Batch 7, Year 4, Semester 1

Group 1

Bottom Navigation

Mobile App Dev I

Lecturer:

Mrs: Hong Mom

IT 420

Team Members

Khorn Sopheap
Team Leader

Siek Reaksmey
Team Member

Sophorn Chiva
Team Member

Rim Soung Socheata
Team Member

Path Nimol
Team Member

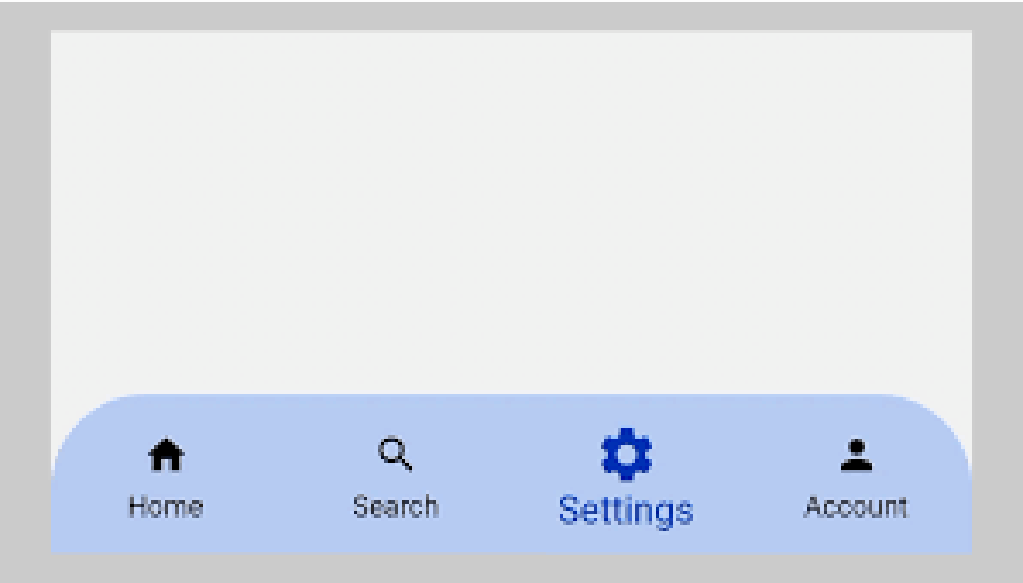
Khloy Vanno
Team Member

Table of Content

Chapter 1:

Understanding Bottom Navigation

1.1 What is Bottom Navigation?

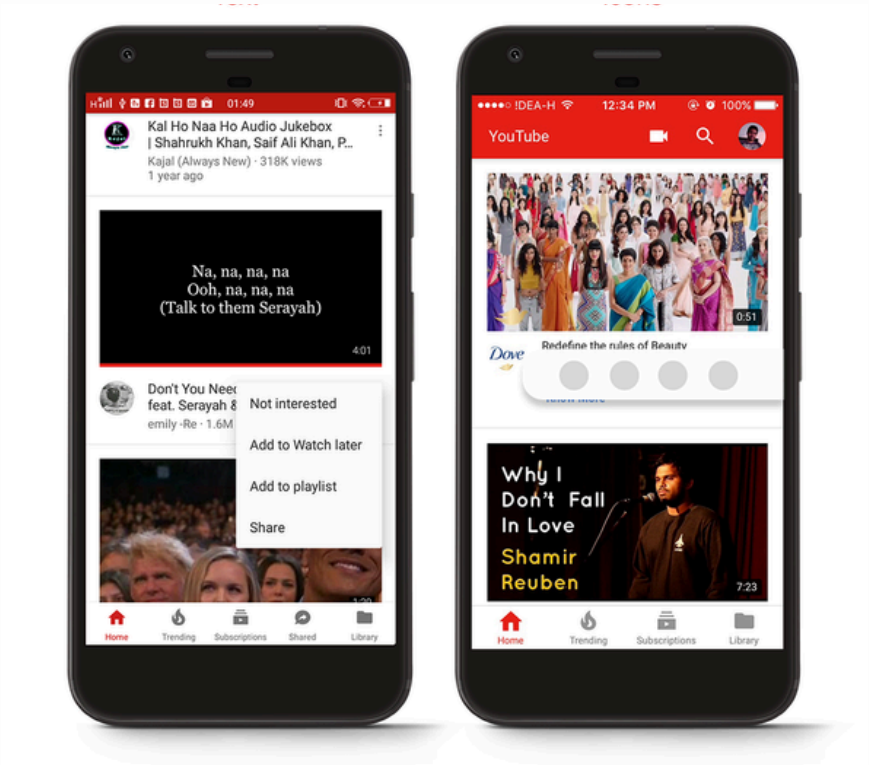


Bottom navigation is a widely used mobile UI component that allows users to switch between the main sections of an application quickly. It is usually placed at the bottom of the screen and contains icons and labels representing different screens.

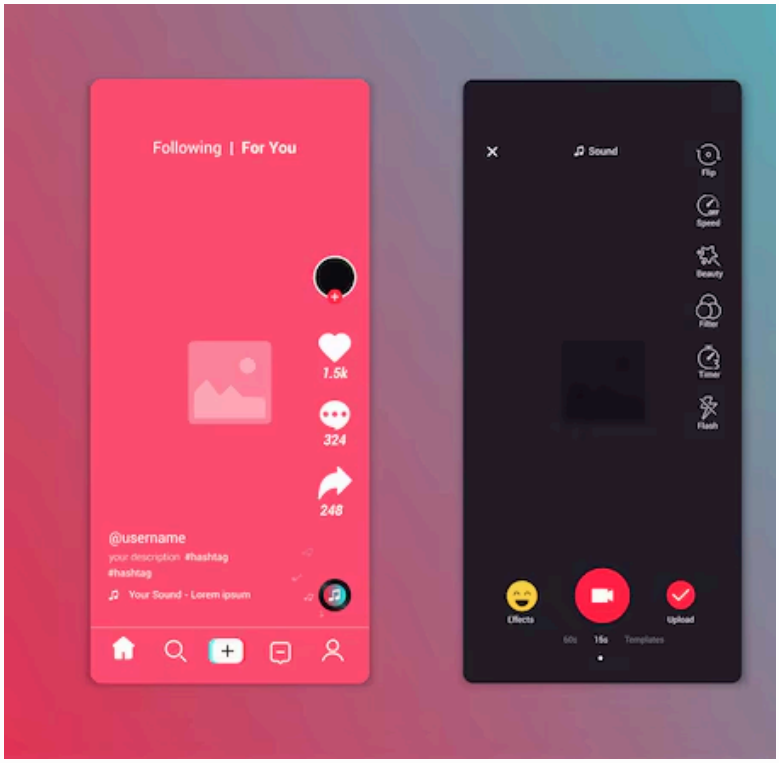
This document presents a complete frontend implementation of a bottom navigation system using Kotlin, XML, and Fragments for an Android application named DrinksOrderApp. The implementation focuses purely on UI and fragment navigation, without backend or database integration.

Bottom Navigation is a popular design interface among the Mobile development. Example: Apps that uses Bottom Navigation includes:

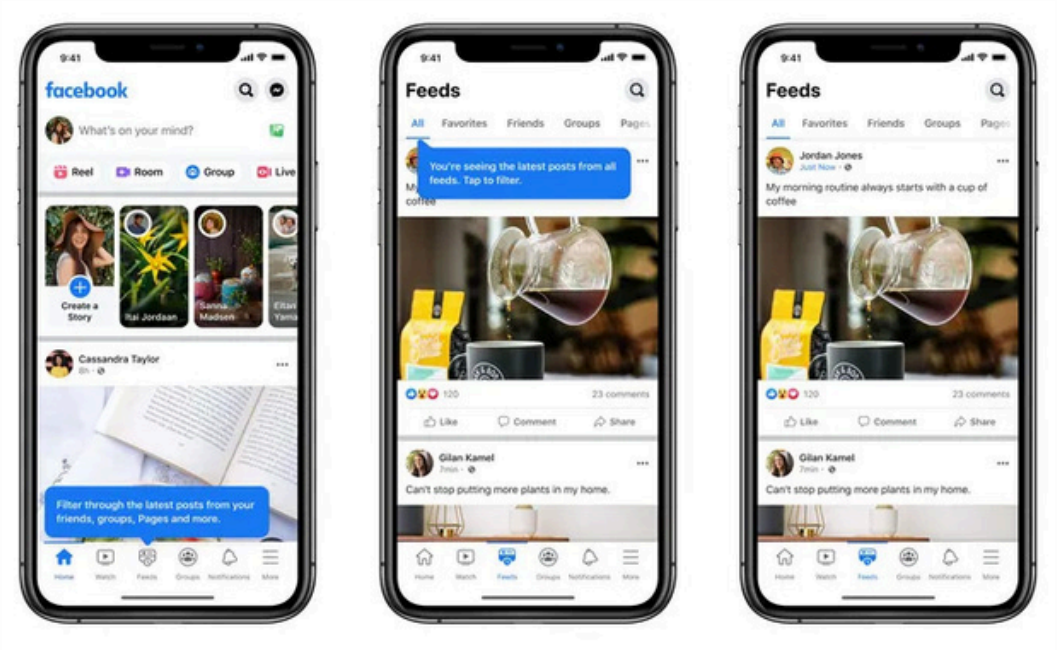
- YouTube



- TikTok



- Facebook



1.2. Objective

The objectives of this assignment are:

- To understand the role of bottom navigation in Android apps
- To implement bottom navigation using Kotlin and XML
- To manage screen navigation using Fragments
- To create a clean and user-friendly UI for DrinksOrderApp

1.3. Benefits of using Bottom Navigation

- **Always Visible**: The main sections are always on screen, so users don't have to search for hidden menus.
- **One-Tap Switching**: You can jump between major parts of the app with a single click, saving time and effort.
- **Location Awareness**: It uses colors or indicators to show exactly which page the user is currently on.
- **Simple Design**: It limits you to 3–5 icons, which prevents the app from looking cluttered and messy.

Chapter 2:

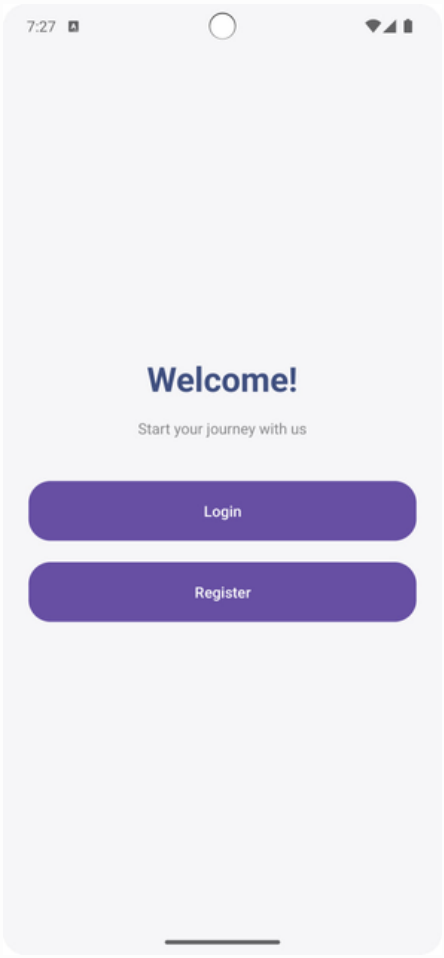
Project Overview

2.1. Scope

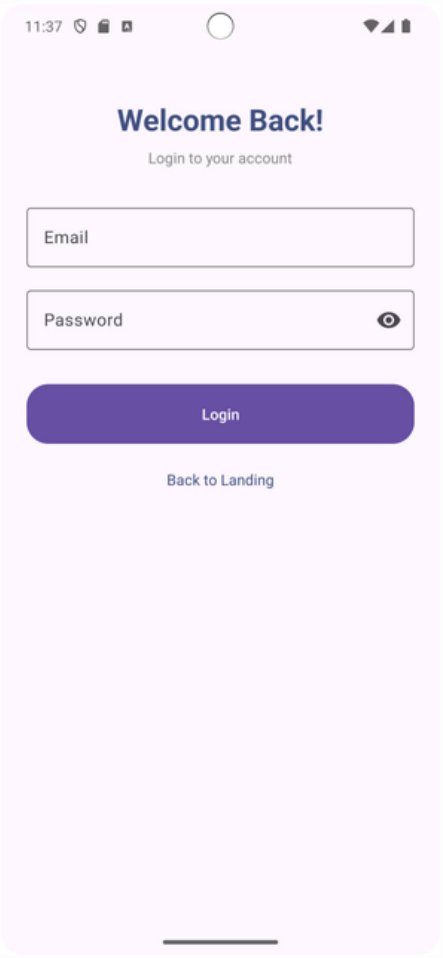
The scope of this project is focused on the frontend architectural design and user interface implementation of an Android application. The specific boundaries include:

- **UI Components**: Development of the BottomNavigationView using Material Design 3 standards.
- **Navigation Logic:** Implementation of a Fragment-based architecture to switch screens without reloading the app.
- **Frontend Design**: Customization of icons, text labels, and active/inactive states for 3–5 destinations.
- **Static Layouts:** Creation of XML layouts for each Fragment to demonstrate a complete user interface.

