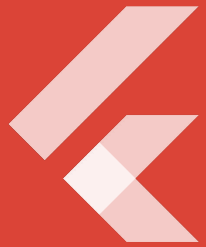
institute of CODING

# Mobile Application Development



## Flutter Navigation Basics

Northumbria University NEWCASTLE

Office for Students OfS

# So far,

We have learnt how to create stunning screens easily using the Flutter’s built-in widgets. What we don’t have is the functionality of navigating to different screens. We had some buttons, yet these do nothing when we pressed. It’s time to make them work!

# Tidy up!

Create a new file in the `lib` directory named `conversations.dart` and put the chat screen we built previously into a class named `ChatScreen`.

## conversations.dart

```dart
import 'package:flutter/material.dart';

class Conversations extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButton: ...,
      appBar: ...,
      body: ...
    );
  }
}
```

# Tidy up!

`main.dart` file will be much shorter now!

# main.dart

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(primarySwatch: Colors.purple),
      home: Conversations(),
    );
  }
}
```

# Revise

We can revise the same functionality of `main.dart` by specifying routes (navigational structure or screens) of our app!

# main.dart

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(primarySwatch: Colors.purple),
      initialRoute: "/",
      routes: {
        "/": (BuildContext context) {
          return Conversations();
        },
      },
    );
  }
}
```

Entry point of the app or route of the homescreen

A function returning the `Conversation` widget.

# Quick note!

If a function just contains a return statement, we can write it in a much shorter way:

```
(BuildContext context) {
  return Conversations();
},

(context) => Conversations(),
```

# contacts.dart

Create a `Contacts` class for selecting a person from the contact list.

# contacts.dart

```
import 'package:flutter/material.dart';

class Contacts extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.teal[700],
        title: Text("Choose a contact"),
      ), body: …
    );
  }
}
```

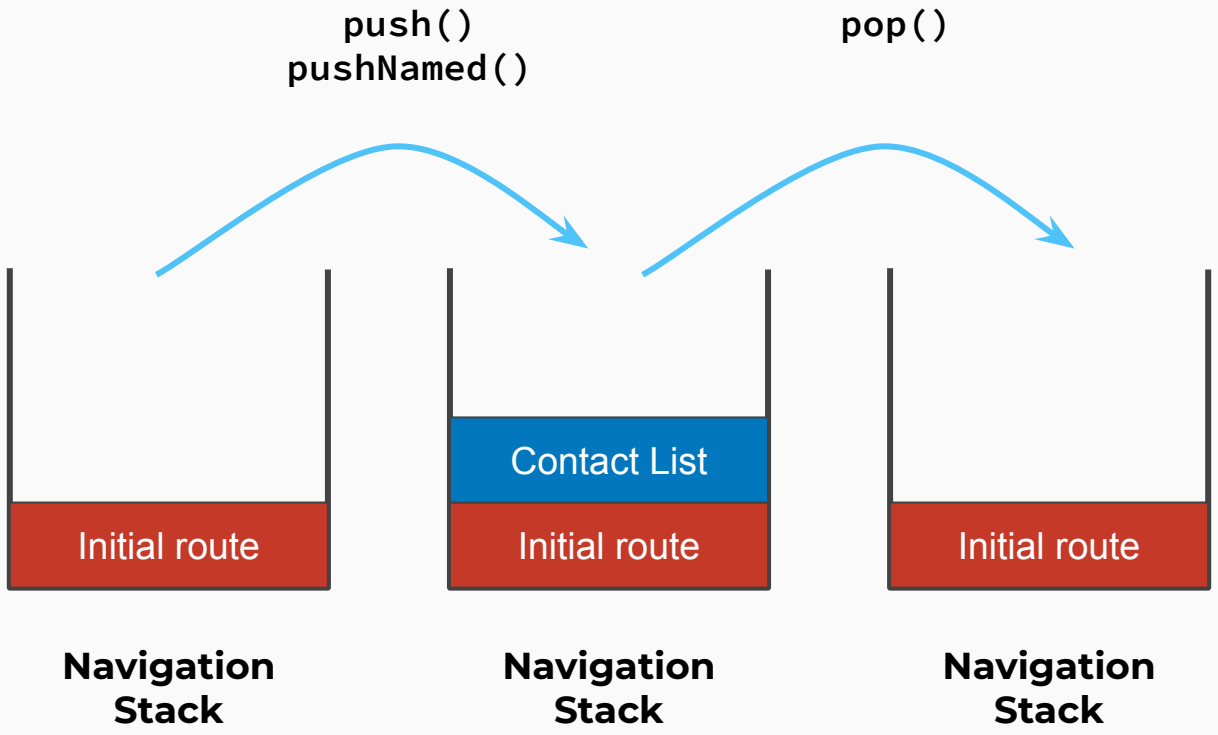
# contacts.dart *(Scaffold body)*

```
body: ListView(
  children: <Widget>[
    ListTile(
        leading: CircleAvatar(
          backgroundImage: NetworkImage("https://i.pravatar.cc/300?u=16"),
        ),
        title: Text("Chester Gatewood")),
    ListTile(
        leading: CircleAvatar(
          backgroundImage: NetworkImage("https://i.pravatar.cc/300?u=17"),
        ),
        title: Text("Daniel Brown"))
  ],
)
```

