



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

**РТУ МИРЭА**

---

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных  
технологий

## **Отчет по практической работе №6**

по дисциплине «Разработка мобильных приложений»

**Выполнил:**

Студент группы ИКБО-20-23

Комисарик М.А.

**Проверил:**

Старший преподаватель кафедры  
МОСИТ

Шешуков Л.С.

Москва 2025 г.

1 ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ .....	3
1.1 BottomBar .....	3
1.2 ActionBar .....	11
1.3 Navigation Drawer .....	16
2 ПРАКТИЧЕСКОЕ ЗАДАНИЕ .....	23
2.1 Боковая панель навигации.....	23
2.1.1 Разметка.....	23
2.1.2 Код .....	27
2.1.3 Тестирование .....	29
2.2 Нижняя панель навигации.....	32
2.2.1 Разметка.....	32
2.2.2 Код .....	34
2.2.3 Тестирование .....	35
ЗАКЛЮЧЕНИЕ .....	38

# 1 ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

## 1.1 BottomBar

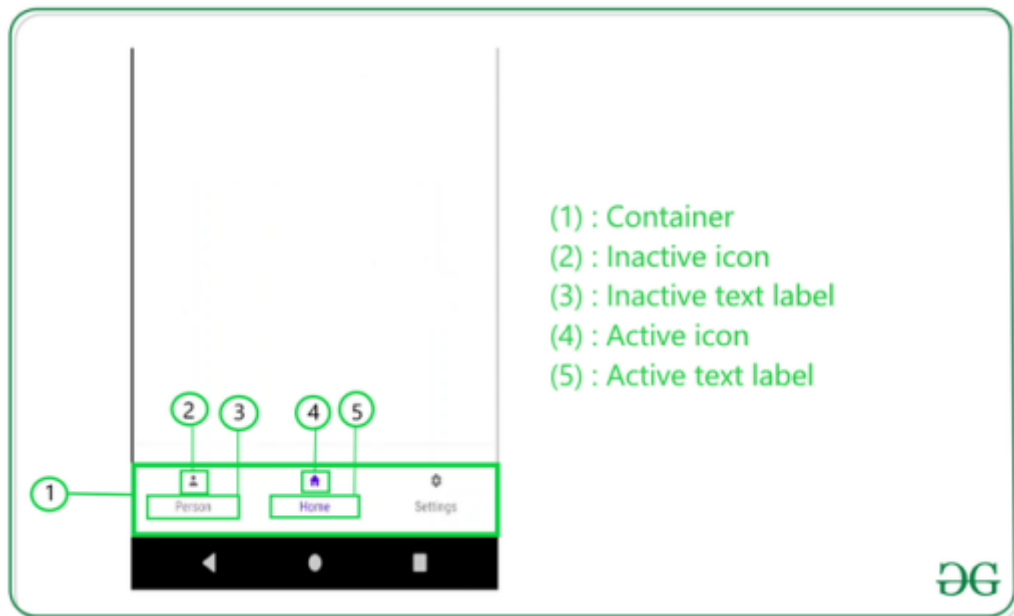
В предыдущих практических работах были рассмотрены переходы между разными экранами в приложении с использованием простых элементов Button. Однако, в контексте современной Android разработки такой метод является не интуитивным и устаревшим. Поэтому, на замену такой навигации были придуманы объекты пользовательского интерфейса BottomBar и Drawer.

BottomBar, или нижняя панель навигации, является ключевым элементом пользовательского интерфейса в мобильных приложениях, работающих под управлением Android. Этот элемент предназначен для улучшения навигации и повышения удобства использования приложения за счёт предоставления быстрого доступа к основным разделам приложения. В контексте перемещения между экранами BottomBar выполняет несколько важных функций:

- улучшение пользовательского опыта: BottomBar делает навигацию по приложению интуитивно понятной и удобной. Располагаясь в нижней части экрана, он легко доступен для пользователя, что особенно важно при использовании устройства одной рукой. Это облегчает переход между ключевыми разделами приложения, не отвлекая пользователя от основного контента;
- повышение производительности: использование BottomBar позволяет сократить количество нажатий, необходимых для перехода между разделами приложения, тем самым ускоряя взаимодействие пользователя с приложением и повышая его общую производительность;
- организация контента: BottomBar помогает организовать контент в приложении, выделяя основные разделы или функции, которые

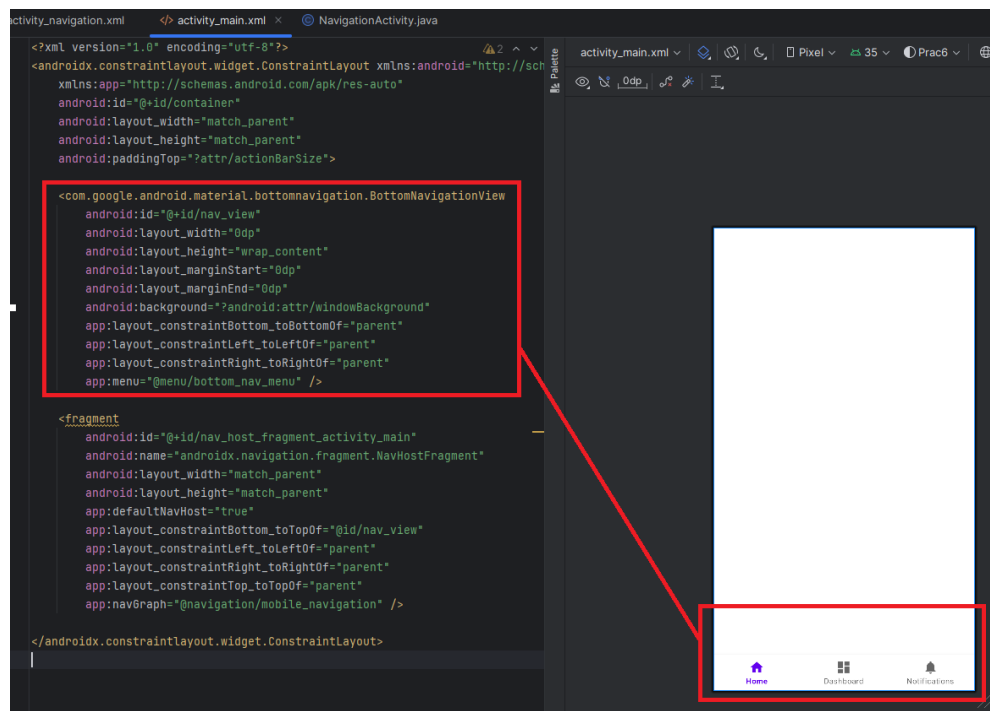
должны быть всегда под рукой. Это упрощает структуру приложения и делает его более понятным для пользователя.

BottomBar представляет собой контейнер, который содержит в себе различные элементы (Рисунок 1).



**Рисунок 1 – Наполнение BottomBar**

Для использования BottomNavigationView необходимо добавить этот элемент в макет активности (Рисунок 2).



**Рисунок 2 – Описание BottomNavigationView в файле разметки**

CoordinatorLayout – это контейнер, который расширяет FrameLayout.

CoordinatorLayout предназначен для двух основных вариантов использования:

- в качестве декора приложения верхнего уровня или макета Chrome;
- как контейнер для определенного взаимодействия с одним или несколькими дочерними представлениями.

После добавления BottomNavigationView в активность необходимо заполнить его элементами (экранами, между которыми будет осуществляться перемещение). Для этого создается XML-файл в папке res/menu (например, bottom\_nav\_menu.xml) и добавляются в него пункты меню. Если папка menu отсутствует в проекте, то сначала ее необходимо добавить.

Для создания новой папки нажимаем правой кнопкой мыши на "res" → "New" → "Android Resource Directory" (Рисунок 3). Вводим название папки, выбираем тип ресурса из выпадающего списка (Рисунок 4).

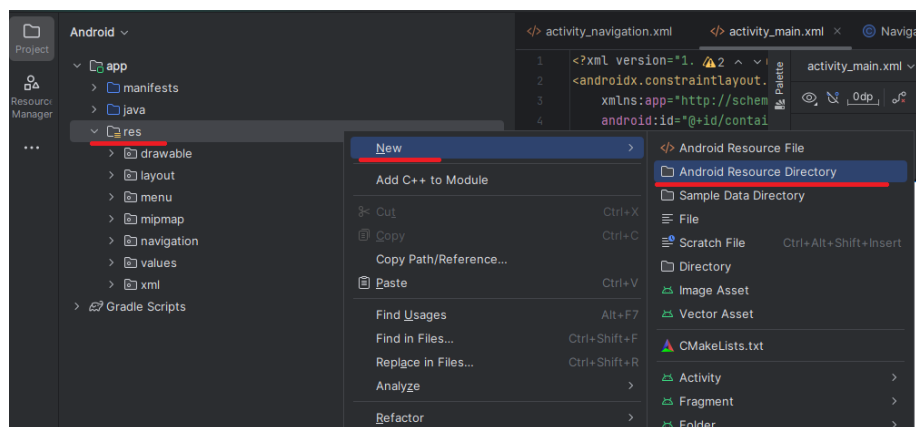


Рисунок 3 – Создание новой папки в проекте

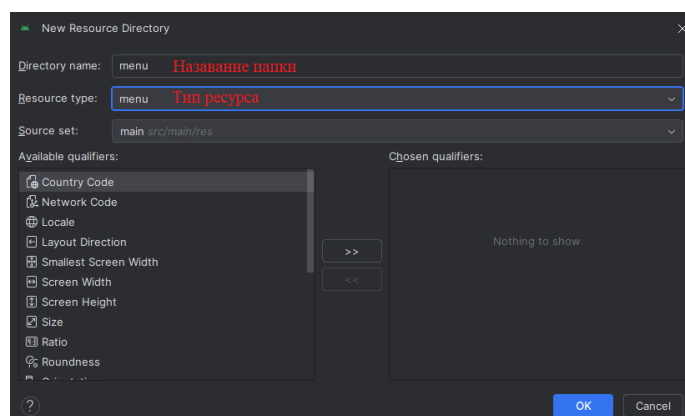
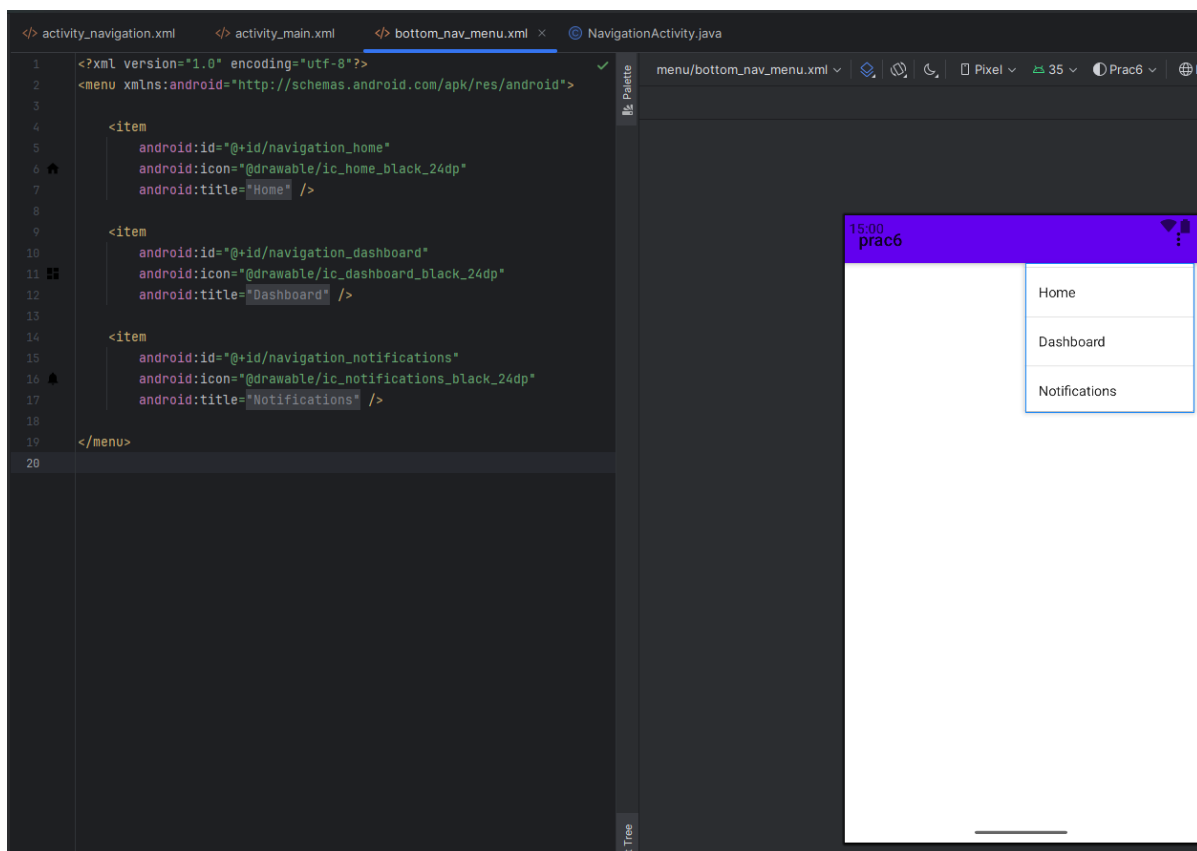


Рисунок 4 – Настройка создаваемой папки

После создания XML-файла его можно наполнить пунктами меню (Рисунок 5).



