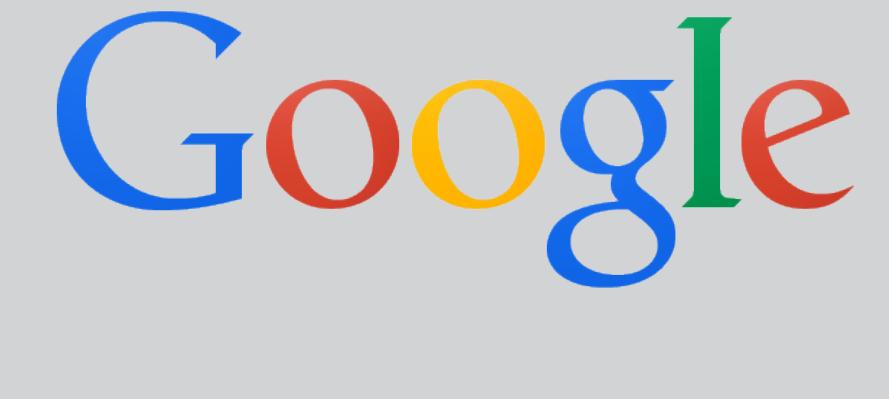
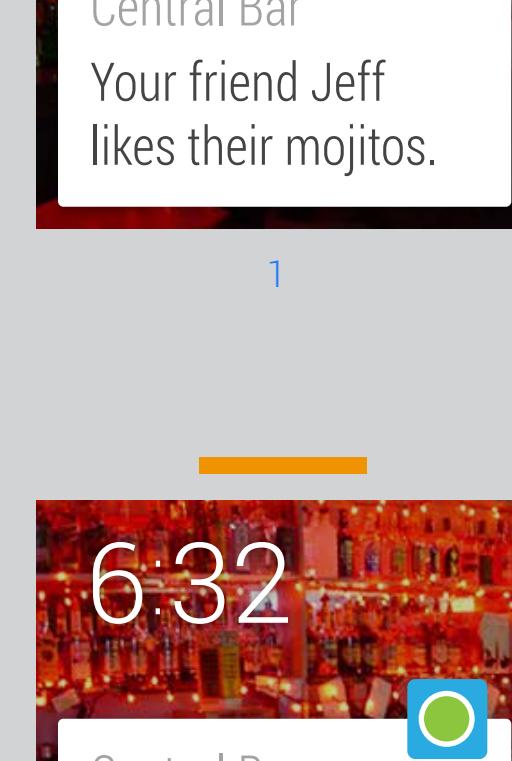


