Mobile Application Development

Assignment 1

Report

Name: Alastair Kho

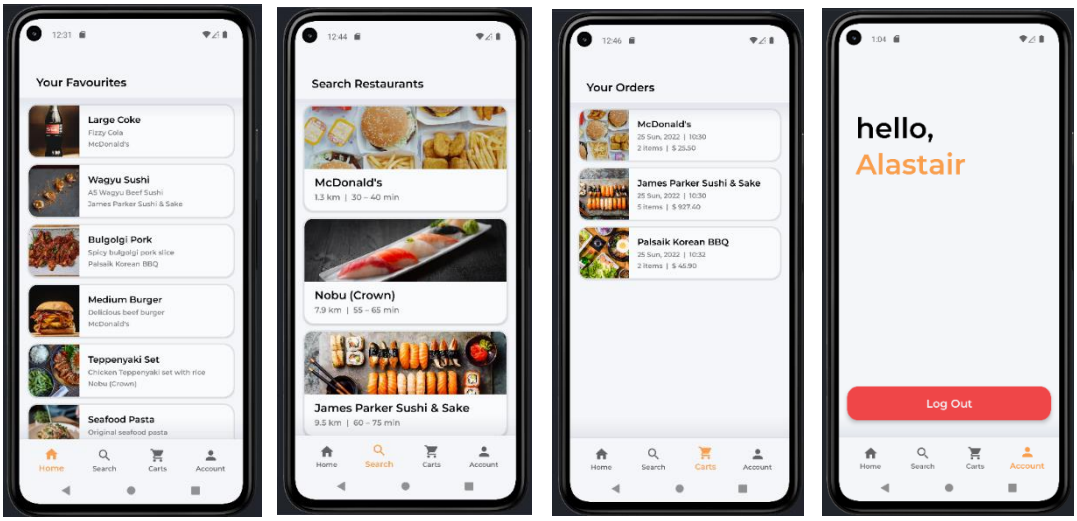
Student ID: 20214878

Introduction

Mobile applications are an essential aspect of society. This application is designed to be a food ordering application that connects users to restaurants, enabling users to order food through the application. The application is made using Android Studio with Java and is intended for newer and faster mobile phones. Users are able to search for items and restaurants on the main UI fragments, such as “home” and “search”, and place an order through a dialog fragment. The finalised order will appear in the order history under the “carts” section. Users can also manage their account (log in or out) in the ”account section. This application only consists of one activity and utilises a navigation bar to navigate between each fragment, and dialog fragments that pop up for login and ordering systems. This application is also functional with both tablets and phones. In the development of this application, I worked by myself from the initial UI drafting and design until the final application and contributed 100% of the work.

