# Activity “Đăng ký”:

## Sau khi nhấn button “đăng ký”sẽ trigger OnClickListener - Đầu tiên sẽ kiểm tra các EditText như Họ tên,Số điện thoại,Mật khẩu,Địa chỉ có null hay không sử dụng thuộc tính length và EditText Email có hợp lệ hay không sử dụng thuộc tính matches. Nếu có EditText rỗng hoặc Email không hợp lệ sẽ hiện thông báo thông qua thuộc tính setError. - Sau khi các Edit Text đều hợp lệ thì sẽ kiểm tra Email đã tồn tại trên database hay chưa thông qua thuộc tính Equals và danh sách User tại MainActivity. - Nếu đã tồn tại thì thông báo lỗi bằng setError. - Nếu chưa tồn tại thì bắt đầu push data trên Database bằng lệnh .push().setValue() và quay lại Activity Đăng nhập.

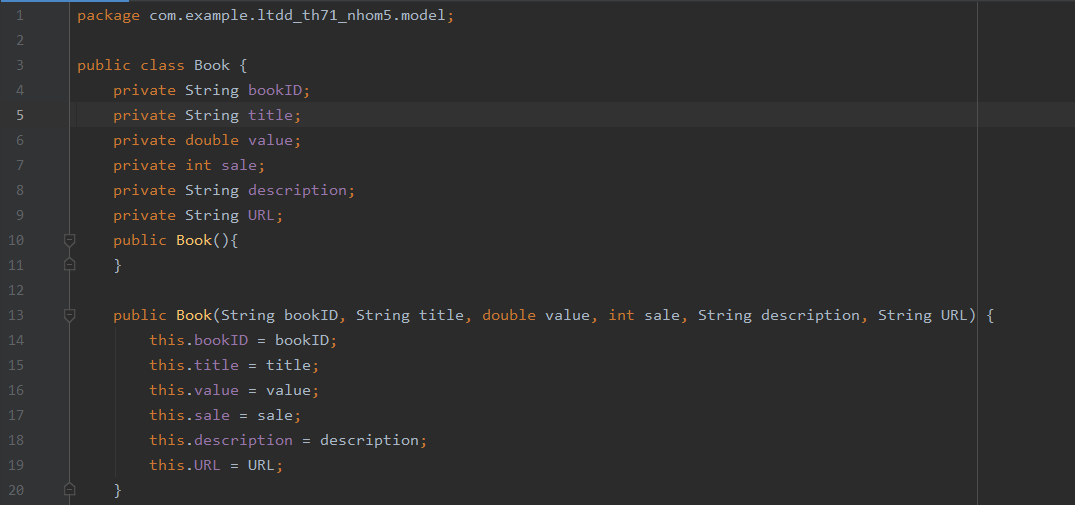
# Activity “Sửa thông tin cá nhân”:

## Thuộc tính của Checkbox được set mặc định là Unchecked. - Nếu unchecked sẽ ẩn các Edit Text để thay đổi mật khẩu. - Nếu checked sẽ hiện các Edit Text để thay đổi mật khẩu sử dụng thuộc tính setVisibility.

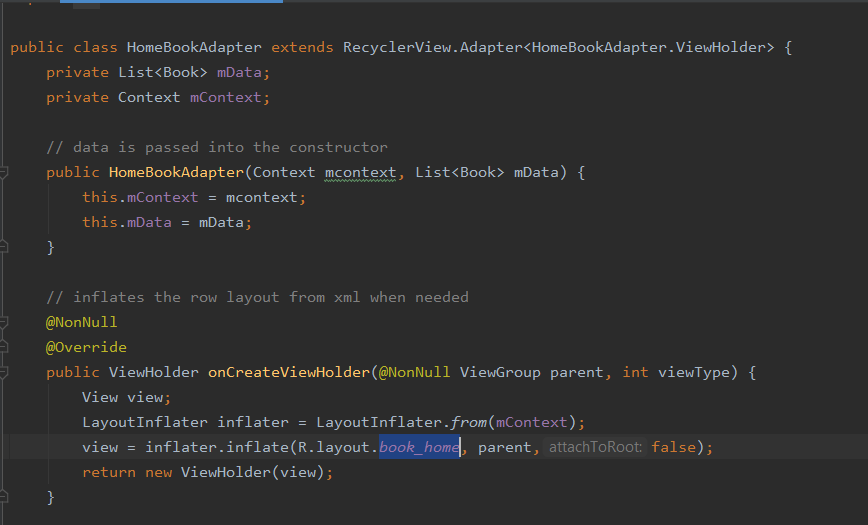
3. Chức năng xem sách:

- Click vào sách muốn xem để thấy chi tiết về sách đó.