ANDROID WEAR NOTIFICATION AND APPLICATION PATTERNS

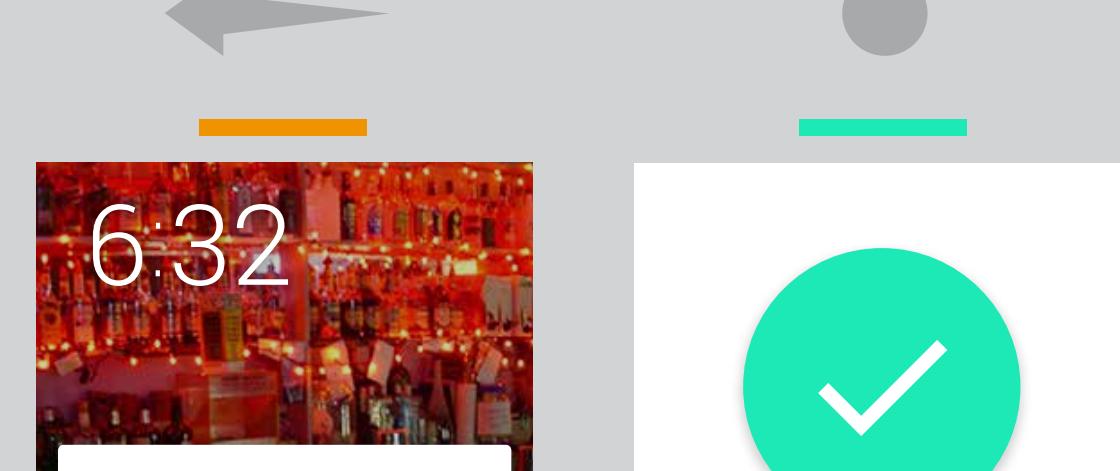


Notification Driven

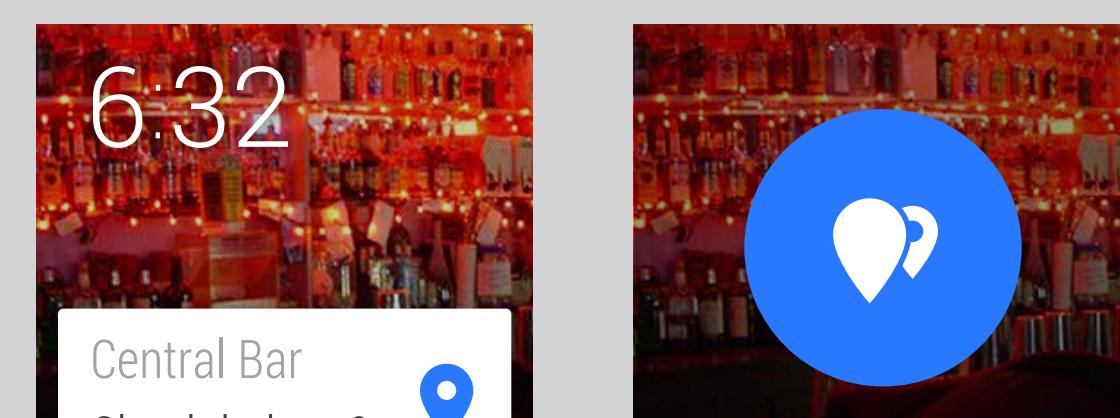
A
A location service could show some useful suggestions when the user enters a venue.



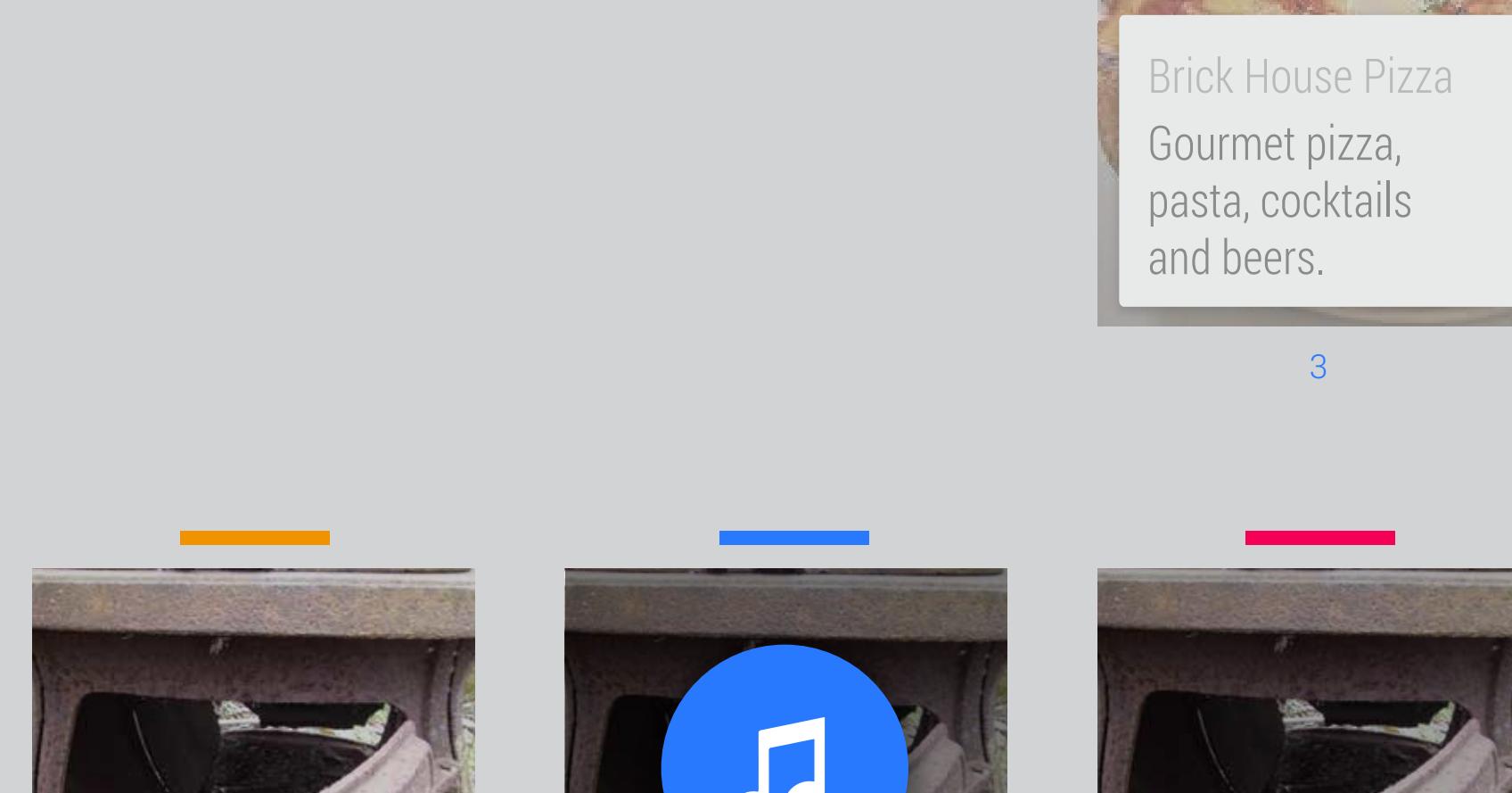
B
We can add an action button to allow the user to tap to take action. In this case to check in at the venue. After tapping, we show a confirmation animation to indicate success.