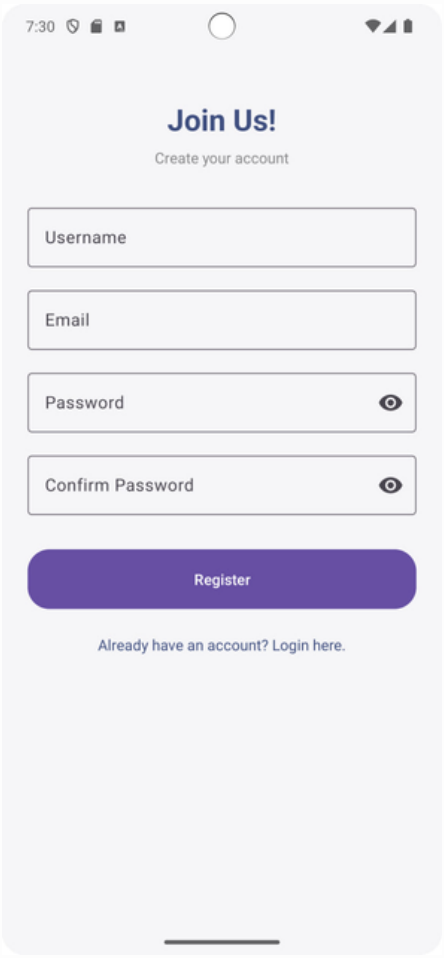
2.2. Project Flow



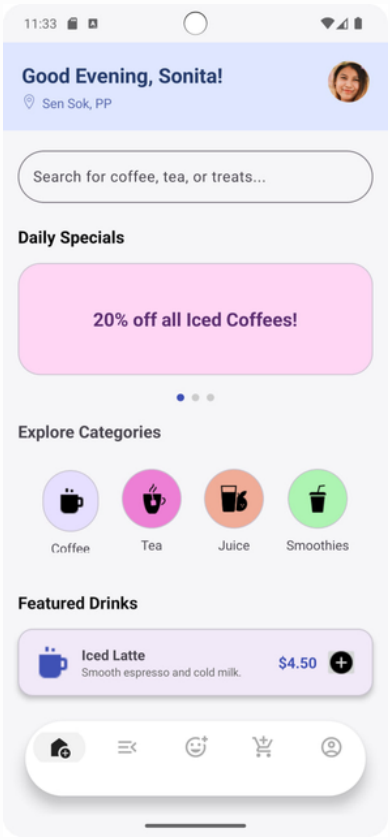
First Page



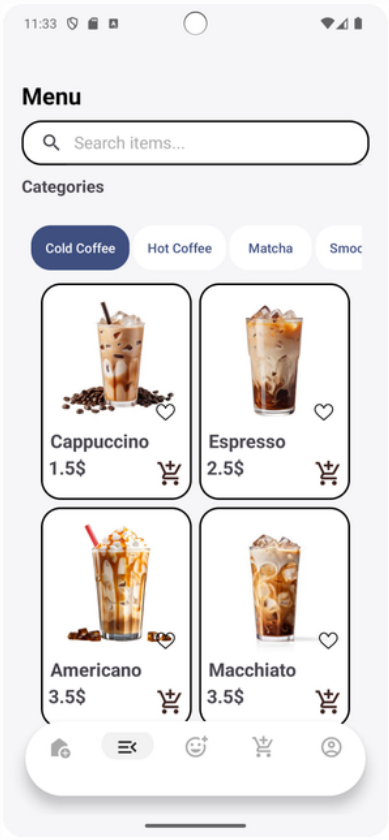
Second Page



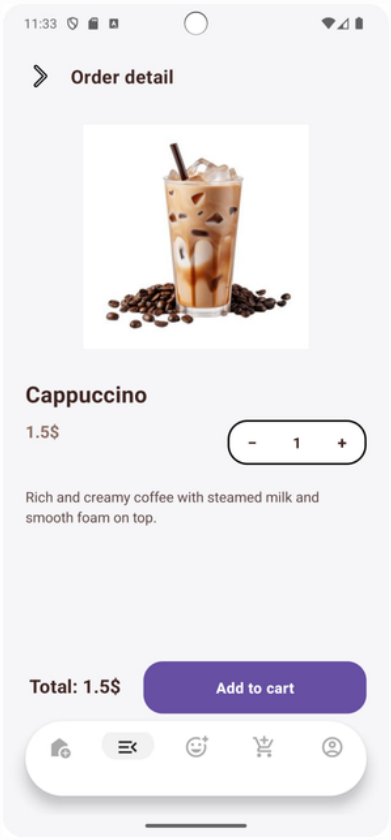
Third Page



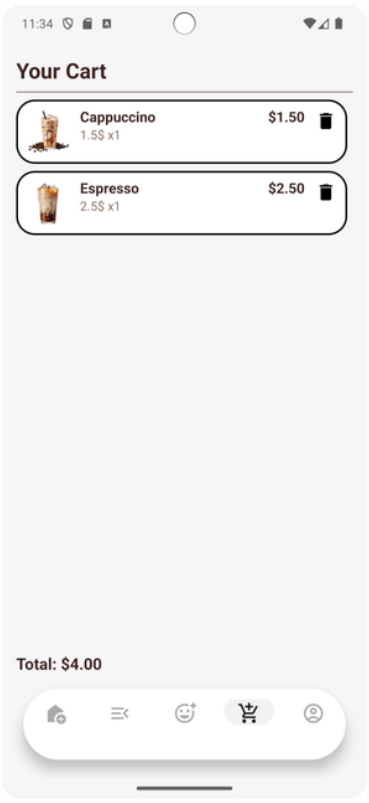
Fourth Page



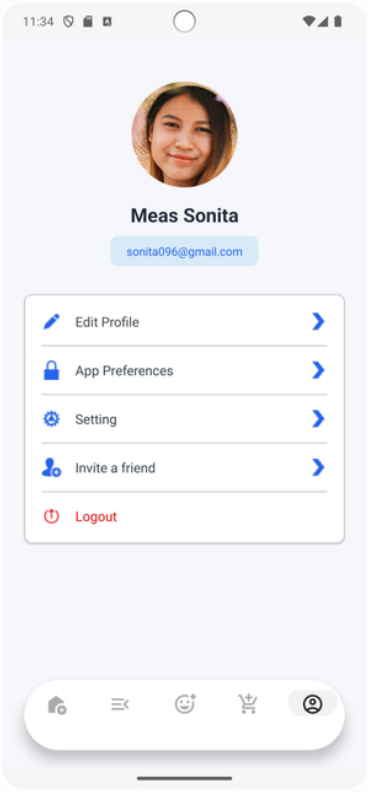
Fifth Page



Sixth Page



Seventh Page



Eighth Page

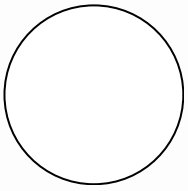
2.3. Technology Stack

Languages

Tools

Color Palette

#3F4F7F

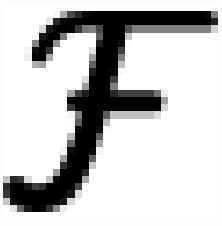


#FFFFFF

Font Size

18dp - 25dp

Typography



Normal & Bold

2.4. Application Navigation Structure

The DrinksOrderApp contains five main sections accessible through bottom navigation:

1. Home
2. Menu
3. Favourite
4. Add to Cart
5. User Profile

Each section is implemented as a Fragment and displayed within a single Activity.

Chapter 3:

Step-by-Step Implementation

Step 1: Create the Main Activity

- Create MainActivity.kt as the single activity of your app.
- Set up View Binding:

```
binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
    val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
    insets
}
```

Step 2: Create Bottom Navigation

```
<item
    android:id="@+id/btnHome"
    .../>
<item
    android:id="@+id/btnMenu"
