

## 11 Final project

### 11.1 Project Idea

This application fetches news from Google News with some query parameters and displays each news into clickable cardview. After clicking news card, it will direct user to webview page using navigation.

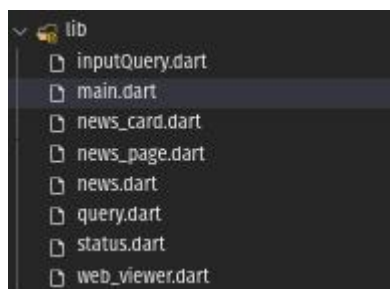
- Source code: <https://gitlab.tamk.cloud/minhdangphuoc/mobile-development-2-final>

### 11.2 System Design

#### 11.2.1 File Structure

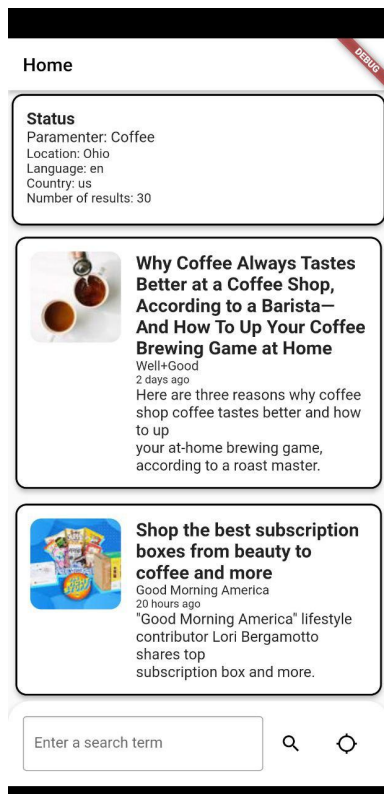
This project contains two pages (main.dart and web\_viewer.dart), 2 classes (news.dart and query.dart), a list view (news\_page.dart), a card view(news\_card.dart), a status view (status.dart).

PICTURE 11.2.1 project lib directory

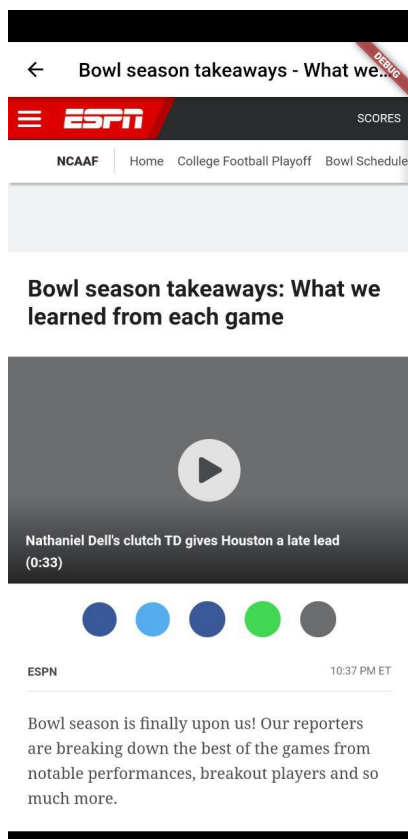


#### 11.2.2 User Interface

PICTURE 11.2.2.a Homepage - default parameter is coffee



PICTURE 11.2.2.b the webview



## 11.3 Code Review

### 11.3.1 Auto-fetching when data changes

PICTURE 11.3.1.a functions response for query values - parameter, location, country, lang, num.

```

class _HomePageState extends State<HomePage> {
  late Future<Query> query = Future(() => const Query(
    parameter: "Coffee",
    location: "Ohio",
    country: "us",
    lang: "en",
    num: 30)); // Query // Future

  void inputHandler({
    String? parameter,
    String? location,
    String? country,
    String? lang,
    int? num,
  }) {
    query.then((value) {
      setState(() {
        query = setNewQuery(Query(
          parameter: parameter != null ? parameter : value.parameter,
          location: location != null ? location : value.location,
          country: country != null ? country : value.country,
          lang: lang != null ? lang : value.lang,
          num: num != null ? num : value.num));
      });
    });
  }

  Future<Query> setNewQuery(Query newQuery) {
    return Future.value(newQuery);
  }
}

