

STEADY

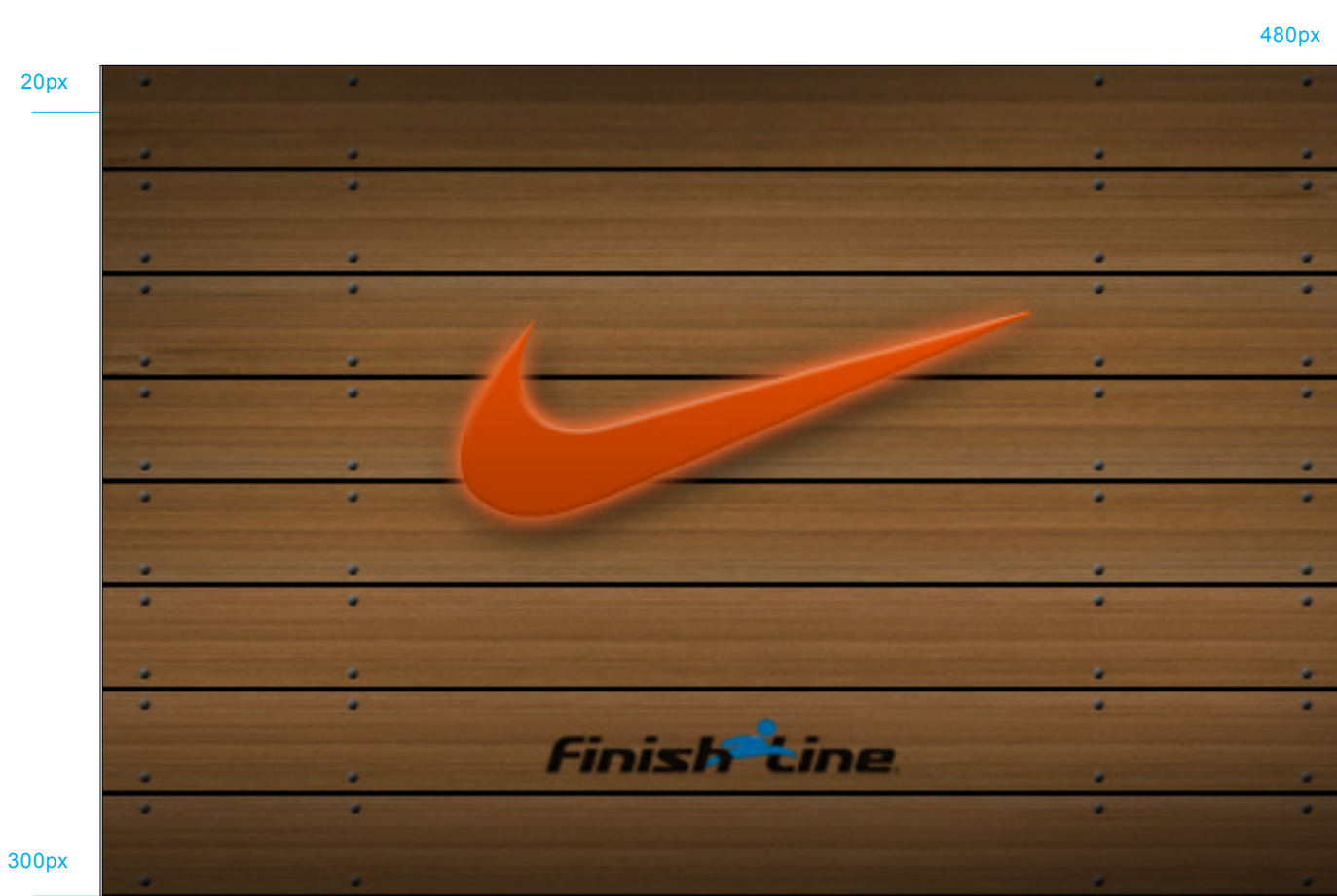
FINISHLINE IPHONE CUSTOMIZATION V5.0

MARCH 2010
Prepared for NIKE

TABLE OF CONTENTS—

INTRO	3
HOME	4
CUSTOMIZE	5
WALK THROUGH.....	6
ATHLETE SELECTION	7
T-SHIRT SIZE SELECTION.....	8
T-SHIRT COLOR SELECTION.....	9
CUSTOMIZATION (LOGO SELECTION).....	10
CUSTOM NAME / NUMBER.....	11
POSITION (LOGO OR NAME / NUMBER).....	12
A CAVEAT.....	13
LOGO COLORIZATION	14
VIEW T-SHIRT.....	15
FACEBOOK.....	16
FLICKR	17
TWITTER.....	18
EMAIL.....	19
SHARE COMPLETE.....	20
BUY	21
POST WALK THROUGH.....	22
HOME.....	23
ATHLETE SELECTION TAB.....	24
T-SHIRT SIZE SELECTION TAB.....	25
T-SHIRT COLOR SELECTION TAB.....	26
LOGO STATUS SCREEN (POSITIONS AVAILABLE) ...	27
LOGO STATUS SCREEN (FULL).....	28
LOGO COLORIZATION.....	29
LOGO ZOOM.....	30
REPOSITION LANDING.....	31
CUSTOM NAME / NUMBER	32
VIEW T-SHIRT / BUY.....	33
HELP CARD(S).....	34
VIDEOS	35
ABOUT	36
ASSOCIATED APPLICATION	37

INTRO—



A brief motion graphics introduction.

20px

480px

300px

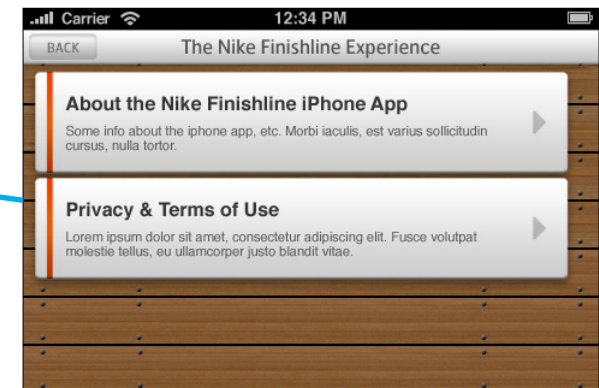
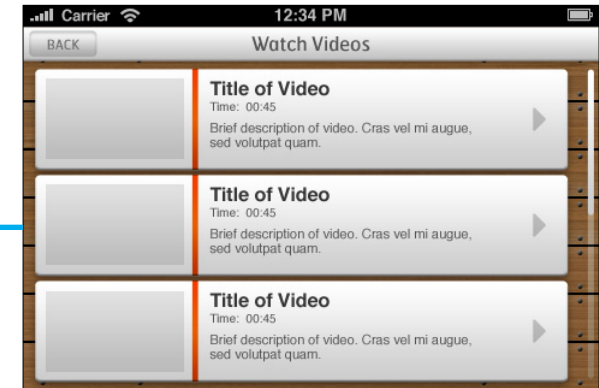
CUSTOMIZE YOUR OWN NIKE T-SHIRT

