x

YOURSTRULY

we deliver the celebration

**Medium-Fidelity README**

Grace Alwan, Alex Crisera, Sasha Moore
CS147- Human-Computer Interaction
Winter 2022
Stanford University

## Med-Fi Prototype Link:

Figma

## Operating Instructions

Because we used Figma, operating our app is almost as simple as using an iPhone. If you try to click a part of the screen that isn't clickable, blue highlighted circles will appear to illuminate where you are able to click. There is a vertical scroll on the "accommodations", "unfinished celebrations", and "your cart" pages, but otherwise all interactions with our app are tapping. Navigation should be familiar and intuitive within the app, with a persistent shopping cart button which looks like a little gift box in the bottom right corner, and labeled backward and forward arrows in the top left and right corners respectively. To begin, select the prototype tab in the left pull-down, click the play button, and be off!

Hypothetical Situation: You are sending a gift to a loved one on their birthday. You wish you could be there in person, but decide to use YoursTruly to stimulate the experience as much as possible. That way, you can attach a video message and a special delivery add-on. You can also add a customized accommodation to accommodate the physical disabilities or other limitations of your loved one. With this in mind, please send a package with your desired specifications.

## Limitations

We encountered multiple limitations with how we could implement our prototype in Figma. First, our background confetti gif was not showing animation, even though other gif files were able to. This will be something we troubleshoot when it comes time to implement our hi-fi prototype. It is also impossible to access a user's camera roll or film and store a video on the

Figma platform, which is applicable to the first simple task. For all tasks, a user may need to input text, which is also not possible on this platform. We were unable to implement horizontal scrolling with our delivery add-ons, so we opted for a click-through navigation instead. When it comes time to checkout, we also cannot verify a user's address or payment method. Lastly, it would have required too many frames to implement a working memory to customize the screens according to each user and the choices they have made thus far in the app.

## Wizard-of-Oz Techniques

One task is to film a video, which would require advanced plugins and libraries. The same task also requires accessing camera roll, which is not possible on Figma for our med-fi. In our prototype, the user will choose a pre-selected video either filmed in app or uploaded. If we were to perform prototype testing, we would have the user record a video on their phone to get a feel for the length of time and effort needed to complete this task. We also cannot actually access a user's camera roll, so in our prototype, this is hard-coded. If we were to test, we would access the user's camera roll externally from the app and plug it into our prototype. Sending a package also requires inputting valid addresses and payment information, the verification of which are not features available on Figma. We could theoretically do this manually taking their address and venmo handle. Their address can be confirmed by typing them into maps to verify with the user that this is where they would like their package sent, and we could send a tester Venmo transaction to make sure their payment method is valid as well. For now, we hard-coded some filler address and payment information.

## Hard-Coded Items

The dynamic confetti background was hard-coded as a static image. For delivery add-ons, we choose a subset of options that may be offered to a specific user, rather than dynamically fetching which options were available at the user's location. For the video task, we hard-coded one of our own camera rolls from which to choose and upload videos. We also hard-coded a personal video to simulate the situation where a user takes a video of themself for the video message. For all tasks that require text input, we used our own pre-populated text to simulate the user entering things like custom add-ons, custom accommodations, etc. Lastly, we made educated guesses at the prices of different add-ons and accommodations to hard-code those into our prototype, rather than fetching them from the retailer and factoring in our labor costs. This also applies to the checkout page, where we hard-coded the input of the user.