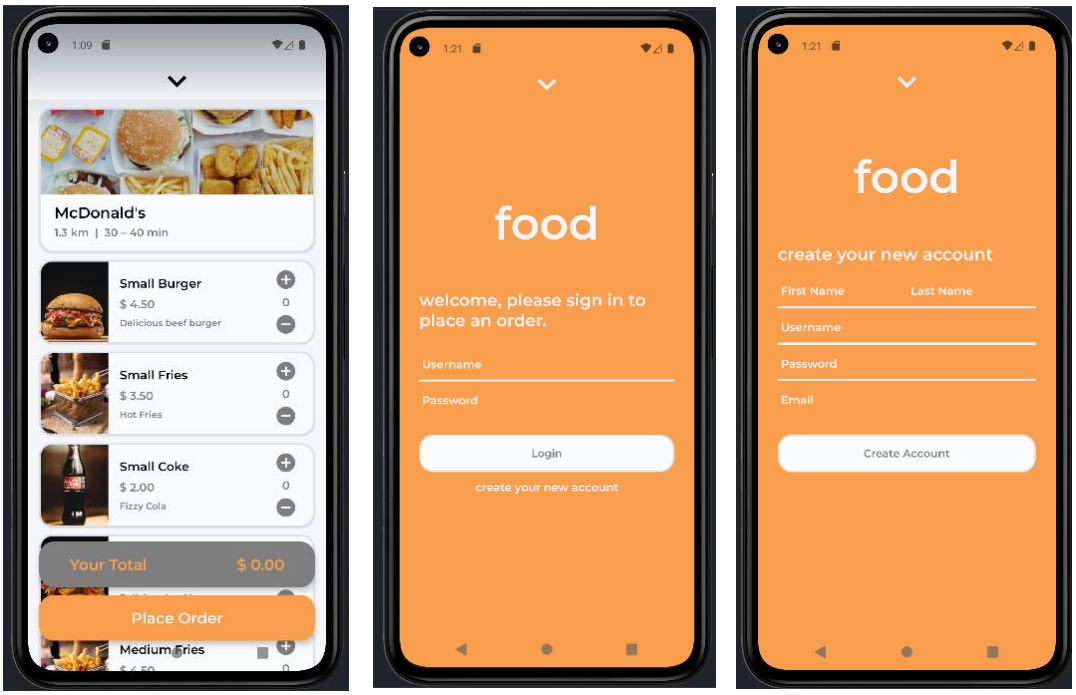
App Navigation

The user must use the bottom navigation bar to navigate between the views. The home screen is the landing page after the splash screen. This home screen presents the users with a random list of items from different restaurants. This set of items will remain the same until the application is closed and rebooted. The application draws all data from an initialised database that inserts data on first boot of the app.



The search section enables the user to search items by restaurant. This section provides the user with a list of restaurants, in which when a restaurant entry is clicked, a dialog for order will appear. The carts section in the navigation bar lets the current user view their past ordering history. Clicking on the entry will bring a dialog fragment up that presents a breakdown of the order. A user must be logged in for their respective order history to show; otherwise, a login dialog will appear, prompting the user to log in. This section will be blank and show no past orders if a user is not logged in. The account section will also prompt a user to log in if there are currently no logged-in users. If a user is logged-in, the customer will be presented with a log out button should they wish to change accounts.

Clicking an entry from the home or search view will reveal a dialog that enables you to order from a restaurant. The user is able to select a food by pressing the plus and minus signs on the right of the entry. The number in between the buttons shows the serving in the current cart. Once the user is happy with their current cart, they can press the place order button, in which will demand for the user to log in (if not already logged-in) or automatically dismiss the dialog tab if the order is processed. If the order is processed, it will appear in the order history (under 'Carts' tab). This dialog is also dismissible by swiping or tapping from the arrow pointing down (situated at the top of the dialog fragment) or pressing the back button (swiping left to right or vice versa with newer API versions). Dismissing any dialog fragments (by any means other than pressing the "place order" button) will result in the cart being cleared, meaning item quantities are required to be re-entered as the cart does not save if the user exits this menu. This is intended to prevent confusion whilst ordering.



The login dialog fragment requires a valid username and password. This fragment is also dismissible via the top arrow button (it is the same function as the ordering dialog fragment). If the user requires, a new account can also be created by pressing "create your new account" under the login button. A new account requires a unique username (not existing in the database), as well as a valid and unique email (must be formatted correctly with "@" and the domain name, i.e., 20214878@student.curtin.edu.au).

Dependencies

This app is compatible with API 31 (Android 12.0 S) and onwards. Although Android Studio suggests this app runs on 13.5% of devices, it is considered as a large number of users given the size of android users. As of 2021, there are 3 billion android users (The Verge, 2021), this places the current marketable audience at 405 million. Furthermore, the push for newer and faster devices results in more recent mobile operating systems being used, thus justifying the app's minimum API version.

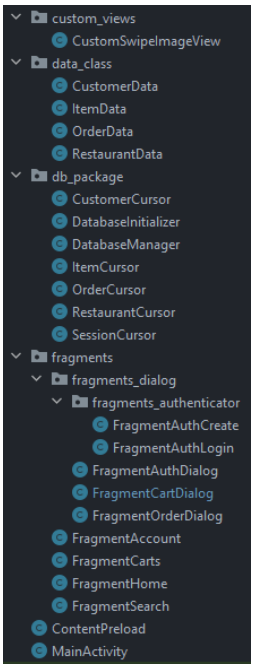
```
implementation 'androidx.appcompat:appcompat:1.5.1'
implementation 'com.google.android.material:material:1.6.1'
implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test.ext:junit:1.1.3'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

/* Other Added Implementations*/
implementation 'androidx.core:core-splashscreen:1.0.0'
```

The app uses the core dependencies added by Android Studio, and additionally a splashscreen dependency. This splashscreen implementation enables the app to greet the users with the app logo and the app's thematic colors (Android Developers, n.d.). Additionally, the device must consist of enough storage space to hold the image content of the images used in the application.