**Рисунок 5 – Графическое и кодовое представление меню навигации**

Тег `<menu>` является корневым узлом файла и определяет меню, состоящее из одного или нескольких элементов `<item>` и `<group>`.

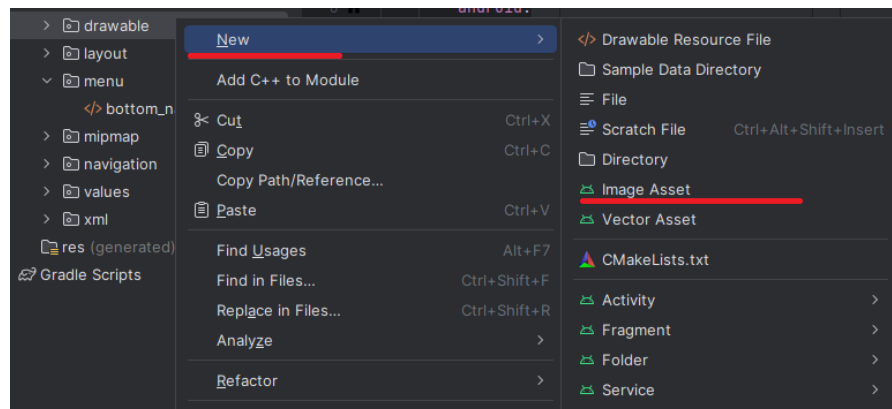
Элемент `<item>` представляет объект `MenuItem`, которой является одним из элементов меню. Этот элемент может содержать внутренний подэлемент `<menu>`, с помощью которого создается подменю.

Элемент `<item>` включает следующие атрибуты, которые определяют его внешний вид и поведение:

- `android:id`: уникальный id элемента меню, который позволяет его опознать при выборе пользователем и найти через поиск ресурса по id;
- `android:icon`: ссылка на ресурс drawable, который задает изображение для элемента (`android:icon="@drawable/ic_help"`);

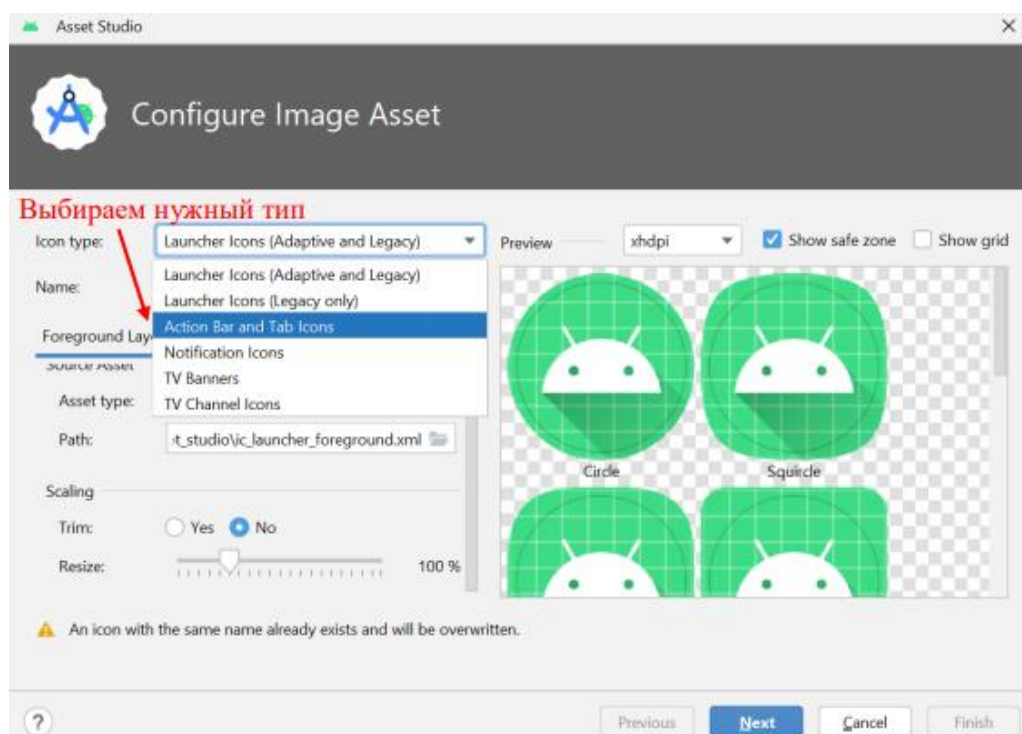
- android:title: ссылка на ресурс строки, содержащий заголовок элемента. По умолчанию имеет значение "Settings";

Теперь пользователь может создавать столько элементов, сколько захочет, в `bottom_nav_menu.xml` файле. Пользователю также необходимо создать значок для отображения каждого из этих элементов. Чтобы создать значок, нужно нажать на "drawable" → "New" → "Image Asset" (Рисунок 6).



**Рисунок 6 – Создание новой иконки**

Далее идёт выбор нужного типа иконки (Рисунок 7).



**Рисунок 7 – Выбор необходимого типа иконки**

В открывшемся окне пользователь может назвать значок как угодно, но он не должен содержать ни одной заглавной буквы. Пользователь может выбрать

нужный значок, выполнив поиск по нему, а когда пользователь закончит, то необходимо нажать "Next" → "Finish" (Рисунки 8-10).

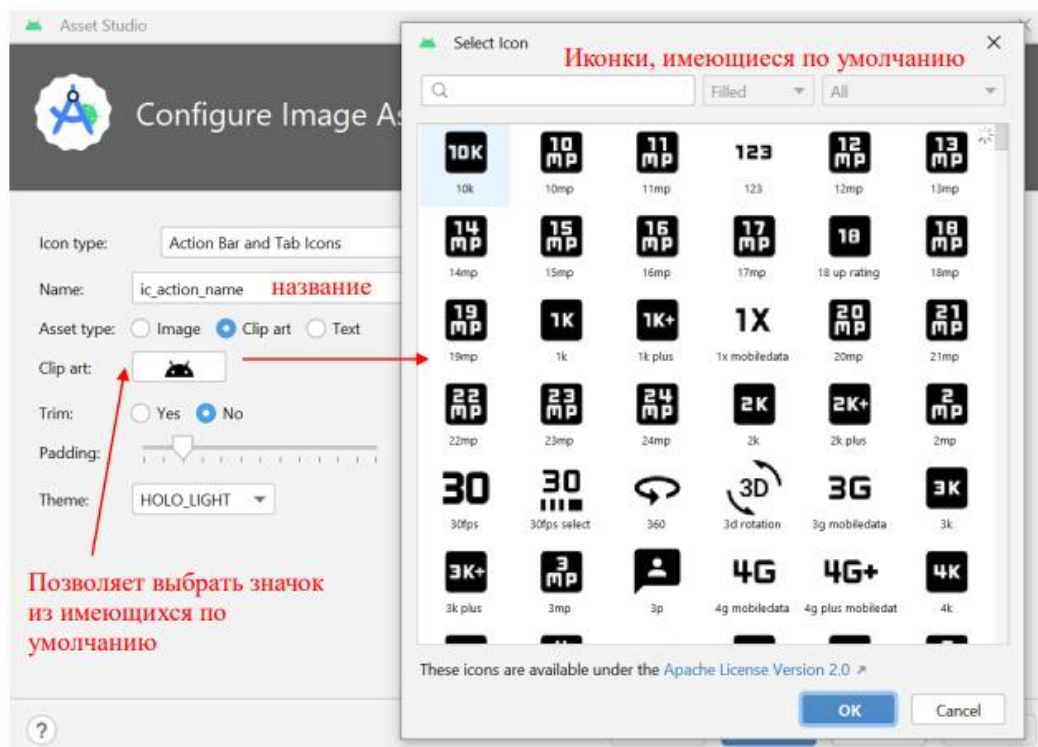
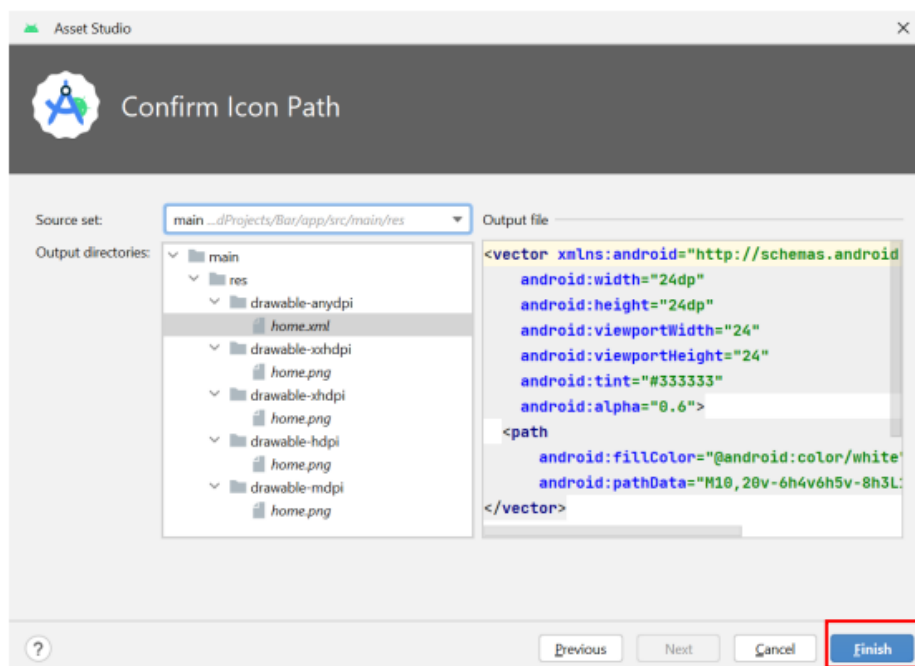


Рисунок 8 – Настройка иконки, часть 1



Рисунок 9 – Настройка иконки, часть 2



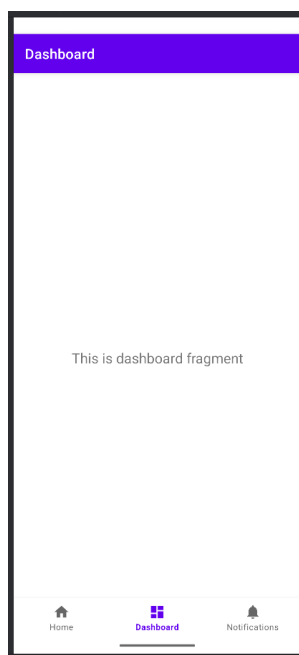


**Рисунок 10 – Настройка иконки, часть 3**

Image Asset поможет создать следующие типы значков:

- иконки лаунчера;
- значки панели действий и вкладок;
- значки уведомлений.

Если запустить сборку проекта, то можно получить отображение меню в нижней части экрана (Рисунок 11).



