Personal Project Report

- Mobile Application Lab 2024 -

Application Name

SmoothieMagic Drinks Application

Application Explanation

- What is your application?
- App usage scenario

- The SmoothieMagic Drink Customization App is an Android application that allows users to browse the SmoothieMagic menu, customize their smoothies according to their preferences, place orders, and track the status of their orders in real-time. It provides a seamless and intuitive user experience, tailored to SmoothieMagic's unique branding and design guidelines.

Users can use the app to:

1) Browse the SmoothieMagic menu and view different types of smoothies.
2) Customize their smoothies by selecting options such as size, type of milk, add-ons, and temperature.
3) Place orders and make payment.

Activity Explanation

- What activities are in your application?
- Role of each activity
- Design of each activity in aspect of UI/UX

The activities included in this application are **MainActivity, MenuActivity, DrinkCustomizationActivity, OrderSummaryActivity**

<u>Role of each activity:</u>

- **MainActivity**: Acts as the dashboard with options to browse the menu or view existing orders.
- **MenuActivity**: Displays a list of available smoothies for selection.
- **DrinkCustomizationActivity**: Allows users to customize their selected smoothies.
- **OrderSummaryActivity**: Shows the summary of the order and allows users to place the order.

<u>Design of each activity in aspect of UI/UX</u>

- **MainActivity**:
  - Dashboard with featured beverages, seasonal offers, and popular drinks.
  - Search bar and filter options.
  - Top navigation bar with the SmoothieMagic logo and menu icon.
- **DrinkCustomizationActivity**:
  - Detailed customization options for each drink, including size, type of milk, add-ons (slices for apple, strawberry, banana, oranges), sweetness level, and temperature.
  - Visual representation of each customization option.
- **OrderSummaryActivity**:
  - Order summary with an itemized list of selected drinks, total cost, payment options, and order tracking feature.
  - "Place Order" button.

API Usage

- From which point your application connects with Open API(or your own server)?
- Why you used it?

The Open API that this application connects with could be a menu_data.txt file with different orders. It can contain drinks id for identification for each different order that can be done, name of order, description for each order, prices that each drink refers to, and the connection for each image to represent the different drink.

The usage of API Usage is from .txt inside the server for using the app. The usage of this API might be done to simulate API response for menu data to develop the application, and demonstrate the purpose of the application if it were to function with orders that might be used in a truly running application for an existing company.

Implementation Challenge

- From where you took time while implementing your application?
- What is your special-care-point?

  (code optimization, user-friendly UI, … / specify code file&line or activity name)

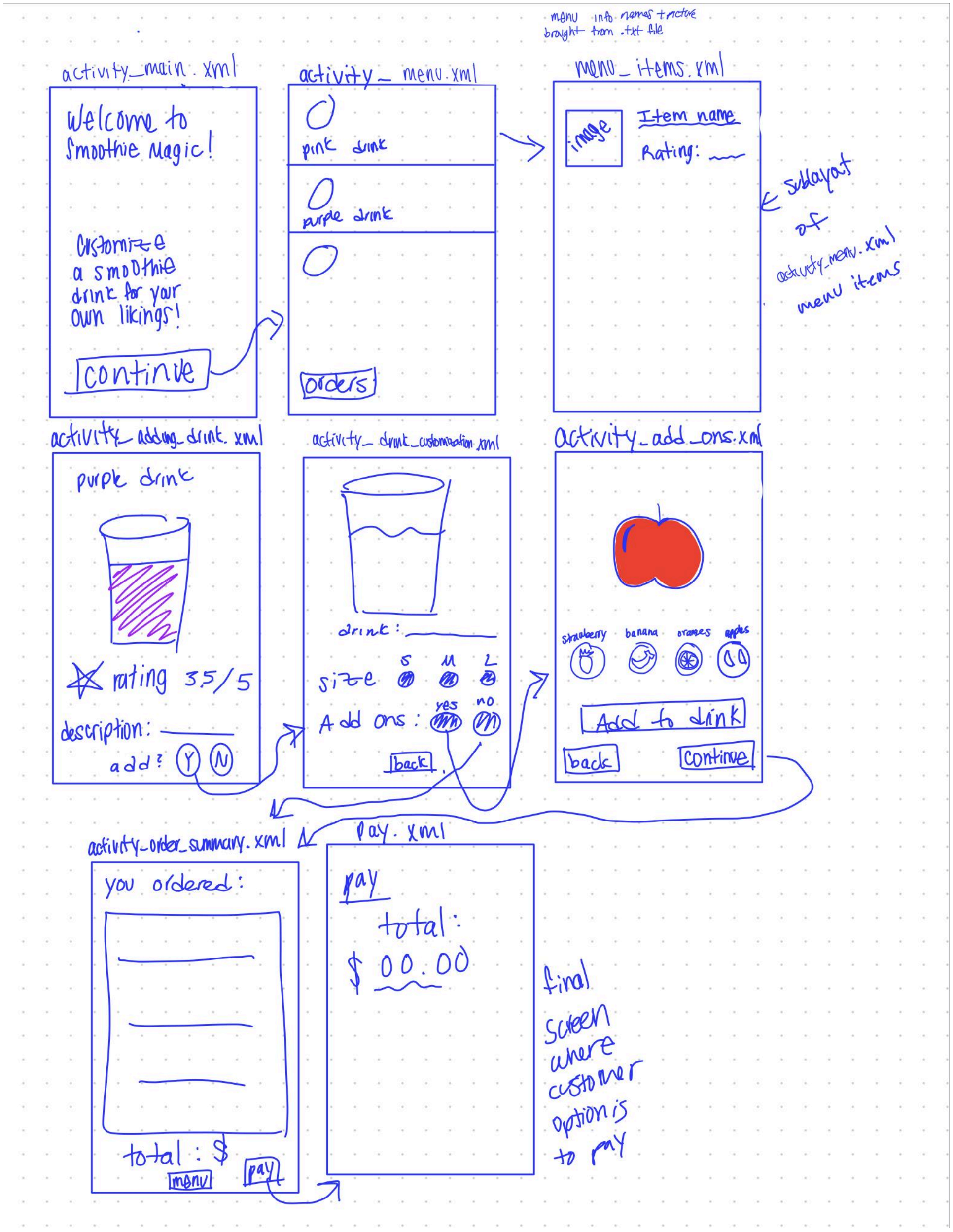
This application was implemented to create an application for customizing a smoothie from a wide range of options and see the changes reflected in creating facilitation to the challenges of ordering a drink in real-time by an application.

