

Project 4 Task 1 Description: Recipe Search App

Name: Diksha Baruah

Andrew ID: dboruah

Description:

My application takes a search string (ideally for a food item) and uses it to fetch and display picture of food item, recipe, cooking time, calories of the requested item from Edamam Recipe Search API.

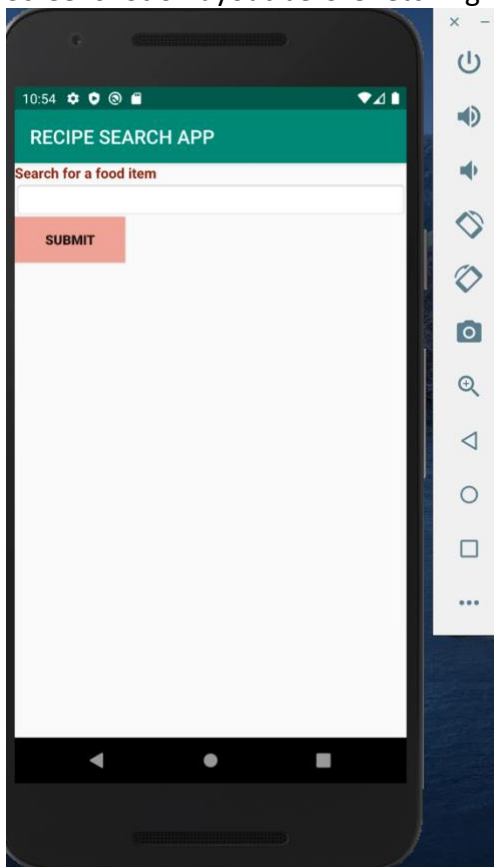
1. Implement a native Android App

App name: project 4 Android

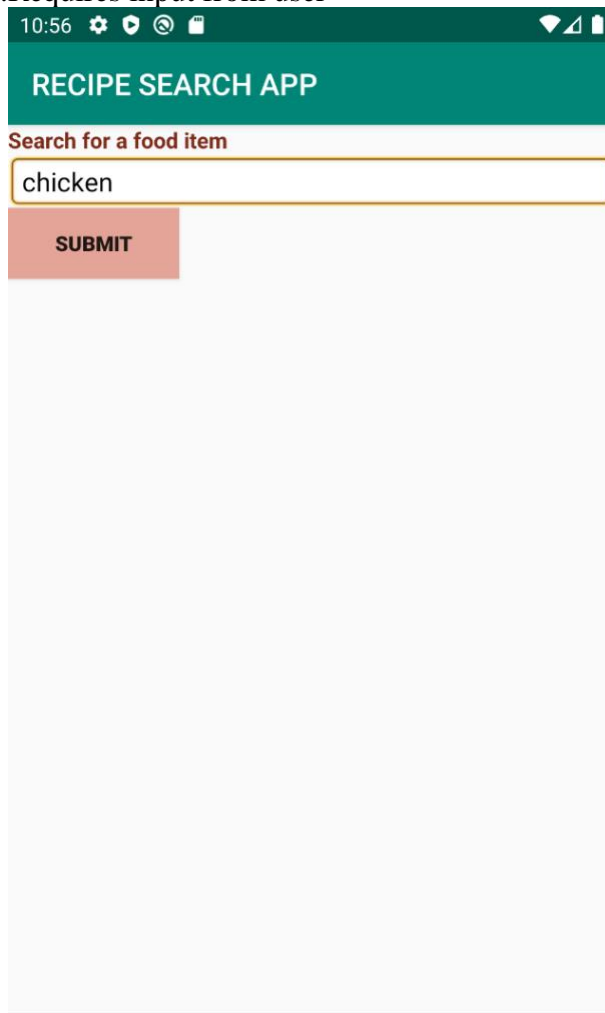
- 1.1. Has at least three different kinds of Views in your Layout (TextView, EditText, ImageView, or anything that extends android.view.View)

My app uses textView, EditText, Button and ImageView.