    .../>
<item
    android:id="@+id/btnLove"
   ... />
<item
    android:id="@+id/btnCart"
   ... />
<item
    android:id="@+id/btnUserProfile"
    ... />
```

Step 3: Create Fragment Classes

Each screen in the application is represented by a Fragment.

- HomeFragment.kt

```
class HomeFragment : Fragment()
```

- MenuFragment.kt

```
class MenuFragment : Fragment()
```

- FavoriteFragment.kt

```
class FavoriteFragment : Fragment()
```

- CartFragment.kt

```
class CartFragment : Fragment()
```

- ProfileFragment.kt

```
class ProfileFragment : Fragment(R.layout.fragment_profile)
```

- Each screen in the application is represented by a Fragment.

```
class HomeFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        return inflater.inflate(R.layout.fragment_home, container, false)
}
```

- XML layouts contain UI elements relevant to each tab.

Step 4: Implement Fragment Replacement Logic

In MainActivity, create a helper function:

```
private fun replaceFragment(fragment: Fragment) {
    supportFragmentManager.beginTransaction()
        .replace(R.id.frameLayout, fragment)
        .commit()
}
```

- Use this function whenever a bottom navigation item is clicked.

Step 5: Configure Bottom Navigation Listener

Set a listener on the BottomNavigationView:

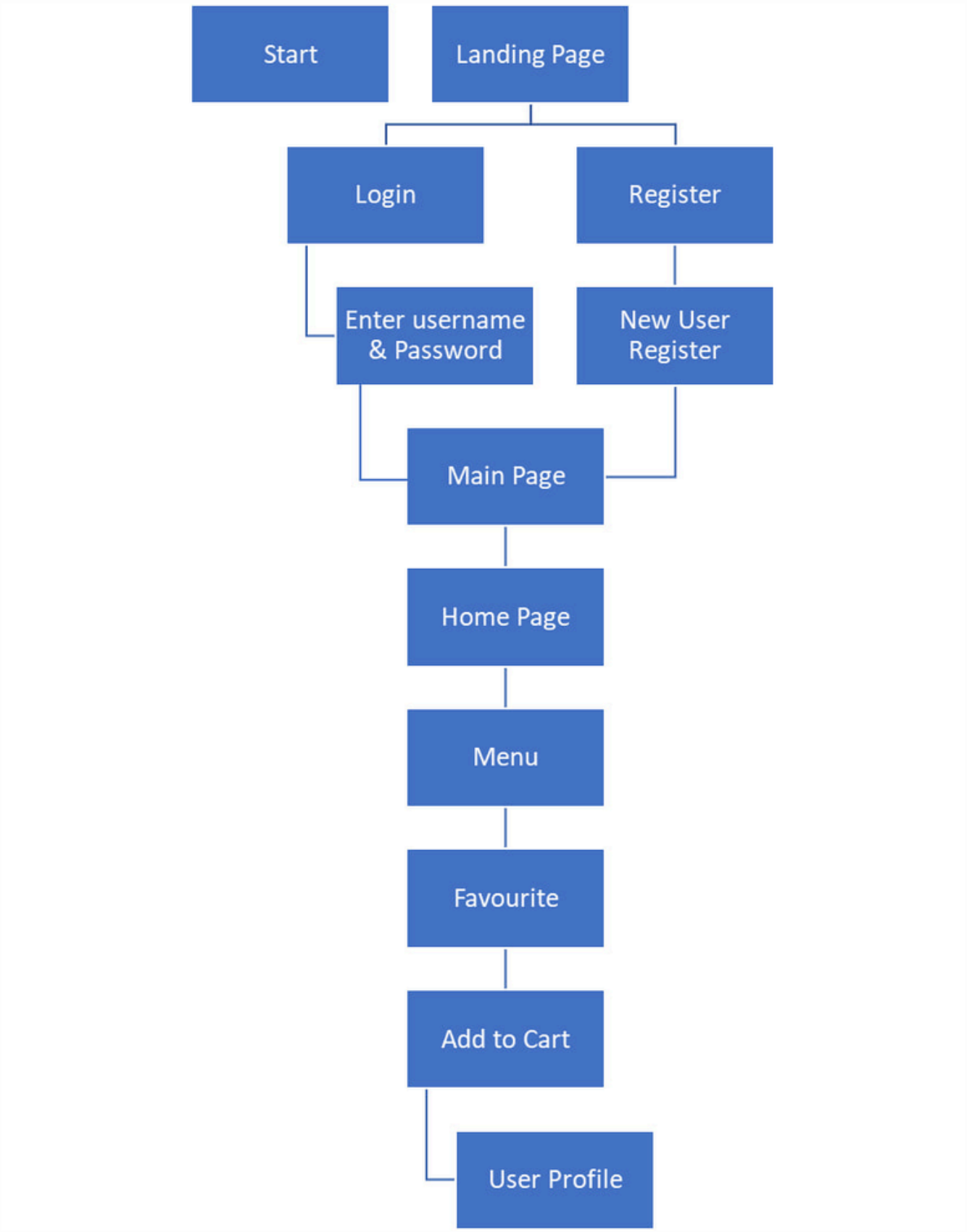
```
binding.btnNavigation.setOnItemSelectedListener { item ->
    when (item.itemId) {
        R.id.btnHome -> { replaceFragment(HomeFragment()); true }
        R.id.btnMenu -> { replaceFragment(MenuFragment()); true }
        R.id.btnLove -> { replaceFragment(FavoriteFragment()); true }
        R.id.btnCart -> { replaceFragment(CartFragment()); true }
        R.id.btnUserProfile -> { replaceFragment(ProfileFragment()); true }
        else -> false
    }
}
```

Step 6: Set Default Fragment

When the app starts, show a default fragment (e.g., LandingFragment or HomeFragment):

```
replaceFragment(LandingFragment())
```

Project Flow Diagram



Chapter 4:

Conclusion & References

Conclusion

This project successfully implements a bottom navigation system using Kotlin, XML, and fragments within a single-activity architecture. The MainActivity is responsible for handling fragment transactions and controlling the bottom navigation, while each fragment represents a different section of the DrinksOrderApp, such as Home, Menu, Favourite, Cart, and User Profile. By using fragments instead of multiple activities, the application provides smooth navigation, better performance, and a consistent user interface. Overall, this implementation follows modern Android development practices and creates a clean, organized, and user-friendly mobile application structure.

References

Research sites

- https://developer.android.com/get-started/overview
- https://github.com/

Research Assistant (AI)

- https://chatgpt.com/
- https://gemini.google.com/app

Resources (Image & Icons)

- Pichon (For Icons)
- Google (For Images)