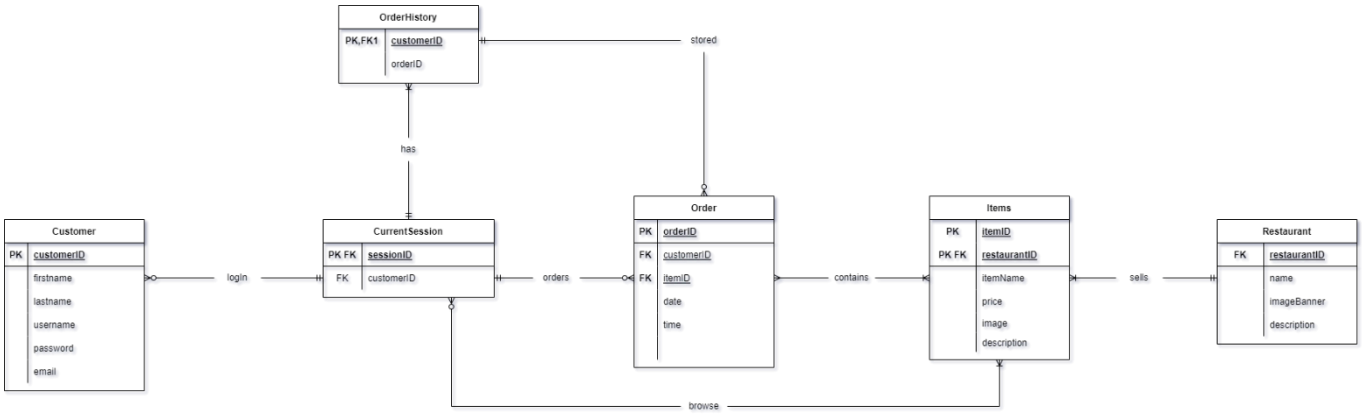
Overall Source Structure

The overall source structure is split into different sub packages. Custom views consist of a custom ImageView that allows a swiping gesture to be registered (used for dialog fragment dismissing button). "data_class" package consists the data object classes used in the application. The "db_package" consists of core database management source code, including setters and getters to interact with the database. The "fragments" package consists of all of the relevant fragments inflated in their corresponding views. The "ContentPreload" java file consists of arrays of items and restaurants that loads data (occurring once at the first launch of app) into the database, and finally the "Main Activity" manages the core navigation view.

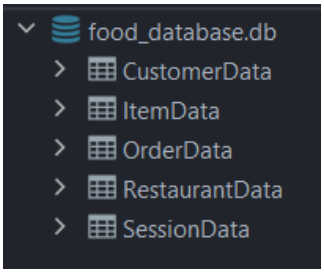


Database Schema

The following diagram presents the database schema used in the database. The OrderHistory entity is not implemented and is presented in the diagram for abstraction purposes.



The implemented database consists of a CustomerData, ItemData, OrderData and RestaurantData tables to hold their corresponding entries. The SessionData table is a single-entry table that keeps a record of the current user. This enables the same user to remain logged on even when the app is closed.



Git Logs

Before the final version was ported to another project, these were the Git Logs.

Adding Preloaded Data

For new restaurants wishing to participate in the app, to add new data, a developer must follow the steps listed below. Each item must be assigned to a restaurant. An item cannot exist without a restaurant as the ItemData contains a constraint that references a foreign key to the restaurant.