Screenshot of layout before fetching results



1.2.Requires input from user



1.3.Makes an HTTP request

My application does an HTTP GET request in GetResults.java. The HTTP req is:

```
"https://whispering-badlands-89043.herokuapp.com/?searchWord="+searchTag;
```

Where searchTag is the search term entered my user in the app.

The search method makes this request, parses the returned JSON to find data on name of dish, ingredients, picture URL, time taken, calories etc. It also fetches picture and returns bit image of the picture.

1.4. Received JSON formatted reply from webservice

```
Project CMU- Adding Catego How to electio joe bid Python electio localh localh Lab 3: CMU- Heroku http x androi us  
whispering-badlands-89043.herokuapp.com/?searchWord=chicken  
Apps inbox canvas Capstone-Data Productivity Chart Live Stage - Anita... Virtual Fair - Carn... Preliminaries - Pyt... Handshake 95702 (3 unread) Inferno | Login  
{ "nameOfDish": "Chicken Vesuvio", "source": "Serious Eats", "ingredients": [ "1/2 cup olive oil", "5 cloves garlic, peeled", "2 large russet potatoes, peeled and cut into chunks", "3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)", "3/4 cup white wine", "3/4 cup chicken stock", "3 tablespoons chopped parsley", "1 tablespoon dried oregano", "Salt and pepper", "1 cup frozen peas, thawed" ], "image": "https://www.edamam.com/web-img/e42/e42f9119813e890af34c259785aefcfb.jpg", "calories": "4228.043058200812", "timeTaken": "60.0 mins" }
```


1.5. Displays new info to user

RECIPE SEARCH APP

Search for a food item

SUBMIT

Here is the result for chicken:



Name of dish: Chicken Vesuvio
Source: Serious Eats
Calories: 4228 kCals
Cooking Time: 60.0 mins
Ingredients: 1/2 cup olive oil, 5 cloves garlic, peeled, 2 large russet potatoes, peeled and cut into chunks, 1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs), 3/4 cup white wine, 3/4 cup chicken stock, 3 tablespoons chopped parsley, 1 tablespoon dried oregano, Salt and pepper, 1 cup frozen peas, thawed,

1.6.Is repeatable


Here is an example when user searches for “French fries”

RECIPE SEARCH APP

Search for a food item

SUBMIT

Here is the result for french fries:



Name of dish: French Fry Waffles Recipe
Source: Serious Eats
Calories: 1055 kCals
Cooking Time: 25.0 mins
Ingredients: 1 1/2 pounds leftover cold french fries, or thawed frozen french fries, Kosher salt, Mayonnaise or ketchup for serving,

2. Implement a web service, deployed to Heroku

The URL of my web service deployed to Heroku is:
whispering-badlands-89043

For example:

<https://whispering-badlands-89043.herokuapp.com/?searchWord=chicken>

- a. Using an HttpServlet to implement a simple (can be a single path) API

In my project:

Model: FoodSearchModel.java

Servlet: FoodAppServlet.java

Schema: FoodAppSchema.java

- b. Receives an HTTP request from the native Android application

FoodAppServlet.java receives HTTP GET request with argument “searchWord” which is passed on to the model

- c. Executes business logic appropriate to your application

FoodSearchModel.java makes HTTP request to:

```
"https://api.edamam.com/search?q="+searchTag.strip()+"&app_id=c1b6ade8&app_key=bd82130b2e3d41a6f8cbab1e151c90f6";
```

