IT2010 – Mobile Application Development
BSc (Hons) in Information Technology
2nd Year
Faculty of Computing
SLIIT

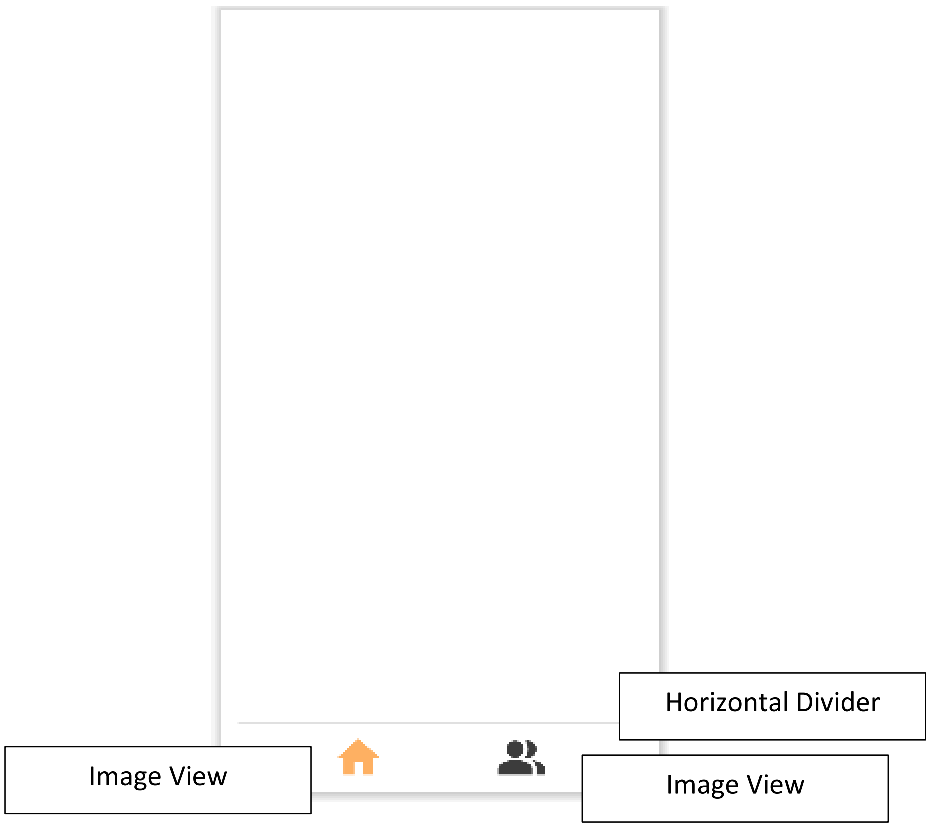
2023 – Lab 05

Fragments

A Fragment represents a reusable portion of your app's UI. A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events. Fragments cannot live on their own--they must be hosted by an activity or another fragment. The fragment's view hierarchy becomes part of, or attaches to, the host's view hierarchy.

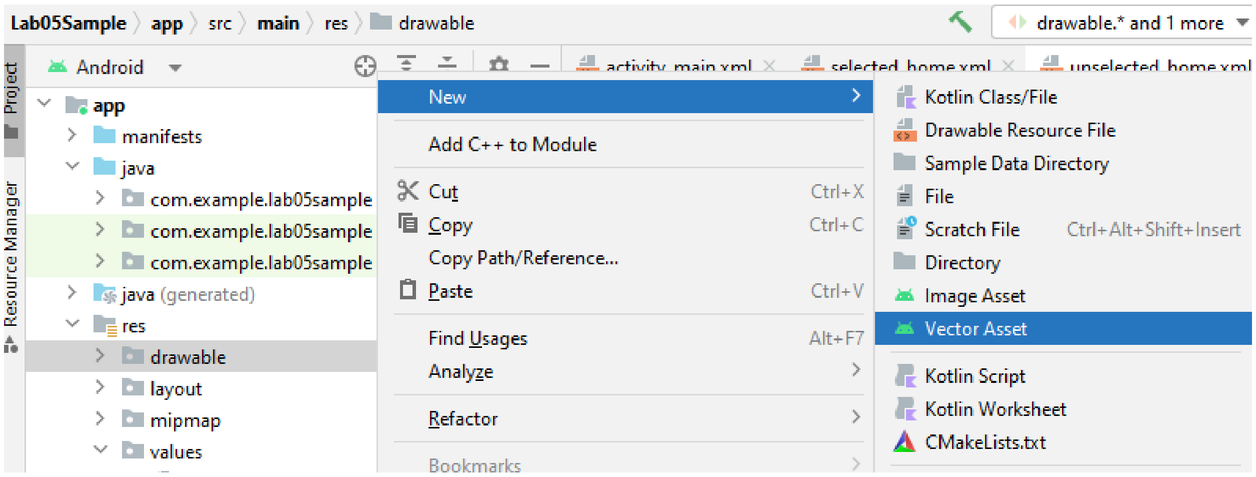
In this exercise, we will create a simple application to use fragments in an Android Application