Adding Restaurants

Steps to add a new Restaurant

1. Locate the ContentPreload Java file in the directory
2. Locate the ArrayList named “preloadedRestaurants”

```
private static final ArrayList<RestaurantData> preloadedRestaurants = new ArrayList<>( Arrays.asList(
        new RestaurantData( restaurantID_PK: "00001", restaurantName: "McDonald's", restaurantDesc: "1.3 km  |  30 - 40 min", imageRef: "pexels_ready_made_4021944"),
        new RestaurantData( restaurantID_PK: "00002", restaurantName: "Nobu (Crown)", restaurantDesc: "7.9 km  |  55 - 65 min", imageRef: "pexels_rajesh_tp_2098143"),
        new RestaurantData( restaurantID_PK: "00003", restaurantName: "James Parker Sushi & Sake", restaurantDesc: "9.5 km  |  60 - 75 min", imageRef: "sushi_set_on_stone_slate"),
        new RestaurantData( restaurantID_PK: "00004", restaurantName: "KFC", restaurantDesc: "0.7 km  |  25 - 30 min", imageRef: "kfc_image"),
        new RestaurantData( restaurantID_PK: "00005", restaurantName: "Ciao Italia", restaurantDesc: "6.9 km  |  55 - 65 min", imageRef: "pexels_eneida_nieves_905847"),
        new RestaurantData( restaurantID_PK: "00006", restaurantName: "Palsaik Korean BBQ", restaurantDesc: "10.9 km  |  70 - 80 min", imageRef: "pexels_becerra_govea_photo_5773968")
));
```

3. Add the image of the restaurant into the drawables folder.
4. Under the final entry in this ArrayList, add a comma after the final entry and in a new line, insert a new RestaurantData object following the same format:
   new RestaurantData(restaurantID_PK, restaurantName, restaurantDesc, imageRef)
5. After this line has been added with the correct arguments, the emulator or mobile phone must wipe the app data (fully clearing out the previous database). This will request the app to create the new database with the new restaurant.

Adding Items

Adding items follow a similar procedure as adding a new restaurant,

1. 1. Locate the ContentPreload Java file in the directory
2. Locate the ArrayList named “preloadedItems”

```
private static final ArrayList<ItemData> preloadedItems = new ArrayList<>( Arrays.asList(
        new ItemData( itemID_PK: "00011", restaurantID_FK: "00001", itemName: "Small Burger", itemDescription: "Delicious beef burger", itemPrice: 4.5, imageRef: "pexels_gonzalo_acuna_10922928"),
        new ItemData( itemID_PK: "00012", restaurantID_FK: "00001", itemName: "Small Fries", itemDescription: "Hot Fries", itemPrice: 3.5, imageRef: "pexelsnikitakrasnov6006652"),
        new ItemData( itemID_PK: "00013", restaurantID_FK: "00001", itemName: "Small Coke", itemDescription: "Fizzy Cola", itemPrice: 2, imageRef: "pexelsjonathanborba2983100"),
        new ItemData( itemID_PK: "00014", restaurantID_FK: "00001", itemName: "Medium Burger", itemDescription: "Delicious beef burger", itemPrice: 5.5, imageRef: "pexels_gonzalo_acuna_10922925"),
        new ItemData( itemID_PK: "00015", restaurantID_FK: "00001", itemName: "Medium Fries", itemDescription: "Hot Fries", itemPrice: 4.5, imageRef: "pexelsnikitakrasnov6006652"),
        new ItemData( itemID_PK: "00016", restaurantID_FK: "00001", itemName: "Medium Coke", itemDescription: "Fizzy Cola", itemPrice: 3, imageRef: "pexelsjonathanborba2983100"),
        new ItemData( itemID_PK: "00017", restaurantID_FK: "00001", itemName: "Large Burger", itemDescription: "Delicious beef burger", itemPrice: 6.5, imageRef: "pexels_gonzalo_acuna_10922926"),
        new ItemData( itemID_PK: "00018", restaurantID_FK: "00001", itemName: "Large Fries", itemDescription: "Hot Fries", itemPrice: 5.5, imageRef: "pexelsnikitakrasnov6006652"),
        new ItemData( itemID_PK: "00019", restaurantID_FK: "00001", itemName: "Large Coke", itemDescription: "Fizzy Cola", itemPrice: 4, imageRef: "pexelsjonathanborba2983100"),
```

3. Add the image of the item into the drawables folder.
4. Under the final entry in this ArrayList, add a comma after the final entry and in a new line, insert a new ItemData object following the same format:
   new ItemData(itemID_PK, restaurantID_FK, itemName, itemDescription, itemPrice (double), imageRef)
6. After this line has been added with the correct arguments, the emulator or mobile phone must wipe the app data (fully clearing out the previous database). This will request the app to create the new database with the new item.