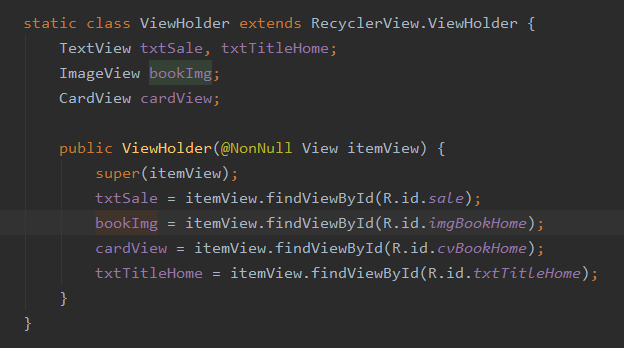
Khi lấy sách từ data về ta tạo cho nó một model để lưu trữ các thuộc tính tương ứng của nó như BookID, Name, …



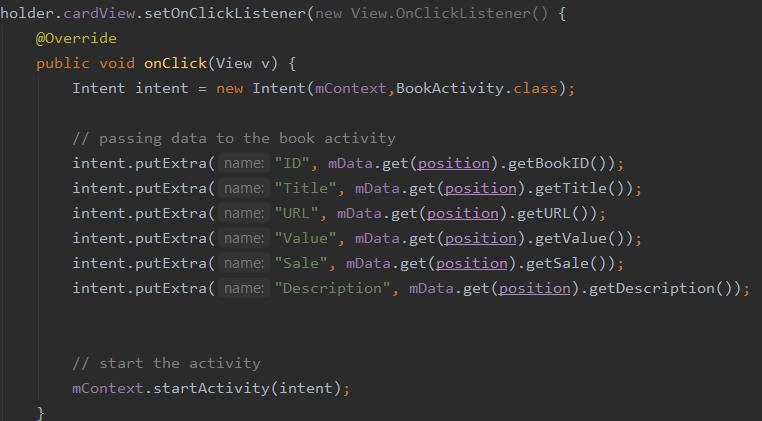
* Các cuốn sách này sẽ được đưa vào lưu trữ trong List với từng danh mục của nó. Sau đó chúng ta sử dụng RecyclerView để hiển thị các List sách này lên màn hình bằng cách tạo adapter cho Recycler View, layout và ViewHolder tương ứng cho các đối tượng sách. Vì chúng ta sử dụng RecyclerView nên adapter tạo ra phải được kế thừa lớp RecyclerView.Adapter



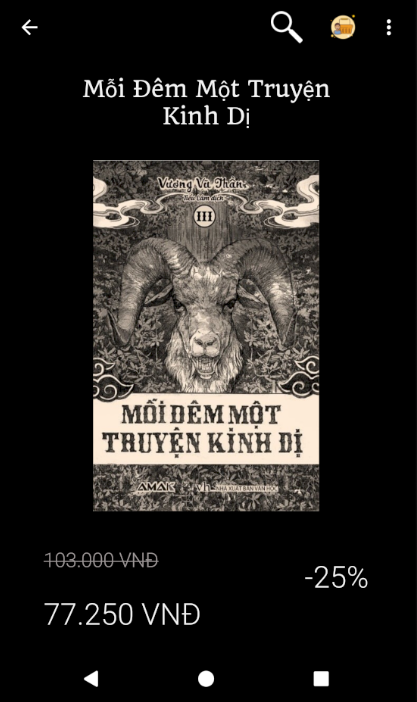




* Tiếp đến là set các giá trị cho các view có trong layout book\_home.xml và bắt các sự kiện các view này, việc này được thực hiện trong hàm onBindViewHolder Override từ lớp RecyclerView.Adapter.
* Ở đầy mình thiết kế layout của mỗi item được chứa trong 1 card view nên mình cần bắt sự onClick trên card view để chuyển đến màn hình chi tiết và them sách vào giỏ hàng. Vì màn hình chi tiết cần 1 vài giá trị để hiện thị lên màn hình nên khi tạo Intent để chuyển Activity mình dùng putExtra để truyền các giá trị này, sau đó bên Activity của màn hình chi tiết sẽ hứng Intent này bằng hàm getIntent và lấy các giá trị bằng cách sử dụng các key tương ứng khi truyền giá trị. Sau đó sử dụng các giá trị này để gán các view trên màn hình chi tiết sách.

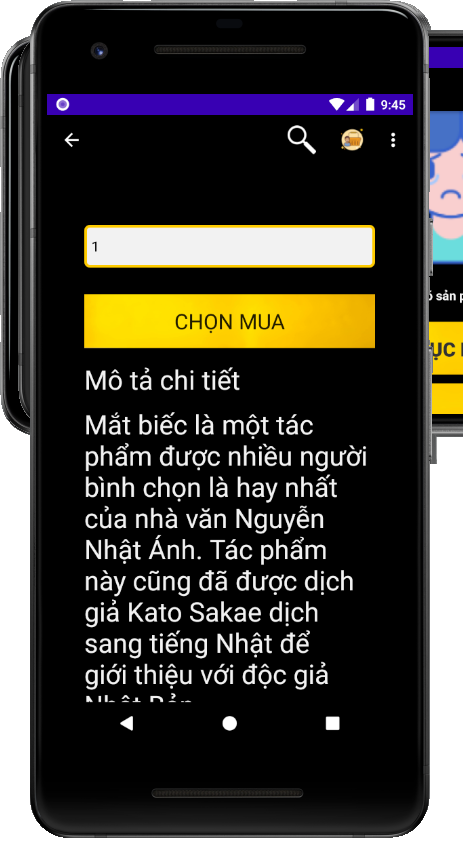


* Và đây là kết quả:

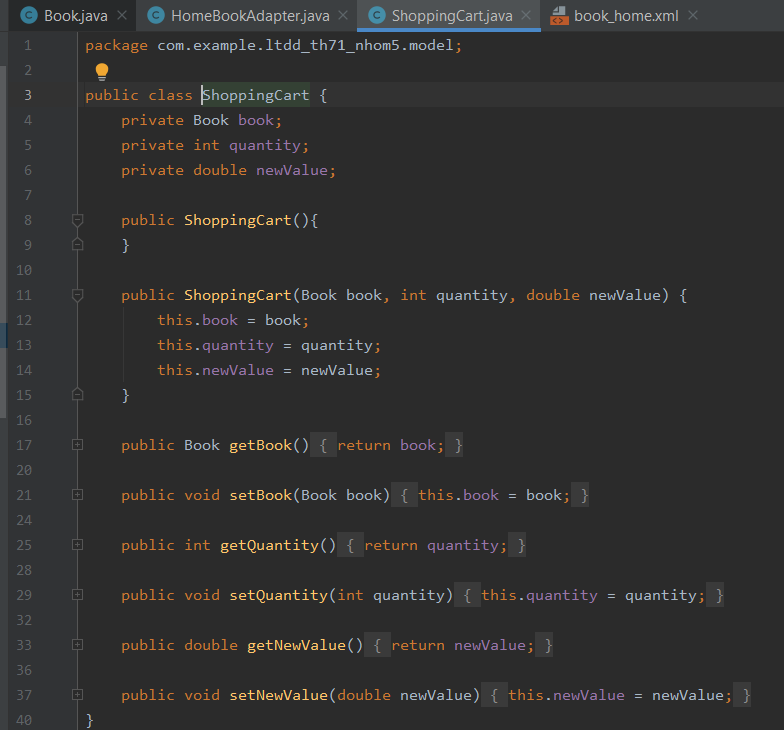


4. Chức năng thêm sách vào giỏ hàng:

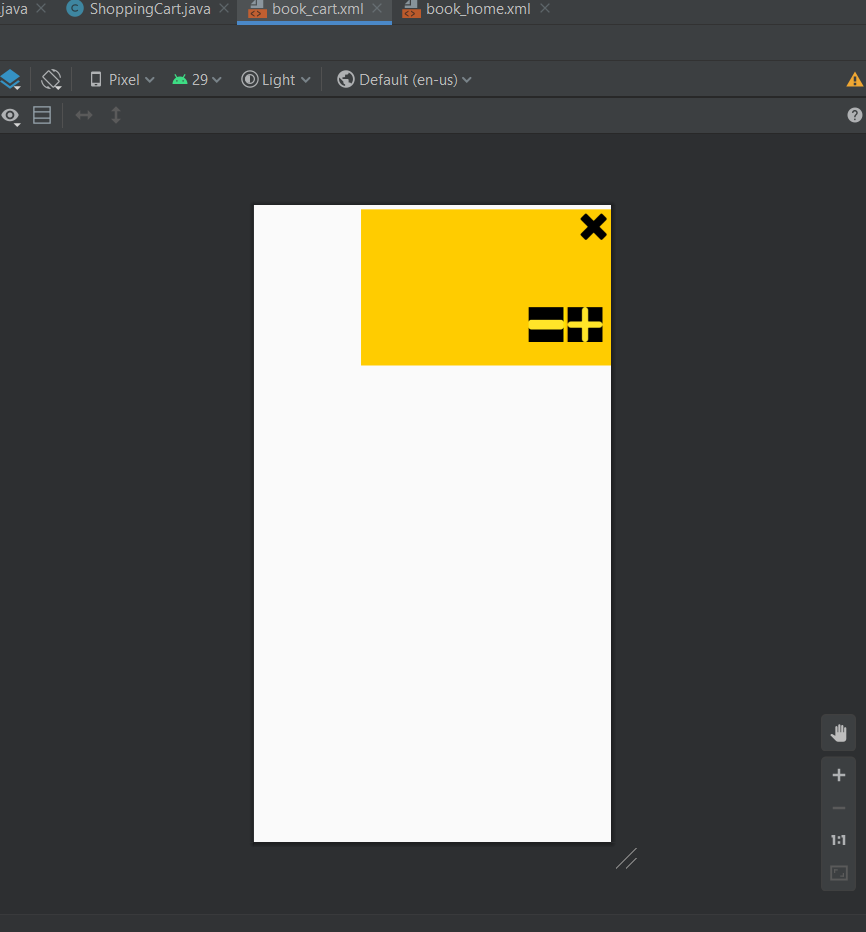
- Sau khi xem sách đã được hiển thị, bạn có thể thêm sách vào giỏ hàng (có thể tùy chỉnh số lượng).