**Рисунок 11 – Отображение нижнего меню навигации**

Однако только отображение не подходит для работы с BottomBar. Необходимо, чтобы при нажатии на элемент, происходили некоторые действия. Для этого необходимо установить слушатель нажатий в классе активности (Рисунок 12).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

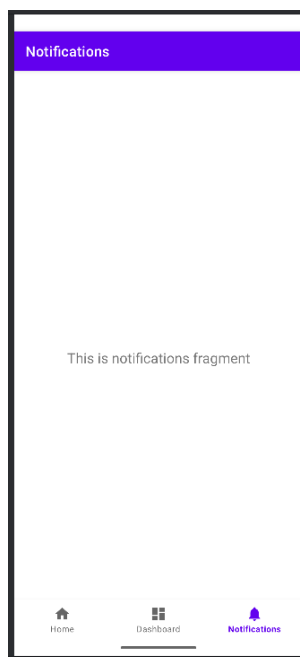
    BottomNavigationView bottomNavigationView = findViewById(R.id.bottom_navigation);
    bottomNavigationView.setOnNavigationItemSelectedListener(
        new BottomNavigationView.OnNavigationItemSelectedListener() {
            1 usage
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                if (item.getItemId() == R.id.home){
                    // Обработка выбора раздела "Домой"
                    Toast.makeText( context: MainActivity.this, text: "Домой", Toast.LENGTH_LONG).show();
                    return true;
                }
                else if (item.getItemId()==R.id.settings){
                    // Обработка выбора раздела "Настройки"
                    Toast.makeText( context: MainActivity.this, text: "Настройки", Toast.LENGTH_LONG).show();
                }
                return false;
            }
        }
    );
}
```

**Рисунок 12 – Реализация логики обработчика событий меню навигации**

Чтобы понять, какой пункт меню выбран, вначале получаем его идентификатор через `item.getItemId()`. Затем пробегаемся в конструкции `if...else` и выбираем нужный вариант и в зависимости от выбора производим определенные действия (Рисунки 13-14).



**Рисунок 13 – Отображение результата возможных событий, часть 1**



**Рисунок 14 – Отображение результата возможных событий, часть 2**

## **1.2 ActionBar**

ActionBar в Android представляет собой верхнюю панель приложения, которая обеспечивает удобный доступ к наиболее важным функциям приложения, а также поддерживает навигацию. Создание и использование ActionBar в связке с нижним и боковым меню может значительно улучшить пользовательский интерфейс и удобство навигации в приложении, например показывая текущий экран, на котором находится пользователь.

ActionBar уже встроен в стандартные темы Activity начиная с API Level 11 (Honeycomb). Для его использования необходимо убедиться, что выбранная Activity наследуется от AppCompatActivity, и используется одну из тем Theme.AppCompat. ActionBar будет отображаться вверху страницы (Рисунок 15).

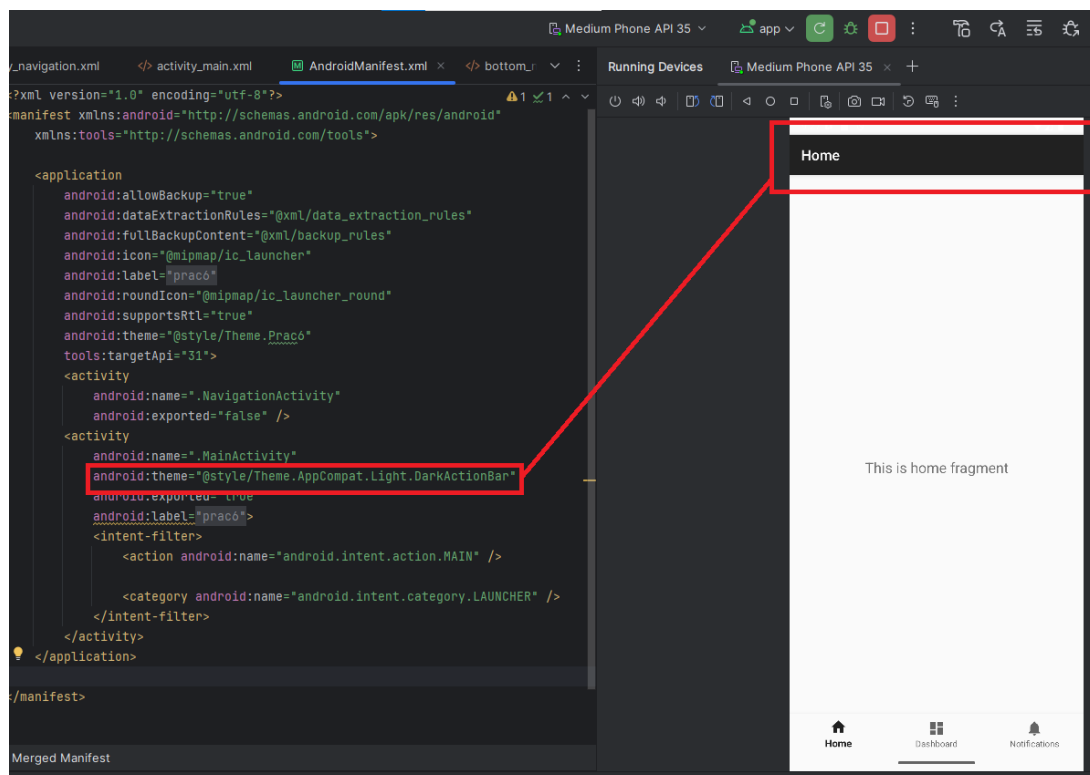


Рисунок 15 – Графическое и кодовое представление ActionBar

Однако такой ActionBar не несет никакой пользы. Для дальнейшей работы его необходимо настроить. В активности можно настроить ActionBar на отображение какого-либо строкового значения (Рисунок 16).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Получаем ActionBar
    ActionBar actionBar = getSupportActionBar();
    if (actionBar != null) {
        // Настройки ActionBar
        actionBar.setDisplayHomeAsUpEnabled(true); // Показать кнопку назад
        actionBar.setTitle("Главная"); // Установить заголовок
        // Другие настройки ActionBar
    }
}
```

Рисунок 16 – Настройка ActionBar в коде

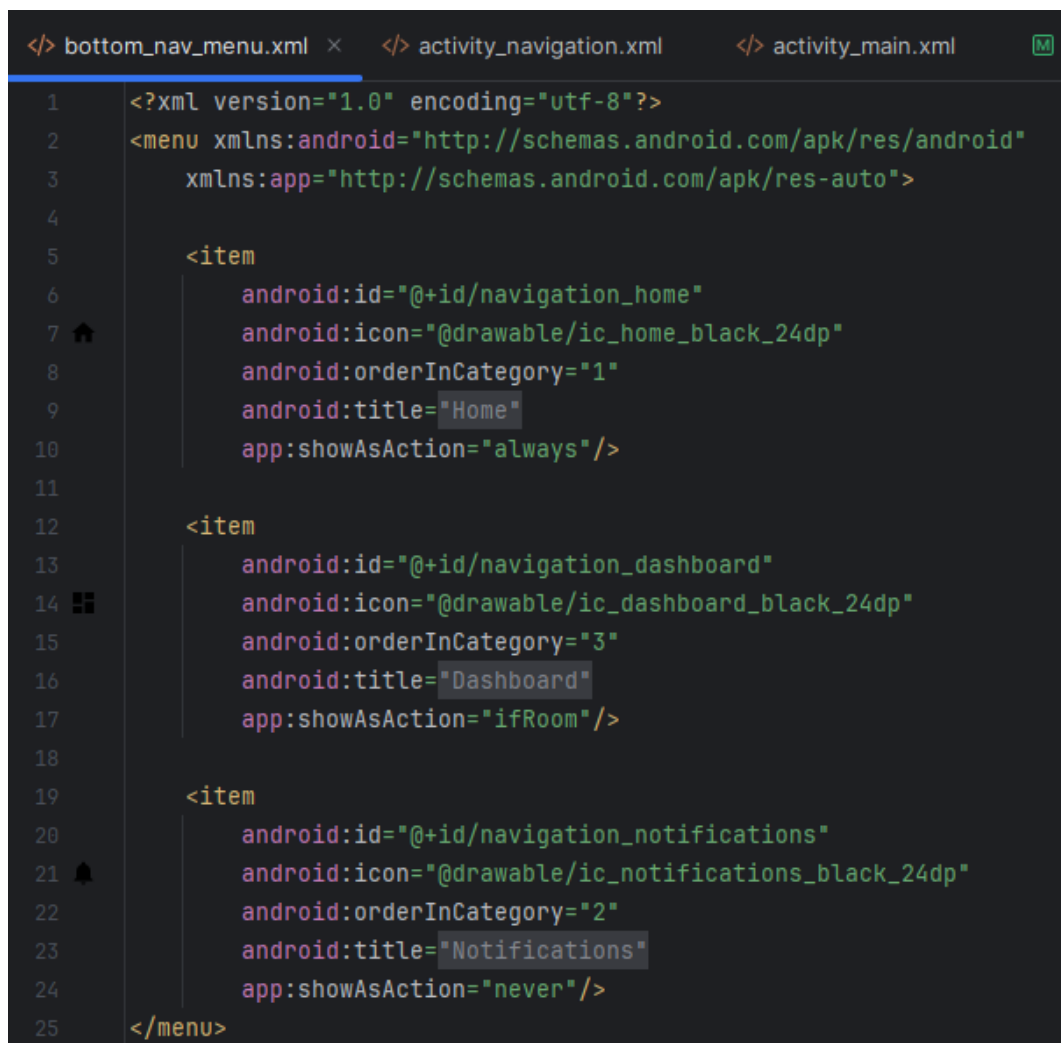
Однако этого тоже может быть недостаточно, так как нам может понадобиться быстрый доступ к некоторым элементам.

Используем ранее созданный файл `bottom_nav_menu.xml`, только немного его изменим, добавив элементам меню некоторые параметры для наглядности (Рисунок 17).

- `android:orderInCategory`: значение этого атрибута определяет положение элемента в `ActionBar`. Есть два способа определить положение различных пунктов меню. Первый - предоставить одинаковое значение этого атрибута для всех элементов, и позиция будет определена в том же порядке, в каком они объявлены в коде. Второй способ - предоставить разные числовые значения для всех элементов, и тогда элементы будут располагаться в соответствии с порядком возрастания значения этого атрибута;
- `app:showAsAction`: этот атрибут определяет, как элемент будет присутствовать на панели действий.

На выбор для значения поля `app:showAsAction` предлагается четыре возможных флага:

- `always`: постоянно отображать элемент на панели действий;
- `ifRoom`: сохранить элемент, если есть свободное место;
- `never`: с этим флагом элемент не будет отображаться в виде значка в `ActionBar`, но будет присутствовать в меню переполнения;
- `withText`: чтобы представить элемент одновременно в виде значка и заголовка, можно дополнить этот флаг флагом `always` или `ifRoom` (`always / withText` or `ifRoom / withText`).



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto">
4
5      <item
6          android:id="@+id/navigation_home"
7          android:icon="@drawable/ic_home_black_24dp"
8          android:orderInCategory="1"
9          android:title="Home"
10         app:showAsAction="always"/>
11
12     <item
13         android:id="@+id/navigation_dashboard"
14         android:icon="@drawable/ic_dashboard_black_24dp"
15         android:orderInCategory="3"
16         android:title="Dashboard"
17         app:showAsAction="ifRoom"/>
18
19     <item
20         android:id="@+id/navigation_notifications"
21         android:icon="@drawable/ic_notifications_black_24dp"
22         android:orderInCategory="2"
23         android:title="Notifications"
24         app:showAsAction="never"/>
25 </menu>

```

**Рисунок 17 – Измененный файл разметки bottom\_nav\_menu.xml**

На рисунке 17 определено меню с тремя элементами, но само определение элементов в файле еще не создает меню. Это всего лишь описание. Чтобы вывести его на экран, надо использовать его в классе Activity. Для этого надо переопределить метод `onCreateOptionsMenu`.

Метод `getMenuInflater` возвращает объект `MenuInflater`, у которого вызывается метод `inflate()`. Этот метод в качестве первого параметра принимает ресурс, представляющий наше описание меню в xml, и наполняет им объект `menu`, переданный в качестве второго параметра.

Теперь, если мы нажмем на любой из пунктов меню, то ничего не произойдет. Чтобы привязать к меню действия, нам надо переопределить в классе `activity` метод `onOptionsItemSelected` (Рисунок 18).

