Foodies

Always know where to go eat!

Arman Berek, Mary Fan, Jeffrey Lin
CSCI 499
2020-05-18

Product Definition

- Where to go when grabbing a meal?
- Save time, money, and compromise in friend groups.

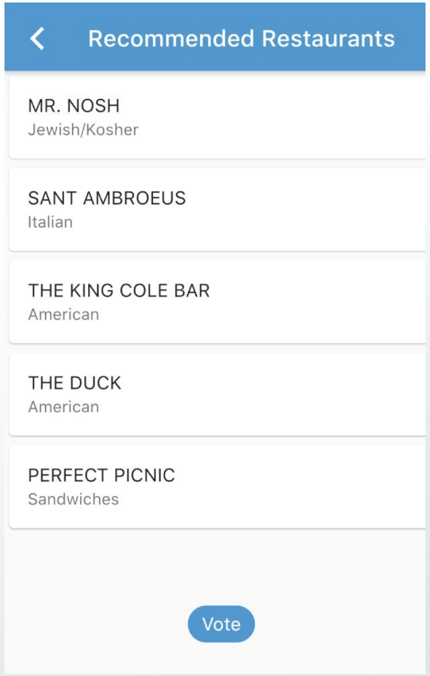
- Foodies is an app for creating events.
- Restaurants are generated for users to vote on.
- Friends vote for their favorite restaurant.
- The winning restaurant is the one we visit!

Demo

Demo

Architecture

- Frontend
  - Register/Login
  - Set preferences
  - Create events
  - Get event details
  - Vote for restaurants
- Backend
  - Stores user preferences and events
  - Recommended restaurants
    - Price
    - Time
    - Location
    - Group Preferences


Recommended Restaurants

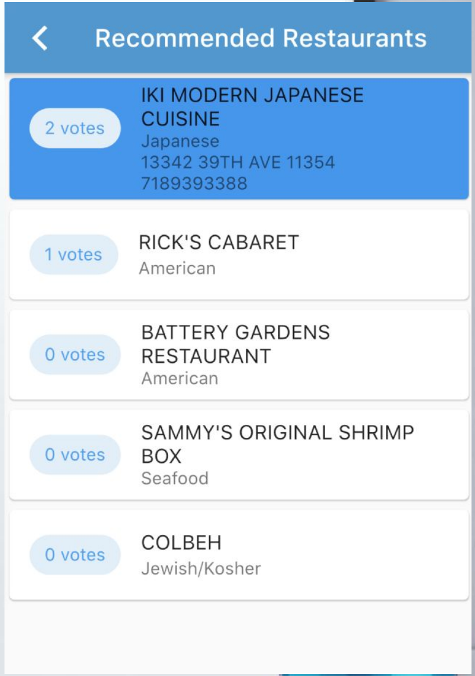
MR. NOSH
Jewish/Kosher

SANT AMBROEUS
Italian

THE KING COLE BAR
American

THE DUCK
American

PERFECT PICNIC
Sandwiches


Recommended Restaurants

2 votes	IKI MODERN JAPANESE CUISINE Japanese 13342 39TH AVE 11354 7189393388
1 votes	RICK'S CABARET American
0 votes	BATTERY GARDENS RESTAURANT American
0 votes	SAMMY'S ORIGINAL SHRIMP BOX Seafood
0 votes	COLBEH Jewish/Kosher

Technologies Used

+ Flutter for the frontend.
+ node JS as webserver and SQLite for storage.
+ NYC OpenData for restaurants, yelp fusion and Google Maps Platform for opening hours and prices, and julia to fetch data.
+ HEROKU for deployment.

