



Nourish your body,  
Grow your mind.

Jasmine X. | Yujen L. | Jade C. | Clare L.

## **README USER MANUAL**

High-Fi Version

# Introduction

BlooME is designed for individuals who struggle to maintain a healthy relationship with food, including those with both clinical and subclinical eating disorder symptoms. Many keep their struggles private due to feelings of shame, stigma, or fear of burdening others. As a community-based platform, BlooME aims to provide a safe and supportive space where people can connect, share experiences, and feel less alone in their recovery journey.

## Our Tasks

More specifically, there are three tasks supported by the hi-fi prototype:

1. Collect the “Daily Message” upon opening the app.
2. Browse the Community feed, open a post, and collect/save it so it appears in your Saved or Pinned posts on the Home page.
3. Create a new post (private/public) using a free-writing or guided reflection prompt.

## Design Tools

Our high-fidelity prototype was built using React Native and Expo to implement core functionality and interface flows, supported by Supabase for backend data management. Ongoing testing occurred through Apple’s Xcode Simulator and physical devices running the Expo Go app. The full toolset included:

- **React Native** as our development framework
- **Expo & Expo Router** for navigation, app framework, and rapid testing
- **Supabase** for our database
- **Visual Studio Code** as our primary development environment
- **Figma** to design interface components and interaction flows
- **GitHub** for version control and collaboration

## Operating Instructions

To access the prototype, please follow the instructions below:



## Limitations

A major limitation of this prototype is that it does not use actual people or a real active user base, so many elements had to rely on Woz techniques to simulate a functional social community. Features such as posts, reactions, and user activity were preconstructed to give the impression of engagement rather than being generated by real users. We were also not able to develop the onboarding tutorial within the timeline, which limits how clearly new users would understand the intended flow and core features of the app. These constraints reflect the rapid pace of prototyping and highlight areas that would need full implementation in a future version. Another limitation is that we could not add personalization and AI based components. We didn't fully develop the different settings, like text and language, that the user can experience BlooME in. We also wanted to originally have AI generated prompts for journaling and also recommending users posts based on their previous saved posts, but these features are not active.

## Accessibility

This prototype lacks screen reading elements, which would improve accessibility, but this was not feasible within our development timeline. We also did not activate a dictation feature, although we would have liked to include it to make journaling easier for people who prefer speaking to typing. The text size is currently fixed rather than adjustable, which can limit readability for those who need larger text. However, we applied Fitts's law where possible to make interactive elements easier to tap and navigate.

## Wizard of Oz Techniques

We used Woz techniques to simulate the peer community in our prototype. The app has a peer user base, but it is actually prepopulated to give the illusion that people are

already active on the platform. Although real reactions and new posts are sent to a database, the initial set of posts was created through Woz to make the experience feel authentic, especially when exploring the community. The terms and conditions screen is also Woz, functioning as a placeholder for a feature we have not fully built yet when a new user signs up. Even the message of the day is Woz, since we are faking that the system selected something meaningful when it is actually predetermined. It is just one message that we have as the illusion of a message of the day being sent to the user chosen from our database of prepopulated story posts. Those are all the WOZ techniques present in our high fidelity prototype.

## **Hard-coded Items**

To simulate key features without implementing full backend infrastructure, we hardcoded a few elements of the prototype. The message of the day is hard coded rather than dynamically generated. The reactions users can choose from come from a fixed, predefined set. The feature itself works, but we do not allow the user to choose their own emojis to react. The tags available on posts are also hard coded, serving more as selectable options than as a true tagging system. We do not have user generated tags and a way to find themes in an algorithmic way, but we force users to choose a tag that fits best. Our daily affirmations and the phrases the app can refresh to are likewise based on a static list rather than any real algorithm. Even the journal prompts are hard coded, so the questions users see are not being generated in real time but drawn from a preset collection created for the prototype. The guiding sub-questions once a journal prompt is picked is also hard coded. The questions that move from acknowledging the scary to growth and positive reframing has been an intentional choice that has been set through development.