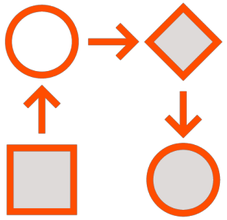
Summary

What are Architecture Components?

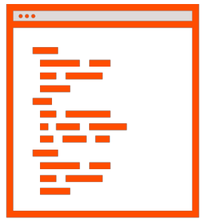
Navigation before the Navigation Component

Navigation Component overview

ViewModel



Works together with LiveData and Lifecycle

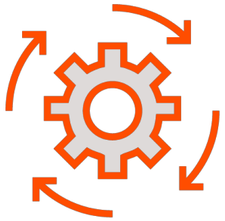


Stores all UI related logic and data

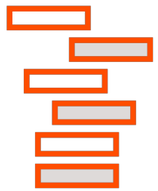
Capable of surviving configuration changes

DataBinding

Object that represents the Views in the UI



Bind data to Views, update automatically



Integrates with ViewModel

Navigation

Simplifies navigation in an Android application

Pass data safely between screens

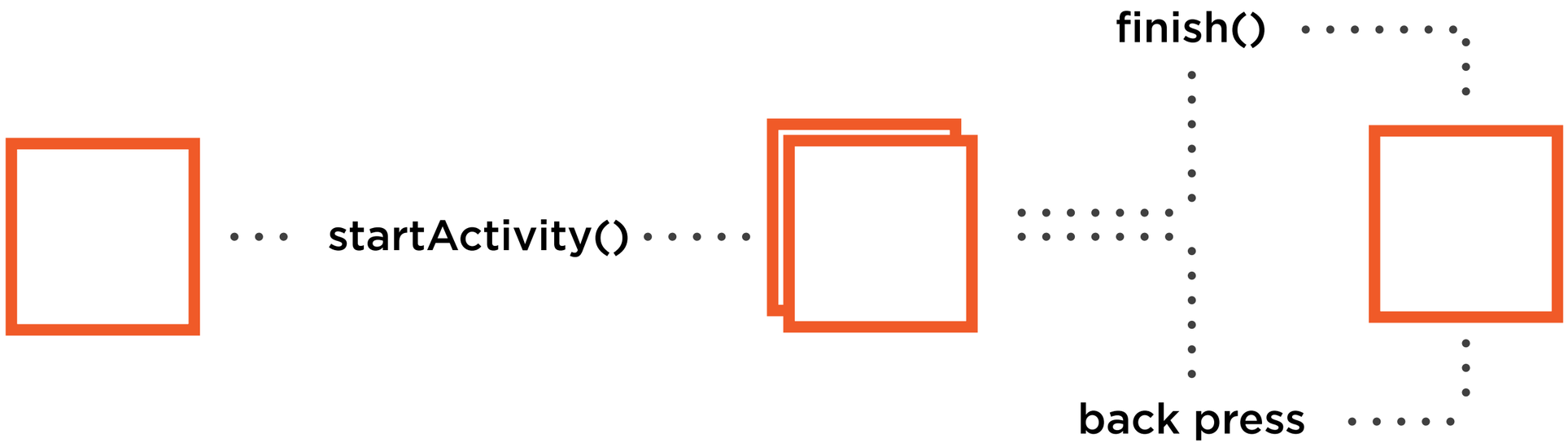
Provide deep links from outside of the app

Navigation before the Navigation Component

Navigation with Activities

```
Intent intent = new Intent(this, MyActivity.class);
intent.putExtra("key", value);
startActivity(intent);
```