Ideas For Future Development

- Push notifications
- Clone past events
- Easier adding of friends

Challenges Faced/Lessons Learned

+ Documentation (Flutter, Google API)
+ Communication (Git)

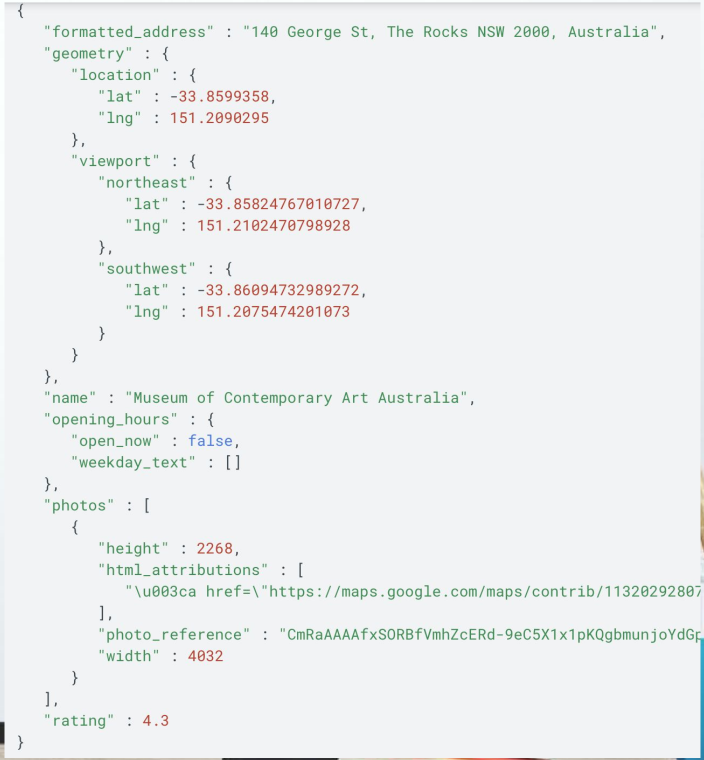
QUESTIONS?

Technical Presentations

- Google Places API
- Flutter

Google Places API

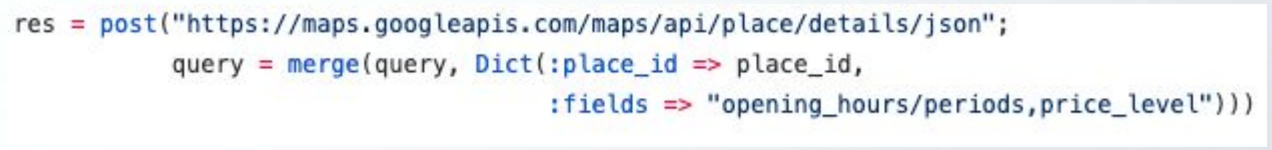
- Search based on an input location or search term
- Provide real-time autocompletion results
- Returns details and photos for each place
- Unique API Key is used to make each request
- Response with XML or JSON



```
{
   "formatted_address" : "140 George St, The Rocks NSW 2000, Australia",
   "geometry" : {
      "location" : {
         "lat" : -33.8599358,
         "lng" : 151.2090295
      },
      "viewport" : {
         "northeast" : {
            "lat" : -33.85824767010727,
            "lng" : 151.2102470798928
         },
         "southwest" : {
            "lat" : -33.86094732989272,
            "lng" : 151.2075474201073
         }
      }
   },
   "name" : "Museum of Contemporary Art Australia",
   "opening_hours" : {
      "open_now" : false,
      "weekday_text" : []
   },
   "photos" : [
      {
         "height" : 2268,
         "html_attributions" : [
            "\u003ca href=\"https://maps.google.com/maps/contrib/1132029280
         ],
         "photo_reference" : "CmRaAAAAfxSORBfVmhZcERd-9eC5X1x1pKQgbmunjoYdG
         "width" : 4032
      }
   ],
   "rating" : 4.3
}
```

Google Places API (in foodies)

- We used the Department of Health’s database to get initial restaurants but we were missing information
- To get opening hours and price range for each of those restaurants, we used Google Places API

