COVID Essentials

Prepared by
Dhiraj Chavva
Kelsey Manglicmot
Ommar Alvarez
Xiangyu Lou

Supervised by
Prof. Ma
MIS 768 Java
UNLV

May 07, 2020

Table of Contents

1. Purpose

The intention of the COVID Essentials application is to provide users with a seamless experience in their search for businesses that continue to offer their services despite quarantine restrictions that may alter their business practice. It is an application with the intention of making the dissemination of information to consumers easier by taking Yelp a step further to only provide results for businesses that are still open during times of quarantine.

2. Back End Development

COVID Essentials can be simplified as using a request-response application design that leverages the power of the current Yelp Fusion (v3) API. Using a private API key distributed by Yelp for open-source application development, our application is able to authenticate API calls to the millions of businesses, across 32 countries, that the Yelp database maintains a record of. We intend to develop a seamless graphical interface in order for users to use our application.

Yelp's Fusion API current consists of six endpoints, though our application will leverage the following five:

1. Business Search—Search for businesses by keyword, category, location, price level, *etc.*
2. Phone Search—Search for businesses by phone number.
3. Business Details—Business data such as name, address, phone number, photos, Yelp rating, price levels and hours of operation.
4. Business Match—Uses business data from other sources to match with Yelp businesses.
5. Reviews—Review excerpts for businesses.

COVID Essentials requires four important criteria to initiate search activity. These criteria consist of the arguments *term*, *location*, *price,* and *limit*. The *term* argument provides the variable of interest that serves as the basis for the business search (*e.g.* restaurant, retailer, or any other service indicator). *Location* and *price* arguments serve as the variables that indicate specific preferences of the client that will serve as parameters for filtration. The *limit* argument indicates the "search limit," or the desired number of hits that the user expects to choose from.

As summarized in the *Figure 1* flowchart, once these criteria are entered into the application's fields, these criteria are tokenized and fed into an HTTP builder, which builds a URL then sends a request to the Yelp Fusion API for information that fits the criteria. The Yelp Fusion API then translates these criteria into a GraphQL query that makes a call to Yelp's vast database for instances that fit these criteria; more specifically, the query calls for the five aforementioned endpoints of instances that suit the criteria. Once the query is complete, the Yelp

database then responds to the COVID Essential application by parsing the queried data into a JSON format, which is then returned back to the client. We intend to format the semi-structured data in the response into an intelligible format that the client can easily interpret.

2.1 Framework

We composed COVID Essentials using IntelliJ IDEA as the interactive development environment of choice, and heavily rely on the Maven core java framework. Maven offers the convenience of easily downloading project dependency libraries, as well as creating builds.

2.2 Modules

We included the following modules in COVID Essentials' development, the majority of which being JAR files relating to API interaction and the GUI:

- Okhttp-3.6.0
- Okio-1.11.0
- Java-json
- Javax.swing

In order for COVID Essentials to utilize the API we are implementing, We use OkHttp as our HTTP client to conduct the data-gathering requests as it loads data quickly, as well as saves bandwidth. Okio is the I/O library COVID Essentials utilizes, as it makes it easier to access, store, and process data. The data that COVID Essentials receives from its API is in JSON format, thus we include the java-json library to manipulate and parse out the data, which is ultimately stylized for user readability. Moreover, we use Java Swing as the GUI API, which builds the desktop interface that the user uses to interact with the console running behind the scenes.

2.3 Entity Relationship in Yelp

In order to understand the entities and its relationship (refer to *Figure 2*) The entities for Yelp are mentioned below along with their attributes.

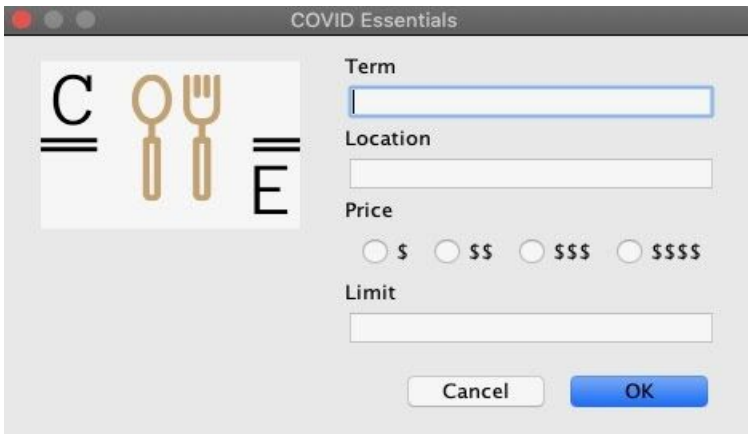
1) Users (UserID, Name, Gender, Age, Profile Picture, Date of birth)
2) Business (Business Attributes - Price, Ratings, Categories, Hours)
3) Reviews (ReviewID, ReviewContent)
4) Check-in (CheckinID, DayID)