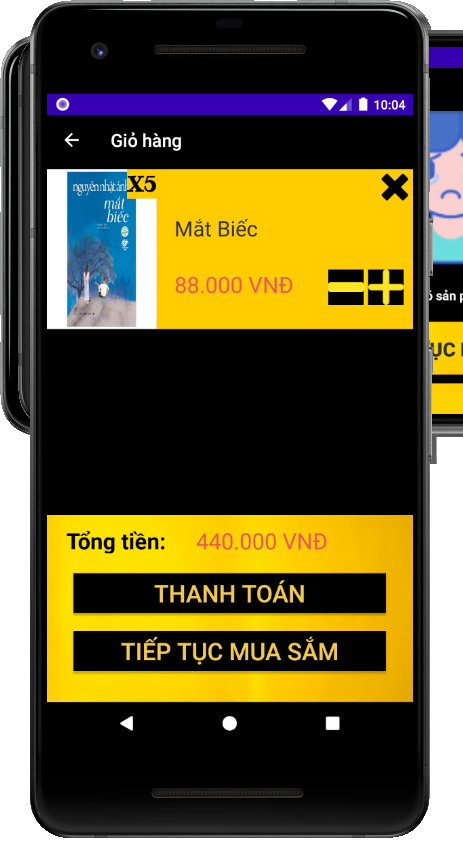
* Ở đây mình sẽ bắt sự kiện click vào Button Chọn Mua, dữ liệu tương ứng sẽ được đổ vào mảng giỏ hàng được khai báo và khởi tạo ban đầu ở MainActivity, mảng này được khai báo dưới dạng public static. Mảng giỏ hàng chứa các đối tượng dạng ShoppingCart bao gồm 1 đối tượng sách, số lượng và giá sau khi được giảm nhân với số lượng (thuộc tính này được sử dụng để tính tổng giá trị các item có trong mảng giỏ hàng)



* Khi bắt sự kiện click trên Button chọn mua mình sẽ kiểm tra xem mảng giỏ hàng đã có dữ liệu hay chưa, nếu chưa thì thêm vào 1 đối tượng shopping cart mới, ngược lại sẽ kiểm tra xem trong giỏ hàng đã tồn tại đối tượng đang them hay chưa, nếu có thì sẽ cộng dồn số lượng và tính giá trị mới cho newValue, trường hợp cuối cùng là mảng giỏ hàng đã có dữ liệu và đối tượng được thêm mới thì làm tương tự với mảng chưa có dữ liệu. Ở đây mình giới hạn số lượng cho mỗi item trong mảng giỏ hàng là 10, nghĩa là khi số lượng tối đa là 10, vượt quá số này sẽ được mặc định chuyển về 10.
* Sau đó bên Activity của giỏ hàng sẽ sử dụng RecyclerView để hiển thị các item có trong giỏ hàng. Điều này cũng tương tự với việc hiển thị sản phẩm ở màn hình trang chủ, suy ra phải thiết kế một Adapter, Layout và ViewHolder tương ứng cho item shopping cart.



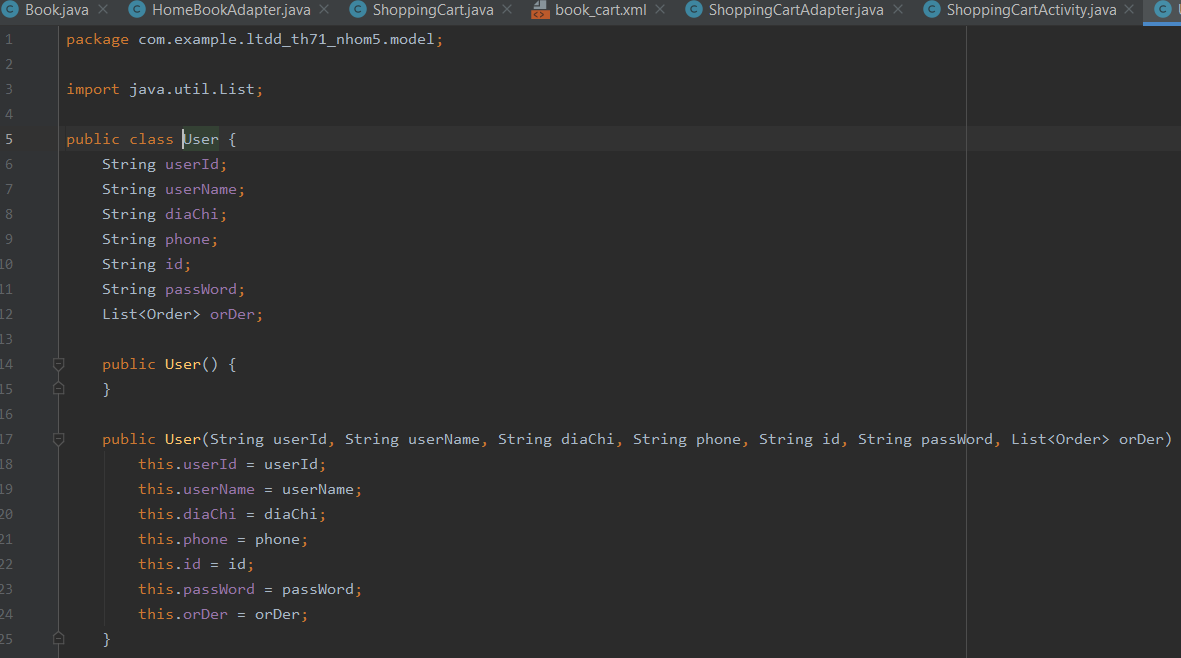
* Layout của item cart mình sẽ hỗ trơ thêm 1 vài button để thêm bớt số lượng cho item, cũng như loại bỏ các item này ra khỏi giỏ hàng. Cách gán giá trị và bắt sự kiện cho view tương ứng tương tự cũng được thực hiện trong hàm onBindViewHolder. Ở sự kiện tang giảm số lượng vẫn sẽ kiểm soát số lượng tối đa và tối thiểu cho từng item ( 0< quantity <= 10)