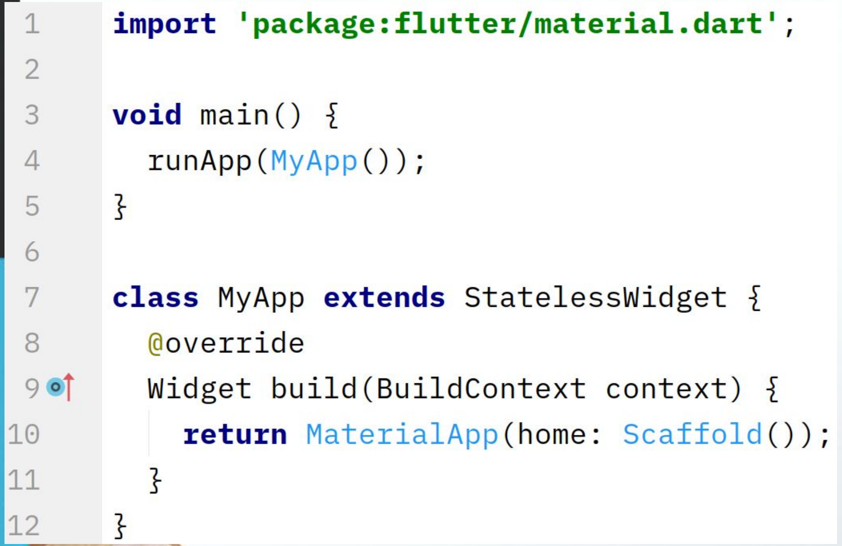


```
res = post("https://maps.googleapis.com/maps/api/place/details/json";
           query = merge(query, Dict(:place_id => place_id,
                                     :fields => "opening_hours/periods,price_level")))
```

Flutter – Synopsis

- Mobile UI framework
  - Material Design (‘material’) or iOS (‘cupertino’) look
- Dart programming language
  - Compiles to native code