```

PICTURE 11.3.2.b Example of changing parameter in query by passing InputHandler

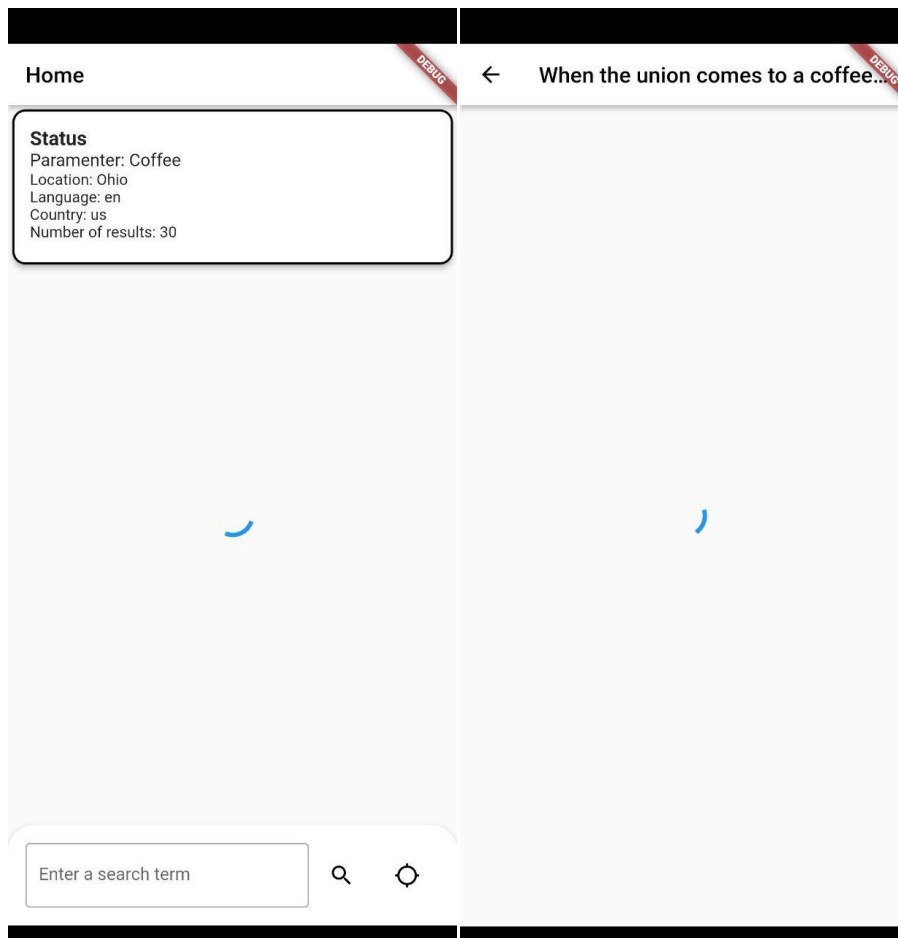
```
child: TextField(
  onSubmitted: (String value) {
    widget.setQuery(parameter: value);
  },
  onChanged: (String value) => {
    setState(
      () {
        parameter = value;
      },
    ),
  ),
  decoration: const InputDecoration(
    border: OutlineInputBorder(),
    hintText: 'Enter a search term',
  ), // InputDecoration // TextField
, // Expanded
circleAvatar(
  Positioned(
    child: Align(
      alignment: FractionalOffset.bottomCenter,
      child: InputQueryWidget(
        setQuery: inputHandler,
      )), // InputQueryWidget // Align // Positioned
  ), // Widget // Stack // Positioned
)
```