The response is in JSON format and looks like:


```
{
  "q": "chicken",
  "from": 0,
  "to": 10,
  "more": true,
  "count": 120230,
  "hits": {
    "recipe": {
      "uri": "http://www.edamam.com/ontologies/edamam.owl#recipe_b79327d05b8e5b83ad6cfd9576b30b6",
      "label": "Chicken Vesuvio",
      "image": "https://www.edamam.com/web-img/e42f9119813e890af34c259785aefcfb.jpg",
      "source": "Serious Eats",
      "url": "http://www.seriouseats.com/recipes/2011/12/chicken-vesuvio-recipe.html",
      "shareAs": "http://www.edamam.com/recipe/chicken-vesuvio-b79327d05b8e5b83ad6cfd9576b30b6/chicken",
      "yield": 4.0,
      "dietLabels": ["Low-Carb"],
      "healthLabels": ["Peanut-Free", "Tree-Nut-Free"],
      "cautions": ["Sulfites"],
      "ingredientLines": [
        "1/2 cup olive oil",
        "5 cloves garlic, peeled",
        "2 large russet potatoes, peeled and cut into chunks",
        "1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)",
        "3/4 cup white wine",
        "3 tablespoons chopped parsley",
        "1 tablespoon dried oregano",
        "Salt and pepper"
      ],
      "ingredients": {
        "text": "1/2 cup olive oil",
        "weight": 108.0,
        "image": null,
        "text": "5 cloves garlic, peeled",
        "weight": 15.0,
        "image": "https://www.edamam.com/food-img/6ee/6eel42951f48aaf94f4312409f8d133d.jpg",
        "text": "2 large russet potatoes, peeled and cut into chunks",
        "weight": 738.0,
        "image": "https://www.edamam.com/food-img/71b/71b3756ecfd3d1efa075874377038b67.jpg",
        "text": "1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)",
        "weight": 1587.5732950000001,
        "image": "https://www.edamam.com/food-img/d33/d338229d774a743f7858f6764e095878.jpg",
        "text": "3/4 cup white wine",
        "weight": 176.39999999999998,
        "image": "https://www.edamam.com/food-img/a71/a718cf3c52add522128929f1f324d2ab.jpg",
        "text": "3 tablespoons chopped parsley",
        "weight": 180.0,
        "image": "https://www.edamam.com/food-img/26a/26a10c4cb4e07bab54d8a687ef5ac7d8.jpg",
        "text": "1 tablespoon dried oregano",
        "weight": 2.999999999999999,
        "image": "https://www.edamam.com/food-img/46a/46a132e96626d7989b4d6ed8c91f4da0.jpg",
        "text": "Salt and pepper",
        "weight": 17.720239769998784,
        "image": "https://www.edamam.com/food-img/694/6943ea510918c6025795e8dc6e6eaaeb.jpg",
        "text": "Salt and pepper",
        "weight": 8.860119884999392,
        "image": "https://www.edamam.com/food-img/c6e/c6e5c3bd83bc15175d9766971a4dlb2.jpg",
        "text": "1 cup frozen peas, thawed",
        "weight": 134.0,
        "image": "https://www.edamam.com/food-img/c91/c9130a361d5c5b279bf48c69e2466ec2.jpg"
      },
      "calories": 4228.043058200812,
      "totalWeight": 2976.8664549004047,
      "totalTime": 60.0,
      "totalNutrients": {
        "ENERC_KCAL": {
          "label": "Energy",
          "quantity": 4228.043058200812,
          "unit": "kcal"
        },
        "FAT": {
          "label": "Fat",
          "quantity": 274.4489059026023,
          "unit": "g"
        },
        "FASAT": {
          "label": "Saturated",
          "quantity": 62.497618998656044,
          "unit": "g"
        },
        "FATRN": {
          "label": "Trans",
          "quantity": 1.047163345382,
          "unit": "g"
        },
        "FAMS": {
          "label": "Monounsaturated",
          "quantity": 147.39060633938868,
          "unit": "g"
        },
        "FAPU": {
          "label": "Polyunsaturated",
          "quantity": 47.35051984782951,