THE NIKE FINISHLINE EXPERIENCE

ABOUT / TERMS+PRIVACY
ADDITIONAL HELP

VIDEOS

iPhone App: Image/Copy FPO



CUSTOMIZE

CUSTOMIZATION:
WALK THROUGH

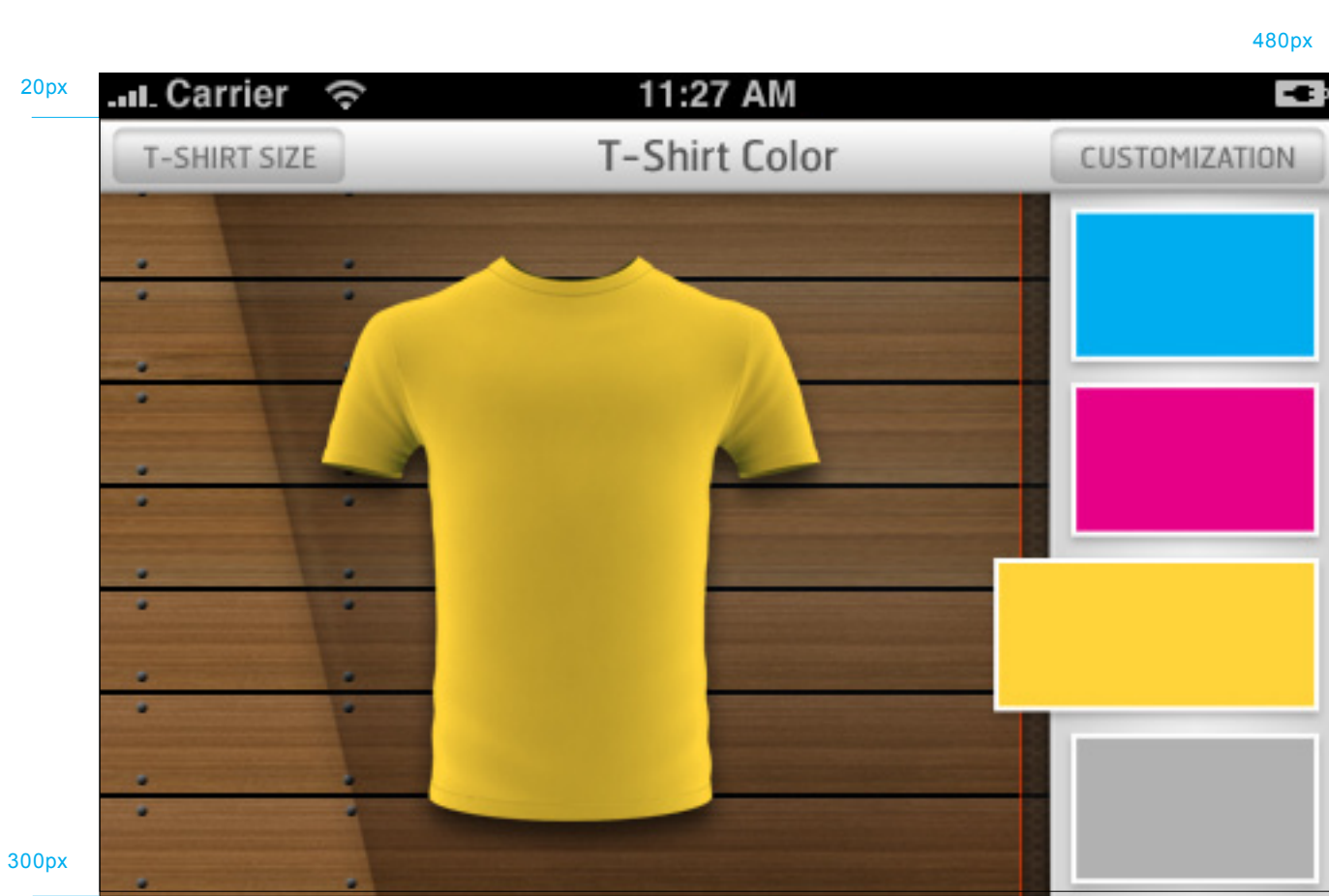
