

## **Problem:**

In the modern digital era, people have access to hundreds of restaurants through countless services at their fingertips. While this abundance of choice may seem beneficial, very often it can lead to decision paralysis, where an individual is overwhelmed by the sheer number of options available to them. This indecision can often waste time, causing frustration and lead to decision fatigue. Behavioral studies have shown that excessive choice can decrease satisfaction and increase stress (Iyengar & Lepper, 2000; Schwartz, 2004). Our user research has shown that people take between 15-20 minutes deciding where to eat, which can lead to dissatisfaction. The root cause of decision paralysis lies in several key factors, including an overload of information, inconsistent and unreliable reviews, and lack of personal recommendations. While many services offer searches and recommendations, they are rarely tailored to individuals. Furthermore, these tools are more focused on presenting options rather than reducing them. This can lead to frustration when making dining choices.

## **Solution:**

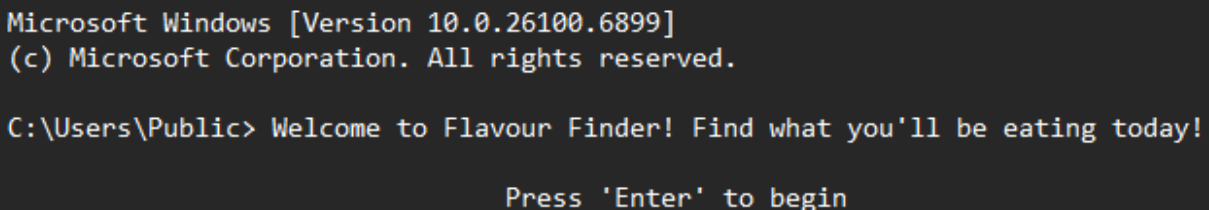
Our solution focuses on reducing the number of choices that a consumer needs to make. This service will randomly generate a restaurant near the user, with consideration of an individual's dietary restrictions before a suggestion is made. This will ensure that the restaurant will have options for the consumer. This drastically reduces the decision-making time from 15 minutes to seconds. It also introduces consumers to a wide variety of cuisine and traditions, which can lead to a deeper appreciation of other cultures. The aim of this is to reduce the number of decisions that a user needs to make.

Our idea is to use a command line interface, which will output a text based result based on the user's inputs.

**Inputs required** will include: Dietary restrictions, location \

**Utilization of Google m**

Upon loading, the user will be met with a welcome message



```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Public> Welcome to Flavour Finder! Find what you'll be eating today!

                        Press 'Enter' to begin
```

The first iteration will include prompting the user for dietary restrictions

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Public> Welcome to Flavour Finder! Find what you'll be eating today!

                        Press 'Enter' to begin

C:\Users\Public> Please List Any Dietary Restrictions:
```

As well as location:

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Public> Welcome to Flavour Finder! Find what you'll be eating today!

                        Press 'Enter' to begin

C:\Users\Public> Please List Any Dietary Restrictions:

C:\Users\Public> Pescetarian

C:\Users\Public> Where is the location you wish to dine in? (Results will be given within a 2km radius)
```

The two kilometer radius is predetermined to limit the choices that the consumer is required to make. There may later be a feature to alter this radius later on.

To implement the proper location features, we will be implementing a Google Maps API, specifically the Geocoding API, which converts named locations (like “Vancouver”) to coordinates. The Geolocation API then gets the user’s current location using IP or device data, Places API searches for nearby restaurants near given coordinates.

The command line interface will then print out a randomly generated restaurant that follows the dietary criteria:

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Public> Welcome to Flavour Finder! Find what you'll be eating today!

                        Press 'Enter' to begin

C:\Users\Public> Please List Any Dietary Restrictions:

C:\Users\Public> Pescetarian

C:\Users\Public> Where is the location you wish to dine in? (Results will be given within a 2km radius)

C:\Users\Public> Richmond

C:\Users\Public> Found: Wo Fung Noodle House, 8088 Park Rd, Richmond, BC V6Y 2C4
```

The program will then print out the additional lines:

```
Enter '1' to generate a new location
```

```
Please do not close the program if you wish to rate this restaurant!
```

The first feature is to allow users to generate another random restaurant should they feel the need to. The primary goal is to limit the number of options that the user is actively seeing, so with this feature we can limit it to one at a time.

The remainder of the program will be implemented as shown:

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Public> Welcome to Flavour Finder! Find what you'll be eating today!

                        Press 'Enter' to begin

C:\Users\Public> Please List Any Dietary Restrictions:

C:\Users\Public> Pescetarian

C:\Users\Public> Where is the location you wish to dine in? (Results will be given within a 2km radius)

C:\Users\Public> Richmond

C:\Users\Public> Found: Wo Fung Noodle House, 8088 Park Rd, Richmond, BC V6Y 2C4.

                        Enter '1' to generate a new location

                        Please do not close the program if you wish to rate this restaurant!

                        Press 'Enter' to rate this restaurant

C:\Users\Public> How would you rate this restaurant out of 5?

                        For 1, press '1'
                        For 2, press '2'
                        For 3, press '3'
                        For 4, press '4'
                        For 5, press '5'

C:\Users\Public> 7

C:\Users\Public> Error, rating can not be greater than 5. Please try again.

C:\Users\Public> 4

C:\Users\Public> Rating saved. Do you wish to save this restaurant to your profile?

                        For 'Yes', press '1'
                        For 'No', press '2'

C:\Users\Public> 1

C:\Users\Public> Restaurant and rating saved.
```

Specific specifications on the users end will include:

- a rating number no greater than 5, but inclusive of 5, no less than 0, but inclusive of 0.
- Unspecific/broad locations, such as “Richmond”, will scan the epicenter of said location within a two kilometer radius
  - If no restaurants exist within this area, a message will be returned indicating that no matches were found

```
C:\Users\Public> No restaurants found within the area
```

- Mistypes of dietary restrictions will be compared for similarity for other existing conditions in the dataset

```
C:\Users\Public> Peskketarian
```

```
C:\Users\Public> Did you mean 'Pescetarian'? (1) Yes, (2) No
```

Confirmations will give an extra layer of security to ensure that the program is understanding what the user is indicating.

The second iterations will occur should the user wish to go to another restaurant, where the above process will repeat. If the user closes the program, all current tasks will terminate. A manual override feature for the user, such as ‘Rate’ where the user can manually enter the restaurant and rate it, may be implemented in the future.