PICTURE 11.3.3.c When query changing value FutureBuilder will automatic run containing code to fetch and process data.

```
@override
Widget build(BuildContext context) {
  return FutureBuilder(
    future: widget.query,
    builder: (context, snapshot) {
      if (snapshot.hasData) {
        Future<List<News>> futureNewsResults = fetchData(snapshot.data!);
        return FutureBuilder<List<News>>(
          future: futureNewsResults,
          builder: (context, snapshot) {
            if (snapshot.hasData) {
              return FractionallySizedBox(
                heightFactor: 0.8,
                widthFactor: 1.0,
                child: ListView.builder(
                  itemCount: snapshot.data?.length,
                  itemBuilder: (context, index) {
                    return NewsCard(news: snapshot.data![index]);
                  }, // ListView.builder // FractionallySizedBox
                );
            } else if (snapshot.hasError) {
              return Text('${snapshot.error}');
            }
            // By default, show a loading spinner.
            return const Align(
              alignment: Alignment.center,
              child: const CircularProgressIndicator(); // Align
            );
          }, // FutureBuilder
        );
      } else if (snapshot.hasError) {
        return Text('${snapshot.error}');
      }
      return const Align(
        alignment: Alignment.center, child: CircularProgressIndicator(); // Align
      ); // FutureBuilder
    },
  );
}
```

### 11.3.2 Loading indicator when task is processing

PICTURE 11.3.2 News ListView and WebView



### 11.3.3 Webview

PICTURE 11.3.3 Passing uri to controller and using WebViewWidget to present the webpage.

```
class _WebViewWidgetState extends State<WebViewWidget> {
  late WebViewController controller;
  bool loading = true;
  late double process = 0;
  @override
  void initState() {
    super.initState();
    controller = WebViewController()
      ..setJavaScriptMode(JavascriptMode.unrestricted)
      ..setBackgroundColor(Color.fromARGB(0, 255, 255, 255))
      ..setNavigationDelegate(
        NavigationDelegate(
          onPageFinished: (String url) {
            setState(() {
              loading = false;
            });
          },
          onWebResourceError: (WebResourceError error) {},
          onNavigationRequest: (NavigationRequest request) {
            if (request.url.startsWith(widget.uri)) {
              return NavigationDecision.prevent;
            }
            return NavigationDecision.navigate;
          },
        ),
      )
      ..loadRequest(Uri.parse(widget.uri));
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(
        widget.title,
        style: const TextStyle(color: Colors.black),
      ), // Text
      iconTheme: const IconThemeData(color: Colors.black),
      backgroundColor: Colors.white,
    ), // AppBar
    body: Center(
      child: loading == false
        ? WebViewWidget(
            controller: controller,
          ) // WebViewWidget
        : const CircularProgressIndicator(),
    ); // Center // Scaffold
}
```

### 11.3.4 Rest Fetching

This project uses 2 fetching origins, for news and dominated language searching.

PICTURE 11.3.4 Example of fetching news and adding them into a list.

```
Future<List<News>> fetchData(Query query) async {
  final response = await http.get(Uri.parse(
    'https://serpapi.com/search.json?q=${query.parameter}&gl=${query.

  if (response.statusCode == 200) {
    final list = <News>[];
    if (jsonDecode(response.body)["search_information"]["news_results_s
      "Fully empty") {
        jsonDecode(response.body)["news_results"].forEach((element) {
          list.add(News.fromJson(element));
        });
      }
    }

    return list;
  } else {
    // If the server did not return a 200 OK response,
    // then throw an exception.
    throw Exception('Failed to load data');
  }
}
```

### 11.3.5 Navigation

PICTURE 11.3.5 By wrapping card into GestureDetector, hence, when user taps a specific card, it will navigate user to the webview page.