# Navigate to contacts list

Now that we have everything ready, two different screens, one for creating a new chat from contacts and a main screen. All we need to do is navigating from main screen to contacts screen when pressing the floating action button.

# Flutter Navigation Stack



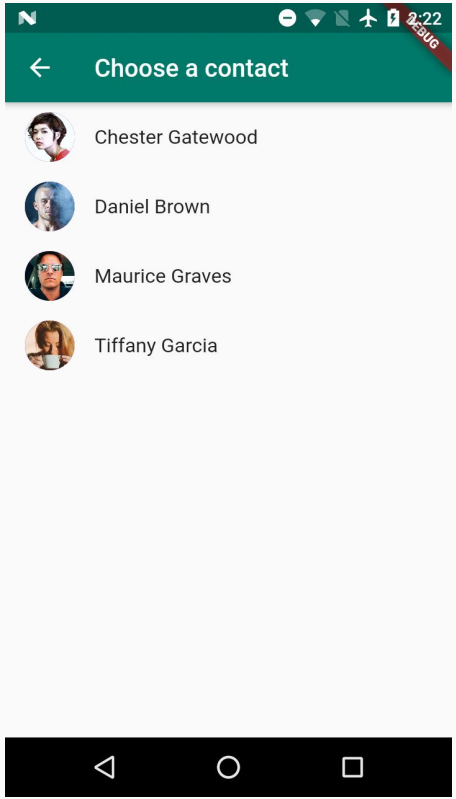
# conversations.dart

To navigate `/new-chat`, we will use `Navigator` and push that screen to navigation stack.

```
floatingActionButton: FloatingActionButton(
  child: Icon(Icons.message),
  backgroundColor: Colors.teal,
  onPressed: () {
    Navigator.pushNamed(context, "/new-chat");
  },
)
```

# Contact list screen

When pressed the floating action button, we will navigate to contacts. Notice that, even if we didn't change anything, AppBar shows a back button to navigate back to main screen.



