droidcon academy

Accessibility in Android Jetpack Compose: Grouping related content

Quintin Balsdon

in In/qbalsdon

Section Overview

- Headings
- Checkboxes and radio groups
- Cards and actions
- Snackbars and toasts

droidcon academy

Grouping content

- Grouping is not purely visual
- What would be easier for our users:
  - Separate the list with headings
  - Tapping the entire card to mark an element complete

droidcon academy

```
modifier = Modifier
      .semantics { heading() },
```

Headings

- Separation of content for users
- Provide navigational anchors

```
modifier = Modifier
    .testTag("Heading")

composeTestRule.onNodeWithText("To do list:")
    .assertIsHeading()

composeTestRule.onAllNodesWithTag("Heading")
    .assertCountEquals(2)
    .assertAll(isHeading())
```

Espresso Testing: Headings

- Built in semantics matcher
- Add tags to find groups of elements
- extensions.kt has a shorthand assertion defined

droidcon academy

```
modifier
    ...
    .toggleable(
        role = Role.Checkbox,
        value = toDoItem.complete,
        onValueChange = {
            updateItem(
                toDoItem.copy(
                    complete = !toDoItem.complete
                ), EditMode.IN_PLACE
            )
        }
    )
```

Toggleable Components

- Give it the appropriate role (Switch could also work)

droidcon academy

```
onNodeWithTag(
    testTag = "OutlinedCard_ToDoListItem",
    useUnmergedTree = true
).apply {
    assertIsToggleable()
    assertIsOff()
}
```

```
onNodeWithTag(
    testTag =
"OutlinedCard_ToDoListItem_Checkbox",
    useUnmergedTree = true
).performClick()

// assert the element state on
onNodeWithTag(
    testTag = "OutlinedCard_ToDoListItem",
    useUnmergedTree = true
).assertIsOn()
```

Espresso Testing: Toggleable

- Built in assertions for state and toggleable

droidcon academy

Radio groups

- Only one element selected at a time
  - need the `selectableGroup()` modifier
- The row becomes selectable, as opposed to the individual **RadioButton** or text

app/source/main/java/com/droidcon/alldone/ui/component/RadioButtonGroup.kt

Actions

- Currently, a screen reader or switch user will take much longer to navigate through the list of items
- Actions help us hide secondary actions until a user is wants to use them

droidcon academy

```
modifier
    ...
    .semantics {
        customActions = listOf(
            CustomAccessibilityAction(editDescription) {
                editAction()
                true
            },
            CustomAccessibilityAction(shareDescription) {
                shareAction()
                true
            },
            CustomAccessibilityAction(addToCalendarDescription) {
                addToCalendarAction()
                true
            }
        )
    }
```

Actions

- Allow events to be attached to containers
- Are applied to a hidden menu available to Switch and TalkBack users
- Can hide the row of secondary actions entirely

droidcon academy

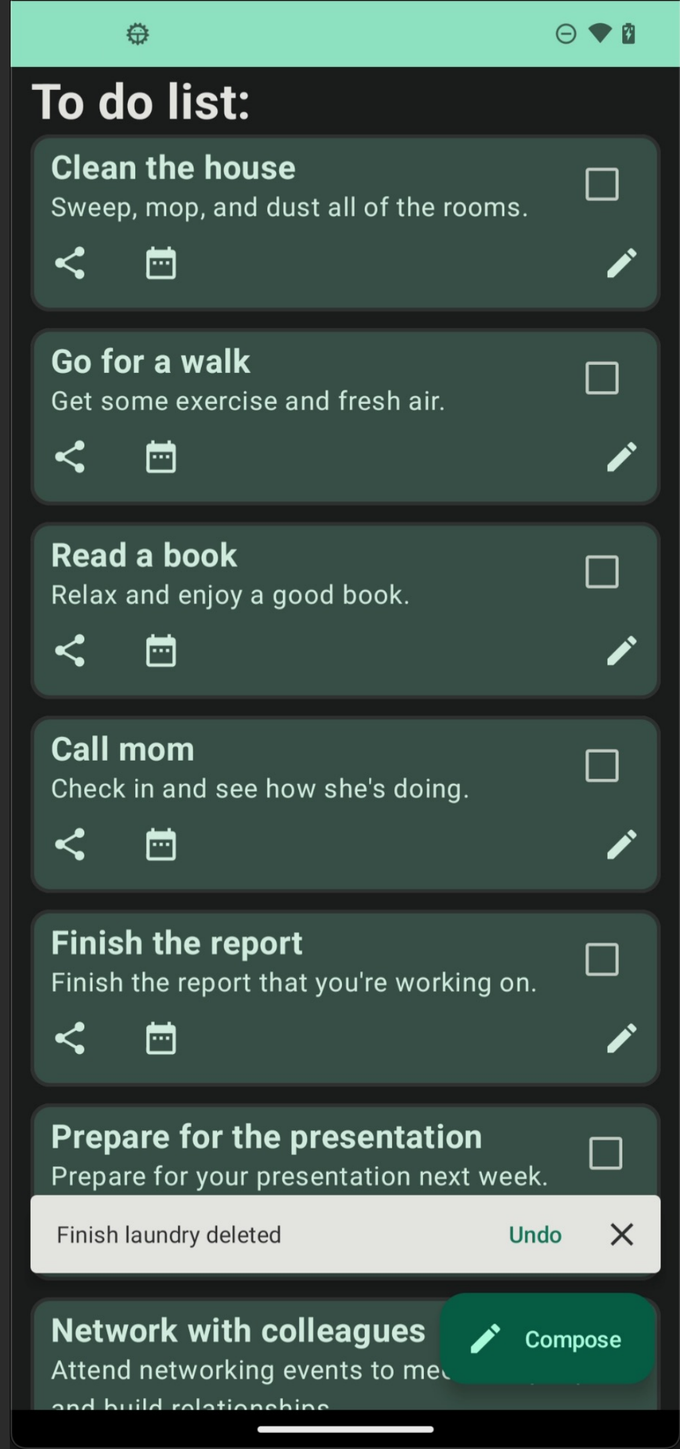
```
fun SemanticsNodeInteraction.assertHasAccessibilityAction(label: String):
SemanticsNodeInteraction {
    return assert(
        SemanticsMatcher.actionIsDefined(label)
    )
}


fun SemanticsMatcher.Companion.actionIsDefined(label: String): SemanticsMatcher {
    return SemanticsMatcher("$label is defined in CustomActions") {
        label in it
            .config[SemanticsActions.CustomActions]
            .map { action -> action.label }
    }
}
```

Espresso Testing: Actions

- app/src/androidTest/java/com/droidcon/alldone/extensions.kt
- Navigate the specific Semantics Properties

Snackbars and Toast Messages

- Use sparingly, and consider users who use the magnifier
  - Use location grouping
- Make users feel like they are in control
- Remember to be fault tolerant!



Section Summary

- Grouping content
  - Visually with headings
  - Checkboxes and radio buttons
  - Cards and actions
- Snackbars and toasts

droidcon academy

Validation, Announcements and Animations

Up Next

droidcon academy