FoodWise

Mai M. Mostafa | Elton C. Manchester | Na-Young Son | Ava E. Love

This README contains an introduction to FoodWise and information surrounding the hi-fidelity prototype, how to access it, and limitations.

• • •

Introduction

FoodWise is an app designed for individuals of any age, gender, or background that are looking to recover from an eating disorder–whether mild or severe. Through a personalized recovery roadmap; mood, stress, and food-noise tracker; toolbox and customizable catalog of coping skills; and emergency help feature, FoodWise provides users with knowledge and habits for sustainable recovery.

• • •

Installation Instructions

Our app is accessible to any iOS or Android device. To use it, first download the app **Expo Go.** FoodWise was designed with Expo SDK 52, which is the latest version at the time of release. If this is now out of date, it may be necessary to install SDK 52 on an emulator at expo.dev/go. Once Expo Go is set up, simply scan the QR code below to run the app.

Limitations

- **Unimplemented Roadmap Modules:** Within the first module of the roadmap, the lesson and lesson journal pages do not actually contain lesson content. Additionally, the first module is the only module currently accessible (subsequent modules have yet to be implemented).
- **Wizard-of-Oz Consequences:** Because the tracker and first module lesson video (see "Wizard-of-Oz" section below) are only simulated, the tracker does not collect or save data from the user when inputted, and the lesson video in the first module is not actually viewable.
- **Erroneous Timer Functionality:** If a user navigates to the emergency help feature and starts the timer in an activity, the timer does not reset if the user exits and returns to the emergency help feature later on.
- **Lack of Notifications:** To enhance the user experience, we originally had intended to include pop-up notifications congratulating the user that they've received "points" once completing an activity/task. This feature, however, has not been implemented.
- **Missing Info Button and Tutorial**: The info button has not been implemented (it is currently nonfunctional), and there is no tutorial for how to use the app.
- **Lack of Personalization:** Most of the "personalized" features (including the personalization of module lessons from the user's intro survey data and specific eating disorder history) are hard-coded, thus making the current prototype's recovery roadmap slightly less user-specific.

Wizard-of-Oz

To provide an experience of a fully functioning app, there are certain functionalities that are only simulated, with a Wizard-of-Oz technique:

- **Tracker:** Several charts are displayed with fabricated user data. We used fabricated data for the charts since a first time user will not yet have inputted data, and we wanted the prototype to reflect the experience of a user who has been using the app for some time.
- **Module Lesson**: The lesson page inside the first module displays a mockup of a video that does not exist and, hence, is not viewable. Generating an actual video was outside the scope of the prototype, so this mockup is meant to show what the interface would look like if there were a video present.
- **Timer:** The emergency help feature includes a timer that is currently not functional. In real implementation, this timer would accurately count down from five minutes, but this implementation was outside the scope of this prototype.

Hard-Coded Items

To provide an experience of a fully functioning app, there are some functionalities we implemented that are hard-coded:

- **User and Profile:** The logged in user (turtle) and the profile page cannot be changed.
- **Text Entries:** All text inputs (including the intro survey, tool editing, and journaling) are hardcoded. This is helpful for the prototype because it reduces the amount of time the user needs to spend on each screen, thus allowing them to focus on usability aspects rather than specific text entries.
- **Toolbox:** When selecting tag(s) for a tool in the toolbox, stress is currently the only selectable tag option. This hard-coding, however, allows the prototype to showcase the tag feature while preventing there being too many options for tags, which could potentially detract from the usability aspect of the toolbox.