

CMPUT 301

Lab 7: Navigation and UI Testing

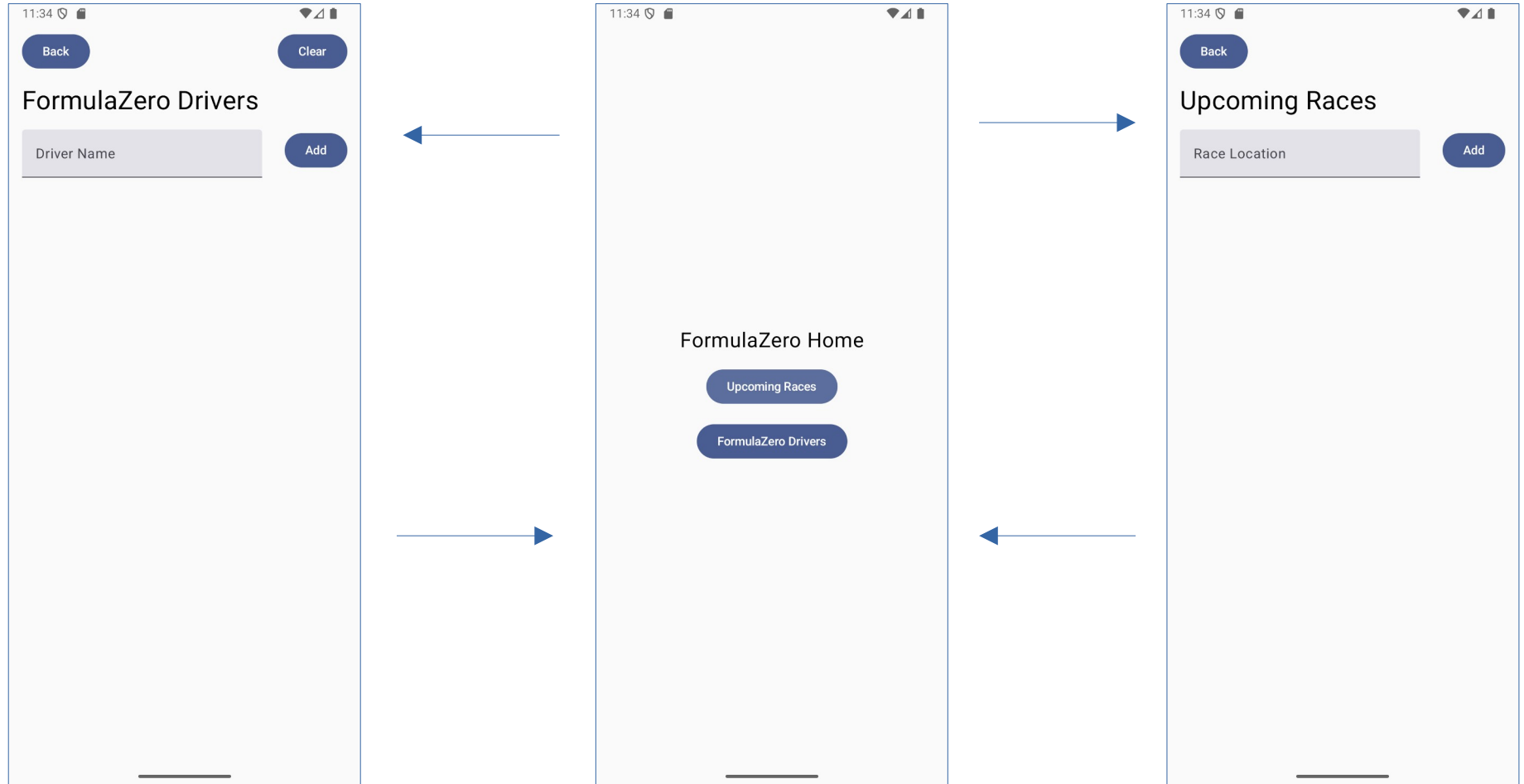
Table of Contents

- Navigation
 - Example
 - Some Concepts
- UI Testing
 - Compose UI Testing
 - UI Testing Versus Unit Testing
- More Information

Navigation

- Navigation in Jetpack Compose lets us move between different composable screens

Navigation Example



Some Navigation Concepts

- A **NavController** manages navigation actions like moving forward, back, and passing data
- A **NavHost** is the container that decides which composable screen is currently displayed

UI Testing

- *Instrumented tests* run tests on a physical or emulated device
- *Most instrumented tests are UI tests*, where the real app UI is tested on a real or emulated device automatically

Compose UI Testing

- Jetpack Compose has its own testing APIs we can use to find elements, verify attributes, and perform actions

UI Testing Versus Unit Testing

- UI testing is similar to the unit testing we did in lab 6
 - Instead of manually performing actions to verify app behaviour, we get the Compose UI testing framework to do it automatically for us
 - But UI testing is often slower than unit testing as we run the UI tests on a physical or emulated device

More Information

- Navigation Overview: <https://developer.android.com/guide/navigation>
- Navigation with Compose: <https://developer.android.com/develop/ui/compose/navigation>
- Compose UI Testing: <https://developer.android.com/develop/ui/compose/testing>
- Compose UI Testing Cheatsheet: <https://developer.android.com/develop/ui/compose/testing/testing-cheatsheet>