ATHLETE SELECTION—



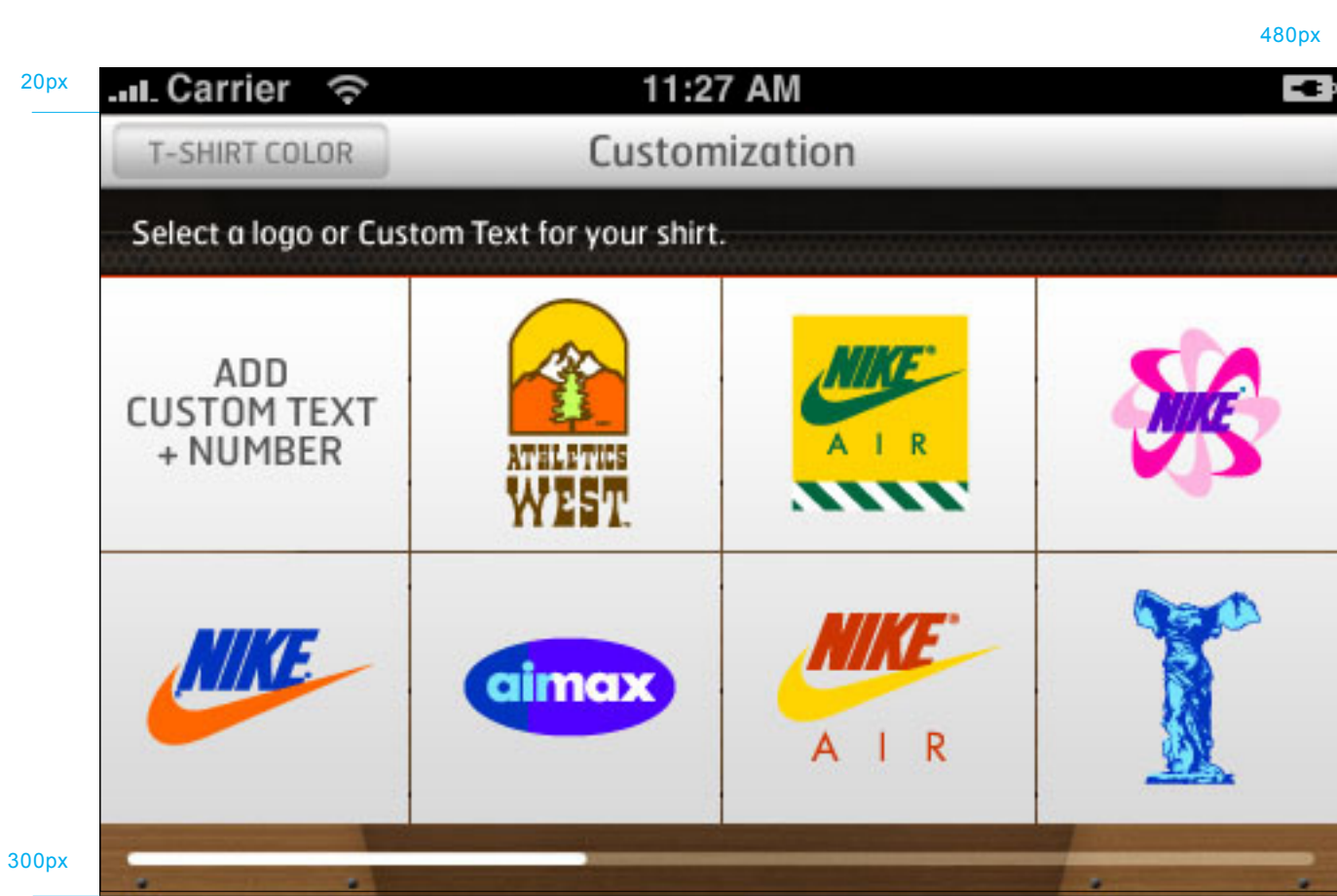
T-SHIRT SIZE SELECTION—



T-SHIRT COLOR SELECTION—

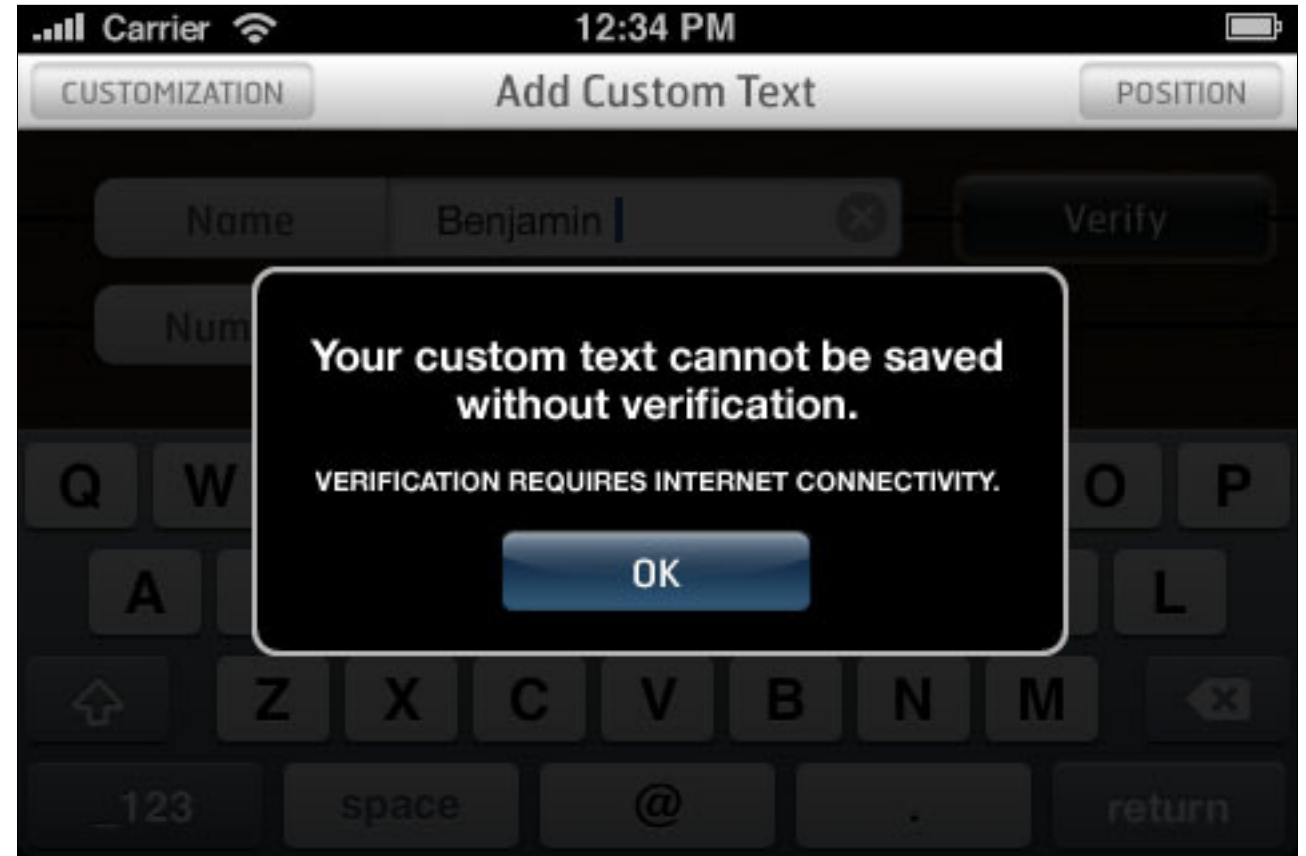
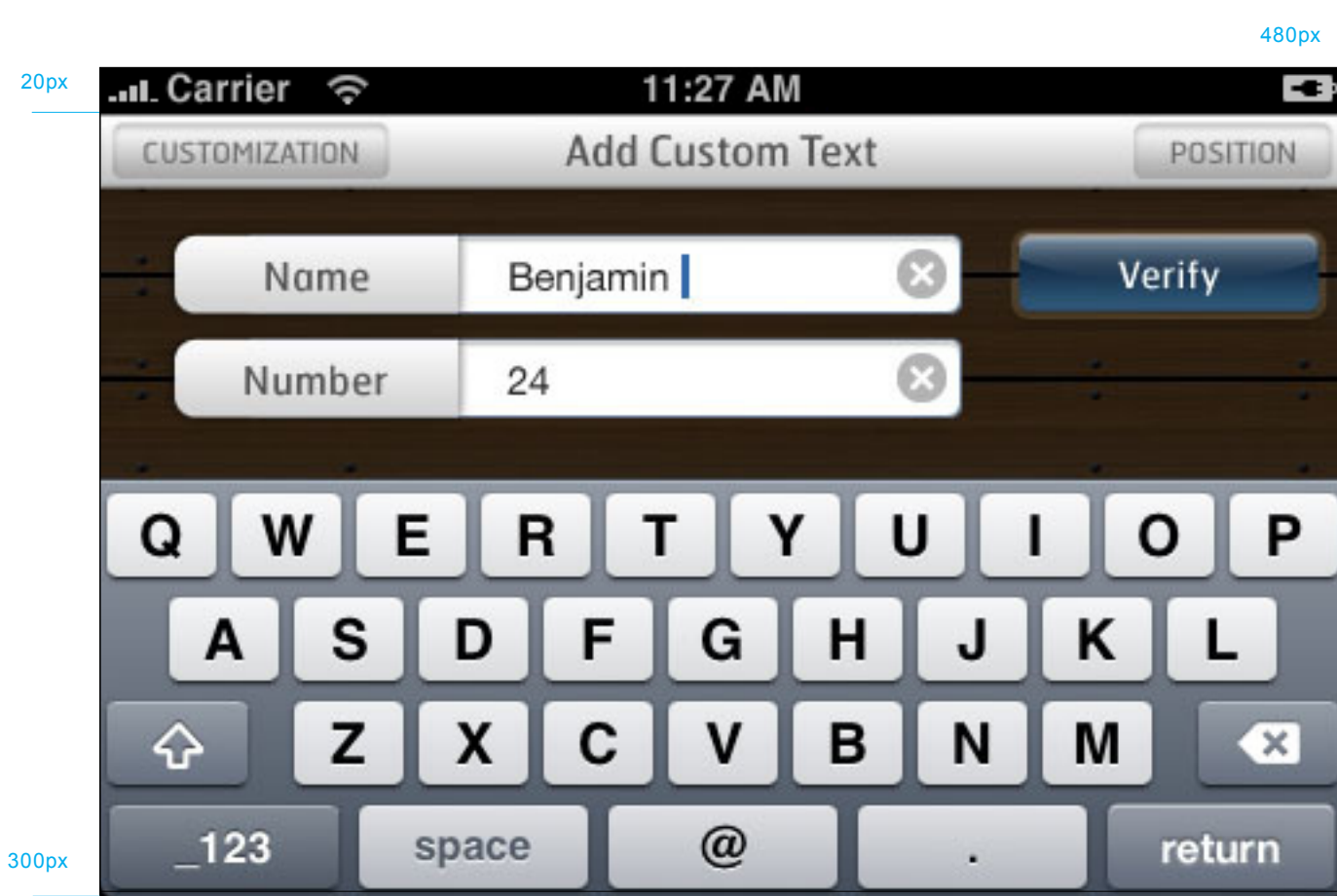