```

19 public class MainActivity extends AppCompatActivity {
24     protected void onCreate(Bundle savedInstanceState) {
27         binding = ActivityMainBinding.inflate(getLayoutInflater());
28         setContentView(binding.getRoot());
29
30         BottomNavigationView navView = findViewById(R.id.nav_view);
31         // Passing each menu ID as a set of IDs because each
32         // menu should be considered as top level destinations.
33         AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
34             R.id.navigation_home, R.id.navigation_dashboard, R.id.navigation_notifications)
35             .build();
36         NavController navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment_activity_main);
37         NavigationUI.setupActionBarWithNavController( activity: this, navController, appBarConfiguration);
38         NavigationUI.setupWithNavController(binding.navView, navController);
39     }
40
41     @Override
42     public boolean onCreateOptionsMenu( Menu menu ) {
43         getMenuInflater().inflate(R.menu.bottom_nav_menu, menu);
44         return super.onCreateOptionsMenu(menu);
45     }
46
47     @Override
48     public boolean onOptionsItemSelected( @NonNull MenuItem item ) {
49         //Определение нажатого элемента
50         if (item.getItemId() == R.id.navigation_home) {
51             // Обработка выбора раздела "Домой"
52             Toast.makeText( context: MainActivity.this, text: "Домой", Toast.LENGTH_LONG).show();
53             return true;
54         } else if (item.getItemId() == R.id.navigation_notifications) {
55             // Обработка выбора раздела "Настройки"
56             Toast.makeText( context: MainActivity.this, text: "Настройки", Toast.LENGTH_LONG).show();
57         }
58         return super.onOptionsItemSelected(item);
59     }
60 }

```

Рисунок 18 – Переопределение необходимых методов класса ActionBar

И при нажатии на троеточие в правом верхнем углу экрана, а после нажатия на “Notifications” появится сообщение “Notifications” (Рисунок 19).

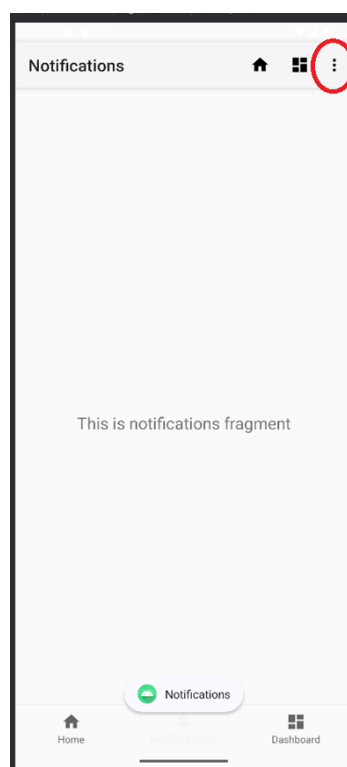
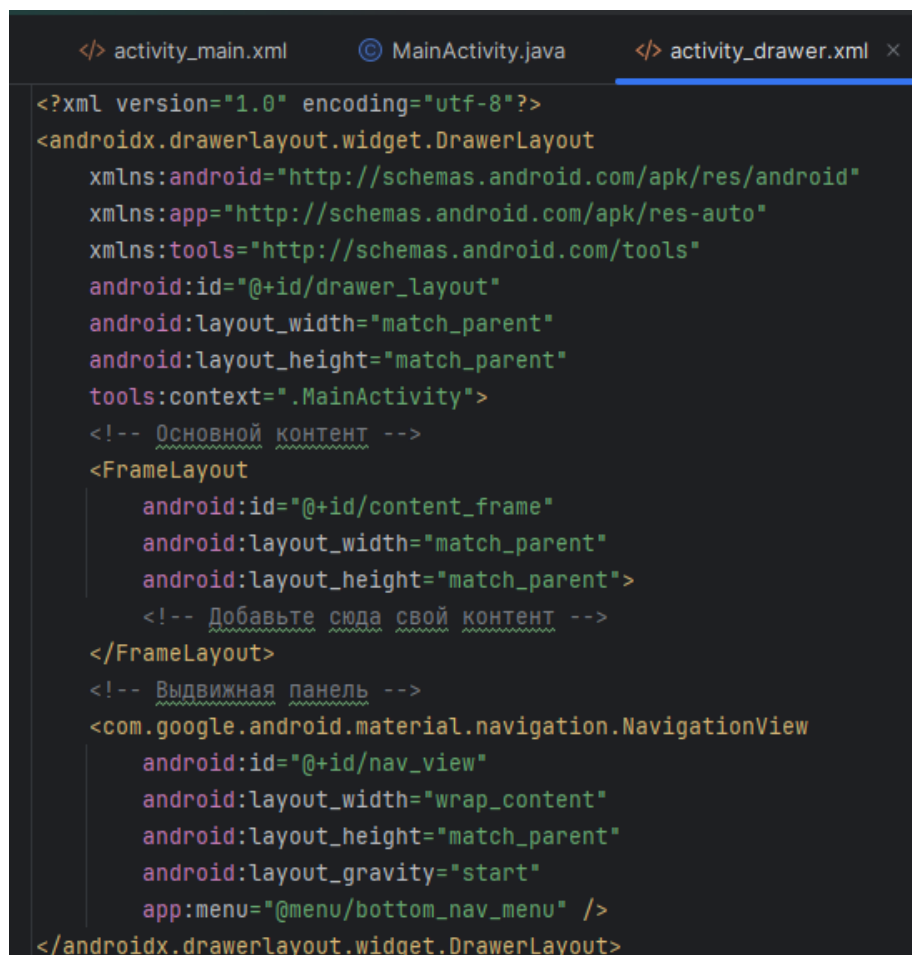


Рисунок 19 – Отображение ActionBar в приложении

## 1.3 Navigation Drawer

Создание выдвижной панели (Drawer) в Android обычно реализуется с помощью DrawerLayout и NavigationView. DrawerLayout используется как корневой контейнер интерфейса, который позволяет разместить основное содержимое приложения и выдвижную панель. NavigationView представляет собой панель навигации, которая отображается в DrawerLayout и содержит элементы меню для навигации по различным разделам приложения.

В файле макета активности необходимо добавить DrawerLayout как корневой элемент. Внутри DrawerLayout разместим основной контент и NavigationView для выдвижной панели (Рисунок 20).



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <!-- Основной контент -->
    <FrameLayout
        android:id="@+id/content_frame"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <!-- Добавьте сюда свой контент -->
    </FrameLayout>
    <!-- Выдвижная панель -->
    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:menu="@menu/bottom_nav_menu" />
</androidx.drawerlayout.widget.DrawerLayout>
```

Рисунок 20 – Код файла разметки activity\_drawer.xml

Создаётся XML-файл в папке res/menu (например, drawer\_menu.xml) и добавьте пункты меню для вашей выдвижной панели (Рисунок 21).



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3      <item
4          android:id="@+id/nav_home"
5          android:icon="@drawable/ic_home_black_24dp"
6          android:title="Home" />
7      <item
8          android:id="@+id/nav_notifications"
9          android:icon="@drawable/ic_dashboard_black_24dp"
10         android:title="DashB" />
11      <item
12          android:id="@+id/nav_setting"
13          android:icon="@drawable/ic_notifications_black_24dp"
14          android:title="Notif" />
15  </menu>

```

Рисунок 21 – Описание элементов в файле drawer\_menu.xml

Затем по аналогии с BottomBar создать слушатель нажатий для каждого элемента меню.

Для интеграции ActionBar с DrawerLayout используется ActionBarDrawerToggle, который добавит иконку меню для открытия и закрытия Drawer и обеспечит анимацию иконки (Рисунок 22).

```

14  public class DrawerActivity extends AppCompatActivity {
15      ActionBarDrawerToggle toggle;
16
17      @Override
18      protected void onCreate(Bundle savedInstanceState) {
19          super.onCreate(savedInstanceState);
20          EdgeToEdge.enable(this);
21          setContentView(R.layout.activity_drawer);
22          ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.drawer_layout), (v, insets) -> {
23              Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
24              v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
25              return insets;
26          });
27
28          DrawerLayout drawer = findViewById(R.id.drawer_layout);
29          toggle = new ActionBarDrawerToggle(
30              activity, drawer, R.string.drawer_open, R.string.drawer_close);
31          if (drawer != null) {
32              drawer.addDrawerListener(toggle);
33          }
34          toggle.syncState();
35          // to make the Navigation drawer icon always appear on the action bar
36          getSupportActionBar().setDisplayHomeAsUpEnabled(true);
37      }
38
39      // обработка нажатия на иконку меню в ActionBar для открытия и закрытия Drawer
40      @Override
41      public boolean onOptionsItemSelected(MenuItem item) {
42          if (toggle.onOptionsItemSelected(item)) {
43              return true;
44          }
45          return super.onOptionsItemSelected(item);
46      }
47  }
48

```

Рисунок 22 – Реализация логики NavigationDrawer

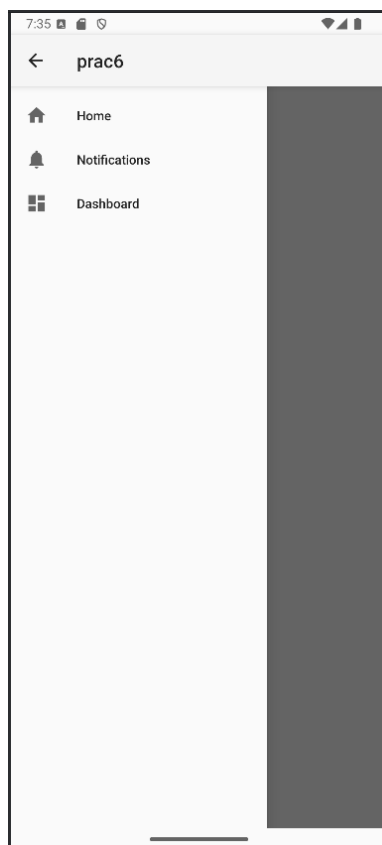
Также на рисунке 22 был добавлен слушатель для обработки нажатия на элементы (onOptionsItemSelected).

В итоге будет отображаться всплывающее меню с выбранными строковыми значениями (Рисунки 23-24).



**Рисунок 23 – Отображение название всплывающего меню, часть 1**

После запуска приложения пользователь нажмёт на иконку всплывающего меню, находящуюся в левом верхнем углу, и увидит содержимое всплывающего меню (Рисунок 24).



**Рисунок 24 – Отображение названия всплывающего меню, часть 2**

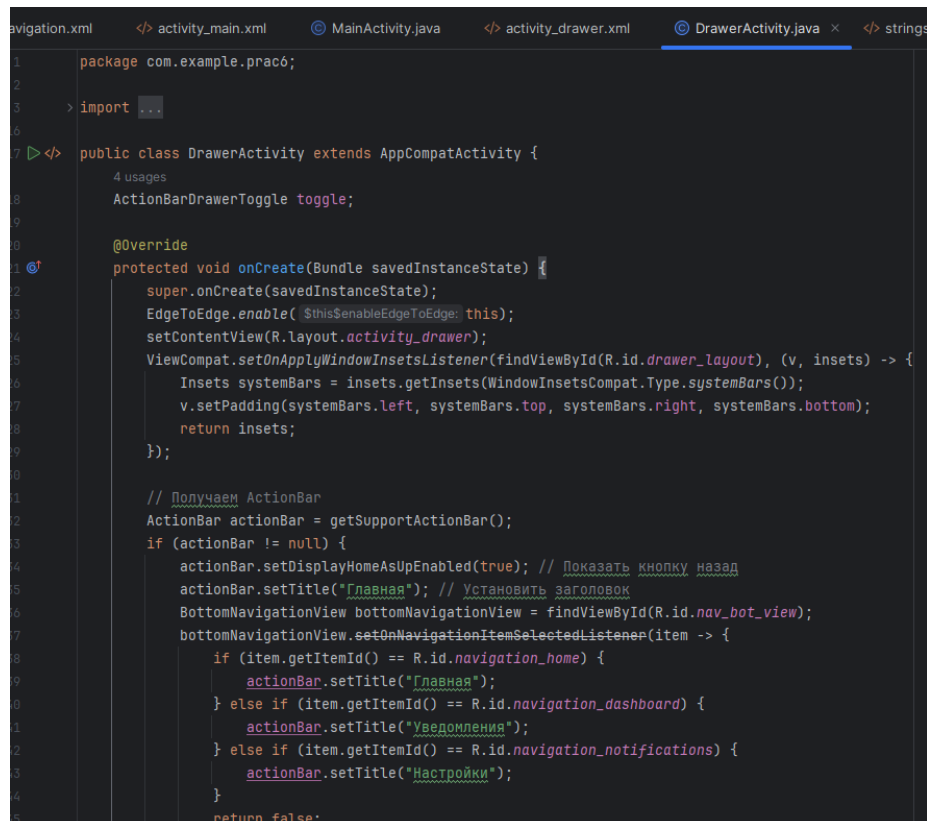
Чтобы использовать `ActionBarDrawerToggle`, нужно создать экземпляр при помощи конструктора со следующими аргументами:

- Activity, в котором размещается боковая панель навигации;
- `DrawerLayout` – drawable ресурс, используемый в качестве индикатора панели. Стандартный навигационный значок панели доступен в [Download the Action Bar Icon Pack](#);
- Строковый ресурс для обозначения открытой панели (для специальных возможностей);
- Строковый ресурс для обозначения закрытой панели (для специальных возможностей).

Теперь, в зависимости от использования класса `ActionBarDrawerToggle` в списке панели навигации, необходимо вызвать `ActionBarDrawerToggle` в нескольких местах жизненного цикла activity.

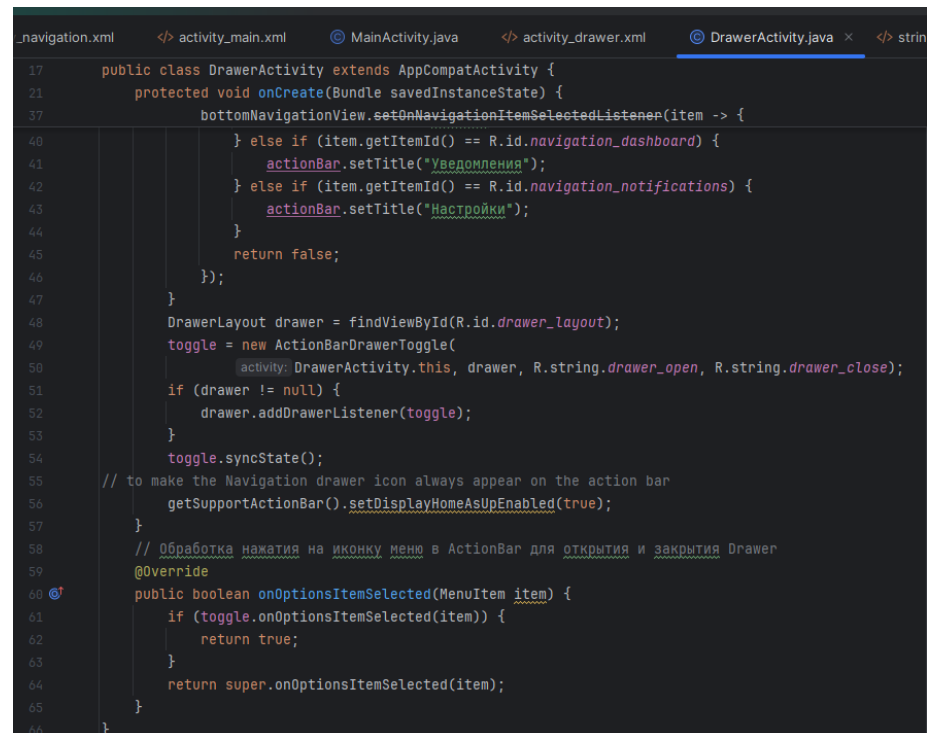
Кроме этого, `ActionBar` можно использовать вместе с `BottomNavigationView` для создания единой системы навигации. В этом случае

ActionBar обычно служит для отображения контекстной информации (например, заголовка текущей страницы), а BottomNavigationView для навигации между основными разделами приложения (Рисунки 25-26).