- Ngoài ra ở Activity của Giỏ hàng còn có một view thể hiện tổng giá trị của giỏ hàng hiện tại. Giá trị này được tính bằng cách cộng dồn giá trị newValue của từng item có trong giỏ hàng, giá này đã được trừ chiết khấu dựa theo % sale.

5. Chức năng đăng nhập: Khi thanh toán nếu người mua chưa đăng nhập thì giao diện sẽ yêu cầu đăng nhập. Sau khi đăng nhập có thể thực hiện thanh toán bình thường.

1. Ở sự kiện đăng nhập này, sau khi nhấn vào button đăng nhập , chuỗi trong 2 edit sẽ được lấy ra so sánh với List User tương ứng được tải về từ Database. (List user chứa các đối tượng có model là User)



2. Nếu trùng khớp sẽ đăng nhập thành công, sau đó có thể thanh toán bình thường.

3. Sau khi thanh toán dữ liệu sẽ được cập nhật lên database, sau đó ta có thể tiếp tục mua sắm

4. Ngoài ra còn bắt một số lỗi về nhập liệu (như để trống, khác định dạng,…)



1. Chức năng đăng kí: Nếu người dùng chưa có tài khoản thì có thể đăng kí tài khoản bằng cách nhấn vào Button “Đăng ký” hoặc “Forgot password” nếu quên mật khẩu.

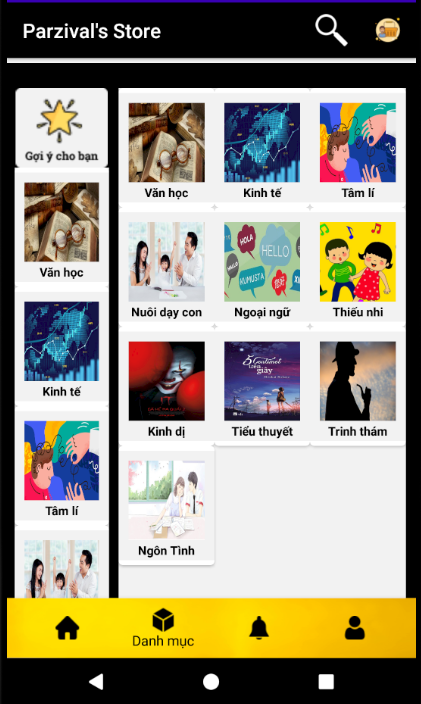
## Sau khi nhấn button “đăng ký”sẽ trigger OnClickListener Đầu tiên sẽ kiểm tra các EditText như Họ tên,Số điện thoại,Mật khẩu,Địa chỉ có null hay không sử dụng thuộc tính length và EditText Email có hợp lệ hay không sử dụng thuộc tính matches. Nếu có EditText rỗng hoặc Email không hợp lệ sẽ hiện thông báo thông qua thuộc tính setError. Sau khi các Edit Text đều hợp lệ thì sẽ kiểm tra Email đã tồn tại trên database hay chưa thông qua thuộc tính Equals và danh sách User tại MainActivity. Nếu đã tồn tại thì thông báo lỗi bằng setError. Nếu chưa tồn tại thì bắt đầu push data trên Database bằng lệnh .push().setValue() và quay lại Activity Đăng nhập.