CUSTOMIZATION (LOGO SELECTION) —



ARROWS FADE AWAY AFTER A MOMENT

CUSTOM NAME / NUMBER—



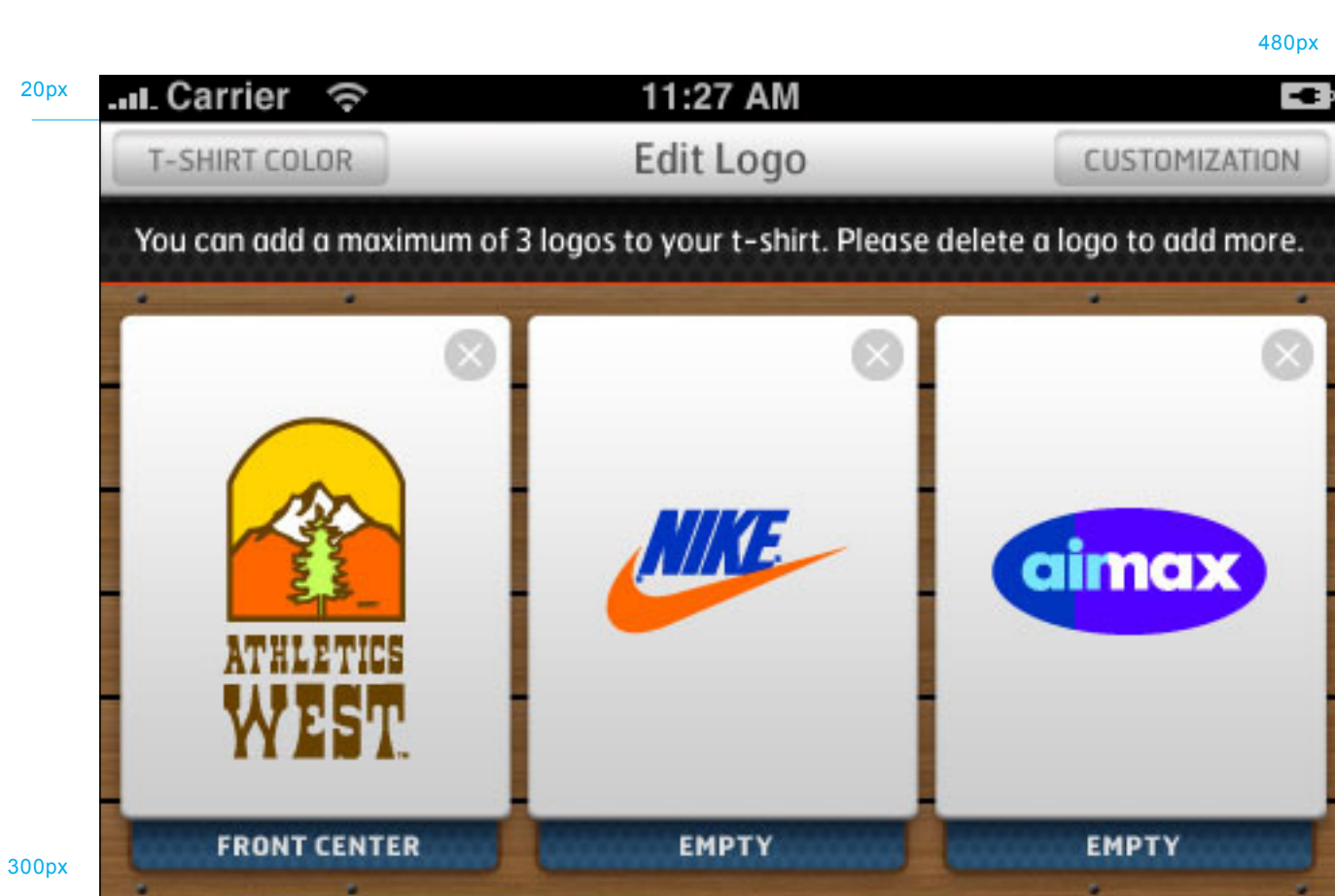
IF USER DOES NOT VERIFY BEFORE LEAVING THIS SCREEN, WE PRESENT AN ALERT.

LOGO OR NAME / NUMBER POSITION—



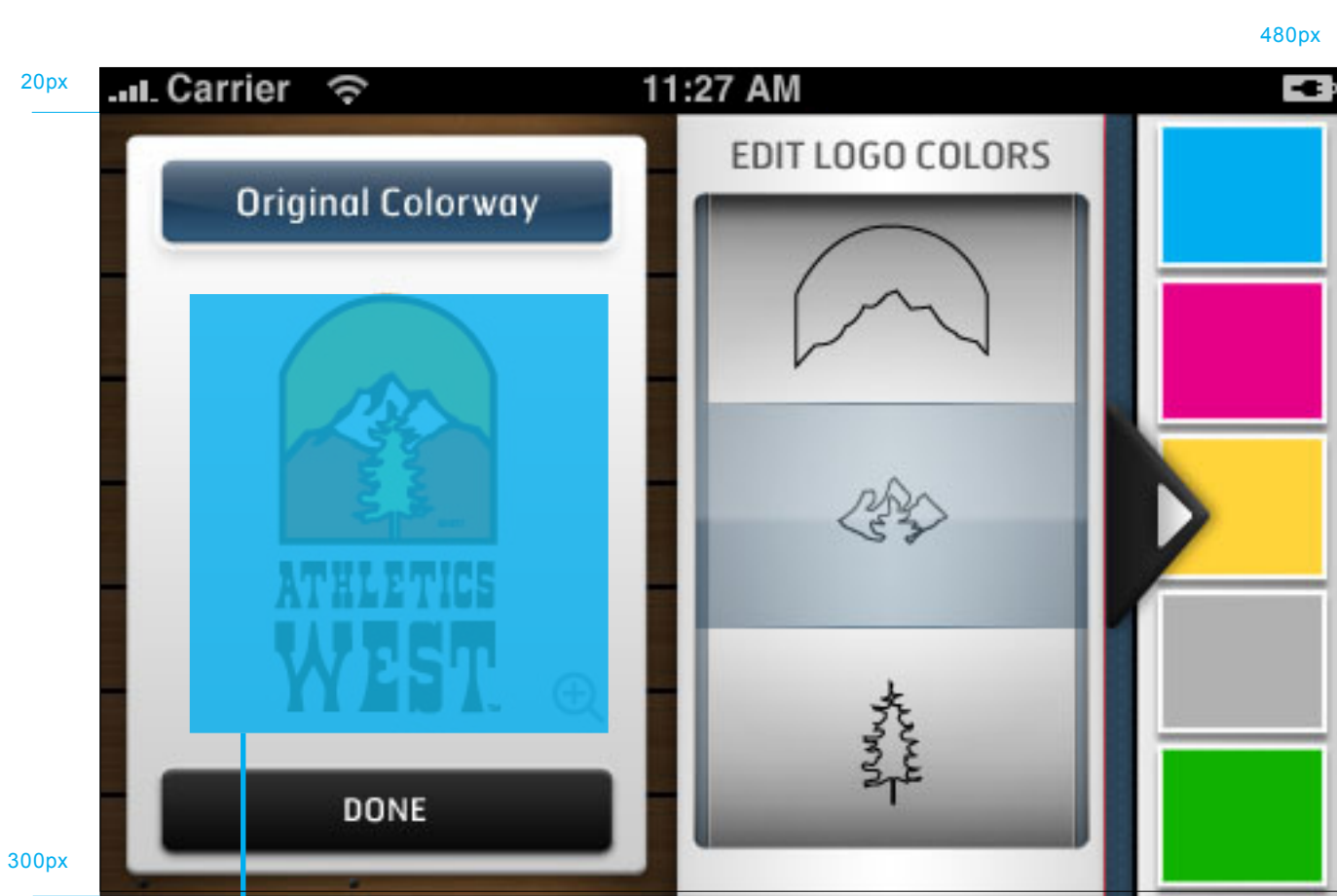
The first time a user arrives at the position screen, T-shirt does a quick 180 to demo spin functionality.

Once the logo is placed, the targets disappear. If the user taps the shirt or logo again, the targets re-appear.

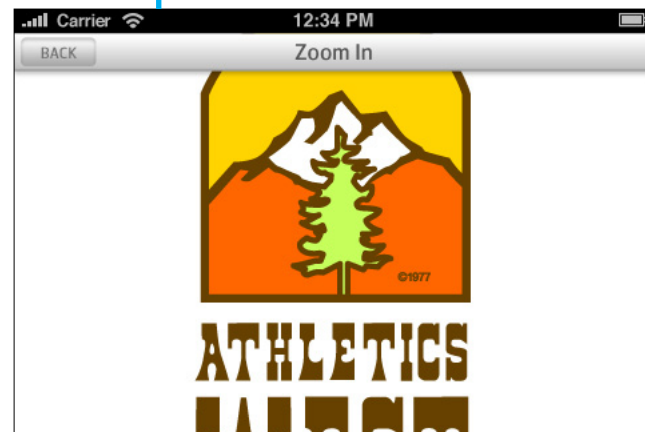
A CAVEAT—

If a user goes back to place another logo before colorizing, they first see the Edit Logo screen before reaching the logo wall. This would prevent the scenario in which a user could select more than 3 customizations.

EDIT COLOR—



Tapping Done completes the walk through and activates the tab bar and "standard" mode.

