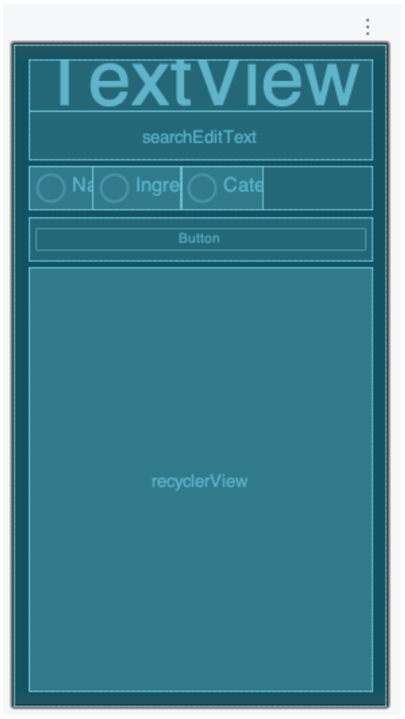
Task 2

- API Name: TheMealDB API
- URL: https://www.themealdb.com/api.php
- My mobile app will allow users to search for recipes by ingredient, category, or name, and then display detailed instructions and ingredient lists from TheMealDB API.

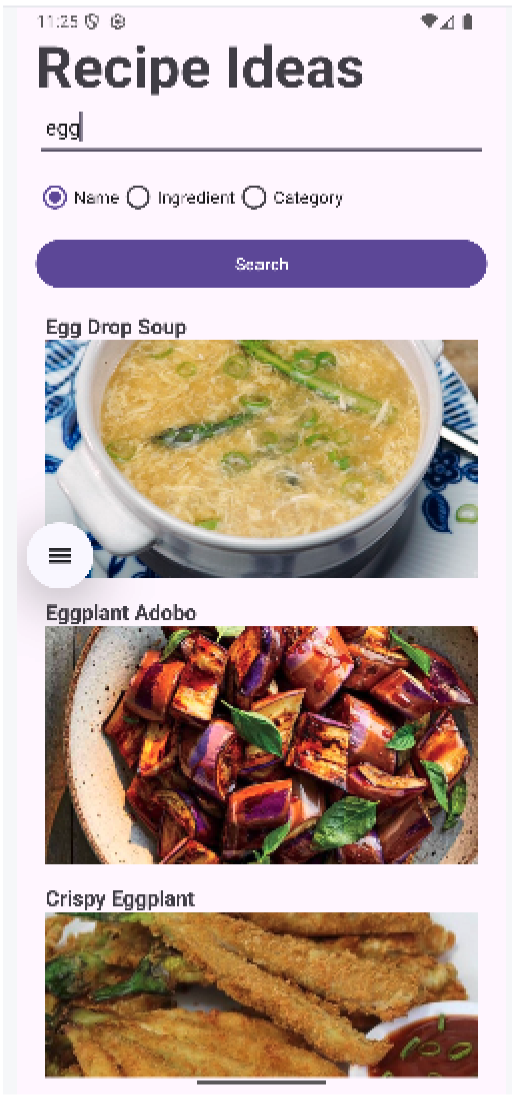
Requirements

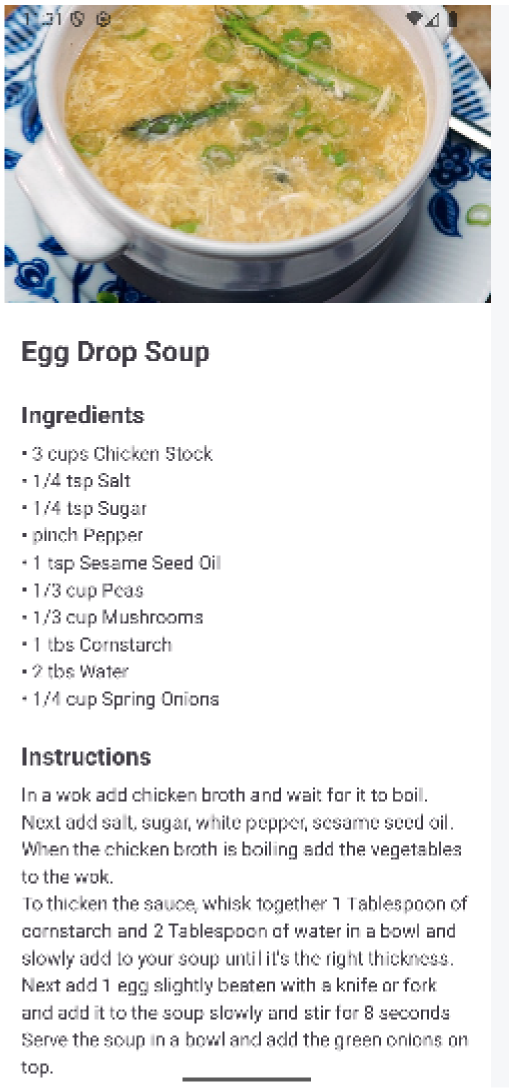
1. Implement a native Android application