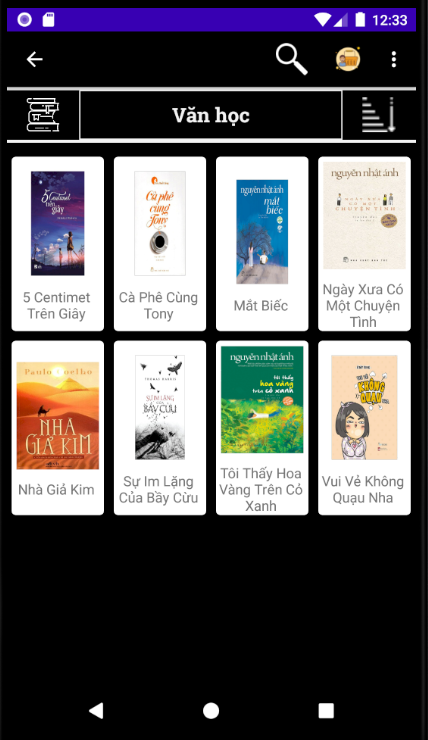
1. Sự kiện quên mật khẩu:
2. Khi người dùng quên mật khẩu, app hỗ trợ lấy lại mật khẩu thông qua email, email được nhập vào edit text, sau khi nhấn vào button lấy lại mật khẩu email này sẽ được so sánh với các tài khoản có trong database, nếu trùng khớp thì sẽ gửi thông báo, người dùng chỉ việc nhấp vào thông báo này để thay đổi mật khẩu mới. (Ý tưởng ban đầu của mình là dựa vào email để gửi link sửa mật khẩu, nhưng do trinh độ chuyên môn chưa đủ nên vẫn chưa hiện thực được)



1. Chức năng hiển thị danh mục: Button danh mục (bên cạnh Button trang chủ) dùng để hiển thị tất cả thể loại sách mà App đang có kèm theo sách theo từng thể loại (có thể sắp xếp sách theo chữ cái tăng hay giảm dần)
2. Cũng giống như việc hiển thị các đối tượng sách, các đối tượng danh mục cũng được tạo sẵn 1 model để lấy dữ liệu về và lưu vào 1 array list. Array List này sẽ được dùng để hiển thị danh mục thông qua RecylerView, Adapter và Layout tương ứng.



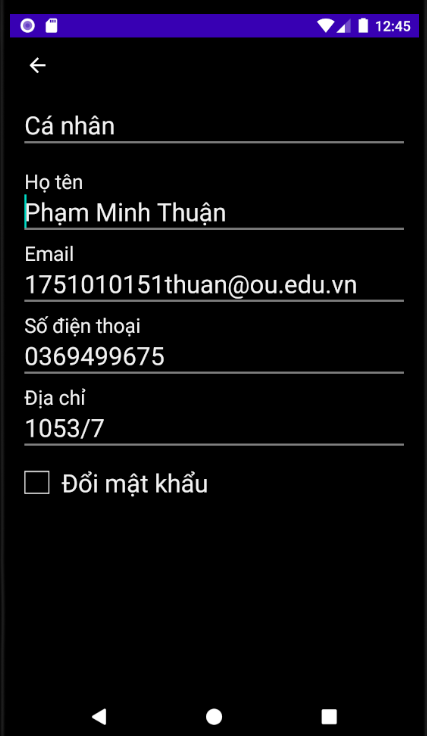
1. Khi nhấp vào các danh mục này sẽ xem được các loại sách tương ứng với từng danh mục



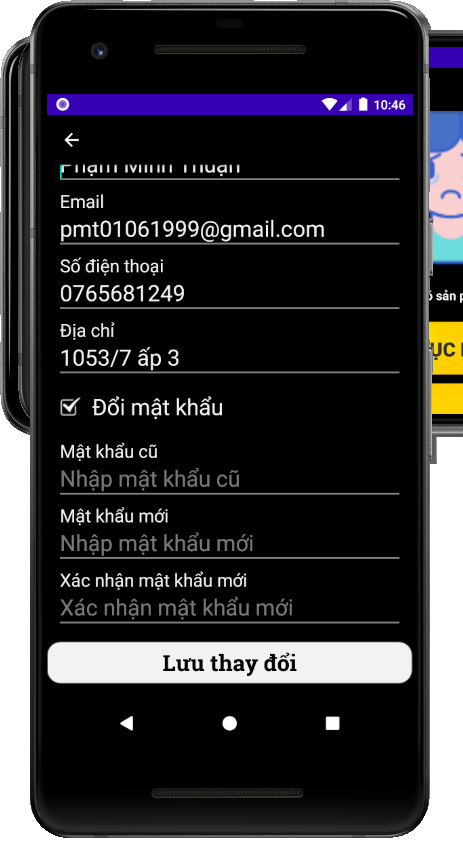
1. Ở màn hình sách theo từng danh mục này mình có hỗ trợ sắp xếp theo tên sách tang dần và giảm dần, khi click vào button sắp xếp ở phía trên bên phải màn hình các đối tượng sẽ được sắp xếp theo thứ tự tương ứng, điều này được thực hiện thông qua lệnh Colllections.sort(…), adapter sau đó sẽ cập nhật lại Data thông qua lệnh adapter.notifyDatasetChanged()
2. Chức năng hiển thị thông tin người dùng:
3. Sau khi đăng nhập thành công, thông tin người dùng sẽ được lưu lại bằng cách tạo 1 biến currentUser để lưu lại (Model User đã được tạo sẵn từ trước).
4. Biến này sẽ dùng để xử lý các sự kiện bên fragment Cá nhân như hiển thị thông tin, hiển thị số hóa đơn đã thanh toán, các sản phẩm đã mua.