          "unit": "g"
        },
        "CHOCDF": {
          "label": "Carbs",
          "quantity": 175.96206666631727,
          "unit": "g"
        },
        "FIBTG": {
          "label": "Fiber",
          "quantity": 19.83181033081862,
          "unit": "g"
        },
        "SUGAR": {
          "label": "Sugars",
          "quantity": 16.239344767255698,
          "unit": "g"
        },
        "PROCNT": {
          "label": "Protein",
          "quantity": 230.72689680763318,
          "unit": "g"
        },
        "CHOLE": {
          "label": "Cholesterol",
          "quantity": 815.06238045,
          "unit": "mg"
        },
        "NA": {
          "label": "Sodium",
          "quantity": 6888.614561646296,
          "unit": "mg"
        },
        "CA": {
          "label": "Calcium",
          "quantity": 400.0807431570531,
          "unit": "mg"
        },
        "MG": {
          "label": "Magnesium",
          "quantity": 478.1771035229573,
          "unit": "mg"
        },
        "K": {
          "label": "Potassium",
          "quantity": 5918.1808352043345,
          "unit": "mg"
        },
        "FE": {
          "label": "Iron",
          "quantity": 22.546435238210286,
          "unit": "mg"
        },
        "ZN": {
          "label": "Zinc",
          "quantity": 18.341531378501646,
          "unit": "mg"
        },
        "P": {
          "label": "Phosphorus",
          "quantity": 2231.0712550999992,
          "unit": "mg"
        },
        "VITA_RAE": {
          "label": "Vitamin A",
          "quantity": 635.3716670147774,
          "unit": "µg"
        },
        "VITC": {
          "label": "Vitamin C",
          "quantity": 103.72979744959534,
          "unit": "mg"
        },
        "THIA": {
          "label": "Thiamin (B1)",
          "quantity": 1.7264528338354403,
          "unit": "mg"
        },
        "RIBF": {
          "label": "Riboflavin (B2)",
          "quantity": 1.9119200245119274,
          "unit": "mg"
        },
        "NIA": {
          "label": "Niacin (B3)",
          "quantity": 86.90416982948213,
          "unit": "mg"
        },
        "VITB6A": {
          "label": "Vitamin B6",
          "quantity": 6.886357390963229,
          "unit": "mg"
        },
        "FOLDFE": {
          "label": "Folate equivalent (total)",
          "quantity": 276.2712108159691,
          "unit": "µg"
        },
        "FOLFD": {
          "label": "Folate (food)",
          "quantity": 276.2712108159691,
          "unit": "µg"
        },
        "FOLAC": {
          "label": "Folic acid",
          "quantity": 0.0,
          "unit": "µg"
        },
        "VITB12": {
          "label": "Vitamin B12",
          "quantity": 3.34660450586,
          "unit": "µg"
        },
        "VITD": {
          "label": "Vitamin D",
          "quantity": 2.1590996812000003,
          "unit": "µg"
        },
        "TOCPHA": {
          "label": "Vitamin E",
          "quantity": 19.62869476856695,
          "unit": "mg"
        },
        "VITK1": {
          "label": "Vitamin K",
          "quantity": 353.31486385948267,
          "unit": "µg"
        },
        "WATER": {
          "label": "Water",
          "quantity": 1738.7966568296217,
          "unit": "g"
        },
        "totalDaily": {
          "ENERC_KCAL": {
            "label": "Energy",
            "quantity": 211.4021529100406,
            "unit": "%"
          },
          "FAT": {
            "label": "Fat",
            "quantity": 422.2290860040035,
            "unit": "%"
          },
          "FASAT": {
            "label": "Saturated",