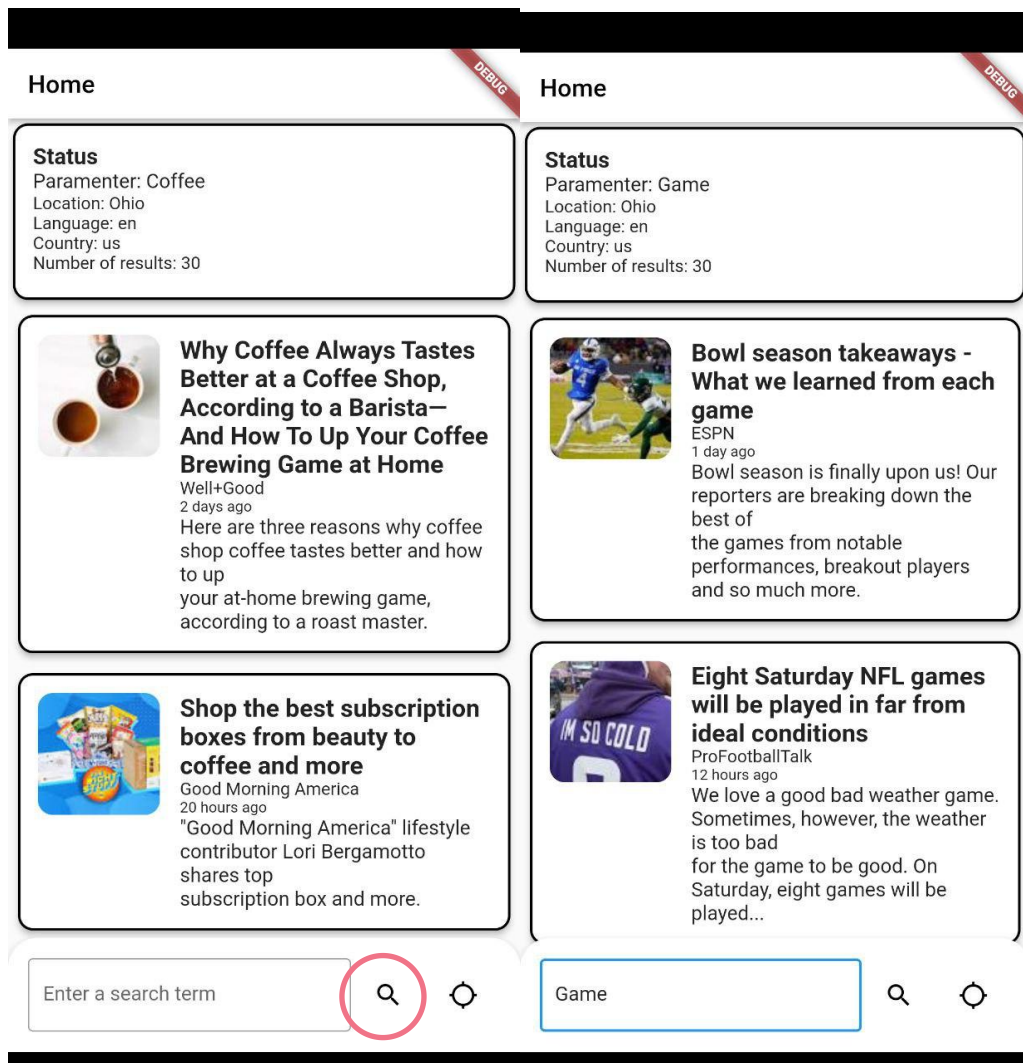
```
return GestureDetector(
  onTap: () => {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) =>
          WebViewerWidget(title: news.title, uri: news.link)),
    ),
  },
  child: Card(
    shape: RoundedRectangleBorder(
      side: BorderSide(width: 2.0, color: Colors.black),
      borderRadius: BorderRadius.circular(12.0),
    ), // RoundedRectangleBorder
  ),
```

## 11.4 Running

### 11.4.1 User Instruction

User can change parameter by insert some text in input field and click searching button or submit button on virtual keyboard.

