## BAIT2073 Mobile Application Development

### Task Description

Name            :        Tang Sharren                                    ID        :21WMR01086
Programme    :        RDS2S2                                           Group :3

Instruction: Answer **ALL** the questions. You may insert extra pages.

**1.        Please briefly describe the module(s)/function(s) you engaged in the assignment.**

My modules are
 (i)All Orders,
 (ii)All Listings ((iii)Edit Listing &  (iv)Add Listing),
 (v)Listing Sold(named as OrderStatusFragment),
(iv)Home Fragment

# (i)All Orders

By clicking on the profile button from the bottom navigation bar, users can access their dashboard. From the dashboard, users can view all orders by clicking on the 'click here to here' for all orders. In all orders pages, it will display all the orders and it is sorted by name ascendingly by default. Users can search the order by product name. Also, they can choose to sort the order by name,carbon footprint saved,price or order date. The auto-scrolling text view under the search bar displays the total carbon footprint saved for all orders excluding canceled orders.

```
//Total carbon saved for all orders excluding canceled order
var totalCarbonSaved = 0.0
for (item in list) {
    if (item.status != "Canceled") {
        totalCarbonSaved += item.carbon!!.toFloat()
    }
}
```

Users can cancel the order if the current order status is either ordered or dispatched.
After cancellation, the product will be available to purchase again.(As after placing order,product in order will be unavailable)
Users can view the product details by clicking on the order, but users can only add the product to cart if the product is available.

### (a)Adapter used in AllOrderFragment
-AllOrderAdapter is used by the recycler view in AllOrderFragment that is populated with all the orders of current users. The layout of the adapter is all_order_item.xml.In the adapter, the click event of the button on each order item is handled.

### (b)Model used in AllOrdersFragment
-AllOrderModel is a data class for each item in the recycler view which is populated with orders to get the orders as an array list of AllOrderModel.

# (ii)All Listings

From the dashboard, users can view all listings by clicking on the 'click here to here' for all listings. . Listings are products that are uploaded by users for other users to purchase. In all listings, it will display all the listings and it is sorted by name ascendingly by default. The recycler view to display the list of listings is populated with the current user's listing(each product have its corresponding seller email)

Users can search the listings by product name or category name. Also, they can choose to sort the order by name,carbon footprint saved, or price.

Users can select a preferred category to display. If there's no listing for the selected category,it will inform the user that no listing for this category is available.

# (iii) Edit Listing

Users can edit the listings by clicking on the edit button or the listing. Users will be able to edit the listing only if the listing(product) is not purchased by a user. In the edit listing page, it will preload all the fields(name,description,carbon footprint saved,price,category,cover image,product image) for the listing user clicked. As all the fields are required, validation will be done after the user clicks the update button so that no fields will be left empty. Also, users can delete old product images by clicking on the red cross on the image. New product images will be added to the end of the list. After clicking the update button, the product record in Firestore will be changed accordingly.

# (iv) Add Listing

In all listings, users can add a new listing by clicking on the + floating button. In the add listing page, users must fill in all fields(name,description,carbon footprint saved,price,category,cover image,product image) before clicking add product. As all the fields are required, validation will be done after the user clicks the update button so that no fields will be left empty. The listing added can be found in all listings and all product pages.

### (a)Adapter used in ListingFragment, EditListingFragment & AddListingFragment
-CategorySearchAdapter is used by the recycler view in ListingFragment and also AllProductFragment that is populated with all the categories available. The layout of the adapter is layout_category_search.xml, so each item in the recycler view is using this layout

-AllListingAdapter is used by the recycler view in ListingFragment that is populated with all listings of current users. The layout of the adapter is all_product_list.xml.

-EditProductImageAdapter is used by the recycler view to display the list of product images in the EditListingFragment. The layout of the adapter is edit_product_image_item.xml, so each item in the recycler view is using this layout. There is a red 'X' in each of the product images, when the 'X' is clicked, it will delete the specific product image.

-AddProductImageAdapter is used by the recycler view in AddListingFragment to display the list

of product images.The layout of the adapter is image_item.xml. It will load the images selected by users from their gallery.

**(b)Model used in ListingFragment, EditListingFragment & AddListingFragment**
-AddProductModel is a data class for each item in the RecyclerView which is populated with products(used by AddListingFragment to store the new listing to firebase and used by AllProductFragment,CategoryFragment,Home Fragment,ListingFragment to get the products as an array list of AddProductModel)

-AddProductModel has some additional properties that the ProductModel does not have(ProductModel is a data class for each item in the RecyclerView that is populated with products which are used by CartFragment and ProductDetailFragment to add the product to cart).These properties are useful for displaying more information about the product or identifying the user who added it. However, they might not be necessary for storing the product in the database, which is why the ProductModel has fewer properties.

-CategoryModel is a data class for each item in the RecyclerView that is populated with categories, it is used by AllProductFragment,EditListing,Home,Listing,AddListing to retrieve the data for categories(containing category name and the icon for the category) from Firebase

# (v)Listing Sold(named as OrderStatusFragment)

From the dashboard, users can view all listings by clicking on the 'click here to here' for all orders. Listings are products that are uploaded by users for other users to purchase.

Users can update the order status for the listing sold. After a user successfully place an order, the order status will be 'Ordered', then the seller user will see this order from the listing sold page. User can dispatch,deliver or cancel orders by clicking on the button.