```
1 package com.example.prac6;  
2  
3 > import ...  
4  
5  
6  
7 public class DrawerActivity extends AppCompatActivity {  
8     ActionBarDrawerToggle toggle;  
9  
10  
11     @Override  
12     protected void onCreate(Bundle savedInstanceState) {  
13         super.onCreate(savedInstanceState);  
14         EdgeToEdge.enable(this);  
15         setContentView(R.layout.activity_drawer);  
16         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.drawer_layout), (v, insets) -> {  
17             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
18             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
19             return insets;  
20         });  
21  
22         // Получаем ActionBar  
23         ActionBar actionBar = getSupportActionBar();  
24         if (actionBar != null) {  
25             actionBar.setDisplayHomeAsUpEnabled(true); // Показать кнопку назад  
26             actionBar.setTitle("Главная"); // Установить заголовок  
27             BottomNavigationView bottomNavigationView = findViewById(R.id.nav_bot_view);  
28             bottomNavigationView.setOnNavigationItemSelectedListener(item -> {  
29                 if (item.getItemId() == R.id.navigation_home) {  
30                     actionBar.setTitle("Главная");  
31                 } else if (item.getItemId() == R.id.navigation_dashboard) {  
32                     actionBar.setTitle("Уведомления");  
33                 } else if (item.getItemId() == R.id.navigation_notifications) {  
34                     actionBar.setTitle("Настройки");  
35                 }  
36                 return false;  
37             });  
38         }  
39     }  
40 }  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