a. Has at least three different kinds of Views in your Layout: TextView, EditText, RecyclerView, etc.



b. Requires input from the user: user input keywords in the search box





Egg Drop Soup

Ingredients

- 3 cups Chicken Stock
- 1/4 tsp Salt
- 1/4 tsp Sugar
- pinch Pepper
- 1 tsp Sesame Seed Oil
- 1/3 cup Peas
- 1/3 cup Mushrooms
- 1 tbs Cornstarch
- 2 tbs Water
- 1/4 cup Spring Onions

Instructions

In a wok add chicken broth and wait for it to boil.
Next add salt, sugar, white pepper, sesame seed oil.
When the chicken broth is boiling add the vegetables to the wok.
To thicken the sauce, whisk together 1 Tablespoon of cornstarch and 2 Tablespoon of water in a bowl and slowly add to your soup until it's the right thickness.
Next add 1 egg slightly beaten with a knife or fork and add it to the soup slowly and stir for 8 seconds
Serve the soup in a bowl and add the green onions on top.

c. Makes an HTTP request (using an appropriate HTTP method) to your web service: this is processed in doInBackground method of the BackgroundTask class

d. Receives and parses an XML or JSON formatted reply from your web service:

- Receiving the JSON reply happens in the doInBackground method of BackgroundTask class:
- Parsing the JSON occurs in the onPostExecute method of MainActivity class

e. Displays new information to the user

- when the keyword is searched, related information will show
- when specific recipe is clicked, the recipe detail will show as well

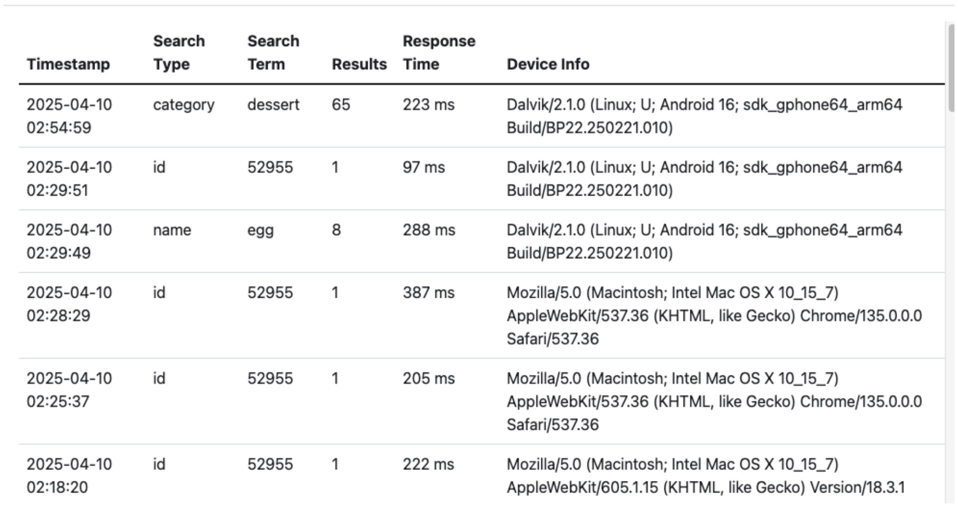
f. Is repeatable: yes, by swiping right, it will go to the result page, and can directly search for next keyword