Some rules applied to the order status are as follows:
Users can dispatch the order only if the current order status is ordered.
Users can deliver the order only if the current order status is dispatched.
Users can cancel the order if the current order status is either ordered or dispatched.

**(a)Adapter used in OrderStatusFragment**
OrderStatusAdapter is used by the recycler view in OrderStatusFragment to display all listings sold. The layout of the adapter is order_status_item.xml. This adapter handles how the content of the button in each order item will change according to the order status.

**(b)Model used in OrderStatusFragment**
AllOrderModel is a data class for each item in the RecyclerView which is populated with orders for the user's listing. It is used to get those orders as an array list of AllOrderModel

# (vi)Home Fragment

This will be the first screen user will see after they sign in to the app. The slider on the top is used to encourage the user to purchase the pre-loved item with some quotes and environmental effects of new products. By clicking on the slider, it will display all products.
Users can choose the product category they want to see. In the category products page, it will display the products for that category only.
Users can see all the products on the home page. By clicking on the product or the price button, it will show the product details. Also, users can see all the products by clicking on the See All, this will navigate to the all products page. In conclusion, users can see all the products from the home page and also by clicking on the slider or See All.

### (a)Adapter used in HomeFragment
-ProductAdapter is used by the recycler view in HomeFragment to display all products. The layout of the adapter is layout_product_item.xml. This adapter handles the click event of the price button and the product item.

-CategoryAdapter is used by the recycler view in HomeFragment to display the list of all categories. The layout of the adapter is layout_category_item.xml.This adapter handles the click event of the category item so that it will navigate to the category products for the selected category.

### (b)Model used in HomeFragment
-AddProductModel
-CategoryModel


**2.     What are the strengths of the modules/functions created by you ?**
-Users can search orders and listings using product names. This helps users avoid scrolling through the long list of products,orders,listings to find what they are looking for.

-Users can sort the result by name,price,carbon footprint saved, and order date ascendingly

-Users can edit their listing based on the details they saved previously, as previous listing details will be preloaded in the edit listing fragment.

<u>**Appendix D**</u>

-In listings sold, the content of the buttons will be changed according to the current order status. If the status is ordered, the button will be CANCEL and DISPATCHED.If it is dispatched, the button will be CANCEL and DELIVERED. When the order is delivered, it will show an unclickable button written with 'ALREADY DELIVERED'. When the order is canceled, it will show an unclickable button written with ; CANCEL'

-In all order and all listing, if the user sorts after searching, then it will sort based on the search result.

-In home fragment, when the user is in offline mode, it will load the page from the cached local database.

**3.      What are the weaknesses of the modules/functions created by you?**
-User cannot choose to sort the result descendingly, it is  only sorted ascendingly for sake of simplicity.

-In EditListingFragment, users can only add 1 product image from their gallery at a time, this is time consuming.

-There is a loading time every time the users update the listing, add new listing, update the order status(in all order or listing sold). This is time consuming.

-If a user deletes his listing(product) when another user is scrolling all products, another user will still see the deleted product.It will only update all products after the user exits the all products page and revisit the product page.

-In all orders, if the user canceled the same item multiple times, the order for the same product(but different order id and order date)will be shown many times. This is confusing for users to see.

-Active internet connection is required to use this app, or else it might force stop as it cannot load the content from firebase.

-The fragment state is not restored when back pressed.

-Deleting the product image will not delete it from the firebase storage, the image will only be deleted from the firestore. The unused product image takes up spaces.

-The listing deleted cannot be restored, there should be a unlist button to make the listing unavailable to purchase only.

4.      **What have you learned in doing this assignment**?

-I learned how to use the navigation component, how to pass data between fragments with safeargs, how to pass data with shared preferences,how to make an auto scrolling text view, how to make a search function and also sort the search result.

-I learned how to use firebase to perform CRUD (create,read,update and delete) the data.

-I understand how to use a recycler view with an adapter and populate it with data from firebase.

-I finally familiarized myself with how to use GitHub to share my project. It is very convenient to see the changes we made since the last commit.I find it helpful that it lets us rollback to the last commit point.

-I learned how to debug the app using the stack trace and logcat with the help of Log.d

**5.      What are the challenges, if any, faced by you while working on this assignment?**
-I don't fully understand the concept by just reading the notes, I have to watch tutorial videos online and do some hands-on practice to familiarize myself.

-I find it challenging to design the app from scratch. After referring to the UI of other apps and some Youtube videos, I finally got some idea of the fragment layout and drafted it on paper.

-It is time consuming to edit a constraint layout, especially when I delete an element from the constraints layout, other elements will be affected and other elements just disappear from the layout until I correct its constraint.

-I find it difficult to debug the app at first,then I figured out I can use Log.d to display the variable we get and also we can get the error by just searching the keyword exception in logcat, it saves me a lot of time.

-I cannot run the app using the emulator in Android Studio due to the low RAM of my laptop, so I must connect my phone via usb to run the app, my laptop still lagged when running the app on my phone, this wasted some time. This makes me unable to run the app on different devices available in Android Studio.

-Sometimes the ADB(Android Debug Driver) is not working as usual, making me unable to run the app on my own device

-I can only debug the app when I have an active internet connection making me unable to code it anytime anywhere.

-I have to clean the projects every time before I run the app else it cannot work as usual.

**<u>Appendix D</u>**

Signature:                                                          Date:14 May 2023