TextView **khongDuLieu**;  
 ArrayList<MonHoc> **data** = **new** ArrayList<>();  
 DBTTChamBai **myDB**;  
 CustomAdapterGV **customAdapterGV**;  
 SwipeRefreshLayout **swipeRefreshLayout**;  
 CoordinatorLayout **coLayoutMH**;  
 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\_dsMH***,container,**false**);  
 **recyclerView** = view.findViewById(R.id.***recyclerViewMH***);  
 **add\_button** = view.findViewById(R.id.***add\_button\_mh***);  
 **img\_empty** = view.findViewById(R.id.***image\_empty***);  
 **khongDuLieu** = view.findViewById(R.id.***no\_data***);  
 **coLayoutMH** = view.findViewById(R.id.***coordinatorLayoutMH***);  
 **swipeRefreshLayout** = view.findViewById(R.id.***swipeToRefreshMH***);  
 **navigationView** = view.findViewById(R.id.***nav\_view***);  
  
 **imgMH** = view.findViewById(R.id.***imgRecyclerViewIDMH***);  
 *//Đổi tên toolbar* String toolbBarTitle = **mContext**.getResources().getString(R.string.***trangMH***);  
 ((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(), ActivityThemMH.**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);  
 **customAdapterMH**.notifyItemRemoved(position);  
 *//long result = myDB.XoaItemTheoIDKH(checkingId);* **final** Snackbar snackbar = Snackbar.*make*(**coLayoutMH**, **"Đã xóa môn học"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Hoàn Tác"**, **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **data**.add(position, recoverItem);  
 **customAdapterMH**.notifyItemInserted(position);  
 **customAdapterMH**.notifyDataSetChanged();  
 **myDB**.ThemMH(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**.XoaMotItemMH(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();  
 **customAdapterMH** = **new** CustomAdapterMH(getActivity(),**mContext**, **data**);  
 **recyclerView**.setAdapter(**customAdapterMH**);  
 **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();  
 }  
 **customAdapterMH**.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** (**customAdapterMH**!=**null**)  
 **customAdapterMH**.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.XoaTatCaMH();  
 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 Thông tin bài thi

- Fragment Thông tin bài thi:

**public class** Fragment\_DSBT **extends** Fragment {  
 RecyclerView **recyclerViewBT**;  
 FloatingActionButton **add\_button\_BT**;  
 ImageView **img\_empty**;  
 TextView **khongDuLieu**;  
 ArrayList<DonHang> **data** = **new** ArrayList<>();  
 DBTTChamThi **myDB**;  
 CustomAdapterBT **customAdapterBT**;  
 SwipeRefreshLayout **swipeRefreshLayoutBT**;  
 CoordinatorLayout **coLayoutBT**;  
 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\_dsBT***,container,**false**);  
 **recyclerViewBT** = view.findViewById(R.id.***recyclerViewBT***);  
 **add\_button\_BT** = view.findViewById(R.id.***add\_button\_BT***);  
 **img\_empty** = view.findViewById(R.id.***image\_empty***);  
 **khongDuLieu** = view.findViewById(R.id.***no\_data***);  
 **coLayoutBT** = view.findViewById(R.id.***coordinatorLayoutBT***);  
 **swipeRefreshLayoutBT** = view.findViewById(R.id.***swipeToRefreshBT***);  
 **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.***trangBT***);  
 ((MainActivity)getActivity()).setActionBarTitle(toolbBarTitle);  
 **return** view;  
 }  
 **private void** setEvent() {  
 **add\_button\_BT**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent(getActivity(), ActivityThemBT.**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);  
 **customAdapterBT**.notifyItemRemoved(position);  
 *//long result = myDB.XoaItemTheoIDKH(checkingId);* **final** Snackbar snackbar = Snackbar.*make*(**coLayoutBT**, **"Đã xóa Thông tin bài thi"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Hoàn Tác"**, **new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **data**.add(position, recoverItem);  
 **customAdapterBT**.notifyItemInserted(position);  
 **customAdapterBT**.notifyDataSetChanged();  
 **myDB**.ThemBT(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**.XoaMotItemBT(recoverItem);  
 }  
 }  
 });  
 snackbar.show();  
 }  
 })) .attachToRecyclerView(**recyclerViewBT**);  
 *//Kéo refresh list* **swipeRefreshLayoutBT**.setOnRefreshListener(**new** SwipeRefreshLayout.OnRefreshListener() {  
 @Override  
 **public void** onRefresh() {  
 hienThiDatabase();  
 **swipeRefreshLayoutBT**.setRefreshing(**false**);  
 }  
 });  
 }  
 **public void** hienThiDatabase(){  
 **myDB** = **new** DBDonDatHang(**mContext**);  
 **data** = **myDB**.listDonHangs();  
 **customAdapterBT** = **new** CustomAdapterBT(getActivity(),**mContext**, **data**);  
 **recyclerViewBT**.setAdapter(**customAdapterBT**);  
 **recyclerViewBT**.setLayoutManager(**new** LinearLayoutManager(**mContext**));  
 **if**(**data**.size() > 0){  
 **recyclerViewBT**.setVisibility(View.***VISIBLE***);  
 **img\_empty**.setVisibility(View.***GONE***);  
 **khongDuLieu**.setVisibility(View.***GONE***);  
 }**else** {  
 **recyclerViewBT**.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();  
 }  
 **customAdapterBT**.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** (**customAdapterBT**!=**null**)  
 **customAdapterBT**.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.XoaTatCaBT();  
 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 Bài Thi"**));  
 entries.add(**new** BarEntry(1f, donHangHoanThanh,**"Hoàn thành"**));  
 entries.add(**new** BarEntry(2f, donHangDangGiao,**"Đang chấm"**));  
 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 Bài Thi"**);  
 barFactors.add(**"Hoàn thành"**);  
 barFactors.add(**"Đang chấm"**);  
 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** DBTTChamThi(**mContext**);  
 **int** BaiThiHoanThanh = **myDB**.getSLBaiThiHoanThanh();  
 **int** BaiThiDangCham = **myDB**.getSLBaiThiDangGiao();  
 **int** BaiThiDaHuy = **myDB**.getSLBaiThiDaHuy();  
 **int** BaiThiTong = **myDB**.getSLBaiThiTong();  
  
 **dataSet**.removeEntry(0);  
 **data**.addEntry(**new** BarEntry(0f, BaiThiTong,**"Tổng Bài Thi"**),0);  
 **data**.addEntry(**new** BarEntry(1f, BaiThiHoanThanh,**"Hoàn thành"**),0);  
 **data**.addEntry(**new** BarEntry(2f, BaiThiDangCham,**"Đang Chấm"**),0);  
 **data**.addEntry(**new** BarEntry(3f, BaiThiDaHuy,**"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 MyHeart Shape 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 Giáo viên, Môn học, Thông tin bài thi
* Sử dụng Recyclerview + Cardview để hiển thị dữ liệu
* Splash Screen có animation, drawable
* Đã có Navigation Drawer
* Đã có Text Input Layout cho các ô input
* Thêm ảnh cho môn học
* Custom Spinner Môn học bao gồm ảnh
* Thêm trạng thái Thông tin bài thi để dễ quản lý
* 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)
* Áp dụng dữ liệu trạng thái Thông tin bài thi để tạo ra bar chart

## 4.2. Kết luận

* Những việc đã làm được
  + 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 giáo viên, Danh sách môn học, Danh sách Thông tin bài thi, Thống kê
  + Liên kết dữ liệu ở 3 bảng chính Giáo viên, môn học, Thông tin bài thi
  + 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
* Những việc chưa làm được
  + Chưa thể nhận được ảnh môn học 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.