2. Implement a web service

a. Implement a simple (can be a single path) API: Fulfilled through RecipeSearchServlet which exposes a single endpoint /recipe/search that handles GET requests.
b. Receives an HTTP request from the native Android application: Fulfilled through the doGet method in RecipeSearchServlet which:
    - Accepts HTTP GET requests
    - Processes query parameters (type and term)
    - Detects the client device through the User-Agent header: deviceInfo = request.getHeader("User-Agent")
c. Executes business logic appropriate to your application. This includes fetching XML or JSON information from some 3rd party API and processing the response:
    - Uses TheMealDB API (a legitimate public API) through MealDBClient
    - Makes HTTP requests to fetch JSON data from TheMealDB API
    - Processes the JSON response and transforms it into Recipe objects
    - Includes error handling for API failures
    - Implements different search types (by name, ingredient, category, or id)
    - Logs search operations to MongoDB for analytics
d. Replies to the Android application with an XML or JSON formatted response. The schema of the response can be of your own design:
    - Returns JSON responses with proper content type: response.setContentType("application/json")
    - Implements JSON response format through createJsonResponse method
    - Uses proper HTTP status codes for different scenarios (200, 400, 404, 502, 503)

4. Log useful information

- Logs 6 pieces of information: timestamp, searchType, searchTerm, resultCount, deviceInfo, responseTime, endpoint
- Includes information about the request from the mobile phone: deviceInfo, searchType, searchTerm
- Captures information about the response to the mobile phone: resultCount, responseTime
- Does not log dashboard operations: the logging is only called from the RecipeSearchServlet, not the DashboardServlet



Timestamp	Search Type	Search Term	Results	Response Time	Device Info
2025-04-10 02:54:59	category	dessert	65	223 ms	Dalvik/2.1.0 (Linux; U; Android 16; sdk_gphone64_arm64 Build/BP22.250221.010)
2025-04-10 02:29:51	id	52955	1	97 ms	Dalvik/2.1.0 (Linux; U; Android 16; sdk_gphone64_arm64 Build/BP22.250221.010)
2025-04-10 02:29:49	name	egg	8	288 ms	Dalvik/2.1.0 (Linux; U; Android 16; sdk_gphone64_arm64 Build/BP22.250221.010)
2025-04-10 02:28:29	id	52955	1	387 ms	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
2025-04-10 02:25:37	id	52955	1	205 ms	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
2025-04-10 02:18:20	id	52955	1	222 ms	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.3.1

5. Store the log information in a database

- Connect
  - Uses MongoDB Atlas (indicated by mongodb+srv:// protocol)
  - Establishes connection in the MongoLogger constructor
  - Connects to a specific database ("recipeApp") and collection ("logs")
- Store
  - fulfilled by logSearch method of MongoLogger class
- Retrieve
  - fulfilled by getAllLogs(), getMostSearchedTerms(), getAverageResponseTime() of MongoLogger class