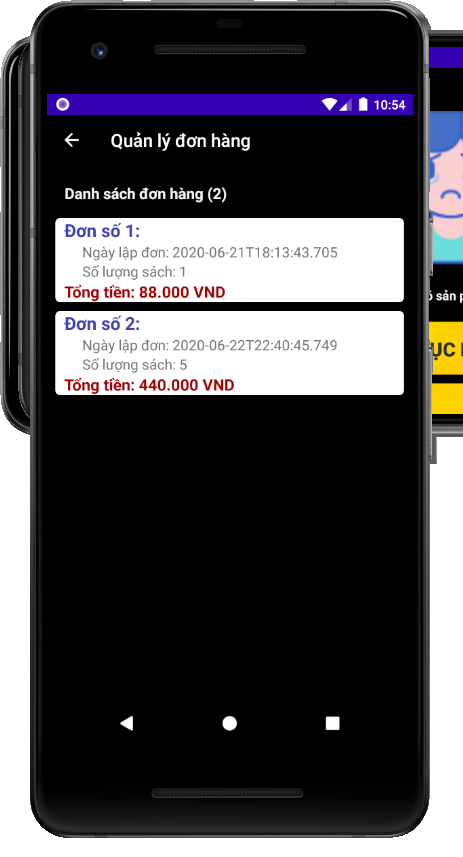
1. Khi nhấp vào biểu tượng hình người phía trên cùng của Layout Fragment Cá nhân, các thông tin tương ứng của người dùng sẽ được hiển thị ra bằng cách lấy thuộc tính của biến currentUser hiển thị lên các View tương ứng. Tại đây ta có thể chỉnh sửa thông tin cá nhân và đổi mật khẩu



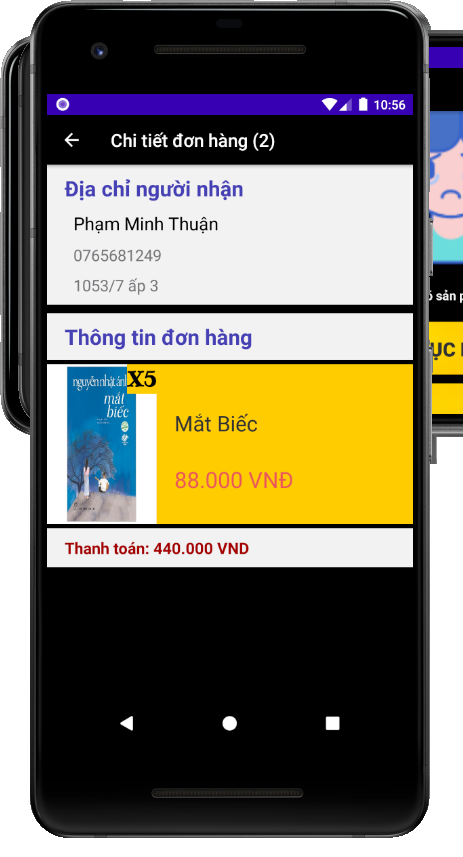
1. Khi check nút đổi mật khẩu sẽ thấy các edit text tương ứng.Khi bấm vào nút Lưu thay đổi, dữ liệu từ các edit text được kiểm tra điều kiện hợp lệ, bao gồm không được trống, định dạng, mật khẩu cũ không trùng khớp, mật khẩu mới trùng mật khẩu cũ và xác nhận mật khẩu mới không trùng khớp. Trường hợp tất cả đều hợp lệ, dữ liệu người dùng sẽ được xử lý và đưa lên database, biến user và listUser cũng được cập nhập lại để xử lý các sự kiện tiếp theo.



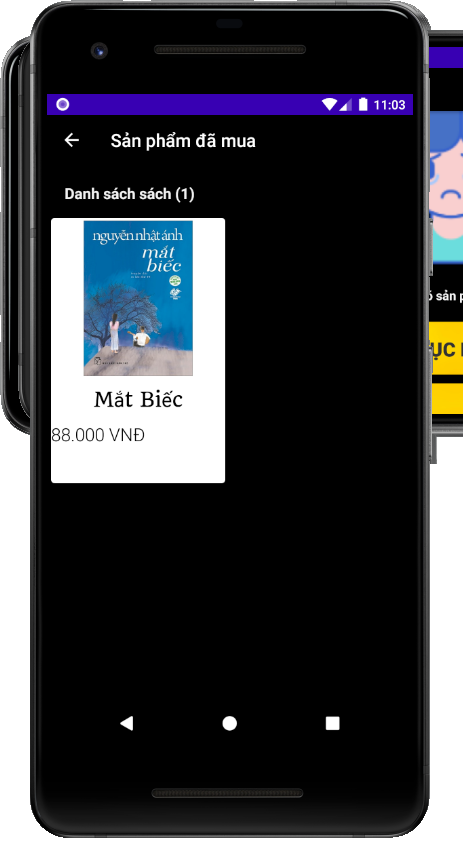
1. Chức năng quản lý đơn hàng:
2. Khi người dùng đăng nhập vào hệ thống, mình sẽ tiến hành lấy list Order tương ứng của người dùng đó, khi nhấn vào mục Quản lý đơn hàng bên dưới thông tin cá nhân, các đơn hàng tương ứng của người dùng đó sẽ hiện ra. Ở sự kiện mình cũng sử dụng recycler view, layout và view holder tương ứng để hiển thị danh sách các đơn hàng