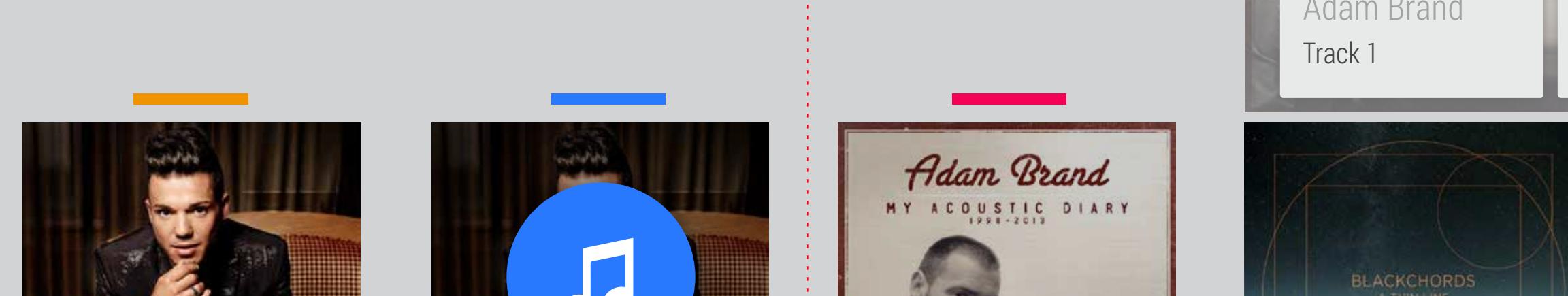
C
If the action is primary to the notification, we can place the action button right on the card.