Рисунок 25 – Реализация логики BottomNavigationView, часть 1



```
17 public class DrawerActivity extends AppCompatActivity {  
21     protected void onCreate(Bundle savedInstanceState) {  
37         bottomNavigationView.setOnNavigationItemSelectedListener(item -> {  
40             } else if (item.getItemId() == R.id.navigation_dashboard) {  
41                 actionBar.setTitle("Уведомления");  
42             } else if (item.getItemId() == R.id.navigation_notifications) {  
43                 actionBar.setTitle("Настройки");  
44             }  
45             return false;  
46         });  
47     }  
48     DrawerLayout drawer = findViewById(R.id.drawer_layout);  
49     toggle = new ActionBarDrawerToggle(  
50         activity, DrawerActivity.this, drawer, R.string.drawer_open, R.string.drawer_close);  
51     if (drawer != null) {  
52         drawer.addDrawerListener(toggle);  
53     }  
54     toggle.syncState();  
55     // to make the Navigation drawer icon always appear on the action bar  
56     getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
57 }  
58 // Обработка нажатия на иконку меню в ActionBar для открытия и закрытия Drawer  
59 @Override  
60 public boolean onOptionsItemSelected(MenuItem item) {  
61     if (toggle.onOptionsItemSelected(item)) {  
62         return true;  
63     }  
64     return super.onOptionsItemSelected(item);  
65 }  
66 }  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

Рисунок 26 – Реализация логики BottomNavigationView, часть 2

У пользователя появится возможность переходить по нижней панели на различные фрагменты (Рисунки 27-28).



**Рисунок 27 – Работа панели навигации, часть 1**



**Рисунок 28 – Работа панели навигации, часть 2**

Это позволяет пользователю быстро переключаться между разделами приложения с помощью нижнего меню, в то время как верхняя панель (ActionBar) отображает контекстную информацию о выбранном разделе или предоставляет дополнительные опции действий, связанных с текущим экраном. Использование ActionBar в сочетании с боковым и нижним меню позволяет создать гибкую и интуитивно понятную систему навигации для пользователей, обеспечивая легкий доступ ко всем важным функциям в приложении.

## 2 ПРАКТИЧЕСКОЕ ЗАДАНИЕ

### 2.1 Боковая панель навигации

#### 2.1.1 Разметка

Для начала создадим проект (Рисунок 30).

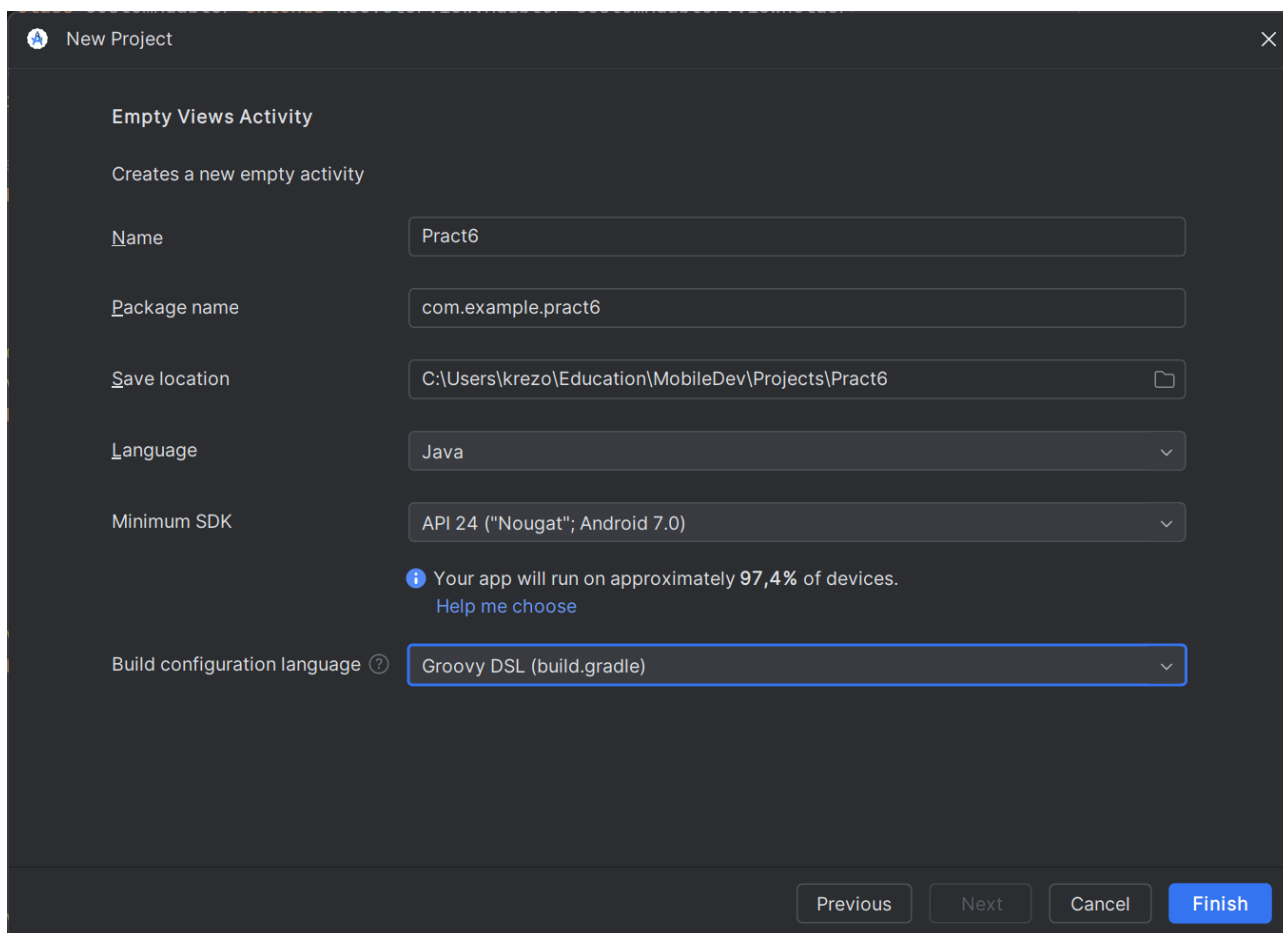
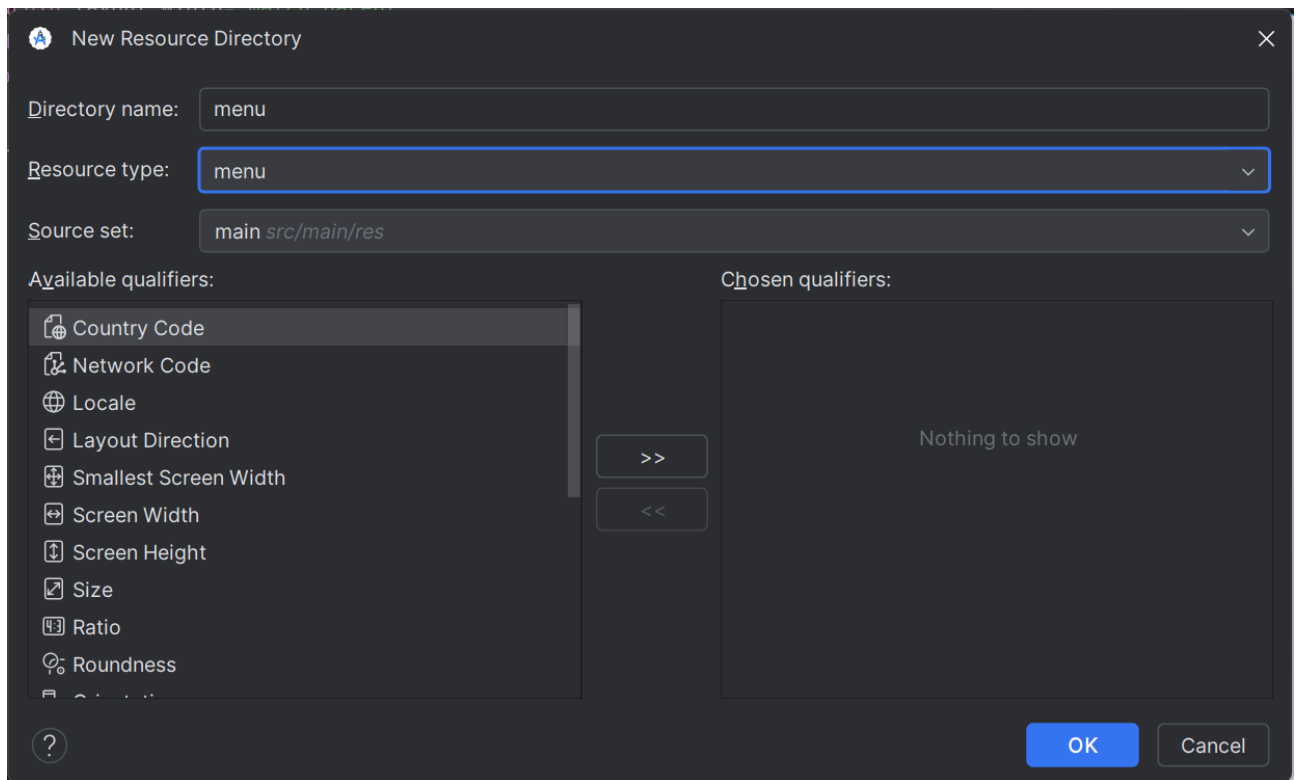
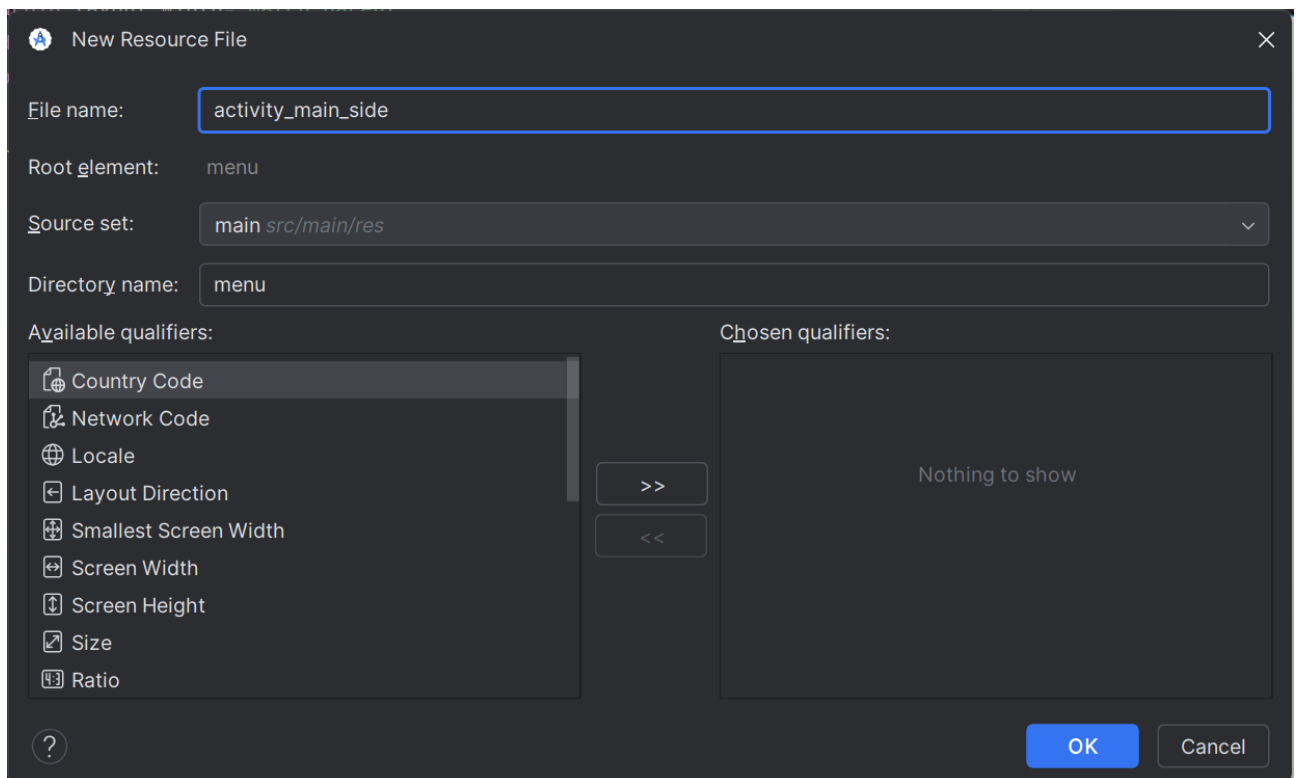


Рисунок 29 – Создание проекта

Создадим файл разметки бокового меню `activity_main_side.xml`, перед этим создав папку ресурсов `main` (Рисунки 30-31).



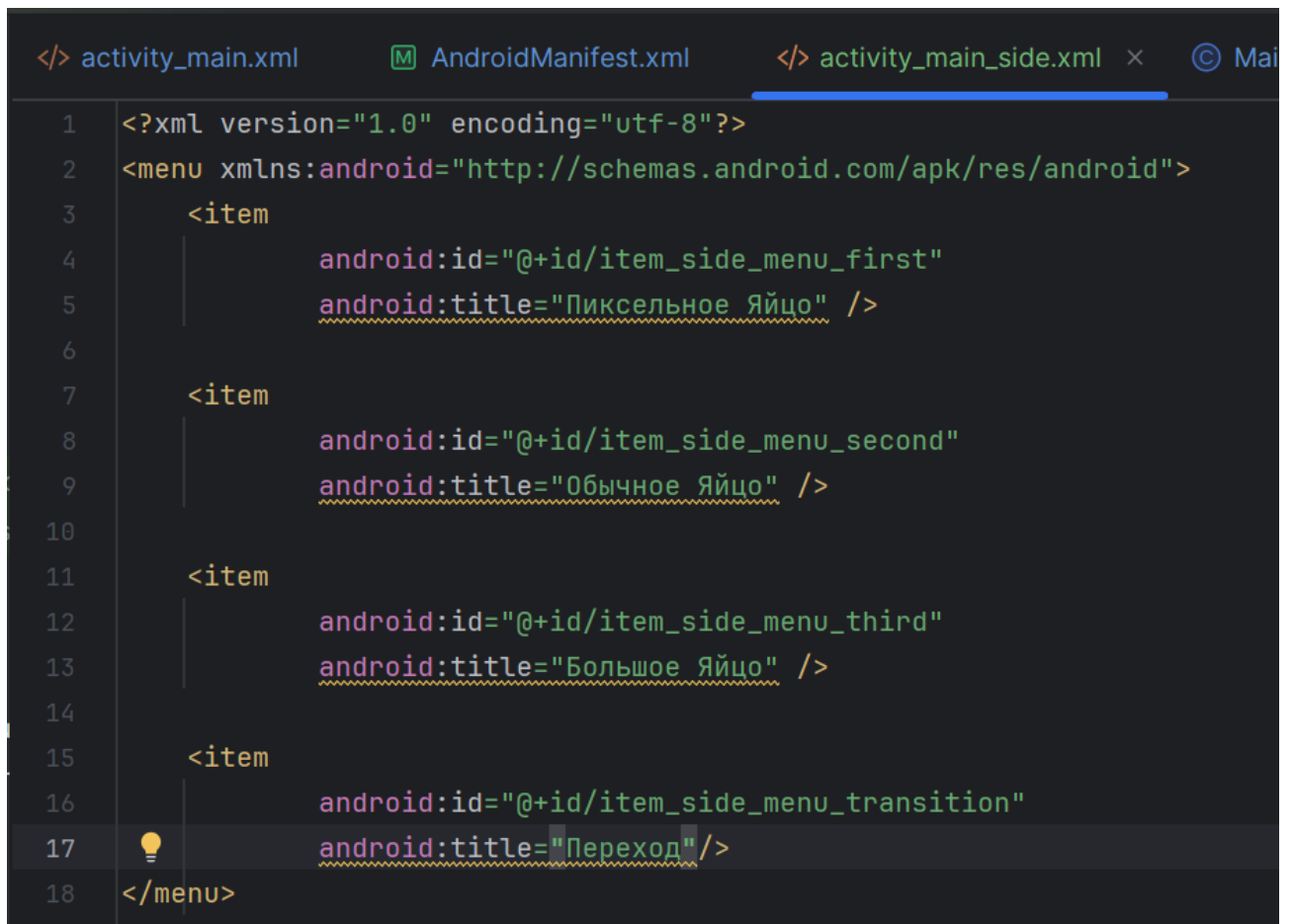
**Рисунок 30 – Создание папки ресурсов меню**



**Рисунок 31 – Создание файла разметки меню activity\_main\_side.xml**

Заполним новый файл разметки меню необходимыми элементами (Рисунок 32).

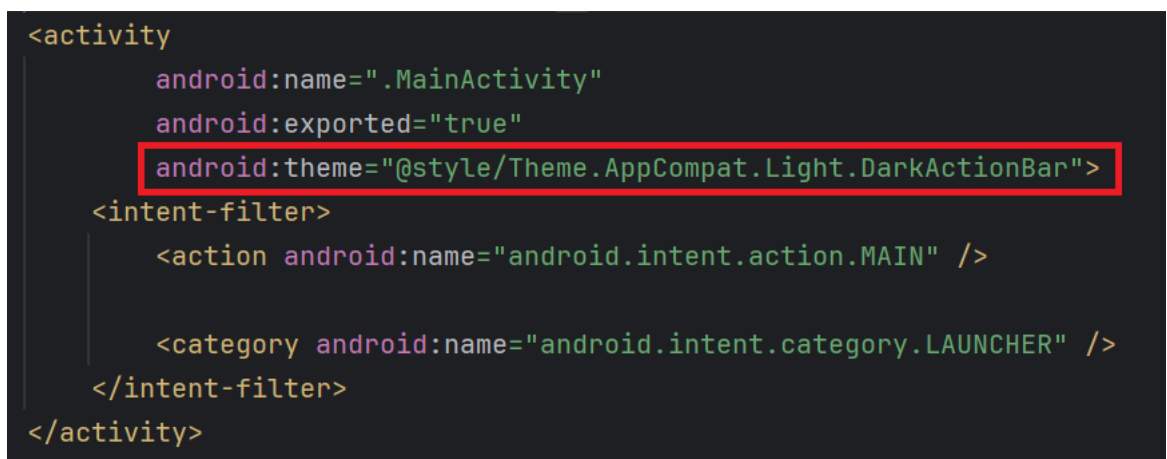




```
</> activity_main.xml  AndroidManifest.xml  </> activity_main_side.xml  x  © Mai
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3      <item
4          android:id="@+id/item_side_menu_first"
5          android:title="Пиксельное Яйцо" />
6
7      <item
8          android:id="@+id/item_side_menu_second"
9          android:title="Обычное Яйцо" />
10
11     <item
12         android:id="@+id/item_side_menu_third"
13         android:title="Большое Яйцо" />
14
15     <item
16         android:id="@+id/item_side_menu_transition"
17         android:title="Переход"/>
18 </menu>
```

Рисунок 32 – Файл разметки activity\_main\_side.xml

Для того, чтобы появился ActionBar, необходимо установить совместимую тему в манифесте (Рисунок 33).

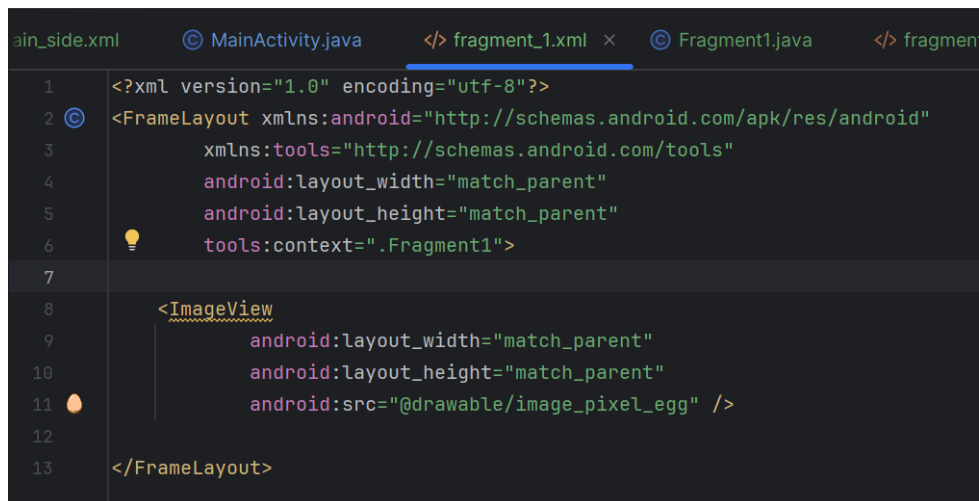


```
<activity
    android:name=".MainActivity"
    android:exported="true"
    android:theme="@style/Theme.AppCompat.Light.DarkActionBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Рисунок 33 – Настройки MainActivity в AndroidManifest.xml

Создадим три фрагмента Fragment1, Fragment2 и Fragment3 и заполним их аналогично фрагменту Fragment1 (Рисунок 34).