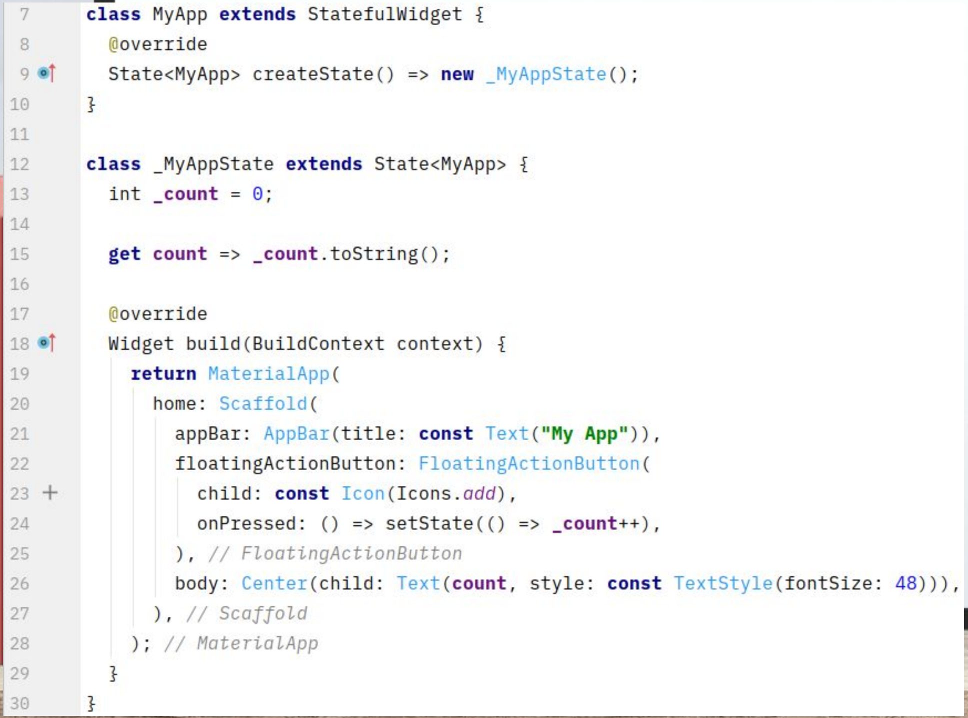
Flutter – Minimal App



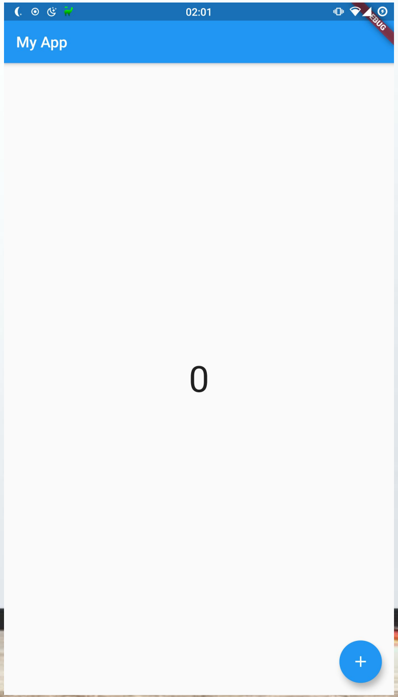
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(home: Scaffold());
  }
}
```

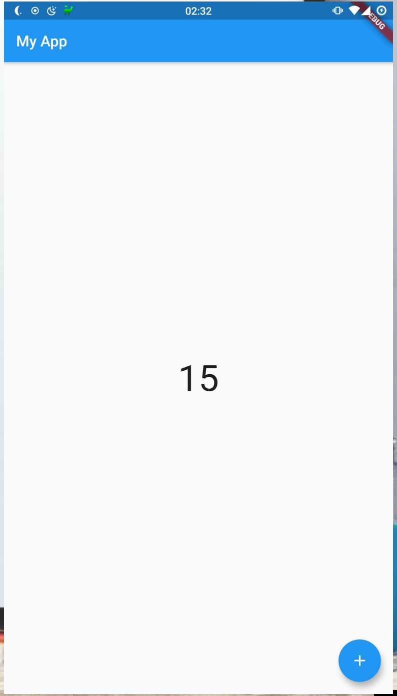
- Import 'material' package
- Create a class extending StatelessWidget
  - Override build()
- runApp() on our widget

Flutter – Scaffold

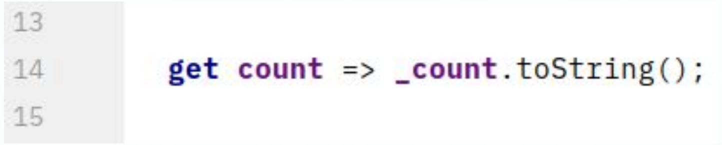


```dart
class MyApp extends StatefulWidget {
  @override
  State<MyApp> createState() => new _MyAppState();
}

class _MyAppState extends State<MyApp> {
  int _count = 0;

  get count => _count.toString();

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: const Text("My App")),
        floatingActionButton: FloatingActionButton(
          child: const Icon(Icons.add),
          onPressed: () => setState(() => _count++),
        ), // FloatingActionButton
        body: Center(child: Text(count, style: const TextStyle(fontSize: 48))),
      ), // Scaffold
    ); // MaterialApp
  }
}
```

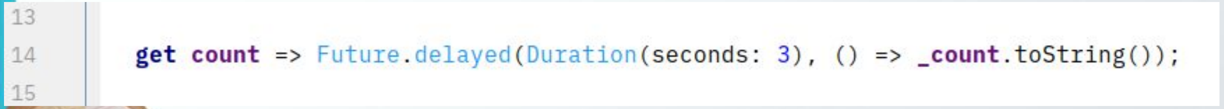




Dart – Futures