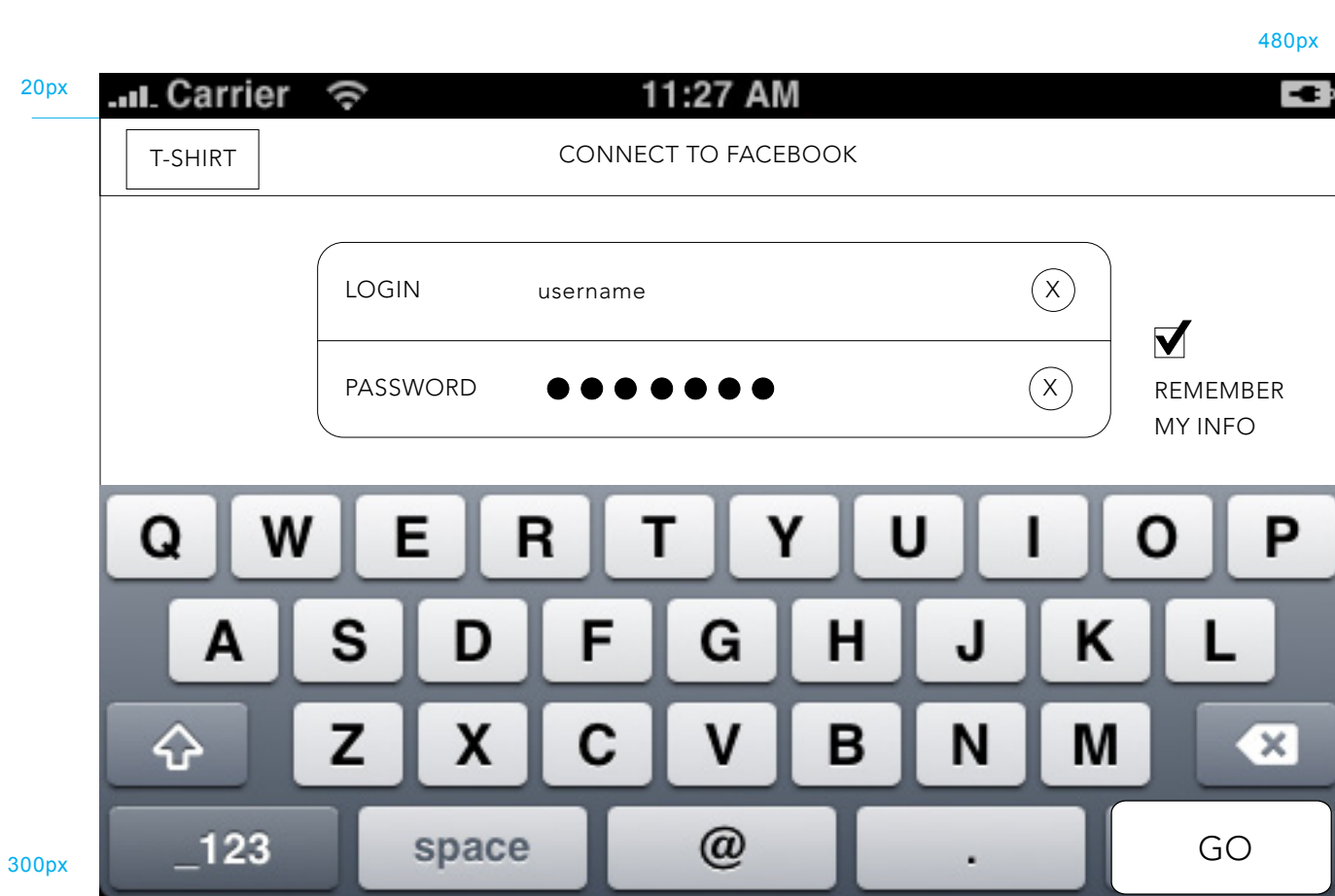


VIEW T-SHIRT—

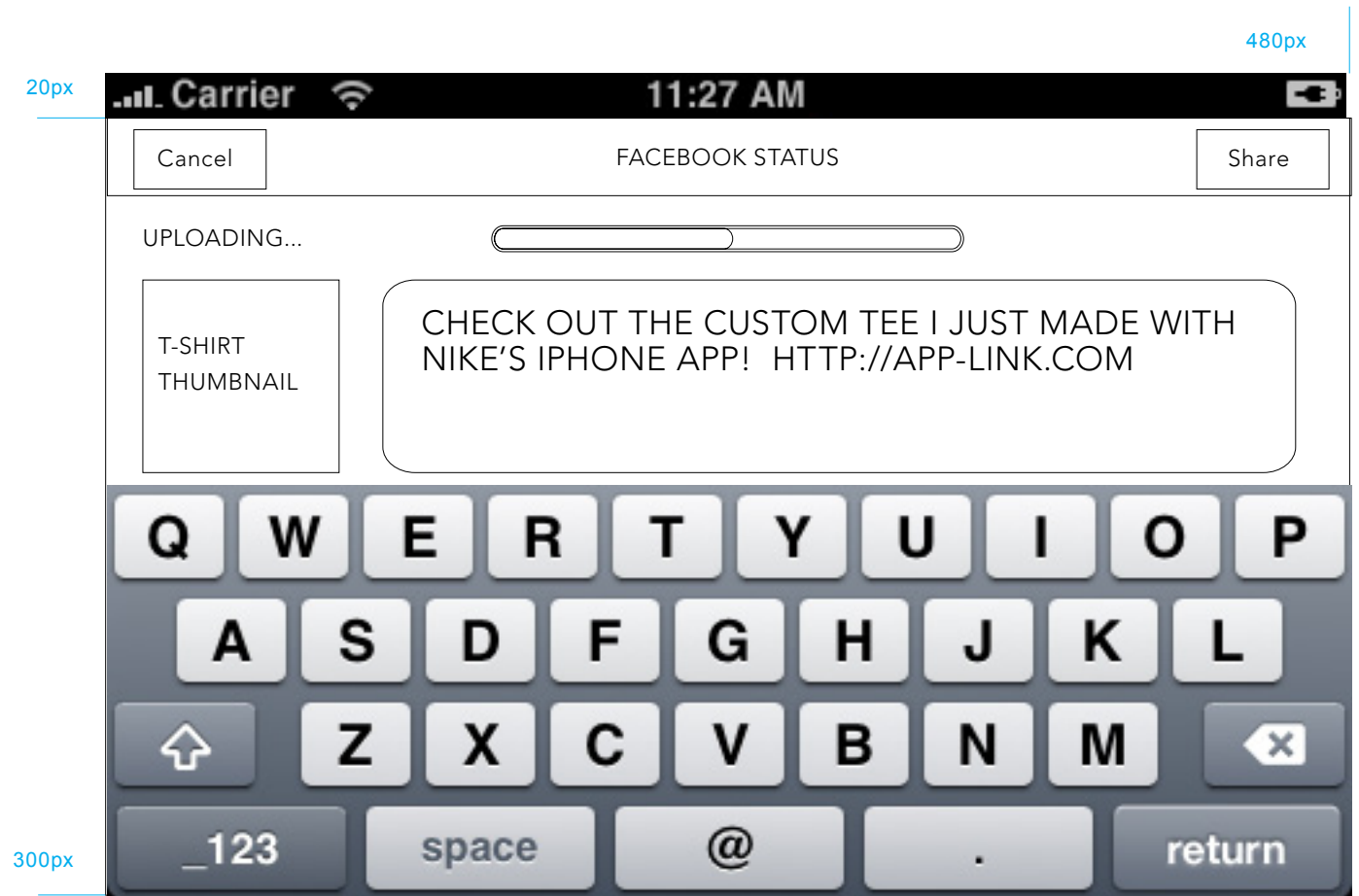


FACEBOOK—

Using the Facebook Connect SDK the user posts photos to their status feed after logging in. The login process is a simple webview popup or we could incorporate our own login UI if needed. To do this we need an API Key/Secret and an app on the Facebook developers site. -Bruno