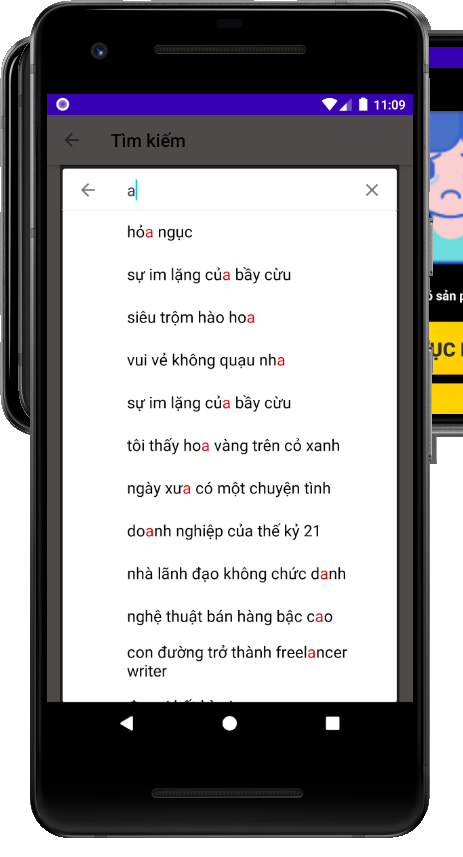
1. Khi nhấp vào đơn hàng sẽ hiển thị ra các sản phẩm tương ứng với đơn hàng đó dựa vào Order ID được truyền từ Form Quản lý đơn hàng sang. Form này hiển thị thông tin người nhận, danh sách sản phẩm, tổng tiền tương ứng với hóa đơn đó.

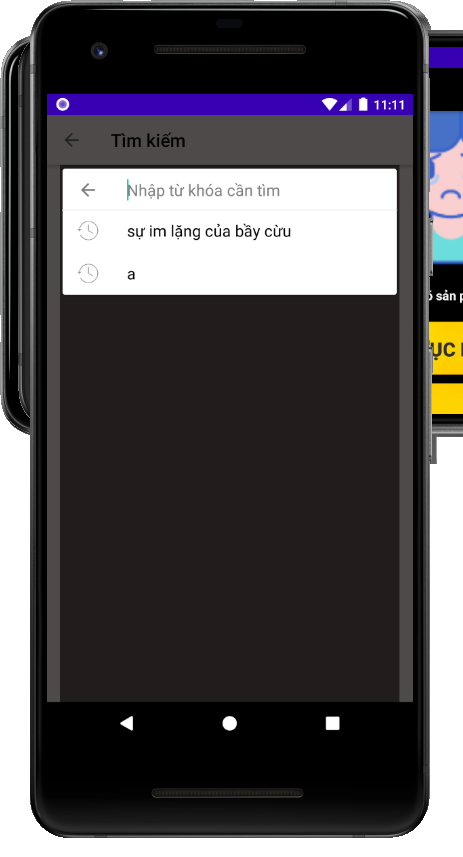


1. Chức năng thống kê các sản phẩm đã mua:
2. Nhấp vào mục Sản phẩm đã mua sẽ hiện ra các sản phẩm đã được giao dịch trước đó, sự kiện này thực hiện dựa trên list Order của User tương ứng, chỉ lấy đối tượng sách và sử dụng SearchResultAdapter để hiển thị lên màn hình.



1. Chức năng tìm kiếm:
2. Chức năng này sử dụng FloatingSearchView có hỗ trợ gợi ý là và lưu lại lịch sử tìm kiếm, để sử dụng được chức năng này ta phải bắt sự kiện tương ứng của nó như setOnQueryChangeListener(oldQuery, new Query) – thay đổi query (cập nhật lại gợi ý), setOnSearchListener - bấm nút search hoặc chọn trên gợi ý, setOnFocusChangeListener – focus trên search view (hiển thị lại lịch sử), setOnBindSuggestionCallback – đổi màu và cập nhật icon cho gợi ý (gợi ý trùng khớp và lịch sử tìm kiếm)





1. Sau khi bấm nút tìm kiếm, query sẽ được gửi sang form tiếp theo để thực hiện tìm kiếm. Ở đây mình sử dụng lại adapter SearchResultAdapter để hiển thị kết quả, chức năng sắp xếp dựa trên tên sách và giá tiền chưa giảm thông qua hàm Collections.sort(..)



1. Chức năng thông báo khuyến mãi mỗi ngày:

* Nghiệp vụ gởi thông báo cho người dùng mỗi khi có thông báo về giảm giá hoặc các sản phẩm mới. Ta có thể tạo 1 đoạn thông báo tại chức năng cloud messaging của firebase để gởi thông báo đến các máy đang sử dụng ứng dụng. Ngoài ra, ta có thể tuỳ chỉnh thời gian, chu kì gởi. Khi nhấp vào thông báo App sẽ tự động được khởi động