Navigation with Activities



Navigation with Activities

Advantages

System manages the stack of Activities

Easy API

Flags to control the stack

Disadvantages

No clear idea of the current application stack

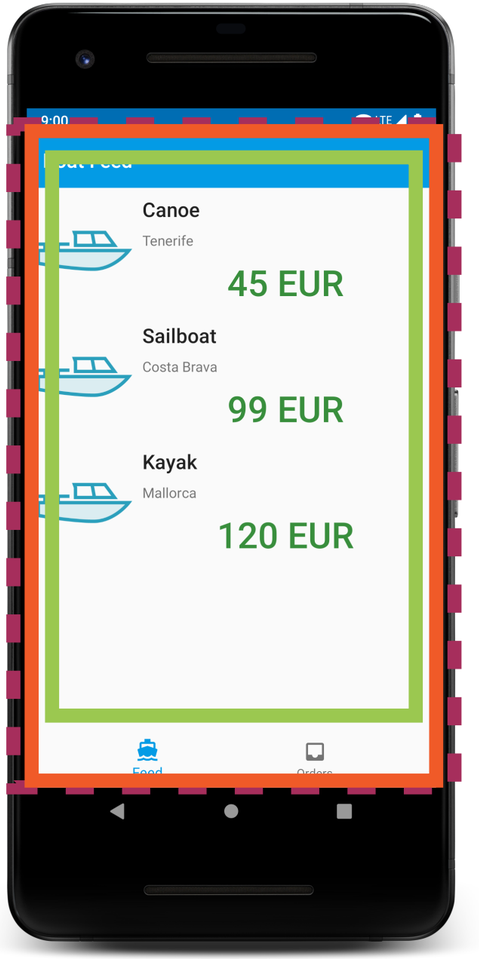
Each screen requires a new Activity

Each Activity needs to be declared in the AndroidManifest

No way to have rich navigation patterns

Activities contain the whole screen

Navigation with Fragments

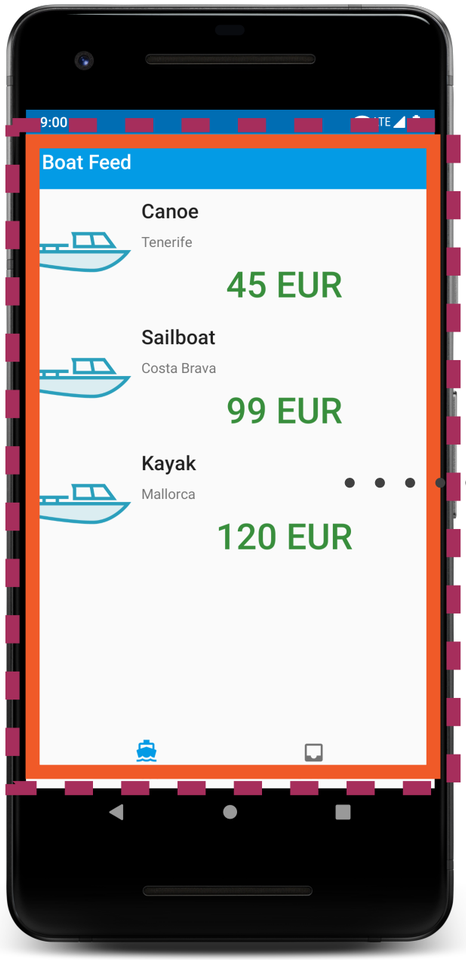


MainActivity

FeedFragment

HomeFragment

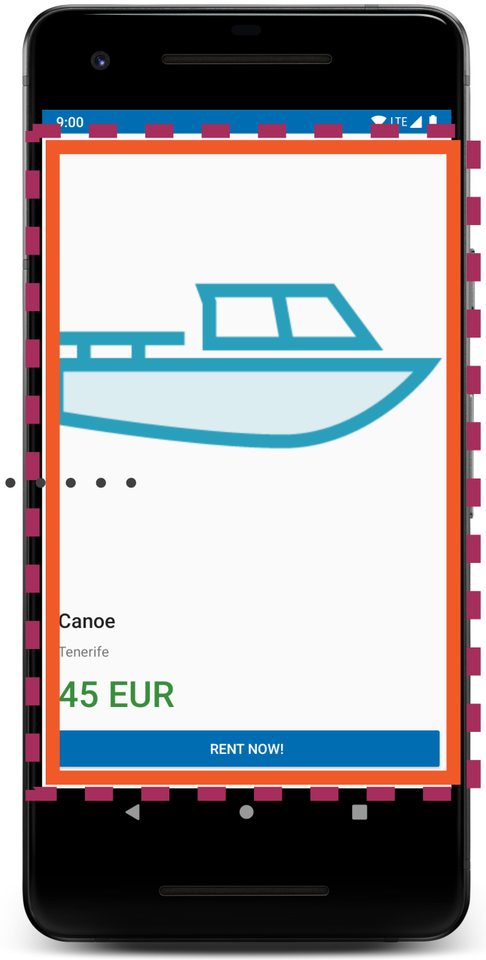
Navigation with Fragments



MainActivity

HomeFragment

Change Fragment



MainActivity

DetailFragment

Fragments

Used by the Navigation Component

- Manage automatically

Before the Navigation Component

- Manage manually
- Add, remove, hide, show, etc.
- FragmentManager

Navigation with Fragments

```
FragmentManager fm = getFragmentManager();
FragmentTransaction ft = fm.beginTransaction();
Fragment fragment = new MyFragment();
ft.add(R.id.fragment_container, fragment);
ft.commit();
```

Activities vs. Fragments

Multiple Activities

Single Activity - Multiple Fragments

Activities vs. Fragments

Multiple Activities

No control over the navigation stack

Poor navigation UX

Does not scale for large apps

Easy API

Single Activity - Multiple Fragments

Activities vs. Fragments

Multiple Activities

No control over the navigation stack

Poor navigation UX

Does not scale for large apps

Easy API

Single Activity - Multiple Fragments

Total control of the navigation stack

Rich navigation UX

Easily extendible and modularizable

Complex to manage

Activities vs. Fragments