            "quantity": 312.48809499328024,
            "unit": "%"
          },
          "CHOCDF": {
            "label": "Carbs",
            "quantity": 58.65402222210575,
            "unit": "%"
          },
          "FIBTG": {
            "label": "Fiber",
            "quantity": 79.32724132327448,
            "unit": "%"
          },
          "PROCNT": {
            "label": "Protein",
            "quantity": 461.4537936152663,
            "unit": "%"
          },
          "CHOLE": {
            "label": "Cholesterol",
            "quantity": 271.68746015,
            "unit": "%"
          },
          "NA": {
            "label": "Sodium",
            "quantity": 287.0256067352624,
            "unit": "%"
          },
          "CA": {
            "label": "Calcium",
            "quantity": 40.08074315705315,
            "unit": "%"
          },
          "MG": {
            "label": "Magnesium",
            "quantity": 113.85169131498982,
            "unit": "%"
          },
          "K": {
            "label": "Potassium",
            "quantity": 125.9187411745603,
            "unit": "%"
          },
          "FE": {
            "label": "Iron",
            "quantity": 125.25797354561271,
            "unit": "%"
          },
          "ZN": {
            "label": "Zinc",
            "quantity": 166.74119435001495,
            "unit": "%"
          },
          "P": {
            "label": "Phosphorus",
            "quantity": 318.7244650142856,
            "unit": "%"
          },
          "VITA_RAE": {
            "label": "Vitamin A",
            "quantity": 70.59685189053081,
            "unit": "%"
          },
          "VITC": {
            "label": "Vitamin C",
            "quantity": 115.25533049955037,
            "unit": "%"
          },
          "THIA": {
            "label": "Thiamin (B1)",
            "quantity": 143.8710694862867,
            "unit": "%"
          },
          "RIBF": {
            "label": "Riboflavin (B2)",
            "quantity": 147.0707711163021,
            "unit": "%"
          },
          "NIA": {
            "label": "Niacin (B3)",
            "quantity": 543.1510614342633,
            "unit": "%"
          },
          "VITB6A": {
            "label": "Vitamin B6",
            "quantity": 529.7197993048637,
            "unit": "%"
          },
          "FOLDFE": {
            "label": "Folate equivalent (total)",
            "quantity": 69.06780270399227,
            "unit": "%"
          },
          "VITB12": {
            "label": "Vitamin B12",
            "quantity": 139.44185441083332,
            "unit": "%"
          },
          "VITD": {
            "label": "Vitamin D",
            "quantity": 14.393997874666669,
            "unit": "%"
          },
          "TOCPHA": {
            "label": "Vitamin E",
            "quantity": 130.85796512377968,
            "unit": "%"
          },
          "VITK1": {
            "label": "Vitamin K",
            "quantity": 294.42905321623556,
            "unit": "%"
          },
          "digest": {
            "label": "Fat",
            "tag": "fat",
            "schemaOrgTag": "fatContent",
            "total": 274.4489059026023,
            "hasRDI": true,
            "daily": 422.2290860040035,
            "unit": "g",
            "sub": {
              "label": "Saturated",
              "tag": "fat",
              "schemaOrgTag": "saturatedFatContent",
              "total": 62.497618998656044,
              "hasRDI": true,
              "daily": 312.48809499328024,
              "unit": "g"
            }
          }
        }
      }
    }
  }
}
```