```
get count => _count.toString();
```

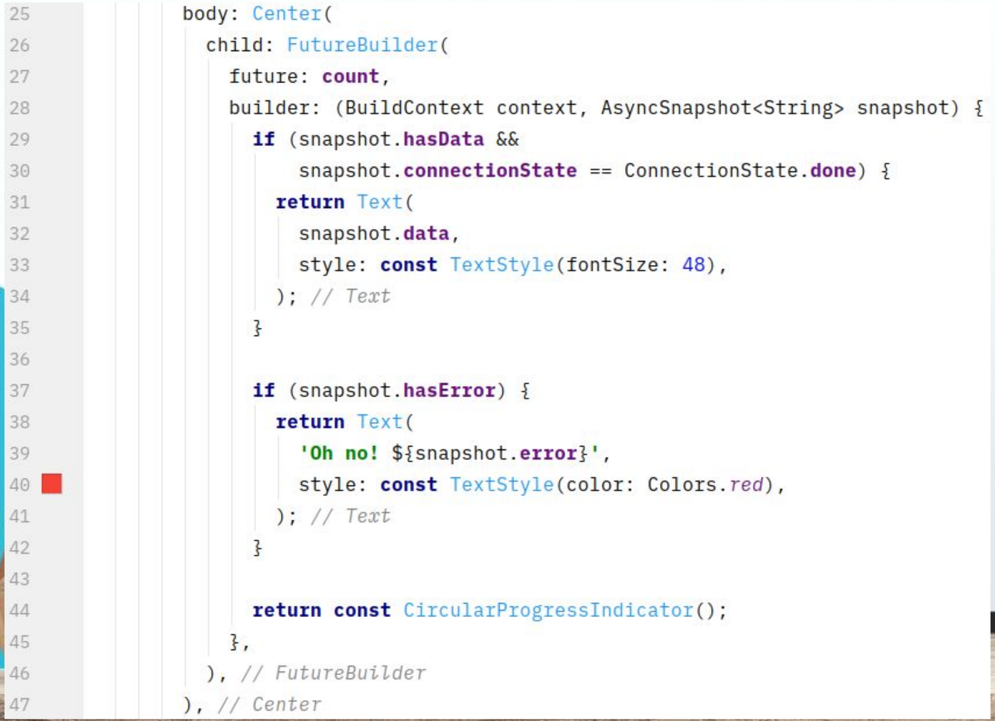


```
get count => Future.delayed(Duration(seconds: 3), () => _count.toString());
```

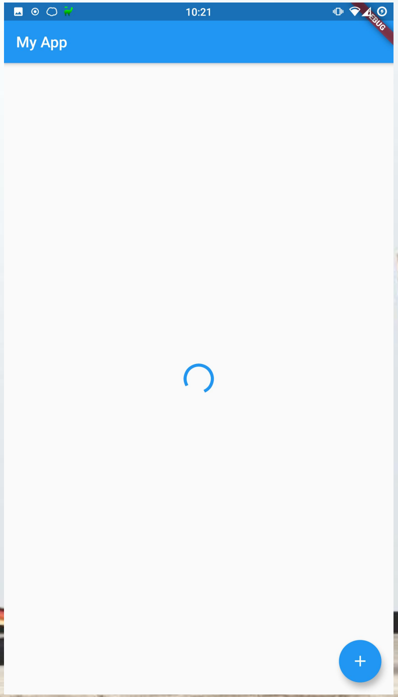
- The app hangs! Bad UX
- Worse: doesn't compile!

- Not unrealistic:
  - Slow network
  - Busy server

Flutter – FutureBuilder



```
body: Center(
  child: FutureBuilder(
    future: count,
    builder: (BuildContext context, AsyncSnapshot<String> snapshot) {
      if (snapshot.hasData &&
          snapshot.connectionState == ConnectionState.done) {
        return Text(
          snapshot.data,
          style: const TextStyle(fontSize: 48),
        ); // Text
      }

      if (snapshot.hasError) {
        return Text(
          'Oh no! ${snapshot.error}',
          style: const TextStyle(color: Colors.red),
        ); // Text
      }

      return const CircularProgressIndicator();
    },
  ), // FutureBuilder
), // Center
```



QUESTIONS?