



# UI

Algorithms/Internal Workings:

Should use some sort of repetition algorithm to give the user variety while not repeating some meals too frequently

Meals should be ranked based on meal information given by the user. The more information provided by the user, the better the algorithm should rank and distribute meals.

## General Features

### User Experience

### Availability

### UI: Main Screen

### UI: Calendar View

### UI: Meal-Management screen

Overview for day

Shows a weekly calendar that displays eating schedule (does not do specific days, but rather just the days of the week)

Uses color-coding for different days of the week.

Can add/edit/disable/remove (in one word, manage) meals.

Upcoming/next-scheduled meal

Can adjust specific planned meals to match times that work for the user

Color-coding for meal significance will change throughout the day if other meals are not eaten.

Meal addition menu should have criteria to help the algorithm "rank" the meal, such as calorie count and other nutrition-facts-related criteria

Daily total (could mean a number of things, still figuring it out)

Can move specific items "down the line", making them appear at a later time.

Uses color-coding to indicate significance of meal (e.g., meals that should be eaten daily vs. ones that are not as important)

Meals will not be assigned to specific days in this screen. It is entirely for "inventory management".

### Downloadable profiles

### Shareable profiles

Possibly cross-platform sync (if there is, for example, an iOS app and a macOS app)

The user should just have to give meal information, and the algorithms should take care of planning for them.

Modern UI design that integrates with system for streamlined, expected-of-an-app user experience.

Accessibility options (such as touch not being the only input method, making keyboard-centric approach if there is a PC version)

Consistent design that should follow typical user expectations (as in, settings/features are not randomly thrown around the app).

### Should be mobile-first (iOS, Android)

Can be cross-platform (Windows, macOS, iOS, Android, etc...). This can be achieved using a cross-platform framework (many popular options available).

### The app should be free for all users regardless of where the app is available.