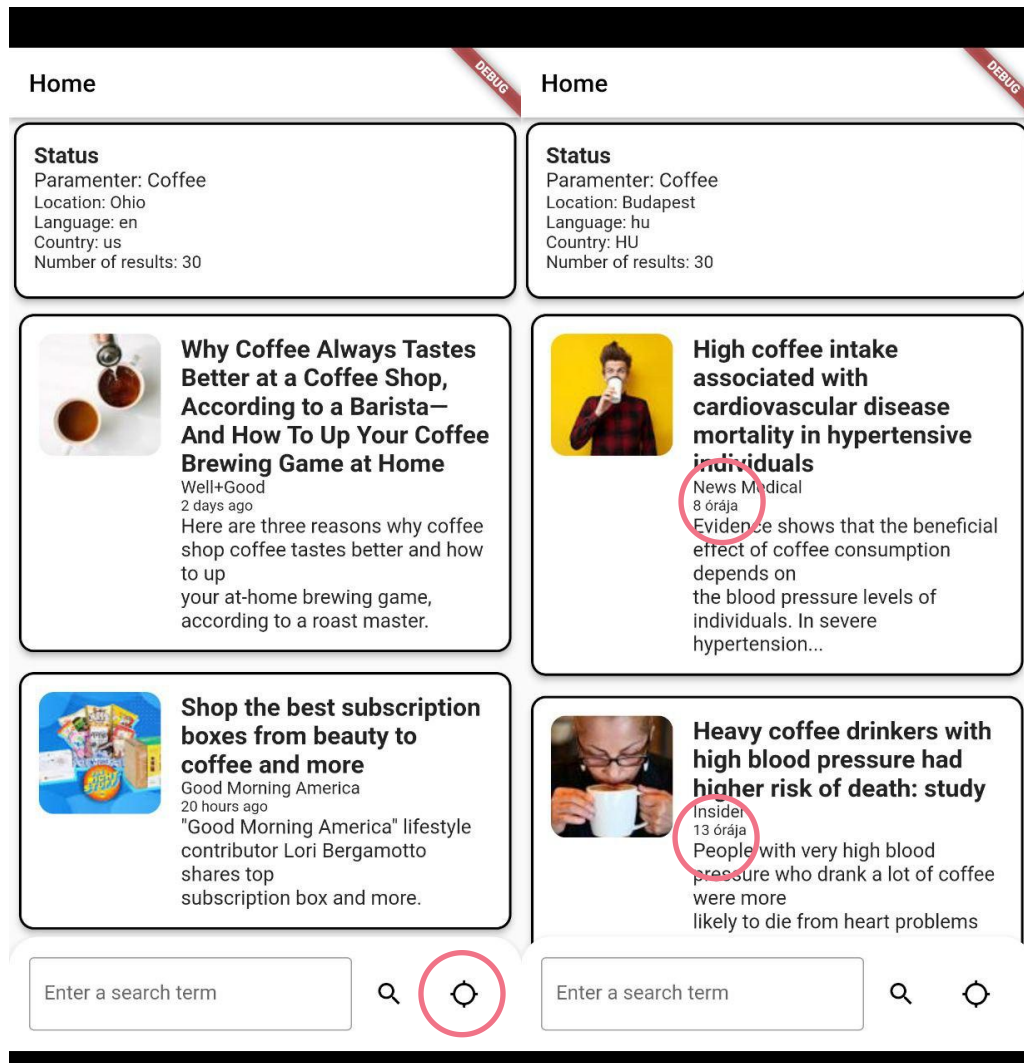
PICTURE 11.4.1.a Before and After change parameter.



After clicking location button, application will re fetch data with new query value and because dominated language in Budapest is Hungarian, then the date format will be changed.



PICTURE 11.4.1.b Before and After click location button.



### 11.4.2 Future Improvement

- User will face some issues when running application due to network performance, typo input, or Android version mismatch.