Flutter

Training Assignments

Using navigation and routing

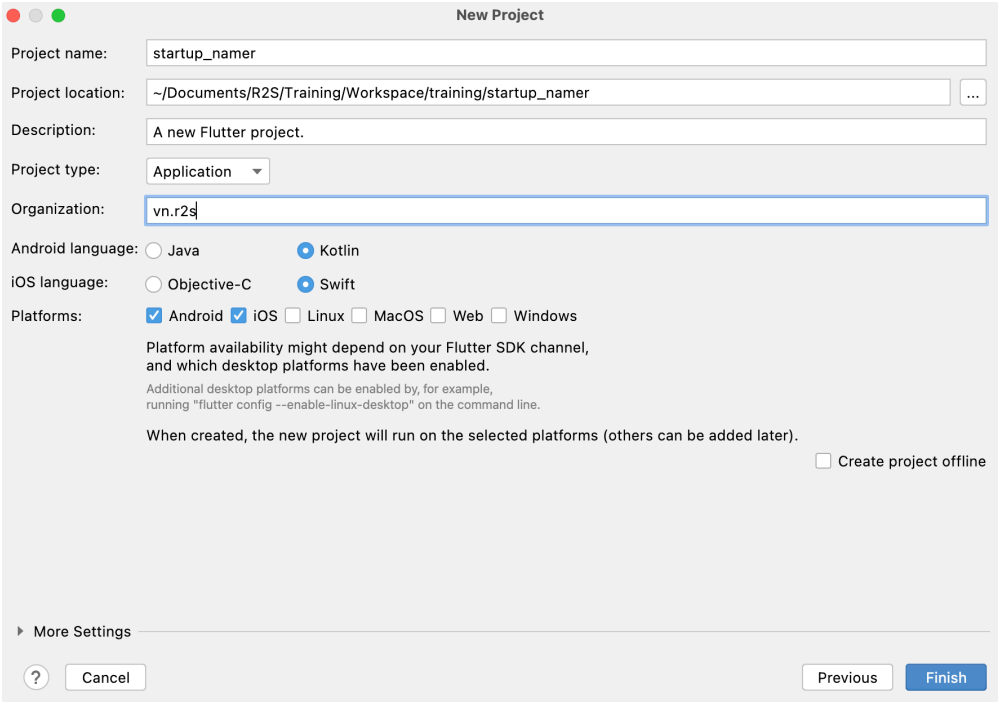
Overview

In this practice, you'll extend a basic, mobile Flutter app to include interactivity. You'll also create a second page (called a route) that the user can navigate to.

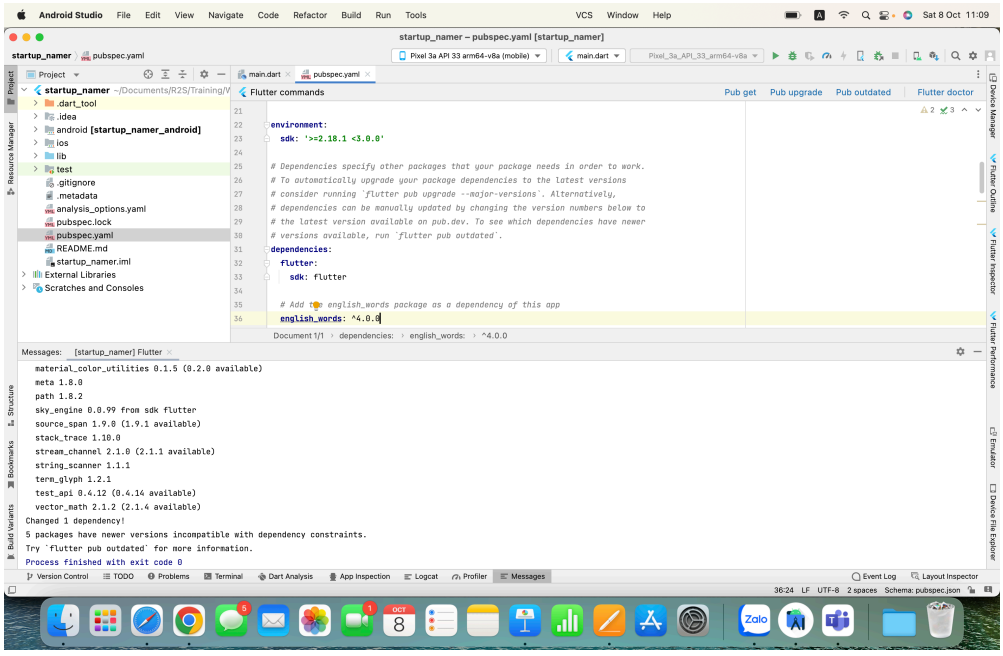
Tasks

1. Get the starting app

+ Open the Android Studio and create a Flutter project called startup_namer



+ Add the **english_words** package as a dependency of this app: **english_words: ^4.0.0**