Multiple Activities

No control over the navigation stack

Poor navigation UX

Does not scale for large apps

Easy API

Single Activity - Multiple Fragments

Total control of the navigation stack

Rich navigation UX

Easily extendible and modularizable

~~Complex to manage~~

Navigation Component can manage it!

Navigation with Architecture Components

```
findNavController().navigate(destination)

// FragmentManager fm = getFragmentManager();
// FragmentTransaction ft = fm.beginTransaction();
// Fragment fragment = new MyFragment();
// ft.add(R.id.fragment_container, fragment);
// ft.commit();
```

Conductor

Not by Google

Alternative to Fragments

Includes its own navigation solution

- Routers: Manage stack
- Controllers: Contain Views

Limited support

How the Navigation Component Can Help You

Single Activity App

Application that only uses a single Activity to display all screens. May use Fragments or other solutions to perform screen changes and navigation.

Creating new Activities is also OK when necessary

Navigation Component

Simplify implementation of navigation

Destination

- Any place you can navigate to
- Fragment
- Activity
- Custom

Navigation Graph

Groups destinations

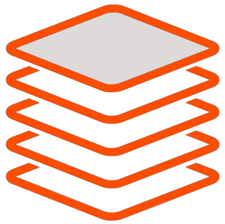
Actions between destinations

1+ graph per app

Transition animations

Passing data

Automatically Managed



Navigation stack fully managed by the Navigation Component library

Restores the navigation stack when the app is destroyed by the system, including any passed parameters

Provide deep links from outside of the app

From Push Notifications, Widgets on the screen, or web links

Summary

What are Architecture Components?