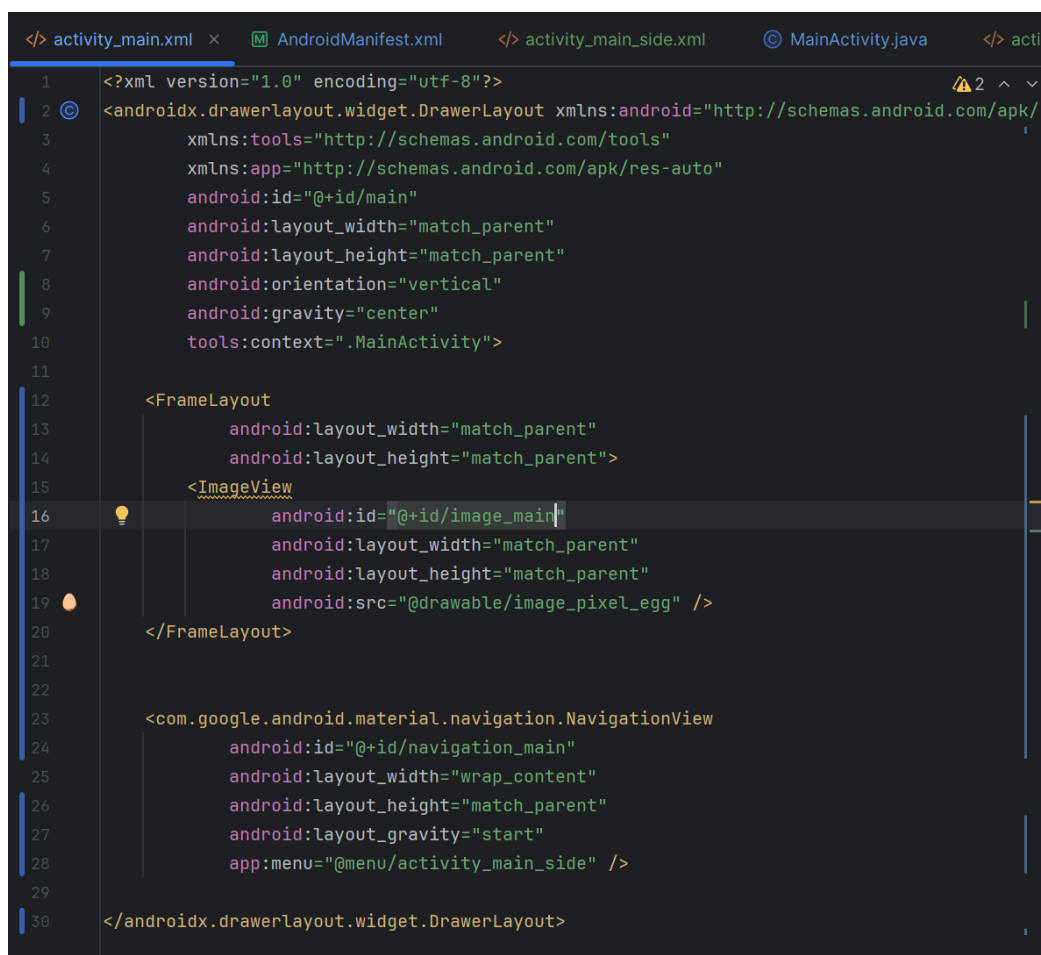
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      tools:context=".Fragment1">
7
8      <ImageView
9          android:layout_width="match_parent"
10         android:layout_height="match_parent"
11         android:src="@drawable/image_pixel_egg" />
12
13 </FrameLayout>

```

**Рисунок 34 – Файл разметки fragment\_1.xml**

Для отображения опций выпадающего меню в файл разметки activity\_main.xml добавим элемент `NavigationView`, где установим ссылку на разметку меню. Также расположим картинку на весь экран (Рисунок 35).



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:orientation="vertical"
9      android:gravity="center"
10     tools:context=".MainActivity">
11
12     <FrameLayout
13         android:layout_width="match_parent"
14         android:layout_height="match_parent">
15
16         <ImageView
17             android:id="@+id/image_main"
18             android:layout_width="match_parent"
19             android:layout_height="match_parent"
20             android:src="@drawable/image_pixel_egg" />
21         </FrameLayout>
22
23     <com.google.android.material.navigation.NavigationView
24         android:id="@+id/navigation_main"
25         android:layout_width="wrap_content"
26         android:layout_height="match_parent"
27         android:layout_gravity="start"
28         app:menu="@menu/activity_main_side" />
29
30 </androidx.drawerlayout.widget.DrawerLayout>

```

**Рисунок 35 – Файл разметки activity\_main.xml**

Создадим класс Activity и назовем его `SecondActivity` (Рисунок 36).

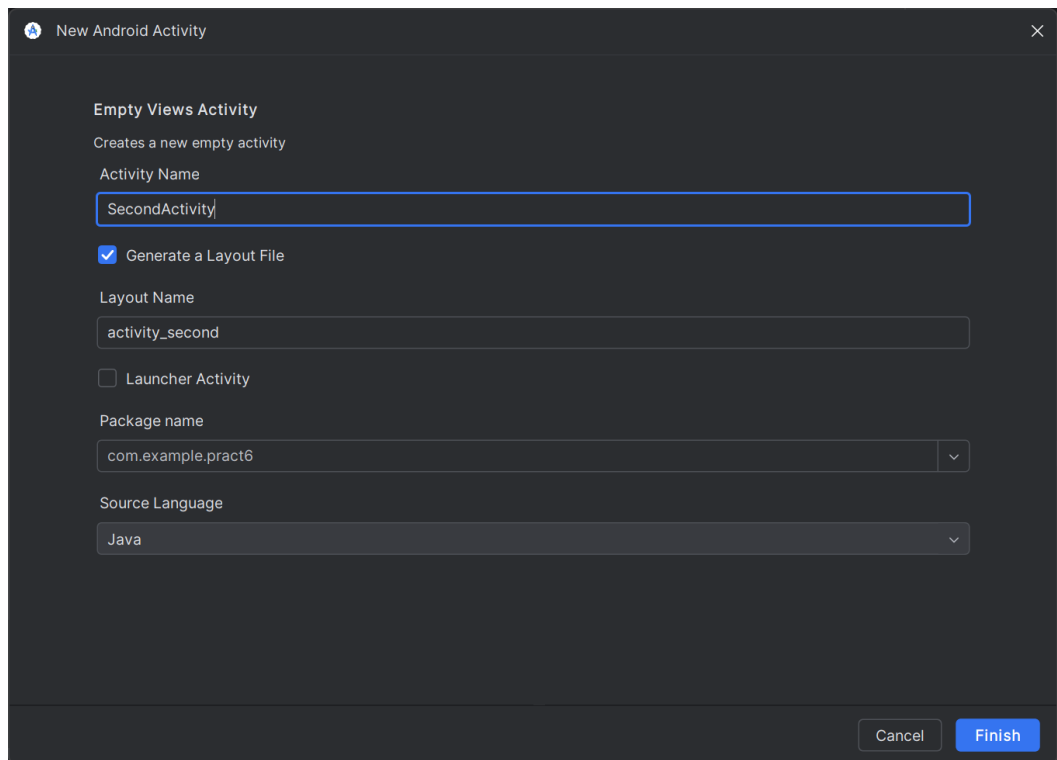


Рисунок 36 – Создание SecondActivity

### 2.1.2 Код

В методе `onCreate()` класса `ActivityMain` создадим кнопку для открытия выпадающего бокового меню и укажем название (Рисунок 37).

```
drawer = findViewById(R.id.main);
toggle = new ActionBarDrawerToggle(activity: MainActivity.this, drawer, R.string.drawer_open, R.string.drawer_close);
drawer.addDrawerListener(toggle);
toggle.syncState();

ActionBar bar = getSupportActionBar();
if (bar == null) return;

bar.setDisplayHomeAsUpEnabled(true);
bar.setTitle("Яйца");
```

Рисунок 37 – Описание метода `onCreate()` класса `MainActivity`, часть 1

В остальной части метода реализуем обработку нажатия на разные кнопки меню (Рисунок 38).

```

NavigationView nav = drawer.findViewById(R.id.navigation_main);
nav.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener()
{
    1 usage
    @Override
    public boolean onNavigationItemSelectedListener(@NonNull MenuItem item)
    {
        FragmentManager fragmentManager = getSupportFragmentManager();
        int id = item.getItemId();
        if (id == R.id.item_side_menu_first)
            changeFragment(new Fragment1(), fragmentManager);
        else if (id == R.id.item_side_menu_second)
            changeFragment(new Fragment2(), fragmentManager);
        else if (id == R.id.item_side_menu_third)
            changeFragment(new Fragment3(), fragmentManager);
        else if (id == R.id.item_side_menu_transition)
        {
            Intent intent = new Intent( packageContext: MainActivity.this, SecondActivity.class);
            startActivity(intent);
        }

        drawer.closeDrawers();
        return true;
    }
});

```

Рисунок 38 – Описание метода onCreate() класса MainActivity, часть 2

Опишем метод changeFragment(), для смены отображаемого фрагмента. Также для работы ActionBarDrawer перегрузим метод onOptionsItemSelected(), в котором требуется обработать нажатие на кнопку, открывающую меню (Рисунок 39).

```

3 usages
private void changeFragment(Fragment newFragment, FragmentManager manager)
{
    manager.beginTransaction()
        .replace(R.id.fragmentContainer_main, newFragment)
        .commit();
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item)
{
    if (toggle.onOptionsItemSelected(item)) return true;

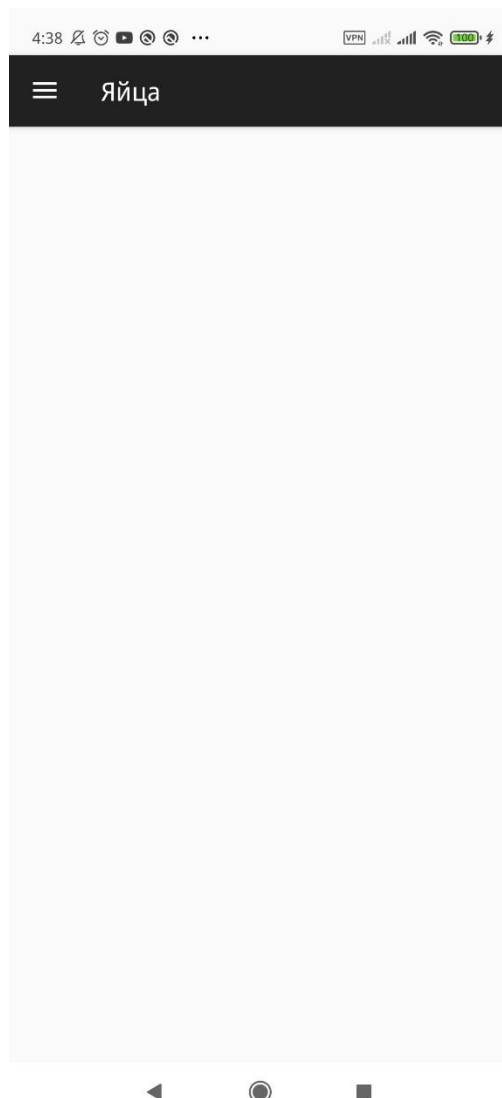
    return super.onOptionsItemSelected(item);
}

```

Рисунок 39 – Описание методов changeFragment() и onOptionsItemSelected() класса MainActivity

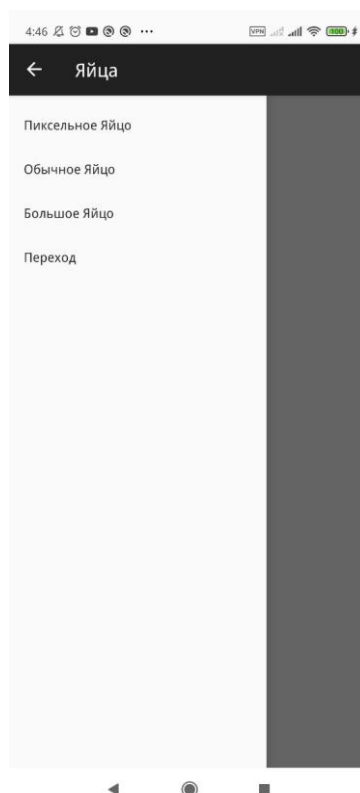
### 2.1.3 Тестирование

Зайдем в приложение. Изначально на экране картинка не выбрана (Рисунок 40).