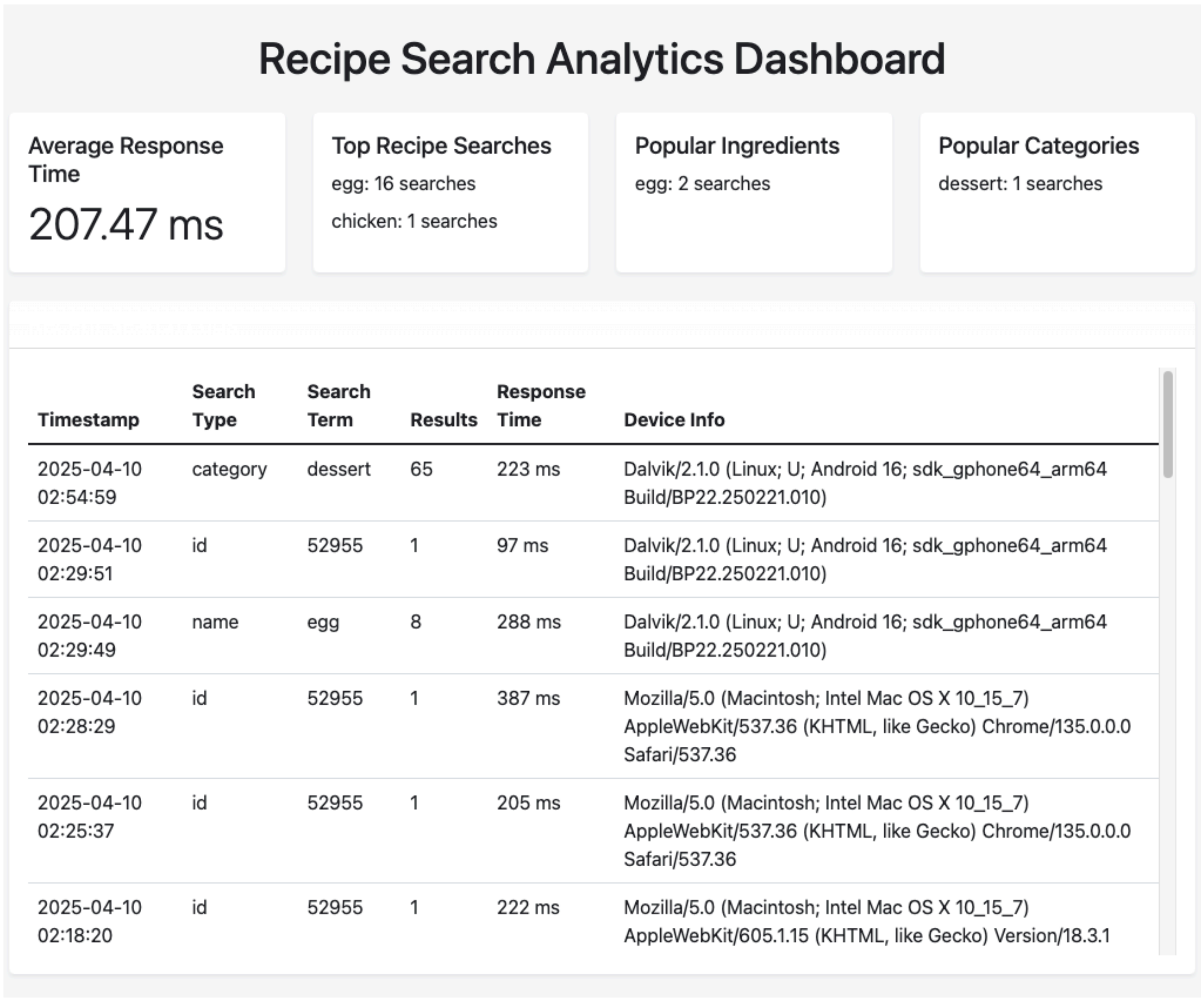
6. Display operations analytics and full logs on a web-based dashboard

a. A unique URL addresses a web interface dashboard for the web service

   https://ominous-space-xylophone-6pj965wpgvvc77g-8080.app.github.dev/dashboard

b. The dashboard displays at least 3 interesting operations analytics: shows Average Response Time, Top Recipe Searches, Popular Ingredients, Popular Categories

c. The dashboard displays formatted full logs



Recipe Search Analytics Dashboard

Average Response Time
207.47 ms

Top Recipe Searches
egg: 16 searches
chicken: 1 searches

Popular Ingredients
egg: 2 searches

Popular Categories
dessert: 1 searches

Timestamp	Search Type	Search Term	Results	Response Time	Device Info
2025-04-10 02:54:59	category	dessert	65	223 ms	Dalvik/2.1.0 (Linux; U; Android 16; sdk_gphone64_arm64 Build/BP22.250221.010)
2025-04-10 02:29:51	id	52955	1	97 ms	Dalvik/2.1.0 (Linux; U; Android 16; sdk_gphone64_arm64 Build/BP22.250221.010)
2025-04-10 02:29:49	name	egg	8	288 ms	Dalvik/2.1.0 (Linux; U; Android 16; sdk_gphone64_arm64 Build/BP22.250221.010)
2025-04-10 02:28:29	id	52955	1	387 ms	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
2025-04-10 02:25:37	id	52955	1	205 ms	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/135.0.0.0 Safari/537.36
2025-04-10 02:18:20	id	52955	1	222 ms	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.3.1

7. Deploy the web service to GitHub Codespaces: Successfully Deployed!