Navigation before the Navigation Component

Navigation Component overview

Summary

Setup

- Android Studio
- Libraries

New Android project

- First navigation graph
- First two screens
- First navigation action

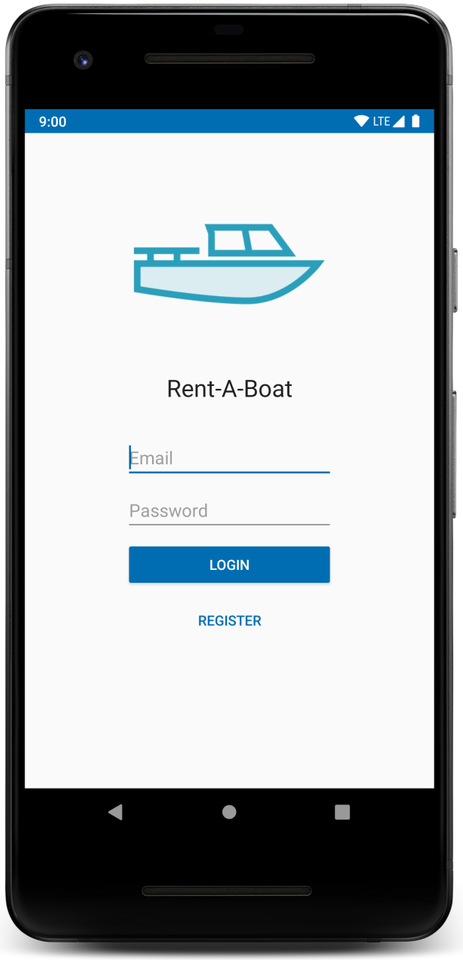
Navigation graph edit tool

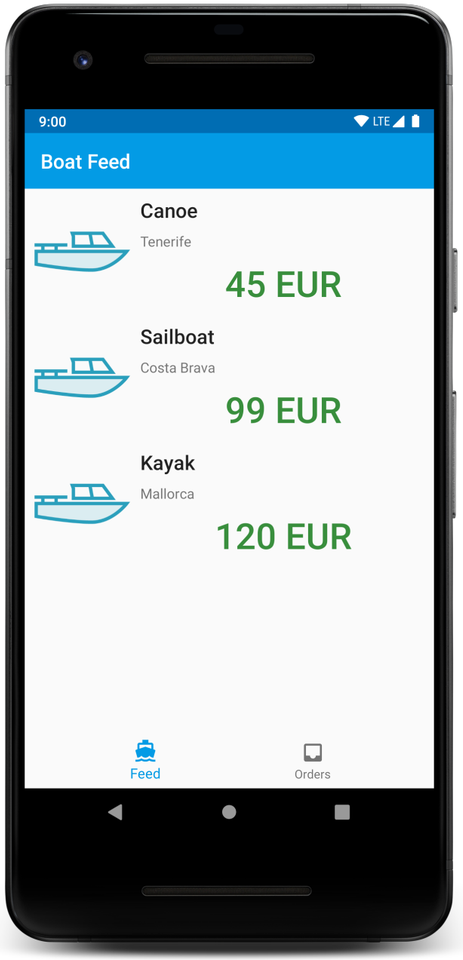
```
implementation "android.arch.navigation:navigation-fragment-ktx:1.0.0-alpha06"

implementation "android.arch.navigation:navigation-ui-ktx:1.0.0-alpha06"
```

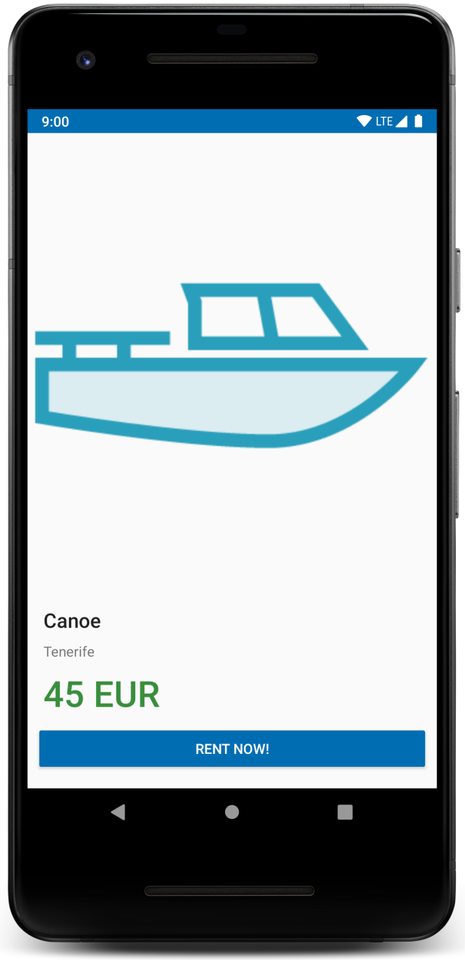
Navigation Component Library

Add the Navigation Component from the Android Architecture Components

Demo Project: Rent-A-Boat







Demo

Enable the navigation editor

Add a navigation graph

Add the first screen

- Setup navigation host

Run the project

- Verify that the screen is displayed

```
<fragment android:id=
  "@+id/nav_container"

  android:name=
    “….NavHostFragment”

  app:navGraph=
    “@navigation/nav_graph"

  app:defaultNavHost=
    "true">
```