# Navigator.push

An alternative way to navigate a screen, especially screens which do not defined in the routes, using the push method and providing a MaterialPageRoute.

```
floatingActionButton: FloatingActionButton(
  child: Icon(Icons.message),
  backgroundColor: Colors.teal,
  onPressed: () {
    Navigator.push(context, MaterialPageRoute(builder: (context) => Contacts()));
  },
)
```

# conversations.dart

In order to go back to the main screen without using the back button, we can pop the navigator. Let's pop when we tap the `ListTile`. `ListTile` has a `onTap` parameter which we can specify the behaviour when user taps.

```
ListTile(
    leading: CircleAvatar(
      backgroundImage: NetworkImage("https://i.pravatar.cc/300?u=16"),
    ),
    onTap: () {
      Navigator.pop(context);
    },
    title: Text("Chester Gatewood"))
```

# Tip!

**Remember the arrow notation. Following statements are equal:**

```
MaterialPageRoute(builder: (context) => Contacts())


MaterialPageRoute(builder: (context) {
  return Contacts();
})
```

# Tip!

Instead of creating widget classes for each screen, we can create small, composable widgets. For example, we can wrap the `FloatingActionButton` as a new `StatelessWidget`, `NewMessageButton`.

# conversations.dart

```
class NewMessageButton extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return FloatingActionButton(
      child: Icon(Icons.message),
      backgroundColor: Colors.teal,
      onPressed: () {
        Navigator.pushNamed(context, "/new-chat");
      },
    );
  }
}
```

# conversations.dart

Then scaffold of conversations.dart became:

```
return Scaffold(
  floatingActionButton: NewMessageButton(),
…
```

# Passing and returning variables between screens

We selected a contact and returned to the main screen and nothing has changed in the main screen. We just navigate to another screen and returned back.

Sometimes we need to pass a result or a variable from one screen to another. In our application, that variable should be which contact is selected from the contacts list.

# Returning a variable

**To return the selected contact:**

```
onTap: () {
  Navigator.pop(context);
},
```

**becomes**

```
onTap: () {
  Navigator.pop(context, "Chester Gatewood");
},
```

Returning name

**To return the selected contact:**

```
onTap: () {
  Navigator.pop(context);
},
```

**becomes**

```
onTap: () {
  Navigator.pop(context, "Chester Gatewood");
},
```

Returning name

# conversations.dart

## Getting the returned contact name:

```
return FloatingActionButton(
  child: Icon(Icons.message),
  backgroundColor: Colors.teal,
  onPressed: () async {
    var contact = await Navigator.pushNamed(context, "/new-chat");
    Scaffold.of(context).showSnackBar(SnackBar(content: Text("$contact selected")));
  },
);
```

# conversations.dart

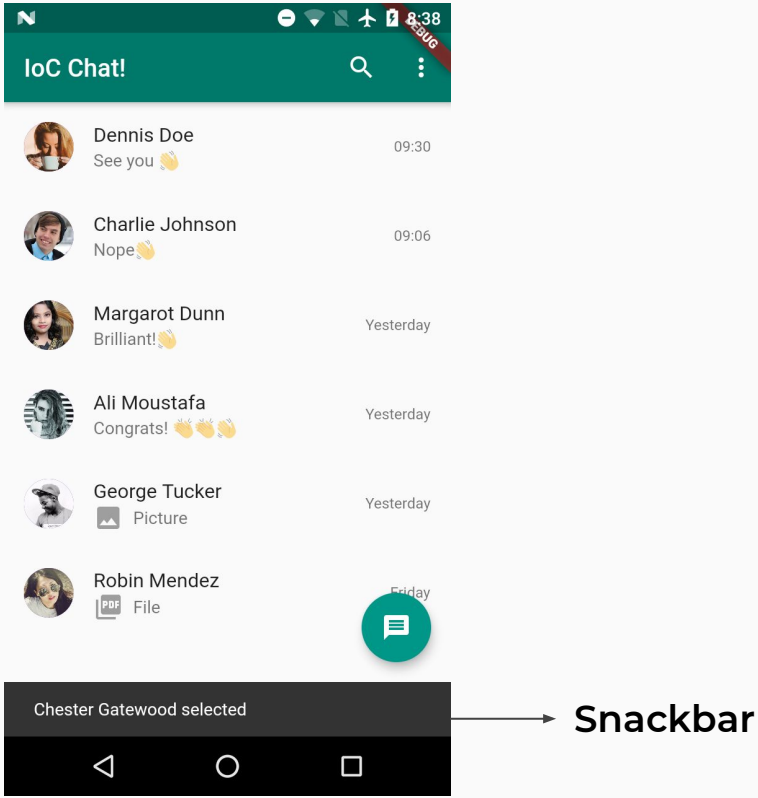
## Getting the returned contact name:

```
return FloatingActionButton(
  child: Icon(Icons.message),
  backgroundColor: Colors.teal,
  onPressed: () async {
    var contact = await Navigator.pushNamed(context, "/new-chat");
    Scaffold.of(context).showSnackBar(SnackBar(content: Text("$contact selected")));
  },
);
```