The special-care-point in this application is making a user-friendly UI where customers can do many multiple selections of fruits they want to include in their smoothie in the drink customization activity.

Also, a challenge that I want to implement is properly managing data loaded according to the .txt file when loading data for the application.

Application Design

- Overall design of your android application
- It can be design pattern, code separation, or so on.


menu info names + picture brought from .txt file
activity_main.xml
Welcome to Smoothie Magic!
Customize a smoothie drink for your own likings!
continue
activity_menu.xml
pink drink
purple drink
orders
menu_items.xml
image
Item name
Rating:
subLayout of activity_menu.xml menu items
activity_adding_drink.xml
purple drink
rating 3.5/5
description:
add? Y N
activity_drink_customization.xml
drink:
size S M L
Add ons: yes no
back
activity_add_ons.xml
strawberry banana oranges apples
Add to drink
back
continue
activity_order_summary.xml
you ordered:
total: $
menu
pay
pay.xml
pay
total:
$ 00.00
final screen where customer option is to pay

activity_main.xml:

What is planned is creating a main intro page for the application with a continue button.

activity_menu.xml:

When clicked, it will direct to the activity_menu.xml screen where it shows different items that a customer can choose. Each item is sub layout by **menu_items.xml**.

With each icon clicked in the menu, it goes to a screen for **activity_adding_drink.xml**. The drinks_details.xml will show the item name, rating, description. This information is brought from a menu_data.txt, where the client can decide whether they want to add it to their order by clicking the “Yes” or “No” button after reading descriptions. If a customer clicks the “Yes” button, it will go to **activity_drink_customization.xml** screen for other details the customer might want to customize in their drinks such as size, and add ons.

activity_drink_customization.xml:

In the part of add on in the activity_drink_customization.xml of drinks, when clicked the “yes” button, it will go to **activity_add_ons.xml** layer where it shows other fruits that customers might add to their drink. There will be four options of fruits that customers might add which are slices for strawberry, banana, oranges, and apples.

activity_add_ons.xml:

The customer can click add to drink, and in the top right corner a box should show how many slices for each fruit are added additionally. If a customer clicks the back button in the activity_add_ons.xml, it erases all add ons to customization of the drink and goes back to activity_drink_customization.xml. If a customer clicks the “continue” button in the activity_add_ons.xml layer, it goes to activity_order_summary.xml where all orders are shown.

activity_order_summary.xml:

When clicked in the back button, it will go to the activity_add_ons.xml layer. If you click the menu button in the summary.xml layer, it will go to the **activity_menu.xml** layer. If you click the pay button, it will finish the order for the customer and go to the **pay.xml** layer showing the customer a total pay for their whole order.

Server Design (Optional)

- (Write only you used your own server like AWS) Overall design of your server
- It can be design pattern, code separation, or so on.

- You can freely add the row in the table above.