2. Write code

+ Delete all of the code from **lib/main.dart**. Replace it with the follow code

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),  // AppBar
        body: const Center(
          child: RandomWords(),
        ),  // Center
      ),  // Scaffold
    );  // MaterialApp
  }
}

class RandomWords extends StatefulWidget {
  const RandomWords({Key? key}) : super(key: key);

  @override
  State<RandomWords> createState() => _RandomWordsState();
}
class _RandomWordsState extends State<RandomWords> {
  final _suggestions = <WordPair>[]; // List
  final _biggerFont = const TextStyle(fontSize: 18);
  final _saved = <WordPair>{}; // Set

  @override
  Widget build(BuildContext context) {
    _suggestions.addAll(generateWordPairs().take(50));

    return Scaffold(
      appBar: AppBar(
        title: const Text('Startup Name Generator'),
        actions: [
          IconButton(
            onPressed: _pushSaved,  // Navigate to a new screen
            icon: const Icon(Icons.list),
            tooltip: 'Saved Suggestions',
          )  // IconButton
        ],
      ),  // AppBar
```

```
    body: ListView.separated(
      itemCount: _suggestions.length,
      itemBuilder: (context, index) {

        final alreadySaved = _saved.contains(_suggestions[index]); // NEW

      return ListTile(
        title: Text(
          _suggestions[index].asPascalCase,
          style: _biggerFont,
        ),  // Text
        trailing: Icon( // Add icons to the list
          alreadySaved ? Icons.favorite : Icons.favorite_border,
          color: alreadySaved ? Colors.red : null,
          semanticLabel: alreadySaved ? 'Remove from saved' : 'Save',
        ),  // Icon
        onTap: () { // Add interactivity
          setState(() {
            if (alreadySaved) {
              _saved.remove(_suggestions[index]);
            } else {
              _saved.add(_suggestions[index]);
            }
          });
        },
      );  // ListTile
    },
    separatorBuilder: (BuildContext context, int index) {
      return const Divider();
    },
  ),  // ListView.separated
);  // Scaffold
}
```

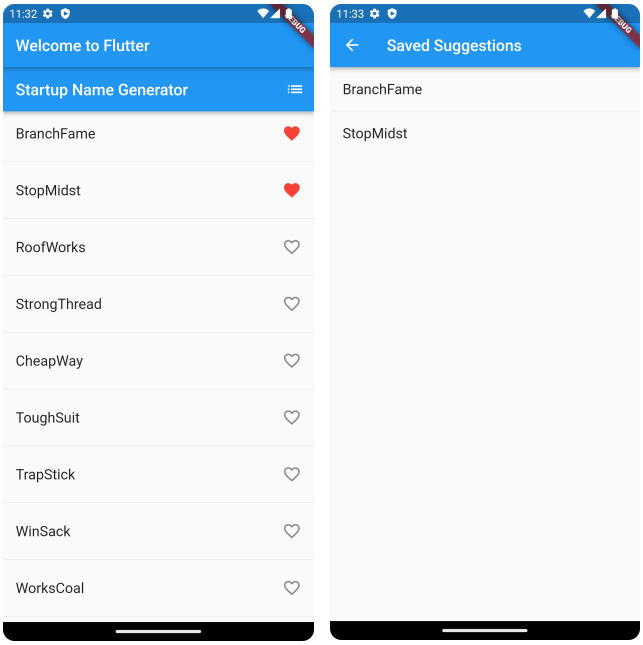
+ Navigate to a new screen

```
void _pushSaved() {
  Navigator.of(context).push(MaterialPageRoute(builder: (context) {
    final tiles = _saved.map((e) {
      return ListTile(
        title: Text(
          e.asPascalCase,
          style: _biggerFont,
        ));  // Text, ListTile
    });

    final divided = tiles.isNotEmpty
        ? ListTile.divideTiles(tiles: tiles, context: context).toList()
        : <Widget>[];