1. Design the following UI

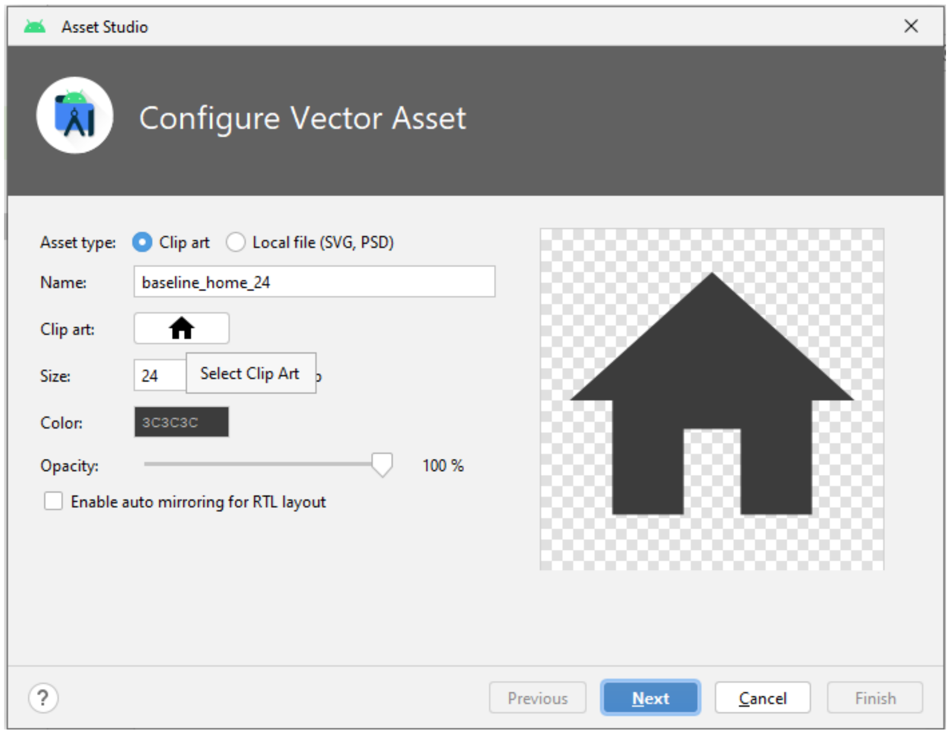


2. Before adding the image views, you need to import the images to the project.
3. For this app we will add 4 images 2 from each icon with #FEB062 color for the selected item and #3C3C3C color for the unselected.

4. To add the image vectors, right click the drawable directory and follow the below steps



5. Then click the clip art to select the needed Icon


Asset Studio

Configure Vector Asset

Asset type: [✓ Clip art] Local file (SVG, PSD)

Name: baseline_home_24

Clip art:

Size: 24

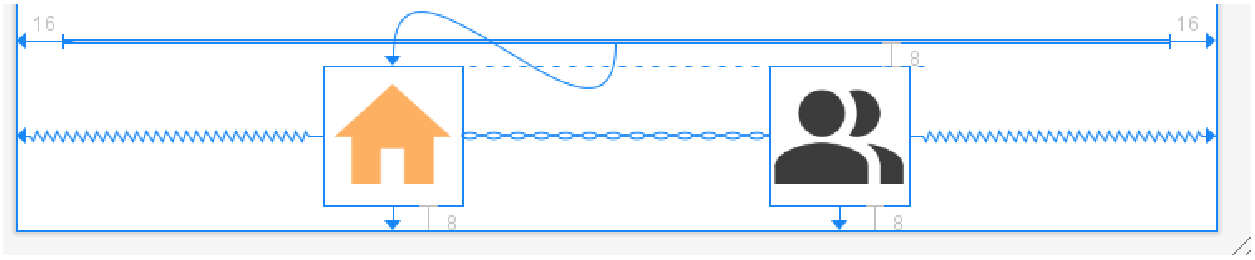
Color: 3C3C3C

Opacity: 100 %

Enable auto mirroring for RTL layout

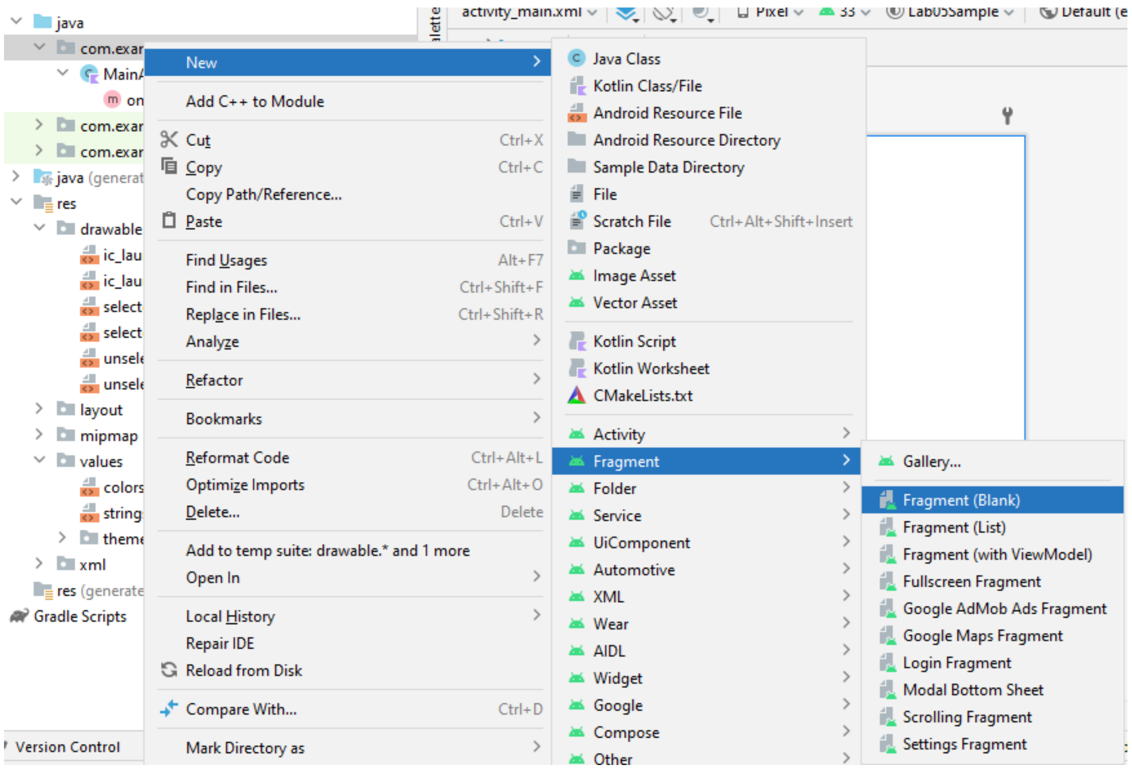
6. Give the relevant name and the color.
    a. selected_home | color - #FEB062
    b. unselected_home | color - #3C3C3C
    c. selected_user | color - #FEB062
    d. unselected_user | color - #3C3C3C
7. And set the size of the icon to 48dp
8. Click next and click finish

9. After creating the four images drawable folder should look as follows. If not get assistance from the instructor
10. Select the initial image vectors as follows and it should be viewed as follows