◀ Implements graph area

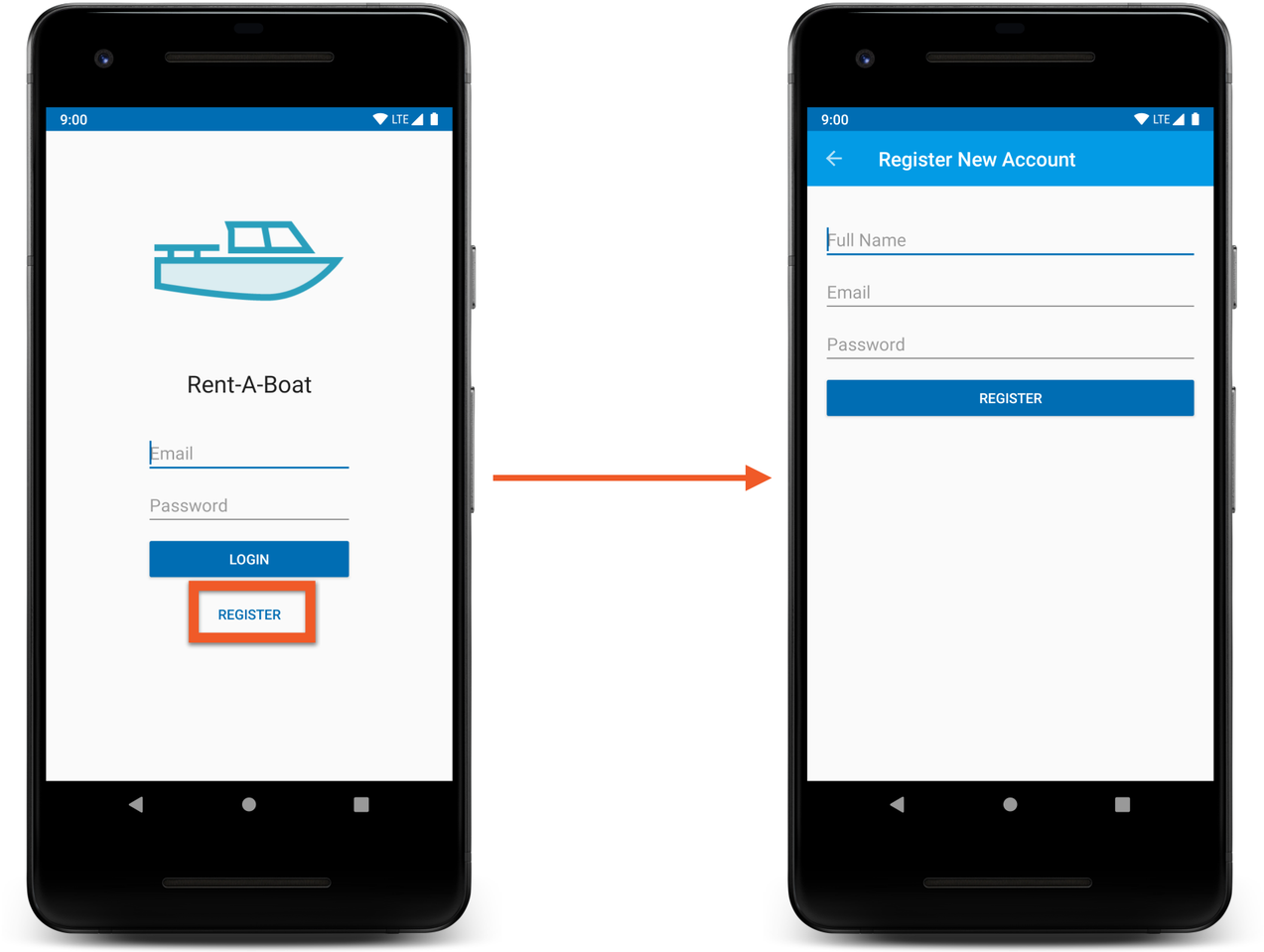
```
<fragment android:id=
  "@+id/nav_container"

  android:name=
    “….NavHostFragment”

  app:navGraph=
    “@navigation/nav_graph"

  app:defaultNavHost=
    "true">
```

