

As the name suggests, WhatToEat is a web application that decides what to eat. Current restaurant apps, and recipe apps bombard the users with options and information. It is up to the user to decide from millions of choices. WhatToEat, on the other hand, identifies the user's information from the login data, and narrows the choices. The main idea is to make the user's life easy by evaluating all possibilities to make one optimal choice. The app's main target is young adults who are comfortable utilizing web applications for choosing restaurants. However, it is open to users of all ages that want to make a good meal choice. Our plan is to use the map API and weather API to identify the user's environment, and then use food API's to suggest a few choices of optimal meal options. Other potential features may provide the distance to location, customer reviews etc. Below is an example of a situation where WhatToEat can come in handy, and the app usage in steps:

- A typical rainy weekend in Vancouver, Joe wants something for brunch that suits the weather. He only has a vague idea. Maybe something warm and with a soup, but can't quite put his finger on one menu. Hamburger and Pizza feels like sunny day food, Steak is too heavy for brunch, and he never had Transylvanian cuisine to know whether it is good for rainy days. He wants to decide what he is going to be eating, before getting out of his bed and get ready for the meal.

step 1: Log in

- The user must be able to log in, so that the user data can be stored, and the user experience can be personalized.
- The application needs to identify the user's location, and the weather of the location.
- The application may need to remember which restaurants have already been visited for later usage.
- If the user chooses to use without logging in, the Status Page would show general info.

step 2: Status Page

- The status page will display the current status of the user's environment.
- The status page can display the weather of the user's location, time of the usage, current location etc.

step 3: Preference Input

- The status page will include an input bar so that the user can use to enter his/her preferences.
- User may use check boxes to select preferences (delivery availability, vegan meal availability etc.)
- Quick search buttons (nearest restaurant, Weather fitting meals, Eat out or Dine in) may be presented.

Step 4: Results Page

- According to the user's preference inputs, the application must be able to present some results.
- Results should not exceed 5 choices. Results can be presented in a list.
- User may choose to like or dislike a restaurant/meal option