References

"Displaying Dialogs With Dialogfragment | Android Developers". Android Developers. Accessed 25 September 2022. https://developer.android.com/guide/fragments/dialogs.

"There Are Over 3 Billion Active Android Devices". The Verge, 2021. https://www.theverge.com/2021/5/18/22440813/android-devices-active-number-smartphones-google-2021.

Images Used In Application

https://www.pexels.com/photo/tasty-junk-food-placed-on-table-4021944/

https://www.pexels.com/photo/sushi-2098143/

https://www.pexels.com/photo/green-vegetable-on-white-ceramic-plate-5773968/

https://www.istockphoto.com/photo/sushi-set-on-stone-slate-gm506502944-84241311

https://www.behance.net/gallery/56754573/KFC-Romania-2017-Menu

https://www.pexels.com/photo/baked-pizza-on-pizza-peel-in-oven-905847/

https://www.pexels.com/photo/close-up-shot-of-a-burger-10922928/

https://www.pexels.com/photo/close-up-shot-of-a-burger-10922925/

https://www.pexels.com/photo/close-up-shot-of-a-mouthwatering-burger-10922926/

https://www.pexels.com/photo/yummy-french-fries-in-basket-surrounded-by-garlic-and-mushrooms-6006652/

https://www.pexels.com/photo/coca-cola-glass-bottle-macro-photography-2983100/

https://www.pexels.com/photo/white-ceramic-plate-filled-with-sushi-3763816/

https://www.pexels.com/photo/a-healthy-food-on-a-ceramic-plate-3763808/

https://www.pexels.com/photo/maki-slices-on-ceramic-plate-shallow-focus-photography-1199987/

https://www.pexels.com/photo/white-scoop-on-white-ceramic-bowl-884600/

https://www.pexels.com/photo/photo-of-vegetables-on-bowl-3297367/

https://www.pexels.com/photo/sushi-on-brown-wooden-board-2098085/

https://www.pexels.com/photo/sashimi-in-bowl-2871756/

https://www.pexels.com/photo/close-up-photo-of-sashimi-2995275/

https://www.pexels.com/photo/food-dinner-lunch-meal-13233519/

https://www.pexels.com/photo/pollo-frito-12118977/

https://www.pexels.com/photo/close-up-photo-of-fried-chicken-and-french-fries-on-yellow-background-12118979/

https://www.pexels.com/photo/close-up-photo-of-fried-chicken-and-french-fries-on-yellow-background-12118979/

https://www.pexels.com/photo/pollo-frito-12118977/

https://www.pexels.com/photo/cutting-board-with-friend-chicken-5652264/

https://www.pexels.com/photo/fried-potatoes-in-paper-on-blue-background-6941042/

https://www.pexels.com/photo/carbonated-drinks-and-fresh-lemon-4161715/

https://www.pexels.com/photo/pizza-on-brown-wooden-board-825661/

https://www.pexels.com/photo/cooked-pasta-with-sliced-tomatoes-and-green-leafy-128408/

https://www.pexels.com/photo/sliced-pepperoni-pizza-in-close-up-view-4109083/

https://www.pexels.com/photo/baking-cheese-cooking-crust-315755/

https://www.pexels.com/photo/closes-up-photo-of-macaroni-1437267/

https://www.pexels.com/photo/food-photography-of-pasta-1438672/

https://www.pexels.com/photo/pasta-dish-on-white-ceramic-plate-4253128/

https://www.pexels.com/photo/shrimp-pasta-served-on-gray-plate-2092906/

https://unsplash.com/photos/gSSWgnBQ3RE

https://www.freepik.com/free-photo/korean-food-jajangmyeon-noodle-with-fermented-black-beans-sauce_13902448.htm#query=jajangmyeon&position=1&from_view=keyword

https://www.pexels.com/photo/shallow-focus-photography-of-meat-dish-and-leaves-1251208/

https://www.pexels.com/photo/close-up-shot-of-a-delicious-korean-food-on-a-plate-5774151/

https://www.pexels.com/photo/close-up-shot-of-delicious-korean-barbeque-in-ceramic-plate-5774093/

https://www.pexels.com/photo/close-up-photo-of-fried-chicken-and-french-fries-on-yellow-background-12118979/

https://www.pexels.com/photo/burger-and-fries-meal-7497221/