**Returning name**

**Showing an indicator at the bottom of the screen**

# Preview


8:38
DEBUG
IoC Chat!
Dennis Doe
See you 👋
09:30
Charlie Johnson
Nope👋
09:06
Margarot Dunn
Brilliant!👋
Yesterday
Ali Moustafa
Congrats! 👏👏👏
Yesterday
George Tucker
Picture
Yesterday
Robin Mendez
File
Friday
Chester Gatewood selected
Snackbar

# Passing a variable

# Passing a variable

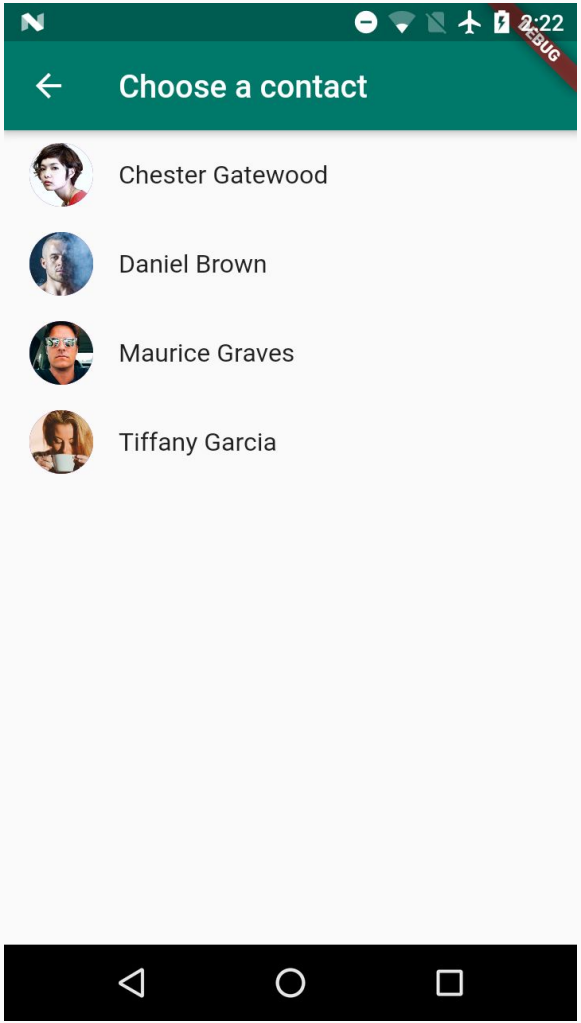
Using the `arguments` keyword, we can pass information to new screen. Let's pass the title for the new screen.

```
Navigator.pushNamed(
  context,
  "/new-chat",
  arguments: "Choose a contact",
);
```

# Getting the passed variable

**We can get the passed information from the new screen.**

```
Widget build(BuildContext context) {
  var title = ModalRoute.of(context).settings.arguments;
  return Scaffold(
      appBar: AppBar(
        title: Text(title),
...
```




Topics covered:

- Navigating to and from screens
- Creating and composing widgets
- Passing and returning variables between pages
- Showing SnackBar, FloatingActionButton and trigger functions.