Fragment implementation

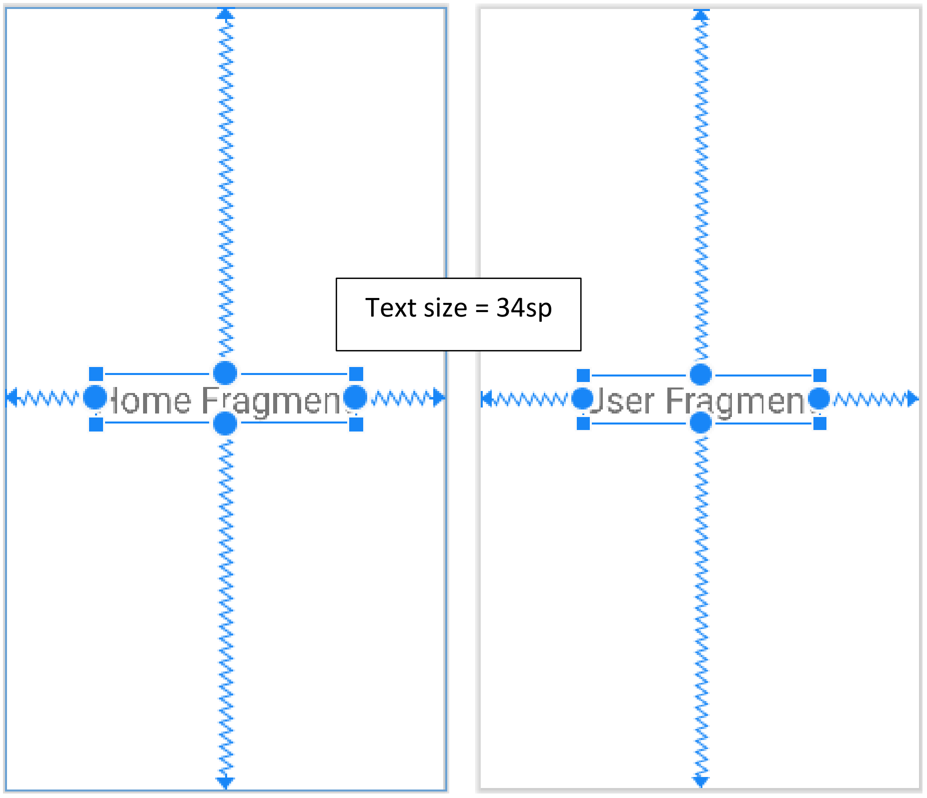
1. Create a fragment in the main directory



2. Name the first fragment as HomeFragment
3. Create another fragement as UserFragement as well.
4. Remove all the other codes except for the onCreateView from the fragment classes.

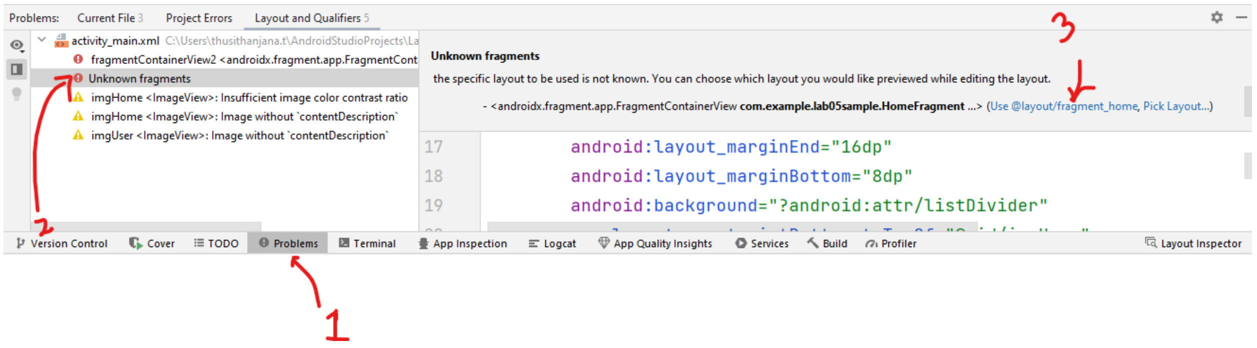
```
class HomeFragment : Fragment() {

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home, container, false)
    }

}
```