- Implements graph area
- Graph to host

```
<fragment android:id=
  "@+id/nav_container"

  android:name=
    “….NavHostFragment”

  app:navGraph=
    “@navigation/nav_graph"

  app:defaultNavHost=
    "true">
```

- Implements graph area
- Graph to host
- Handle backpress

Navigation between Two Screens



Demo

Add a second screen

Add navigation action

- From one screen to another

Run the project

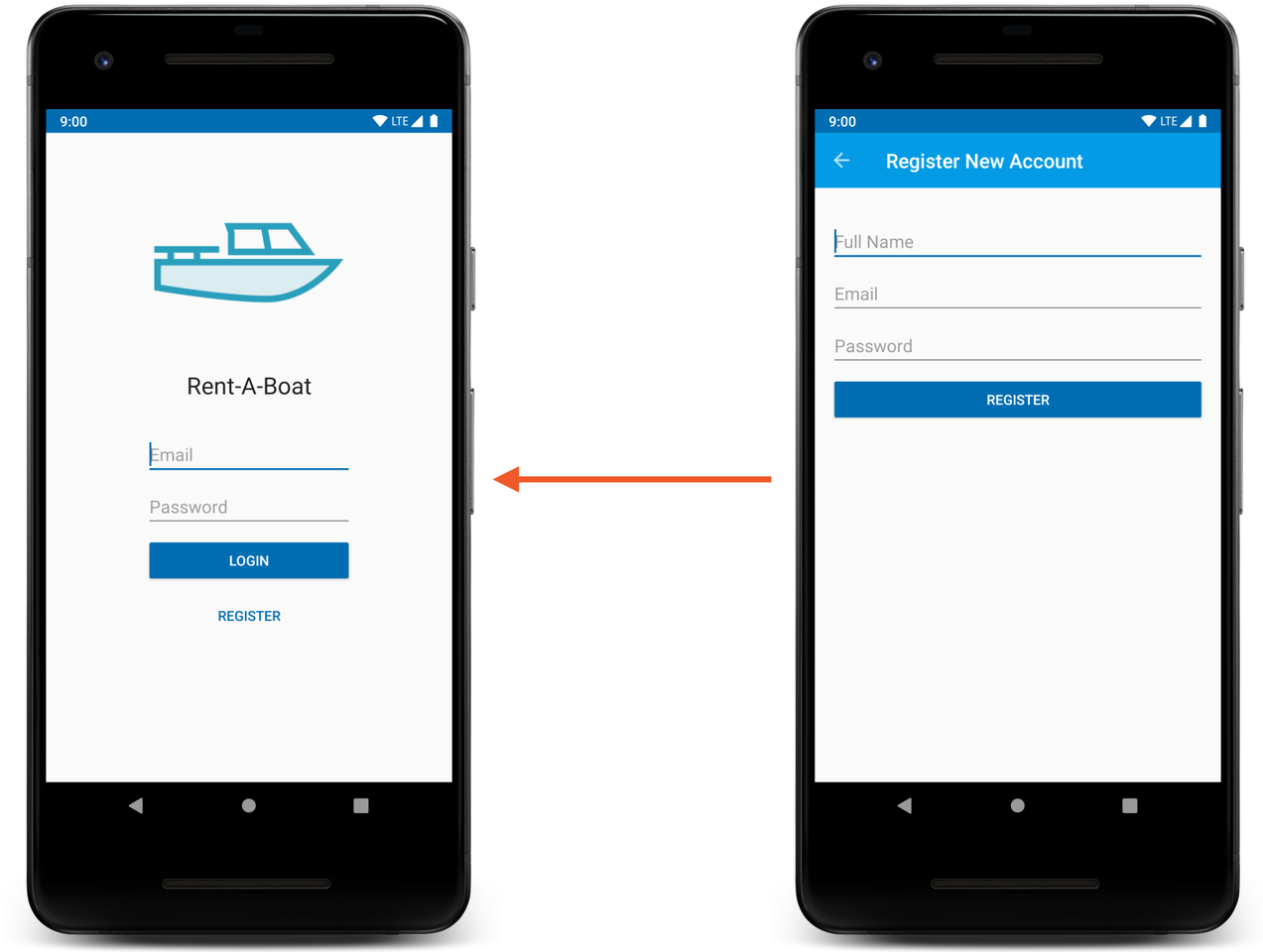
- Verify that navigation works

```
button.setOnClickListener(
  Navigation.createNavigateOnClickListener(action)
)
```