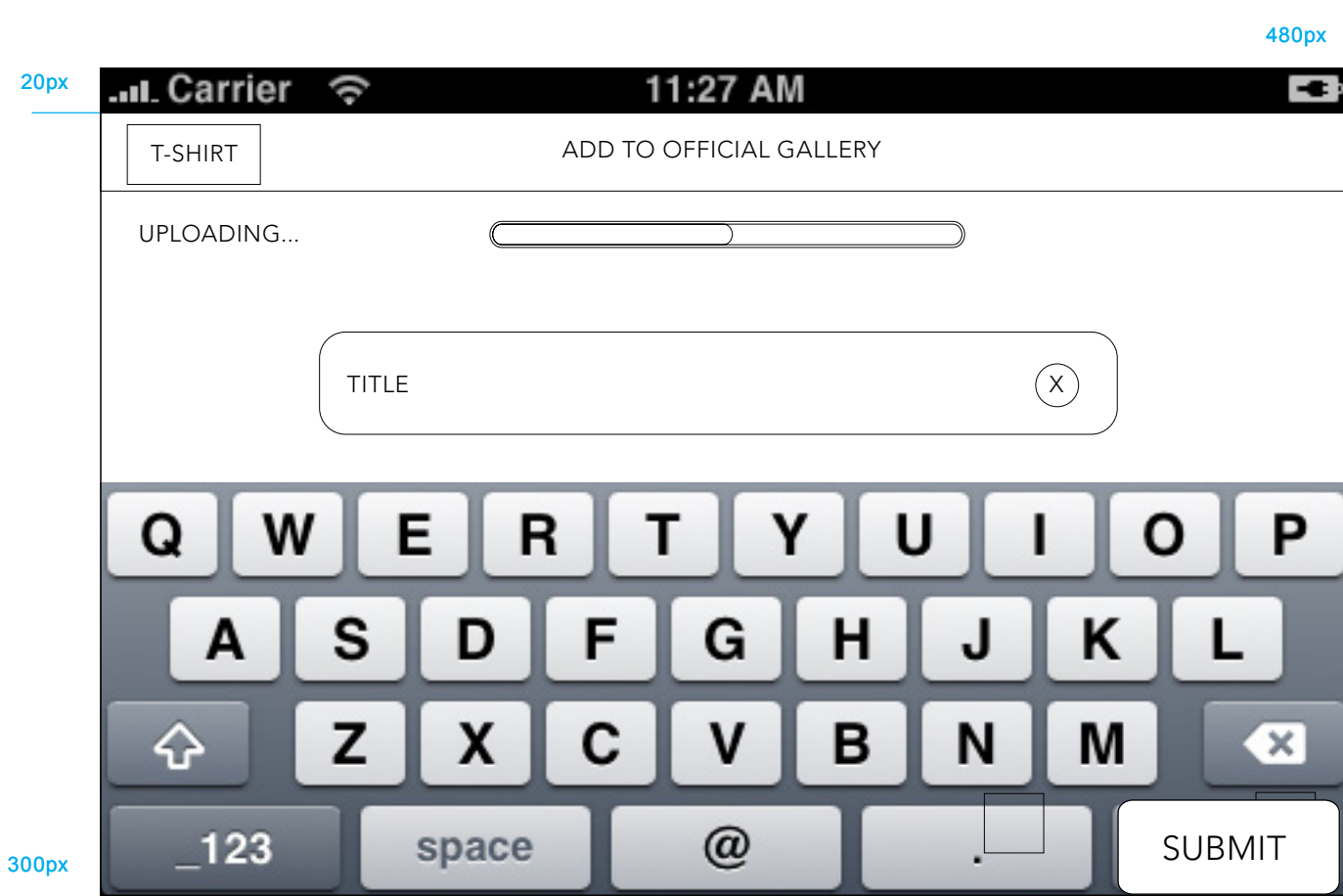


Users login to their Facebook account. We save info by default, but users could opt out.



Once logged in, the image begins to upload with progress noted (text or progress bar. The status field is pre-populated with text and a link to download the app.

FLICKR—



For flickr the options would be to allow users to upload their shirt designs to a Public pool on the Nike flickr account or their own personal flickr accounts. Ben and I were talking about the possibility of a single Nike Flickr account in which all the images were uploaded to. With this approach it would be easier to keep track of all the images in a single gallery for later downloading to the device.

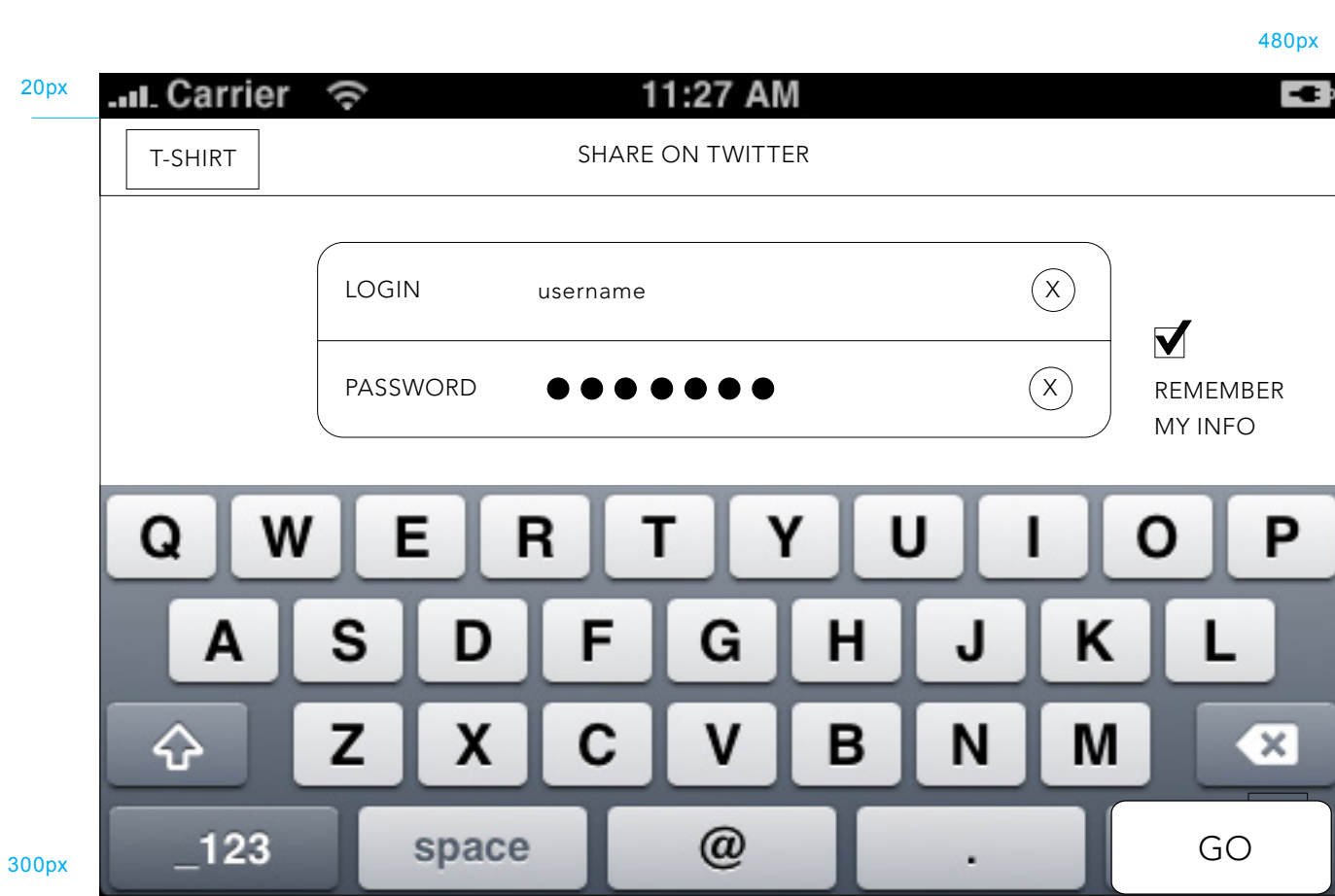
There are two possible approaches for uploading images to flickr from the device. The first would be to obtain a Flickr API/Key and create flickr application (through the flickr account). The experience on the device would direct the user to the web in order to sign-in and pair their device udid with flickr. After the pairing photos can be uploaded normally and any of the Flickr API calls can be used.

Flickr also supports email uploads. In this case we can create the emails in the background and upload to the Flickr pool without any login process. Because this approach does not pair the device udid with flickr, we wouldn't be able to use any of the Flickr API calls, unless we did the sign in/pair process. In both of these cases you would probably need a flickr pro account. Currently there is no official SDK - We can parse the images with a JSON framework or use a framework called objectiveflickr <http://code.google.com/p/objectiveflickr/>, which can handle most of the flickr API.