5. Design the two fragments as follows



6. Add fragment container View to the MainActivity

7. You might face an issue that the screen will not render properly, to fix that, select the fix as follows.



activity_main.xml C:\Users\thusithanjana.t\AndroidStudioProjects\La

fragmentContainerView2 <androidx.fragment.app.FragmentCont

Unknown fragments

imgHome <ImageView>: Insufficient image color contrast ratio

imgHome <ImageView>: Image without `contentDescription`

imgUser <ImageView>: Image without `contentDescription`

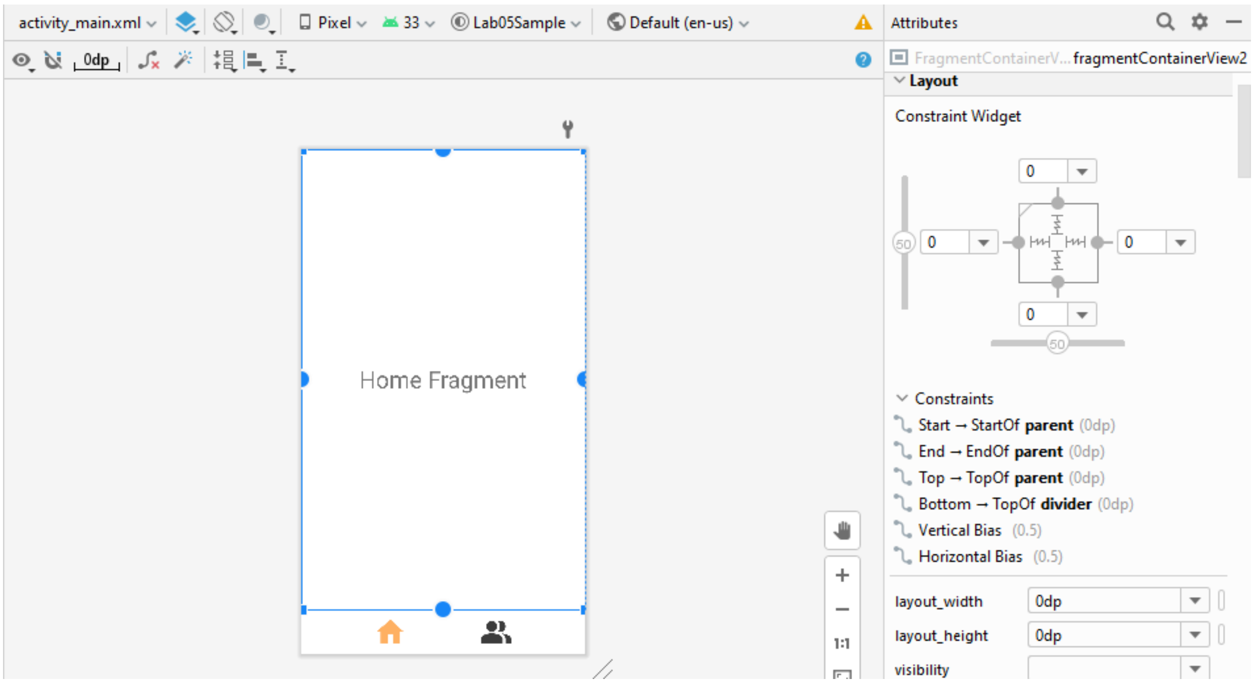
Unknown fragments

the specific layout to be used is not known. You can choose which layout you would like previewed while editing the layout.

- <androidx.fragment.app.FragmentContainerView **com.example.lab05sample.HomeFragment** ...> (Use @layout/fragment_home, Pick Layout...)

```
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="8dp"
        android:background="?android:attr/listDivider"
```

8. After that implement the UI as follows



FragmentContainerV... fragmentContainerView2

Layout

Constraint Widget

0 0 0 0 50 50

Constraints

Start → StartOf **parent** (0dp)

End → EndOf **parent** (0dp)

Top → TopOf **parent** (0dp)

Bottom → TopOf **divider** (0dp)

Vertical Bias (0.5)

Horizontal Bias (0.5)

layout_width 0dp

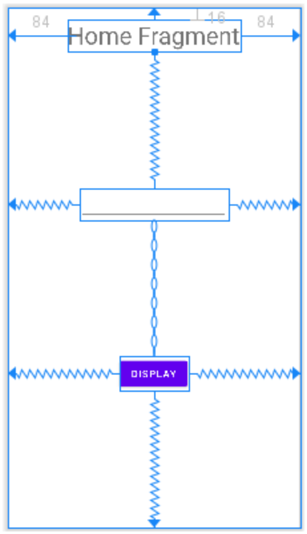
layout_height 0dp

visibility

9. Let’s implement the code in the MainActivity.

```
val imgHome:ImageView = findViewById(R.id.imgHome)
        val imgUser:ImageView = findViewById(R.id.imgUser)
        val fragmentHome = HomeFragment()
        val fragmentUser = UserFragment()

        imgUser.setOnClickListener {
            imgHome.setImageResource(R.drawable.unselected_home)
            imgUser.setImageResource(R.drawable.selected_user)

            supportFragmentManager.beginTransaction().apply {
                replace(R.id.fragmentContainerView2, fragmentUser)
                commit()
            }
        }

        imgHome.setOnClickListener {
            imgHome.setImageResource(R.drawable.selected_home)
            imgUser.setImageResource(R.drawable.unselected_user)

            supportFragmentManager.beginTransaction().apply {
                replace(R.id.fragmentContainerView2, fragmentHome)
                commit()
            }
        }
```

10. Now, let’s do some actions from one fragment.
11. For that modify the fragment_home layout as follows.



12. After the design implement the following code in the Home Fragment

```
override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.fragment_home, container, false)
        val btnDisplay = view.findViewById<Button>(R.id.btnDisplay)
        val edtName = view.findViewById<EditText>(R.id.edtName)

        btnDisplay.setOnClickListener {
            Toast.makeText(context,"Hello, ${edtName.text.toString()}",Toast.LENGTH_LONG).show()
        }

        return view
    }
```