

MOSIS – Ispravke

U dosadašnjim vežbama je sva navigacija između fragmenata rađena korišćenjem akcija. U mnogim situacijama je mnogo optimalnije vršiti vraćanje na pozivajući fragment korišćenjem funkcije *findNavController().popBackStack()*. Da bi adekvatno bila ažurirana logika navigacije, potrebno je obrisati nepotrebne akcije u *NavGraph*-u.

1. Izmeniti *NavGraph* tako da izgleda kao na slici.



2. Izmeniti u *ViewFragment.kt* sadržaj klika na dugme.

Android

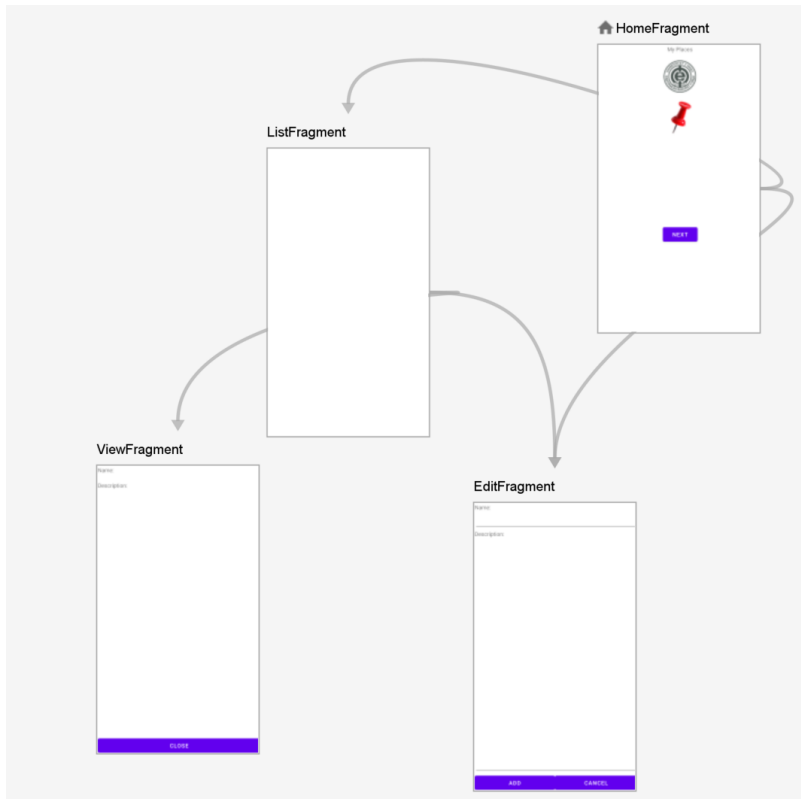
```
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
    super.onViewCreated(view, savedInstanceState)  
    binding.viewmyplaceNameText.text=myPlacesViewModel.selected?.name  
    binding.viewmyplaceDescText.text=myPlacesViewModel.selected?.description  
    binding.viewmyplaceFinishedButton.setOnClickListener { it: View! -> {  
        findNavController().popBackStack()  
    }}  
}
```

3.

4. U *EditFragment.kt* fragmentu zameniti pozive u navigacijama koje koriste akcije *NavGraph-a* tako da koriste *findNavController().popBackStack()* a potrebno je ukinuti pristup meniju u zaglavlju aplikacije kako bi navigacija ostala konzistentna.

```
package elfak.mosis.myplaces  
  
import ...  
  
class EditFragment : Fragment() {  
    private val myPlacesViewModel: MyPlacesViewModel by activityViewModels()  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setHasOptionsMenu(true)  
    }  
    override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {  
        inflater.inflate(R.menu.menu_main, menu)  
    }  
    override fun onCreateView(  
        inflater: LayoutInflater, container: ViewGroup?,  
        savedInstanceState: Bundle?,  
        rootId: Int): View? {  
        val desc: String = editDesc.text.toString()  
        if(myPlacesViewModel.selected!=null){  
            myPlacesViewModel.selected?.name = name  
            myPlacesViewModel.selected?.description = desc  
        }  
        else  
            myPlacesViewModel.addPlace( MyPlace(name, desc))  
        findNavController().popBackStack()  
        val cancelButton: Button = requireView().findViewById<Button>(R.id.editmyplace_cancel_button)  
        cancelButton.setOnClickListener { it: View! -> {  
            findNavController().popBackStack()  
        }}  
    }  
    override fun onDestroyView() {  
        super.onDestroyView()  
        myPlacesViewModel.selected = null  
    }  
    override fun onOptionsItemSelected(item: MenuItem): Boolean {  
        return when (item.itemId) {  
            R.id.action_my_places_list -> {  
                this.findNavController().navigate(R.id.action_EditFragment_to_ListFragment)  
                true  
            }  
            else -> super.onOptionsItemSelected(item)  
        }  
    }  
    override fun onPrepareOptionsMenu(menu: Menu){  
        super.onPrepareOptionsMenu(menu)  
        val item = menu.findItem(R.id.action_new_place)  
        item.isVisible = false  
    }  
}
```

5. Proveriti da li navigacija u aplikaciji i dalje dobro funkcioniše. Nakon promena NavGraph treba da izgleda na sledeći način.



6. Dodati ikonu plus-a na *FloatingActionButton* kako bi bila adekvatna funkcionalnosti.

```
<include layout="@layout/content_main" />
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_marginEnd="16dp"
    android:layout_marginBottom="16dp"
    app:srcCompat="@android:drawable/ic_input_add" />
```



```
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

7. Ukloniti *Next* dugme sa *HomeFragment.kt*-a.

```
    else -> super.onOptionsItemSelected(item)
}

}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)

    binding.bottomSheet.setOnClickListener { it: View!
        findNavController().navigate(R.id.action_homeFragment_to_ListFragment)
    }
}

override fun onDestroyView() {
```

Android

The image shows a snippet of an Android XML layout file. It contains three views: a TextView, a Button, and an ImageView. The Button element is enclosed in a red rectangle with a large red 'X' drawn over it, indicating it is to be removed or is incorrect. The TextView has the text "My Places". The Button has the text "Next". The ImageView has the source "@drawable/elfak".

```
fragment.kt x fragment_home.xml x activity_main.xml x AndroidManifest.xml x nav_grap
}
<TextView
    android:id="@+id/textview_first"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="My Places"
    app:layout_constraintBottom_toTopOf="@id/elfak_image_view"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/button_first"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Next"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/app_logo_image_view" />

<ImageView
    android:id="@+id/elfak_image_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:src="@drawable/elfak"
    app:layout_constraintEnd_toEndOf="parent"
```

8.