-Bruno

TWITTER

For Twitter we also need an API Key/Secret. For the content we can generate a bitly like url and link to the image in the Flickr gallery, TwitPic or another server. Since we are just posting tweets, we probably don't want to incorporate a separate framework - unless we want to start receiving tweets for parsing. I have a class written for sending a tweet from the device with a hardcoded Username/Password. We can create a UI for this too, maybe something similar to the Facebook one. - Bruno

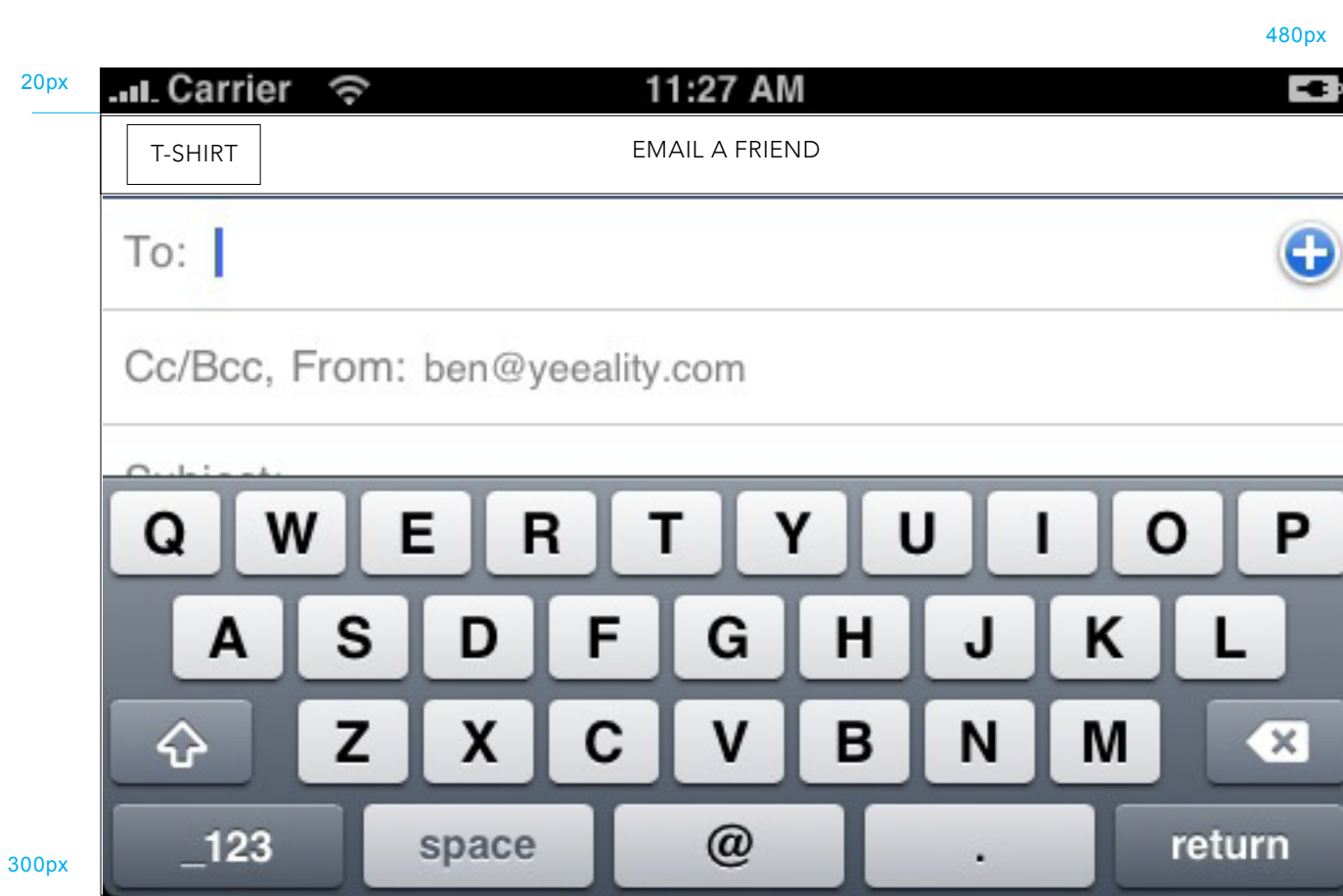


Users login to their Facebook account. We save info by default, but users could opt out.

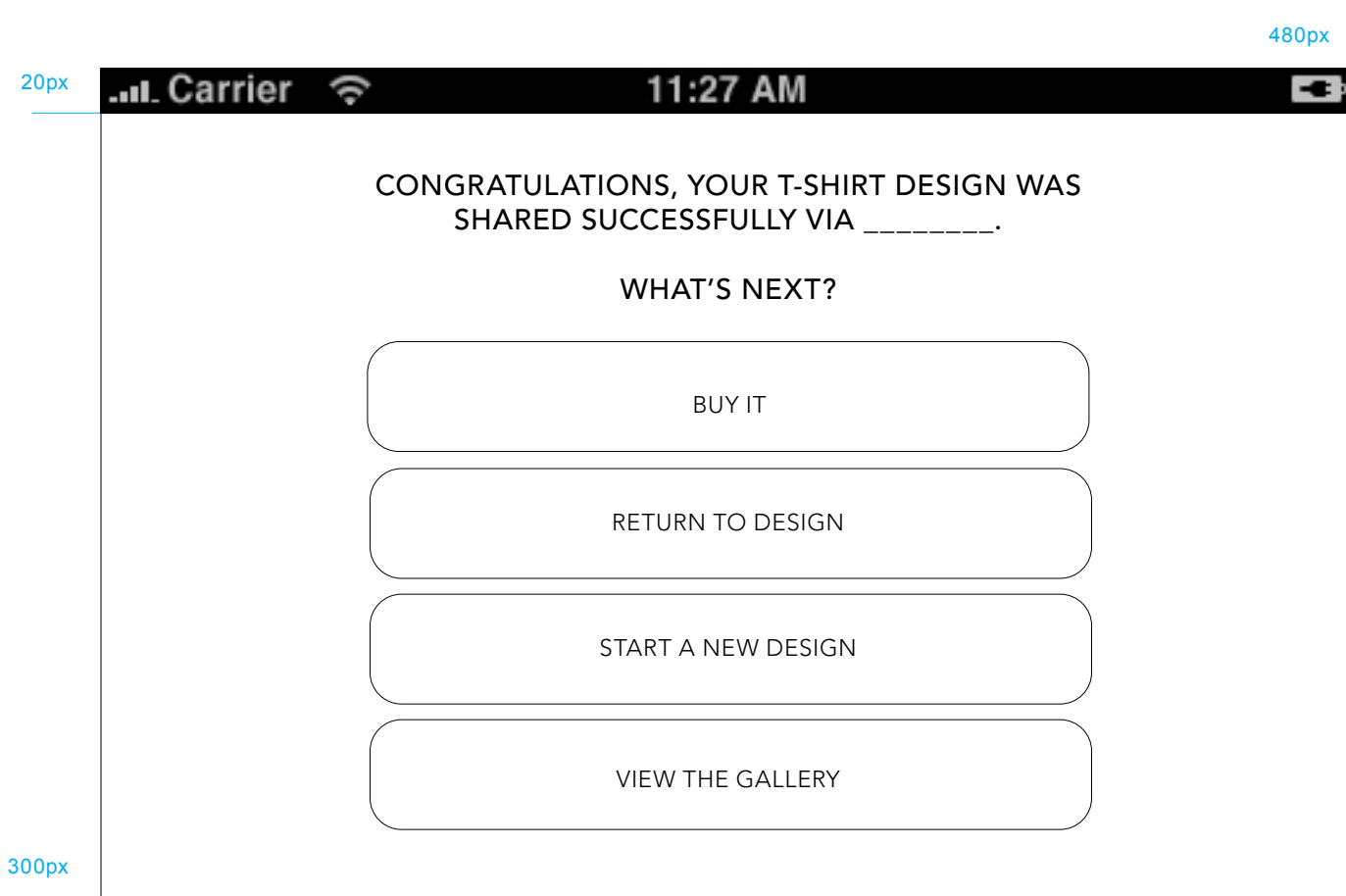


Once logged in, the image begins to upload with progress noted (text or progress bar. The status field is pre-populated with text and a link to download the app.