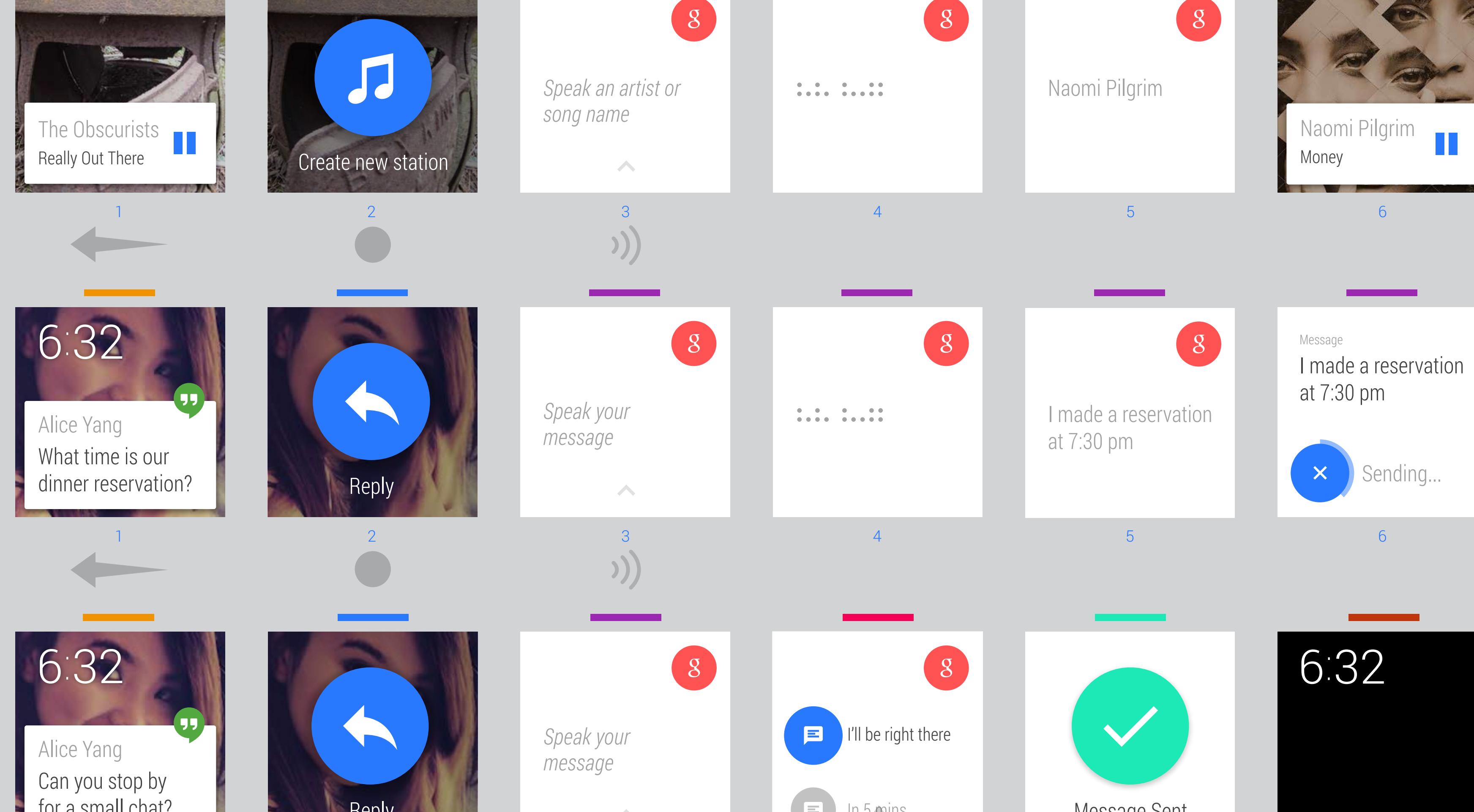
D
We can add the ability to choose a different location by launching a Picker widget when the user taps on an action button. The Picker temporarily takes over the screen and allows the user to choose from a list of results.



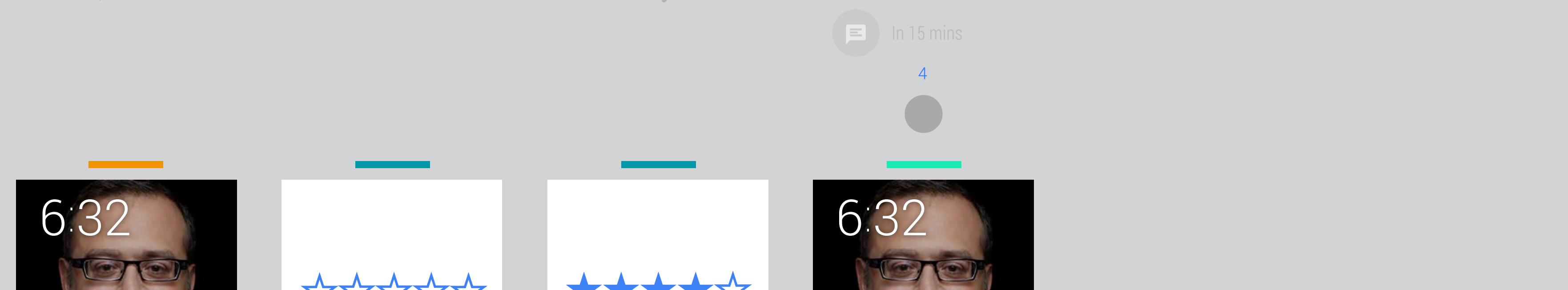
E
Performing an action or making a choice can also affect the original card. In this case, no explicit confirmation animation is needed.



F
The Picker can optionally include a second dimension, allowing users to swipe both vertically and horizontally. This allows for a matrix of options, such as songs for a set of albums.