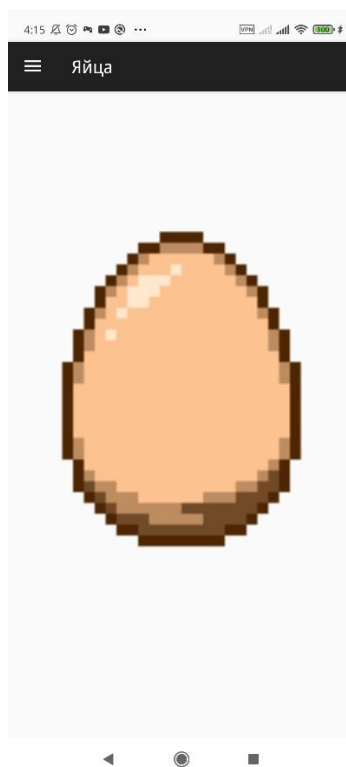
**Рисунок 40 – Начальный экран приложения**

При нажатии на иконку в левом верхнем углу открывается левая панель навигации (Рисунок 41).

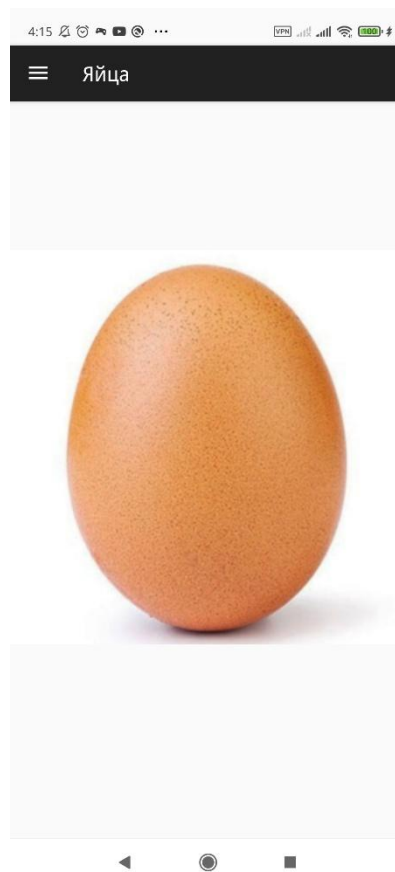


**Рисунок 41 – Левая панель навигации**

На рисунках 42-45 представлен результат нажатия на опции с 1 по 4 соответственно.



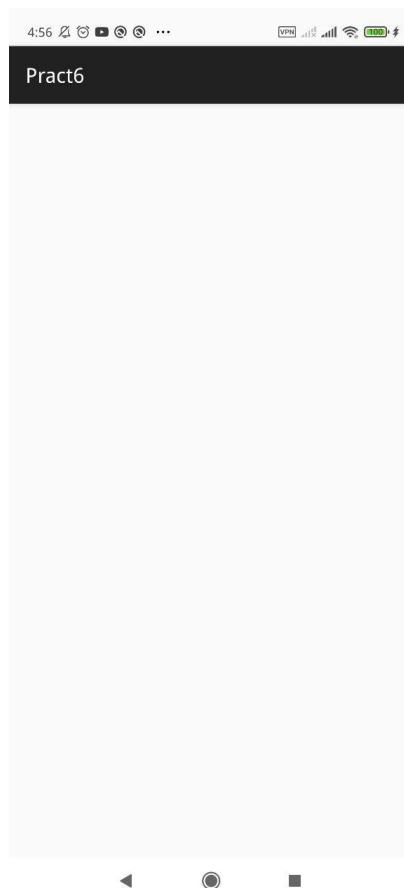
**Рисунок 42 – Нажатие на опцию «Пиксельное Яйцо»**



**Рисунок 43 – Нажатие на опцию «Обычное Яйцо»**



**Рисунок 44 – Нажатие на опцию «Большое Яйцо»**



**Рисунок 45 – Нажатие на опцию «Переход»**

При нажатии на опцию «Переход» происходит переход на вторую Activity.

## **2.2 Нижняя панель навигации**

### **2.2.1 Разметка**

Для начала создадим три новых фрагмента Fragment4, Fragment5 и Fragment6 по аналогии с Fragment1. Потом создадим файл разметки нижнего меню activity\_second\_bottom.xml (Рисунок 46).



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android">
3      <item
4          android:id="@+id/item_bottom_menu_bro1"
5          android:title="Услышал тебя"
6          android:icon="@drawable/ic_button_1"/>
7
8      <item
9          android:id="@+id/item_bottom_menu_bro2"
10         android:title="Не слышу тебя"
11         android:icon="@drawable/ic_button_2"/>
12
13     <item
14         android:id="@+id/item_bottom_menu_grub"
15         android:title="Гусеница"
16         android:icon="@drawable/ic_button_3"/>
17 </menu>

```

Рисунок 46 – Файл разметки activity\_second\_bottom.xml

После этого создадим необходимые иконки и заполним файл разметки activity\_second.xml (Рисунок 47).

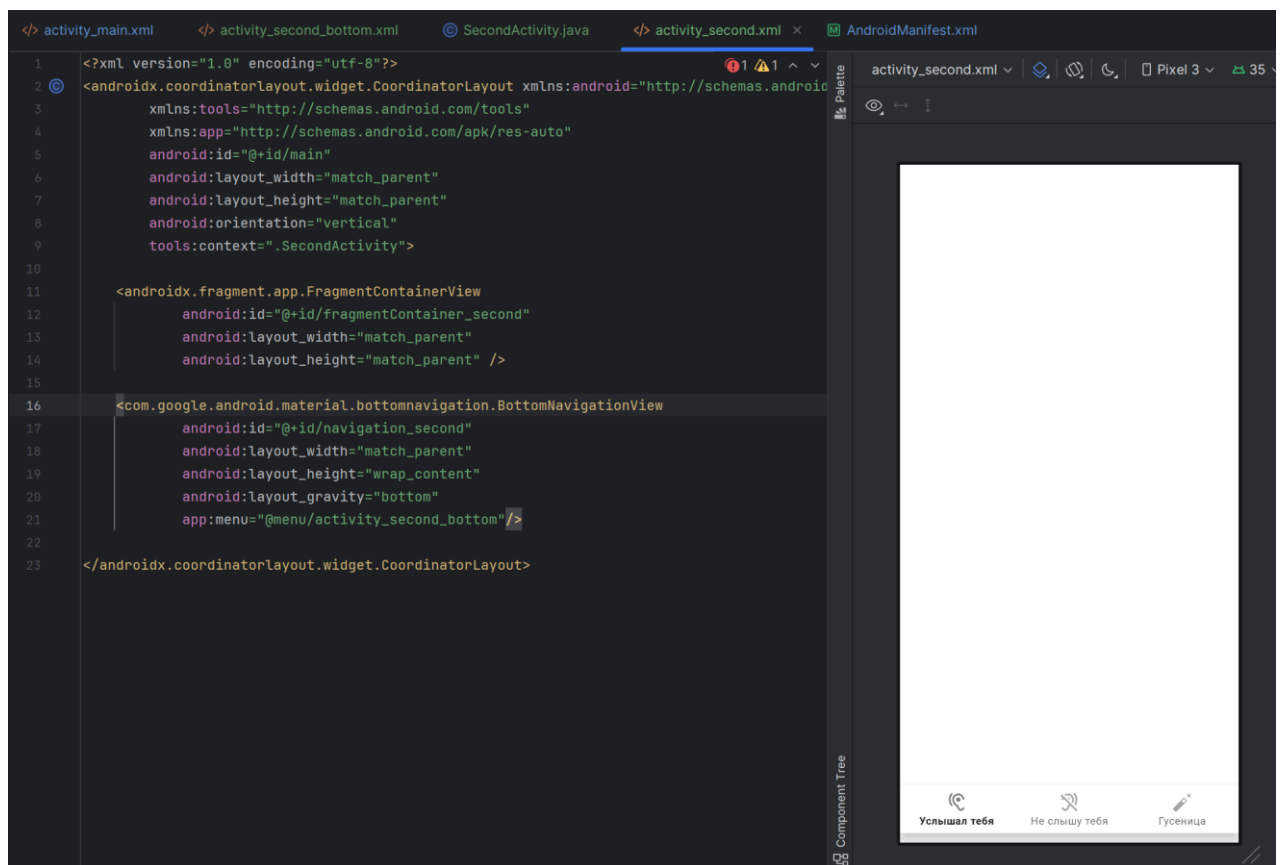


Рисунок 47 – Файл разметки activity\_second.xml

### 2.2.2 Код

В методе onCreate() класса SecondActivity аналогично классу MainActivity реализуем логику работы нижнего меню (Рисунок 48).

```
ActionBar bar = getSupportActionBar();
if (bar == null) return;

bar.setDisplayHomeAsUpEnabled(true);
bar.setTitle("Вторая страница");

BottomNavigationView bottomNav = findViewById(R.id.navigation_second);
bottomNav.setOnItemSelectedListener(new NavigationBarView.OnItemSelectedListener()
{
    1 usage
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item)
    {
        FragmentManager fragmentManager = getSupportFragmentManager();
        int id = item.getItemId();
        if (id == R.id.item_bottom_menu_bro1)
        {
            bar.setTitle("Услышал");
            changeFragment(new Fragment4(), fragmentManager);
        }
        else if (id == R.id.item_bottom_menu_bro2)
        {
            bar.setTitle("Не слышу");
            changeFragment(new Fragment5(), fragmentManager);
        }
        else if (id == R.id.item_bottom_menu_grub)
        {
            bar.setTitle("Гусеница");
            changeFragment(new Fragment6(), fragmentManager);
        }

        return true;
    }
});
```

Рисунок 48 – Описание метода onCreate() класса SecondActivity

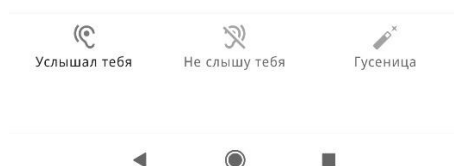
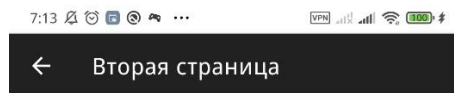
Описание метода changeFragment() представлено на рисунке (Рисунок 49).

```
3 usages
private void changeFragment(Fragment newFragment, FragmentManager manager)
{
    manager.beginTransaction()
        .replace(R.id.fragmentContainer_second, newFragment)
        .commit();
}
```

Рисунок 49 – Описание метода changeFragment() класса SecondActivity

### 2.2.3 Тестирование

Сначала запустим приложение и перейдем на вторую страницу с помощью последней опции бокового выпадающего меню (Рисунок 50).



**Рисунок 50 – Начальный вид второй страницы**

После этого нажимаем на кнопки нижнего меню (Рисунки 51-53).



**Рисунок 51 – Вторая страница после нажатия на «Услышал тебя»**



**Рисунок 52 – Вторая страница после нажатия на «Не слышу тебя»**



**Рисунок 53 – Вторая страница после нажатия на «Гусеница»**  
На панели ActionBar корректно отображается желаемое название.

## **ЗАКЛЮЧЕНИЕ**

В ходе данной практической работы были получены знания связанные с `NavigationDrawer`, `ActionBar` и `BottomBar`. Полученные знания были закреплены путём выполнения практического задания.