d. Replies to the Android application with an XML or JSON formatted response. The schema of the response can be of your own design.

Replies to Android app in JSON formatted response in schema designed by me:

```
{
  "nameOfDish": "Chicken Vesuvio",
  "source": "Serious Eats",
  "ingredients": [
    "1/2 cup olive oil",
    "5 cloves garlic, peeled",
    "2 large russet potatoes, peeled and cut into chunks",
    "1 3-4 pound chicken, cut into 8 pieces (or 3 pound chicken legs)",
    "3/4 cup white wine",
    "3 tablespoons chopped parsley",
    "1 tablespoon dried oregano",
    "Salt and pepper",
    "1 cup frozen peas, thawed"
  ],
  "image": "https://www.edamam.com/web-img/e42f9119813e890af34c259785aefcfb.jpg",
  "calories": 4228.043058200812,
  "timeTaken": 60.0
}
```

3. Handle error conditions

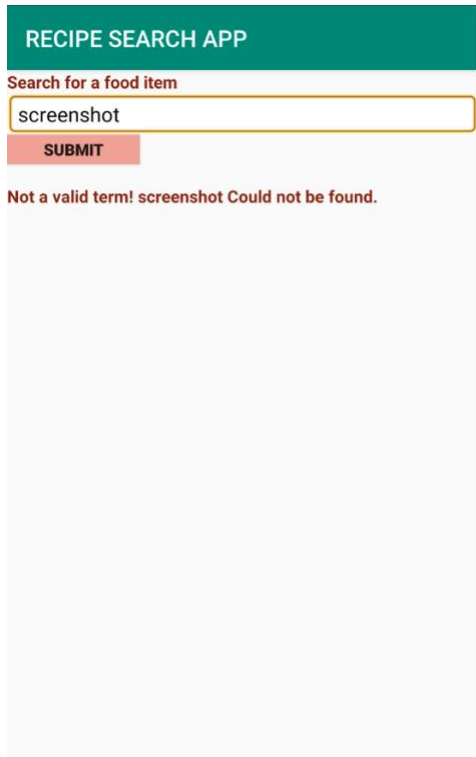
Your application should test for and handle gracefully:

- Invalid mobile app input

Invalid input1:

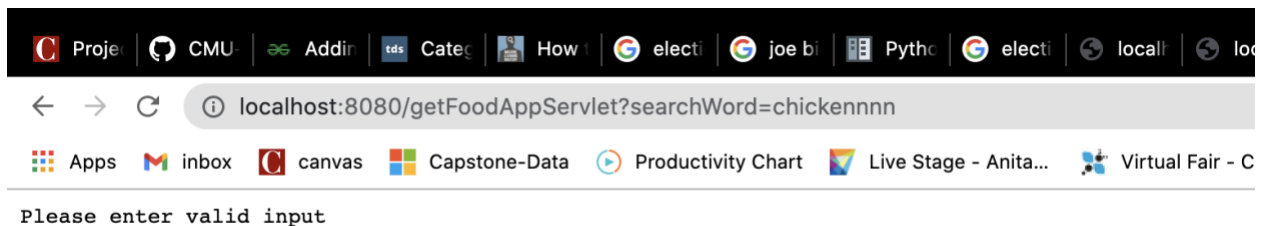
RECIPE SEARCH APP	RECIPE SEARCH APP
<p>Search for a food item</p> <input type="text" value="chickennnn"/>	<p>Search for a food item</p> <input type="text"/>
<p>SUBMIT</p>	<p>SUBMIT</p>
	<p>Not a valid term! chickennn Could not be found.</p>

Invalid input 2:

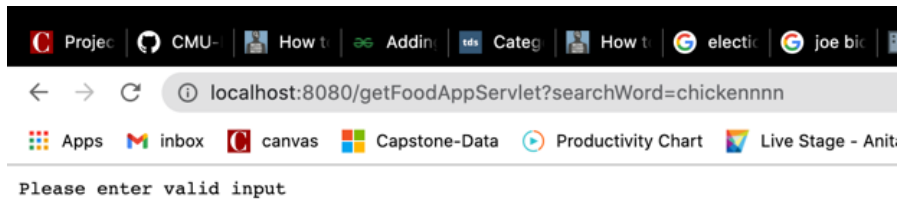


- Invalid server-side input (regardless of mobile app input validation)

Error handled in code when there is invalid or null output, also displaying “Please enter valid input” on browser.



Same for Heroku app:



- Mobile app network failure, unable to reach server

Handled by implementing try catch in search method of GetResults.java

- Third-party API unavailable

Handled by implementing try catch in fetch method of FoodSearchModel.java

- Third-party API invalid data

The API displays results for words that do not make sense in terms of food as well, eg: when user searches volume, the API displays:



Handled this condition in GetResults.java:

RECIPE SEARCH APP

Search for a food item

SUBMIT

Not a valid term! volume Could not be found.