3. Front End Development

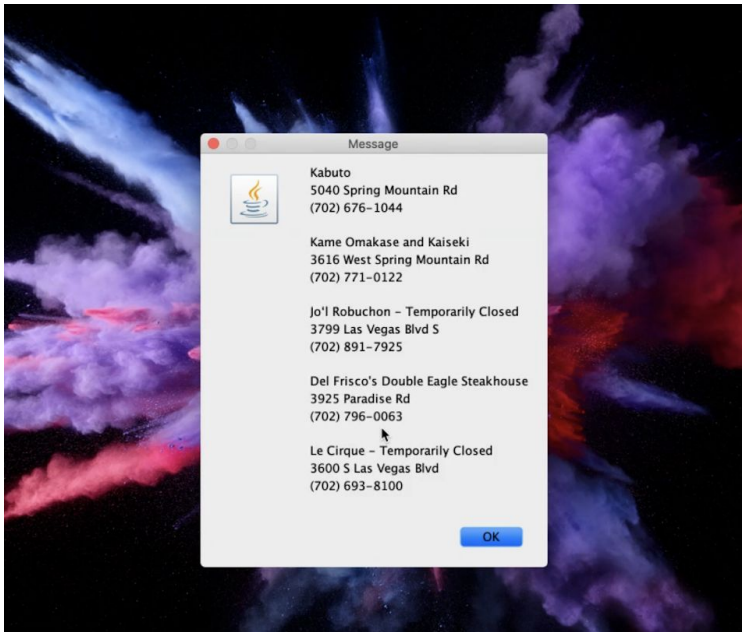
The COVID Essentials application sits on top of the IntelliJ console, but other than having to launch the program from the IDE, the user interacts with the application through the GUI developed using the Java Swing toolkit. Upon launching, the JOptionPane presented on the desktop contains JTextFields which are responsible for receiving text input from the user—specifically input used for the *term*, *location*, and *limit* arguments. As for the *price* argument, the user interacts with a panel of mutually exclusive radio buttons that are responsible for receiving the price-point preference of the user. Much like on the Yelp website and application, price-point is determined on a scale of {$, $$, $$$, $$$$}, which translates roughly to inexpensive, moderately inexpensive, moderately expensive, and expensive respectively. It should be noted that a user can freely input any string of text into the JTextField for search, thus there is no failsafe in case users input nonsensical search parameters. Once the arguments are collected from the user, the inputs are tokenized and passed onto the HTTP builder, which leads us back to the back-end of COVID Essentials' development

3.1 User Interface

The image below displays the JOptionPane responsible for containing JTextFields belonging to the *term*, *location*, and *limit* arguments, as well as the *price* radio button panel. The image featured on the left-side of the JOptionPane proudly displays our COVID Essentials logo.



The image below shows the output results for the client searching for sushi restaurants located in Las Vegas, with moderately expensive price preference, and search limited to the top 5 businesses in the desired area.



4. Project Planning and Schedule

Project planning and its schedule was carried out in two phases. We were able to finish a working prototype of COVID Essentials, hence remaining on our original schedule. The following details the two major phases of development:

4.1 Phase 1: Application Development

By the end of April, we finished two parts of the application. The first part was the backend of the application where COVID Essentials will gather the data needed using the Yelp API and parsing it out. The second part was creating the user interface with which the user was able to interact with and could input their desired filters. We made sure each one can independently work well.

4.2 Phase 2: Application Deployment

By the end of the first week of May, we planned to smooth the communication between the Yelp API and our graphical user interface to ensure that a seamless user experience is provided. Moreover, we will attempt to connect any potential errors that may arise from the user-side of the application if we face any issues with our working program. Before the mid of May, the

anonymous testing will be administered to a control group as a quality-assurance measure prior to the final debut.

4.3 Expected Project Delivery Time

We were able to deliver the project by May, we are gonna officially debut this app into the public after our testing phase with the control group. We expect to get positive feedback as much as possible.

5. Appendix

Figure 1: Covid Essentials Architecture

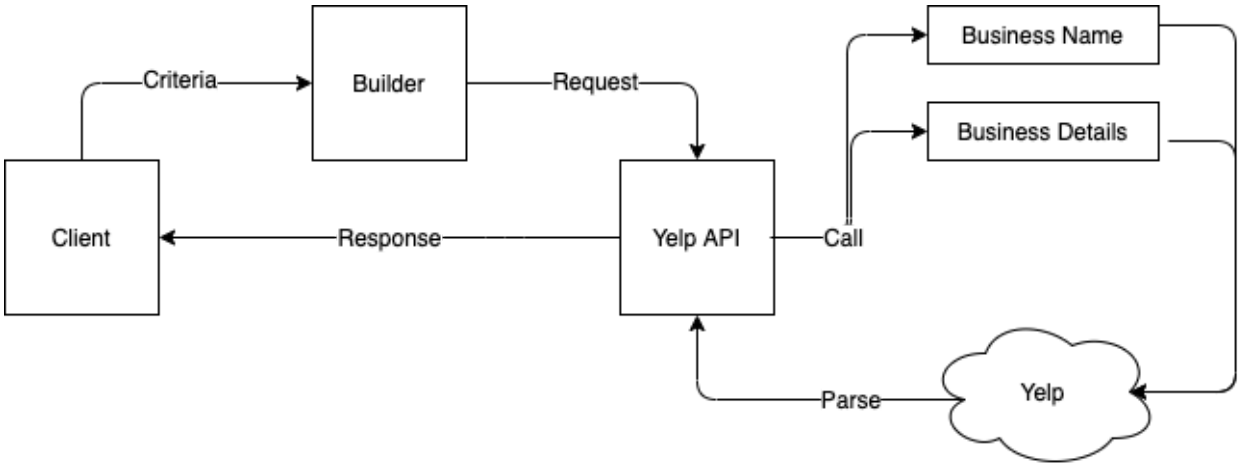


Figure 2: ER Diagram for Yelp