    return Scaffold(
      appBar: AppBar(
        title: const Text('Saved Suggestions'),
      ),  // AppBar
      body: ListView(
        children: divided,
      ),  // ListView
    );  // Scaffold
  }));  // MaterialPageRoute
}
```

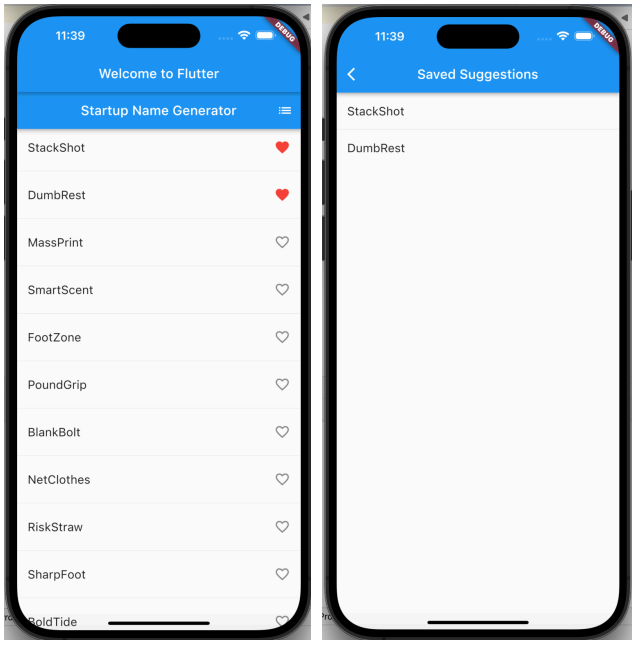
+ Final code

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

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {...}
}

class RandomWords extends StatefulWidget {
  const RandomWords({Key? key}) : super(key: key);

  @override
  State<RandomWords> createState() => _RandomWordsState();
}

class _RandomWordsState extends State<RandomWords> {
  final _suggestions = <WordPair>[]; // List
  final _biggerFont = const TextStyle(fontSize: 18);
  final _saved = <WordPair>{}; // Set

  @override
  Widget build(BuildContext context) {...}

  // Navigate to a new screen
  void _pushSaved() {...}
}
```

3. Run the app



Android emulator



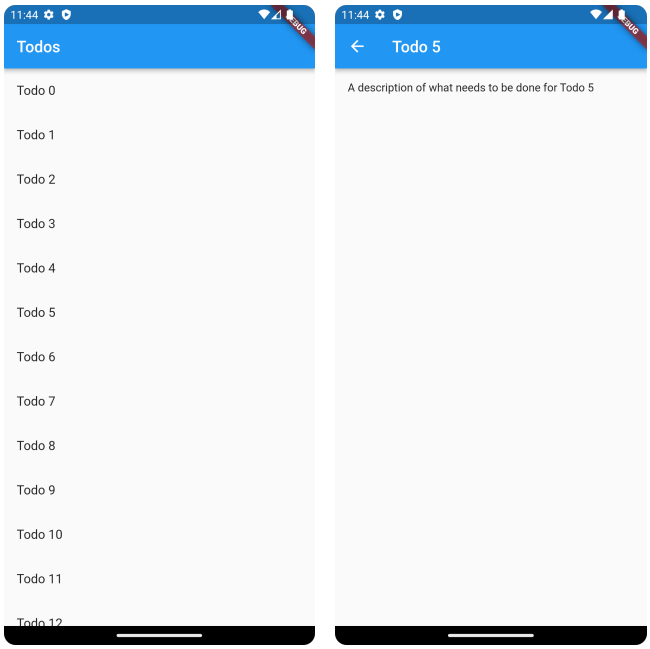
iOS simulator

Extra tasks

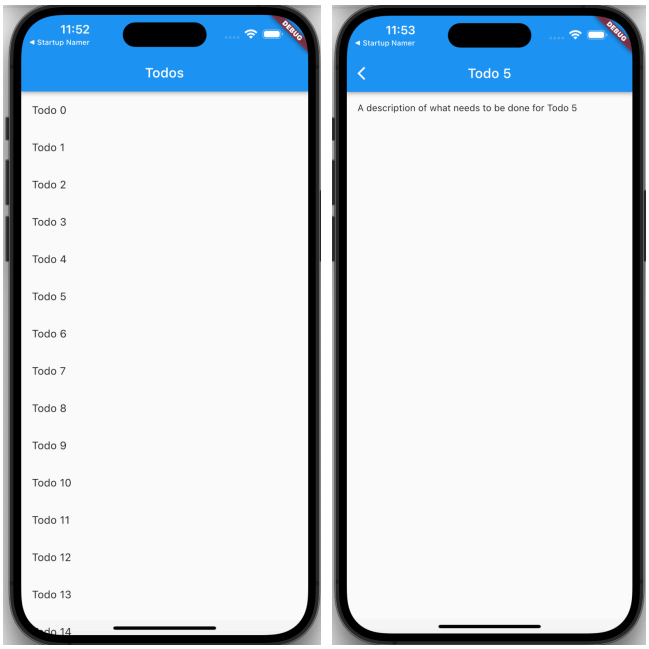
+ In this tasks, create a list of todos. When a todo is tapped, navigate to a new screen (widget) that displays information about the todo. This recipe uses the following steps:

1. Define a todo class.
2. Display a list of todos.
3. Create a detail screen that can display information about a todo.
4. Navigate and pass data to the detail screen.

+ Run the app



Android emulator



iOS simulator

--THE END—