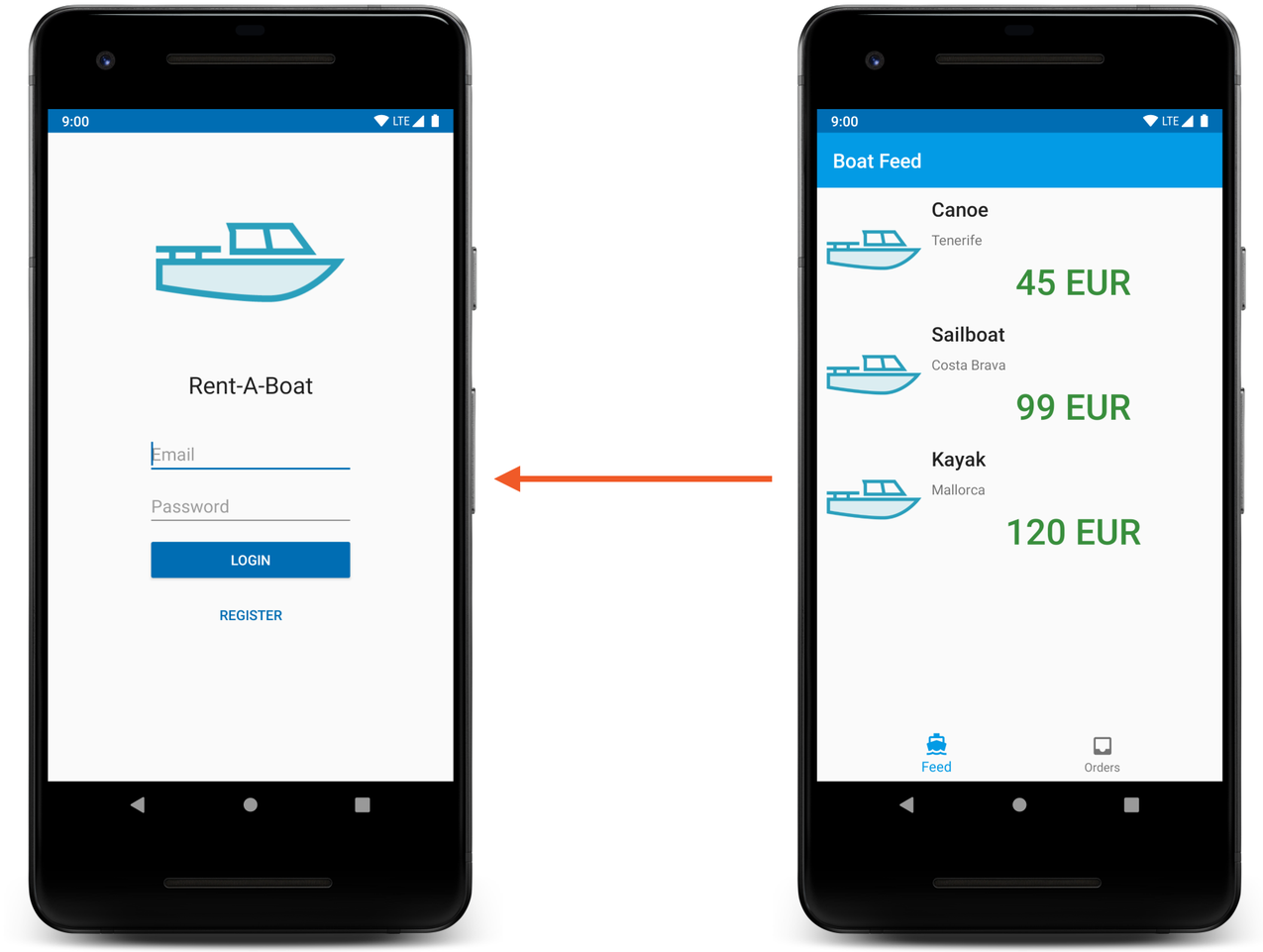
Click Listener Methods

Generate the onClickListener methods that will navigate to a destination

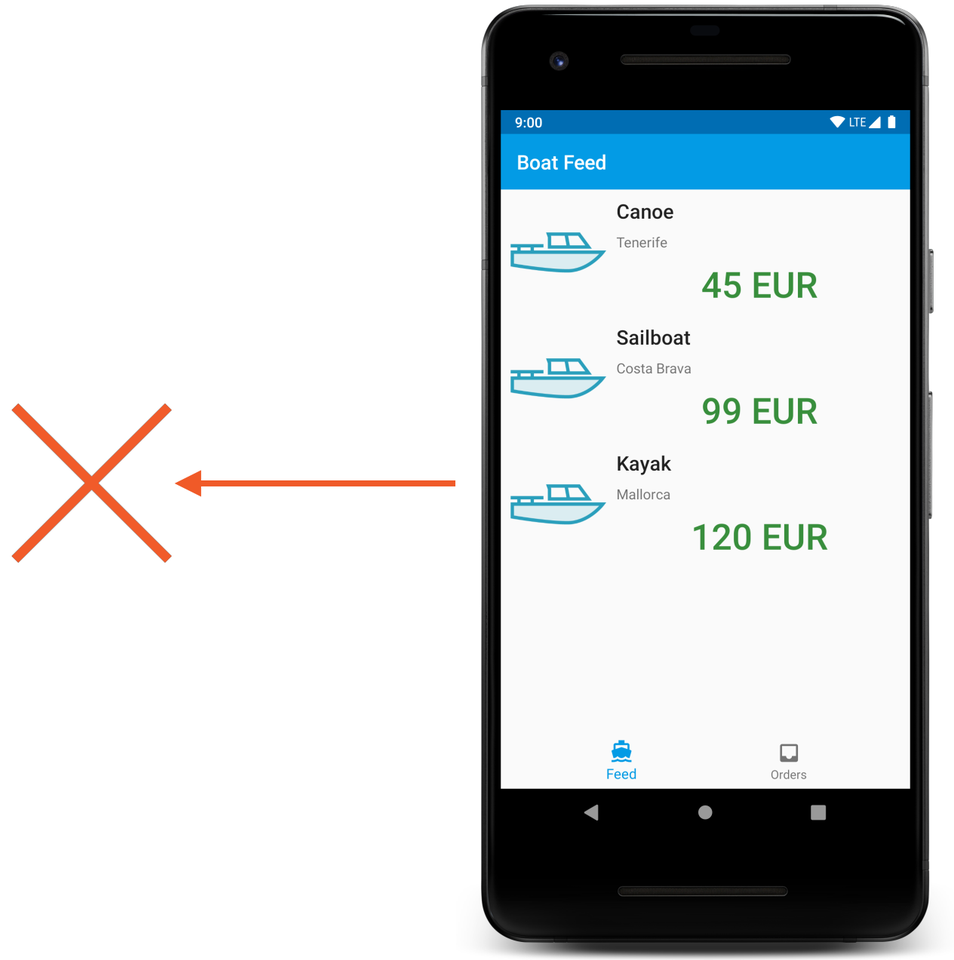
Backpress Handling



Pop Behavior



Pop Behavior: Inclusive is True



SingleTop

At most one copy of a given destination on the top of the back stack

Similar to how FLAG_ACTIVITY_SINGLE_TOP works with Activites

Summary

Configure the project

- Android Studio 3.2
- Added Navigation Component

Created Navigation Graph

Created two screens

- Implemented navigation between them

Look at the options in navigation editor