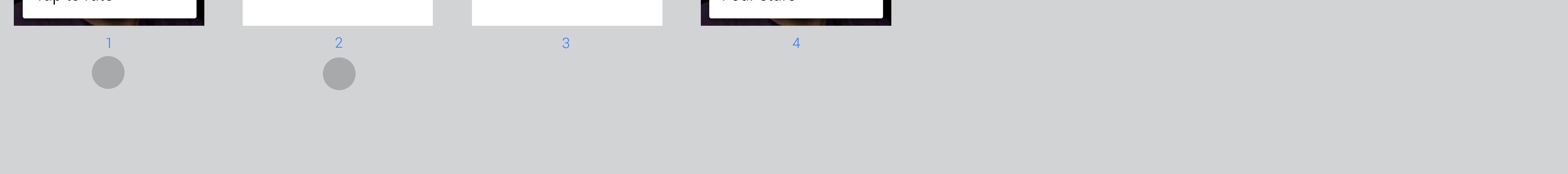
G
Where free text input is more appropriate than choosing from a list, a speech input UI can be introduced. Transcribed text is passed to your app to act upon, for example by playing a given artist.



All of these UIs can be achieved just by posting notifications. Easy, right?

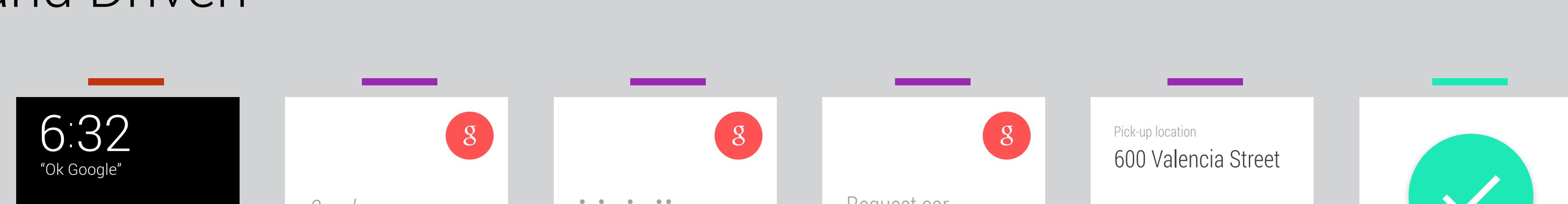
Congratulations, you've graduated to building full apps! They look and feels like part of the stream, the Picker is actually an app that gets overlaid on top of the main stream.

H
If the voice input is not immediately reflected in an updated card, a brief countdown screen should be shown that includes a structured breakdown of how the query was interpreted.



Google's services do the heavy lifting of converting user speech into text, and then that text gets sent to your app to process: you can use it to post a status update, make a reservation, send a message. In some cases, you might want to give the user an opportunity to confirm the transcription. We'll look at that in the next example.

I
A list of common options may also be presented along with speech input. The user can scroll through the list, tapping on the centered item to select it. Since the tap is an explicit act, no countdown screen is needed.



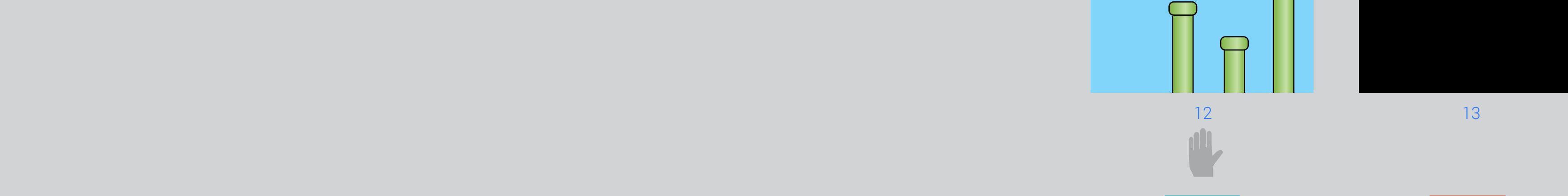
Countdowns allow the user to process transcriptions before they're processed. Use countdown screens for actions that are potentially costly if mistaken!

User Demand Driven

K
In this example a voice action is used to send a request to call a taxi. Since they may need time to process the request fully, the user is held indefinitely on a countdown screen. Instead, status of the request is posted as a card in the stream, with the card reporting itself with updated status when relevant.



L
Full-screen apps can be launched by touch or voice. In cases where a full screen app can't reasonably auto-quit on task completion, you should show an option to quit at logical checkpoints in the interaction flow, for example at the end of a level in a game.



All apps must reserve long press anywhere on the screen for displaying an option to quit.

At any time, covering the screen with a palm will quit any open app, return to the home state, and enter ambient mode.

Full screen apps that don't require swipe gestures for their interactions should also allow themselves to be quit by swiping them away.

M
Lists can also be invoked independently where choosing from a set of options is needed.