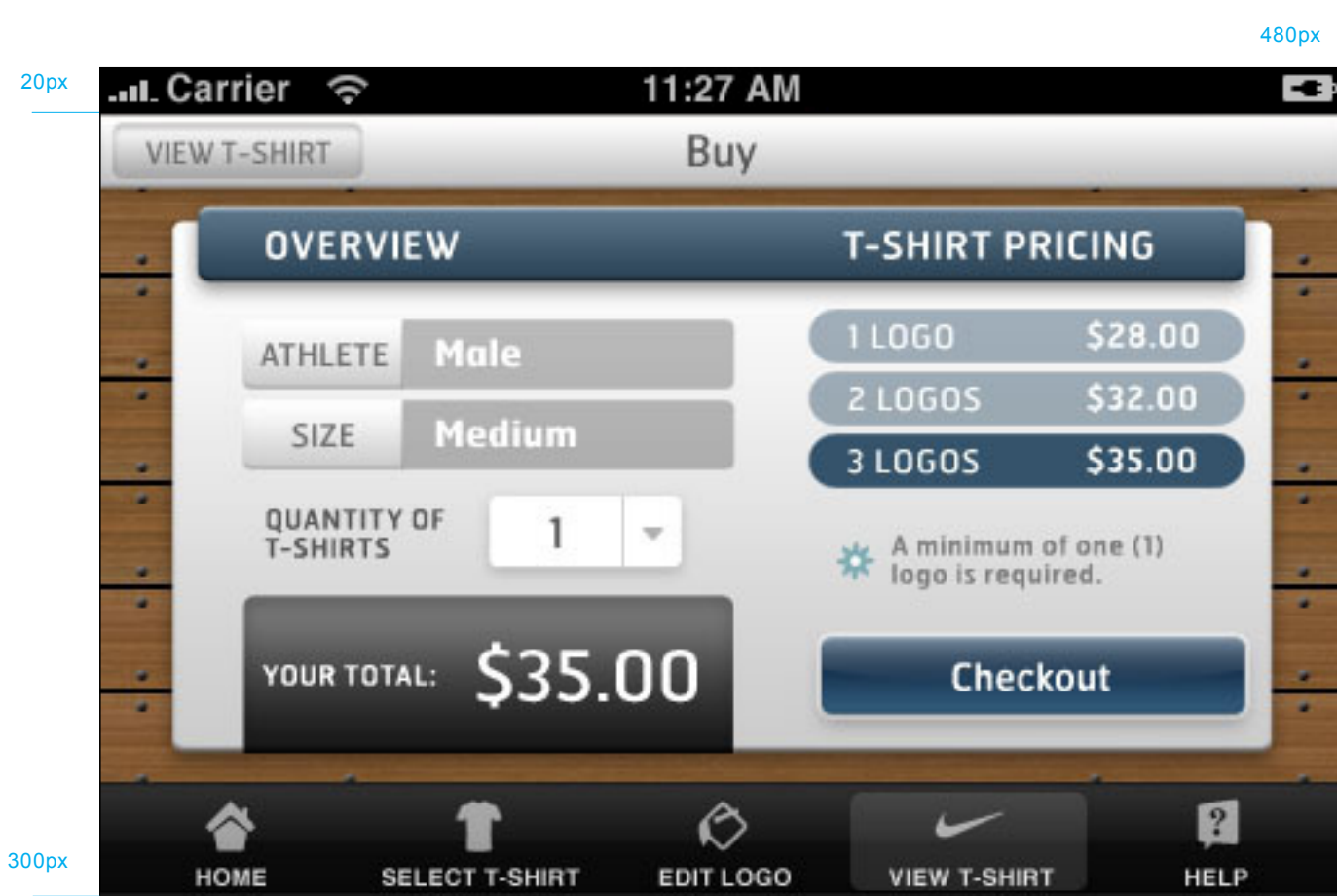
EMAIL—



SHARE COMPLETE—

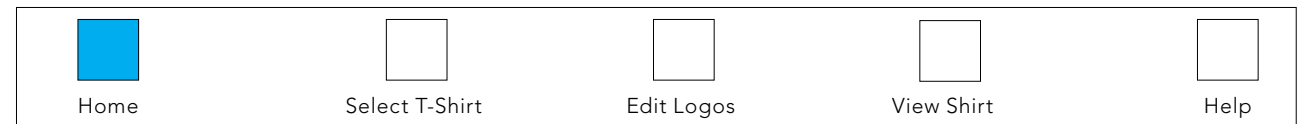


BUY—



CUSTOMIZATION:
POST WALK THROUGH

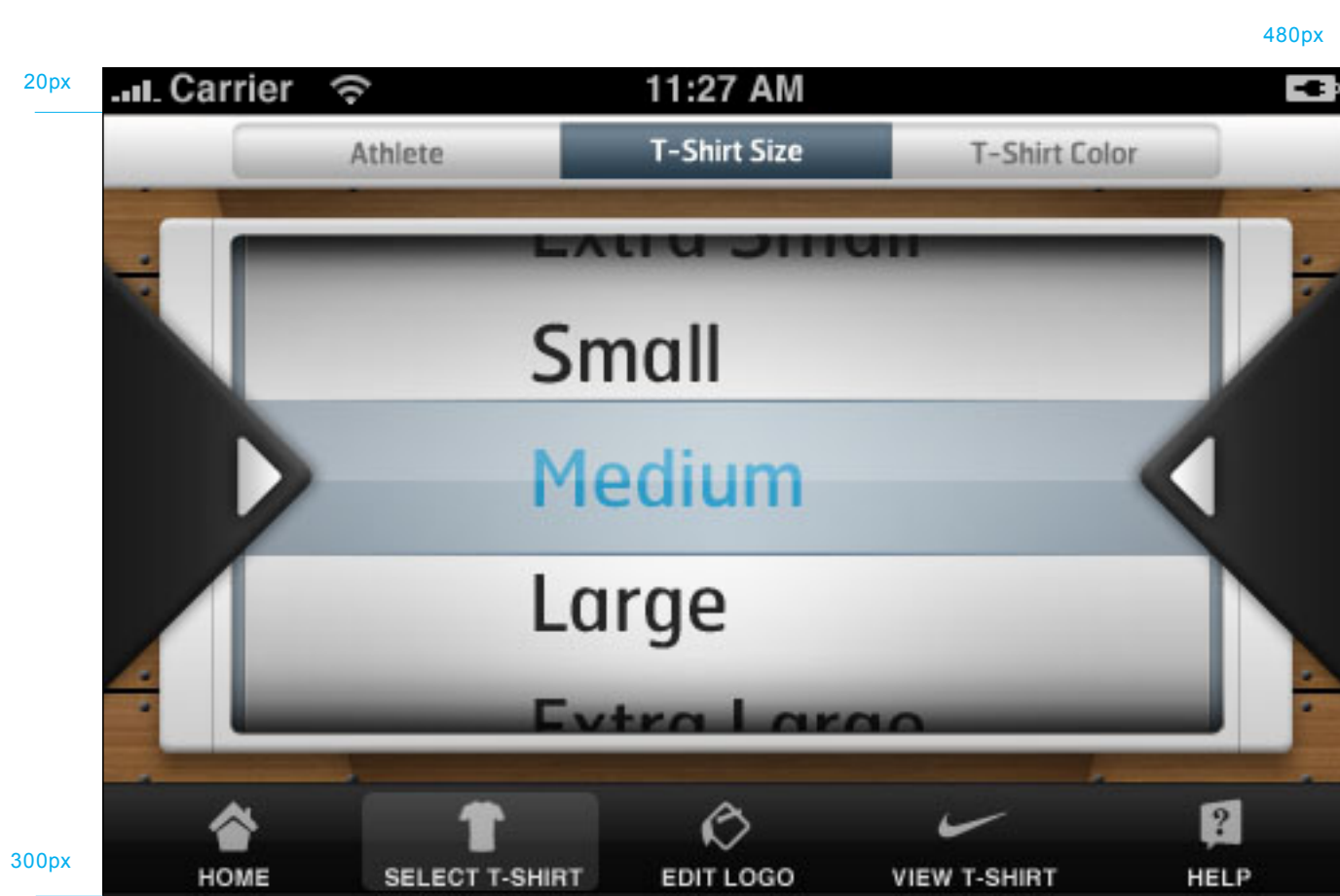
HOME—



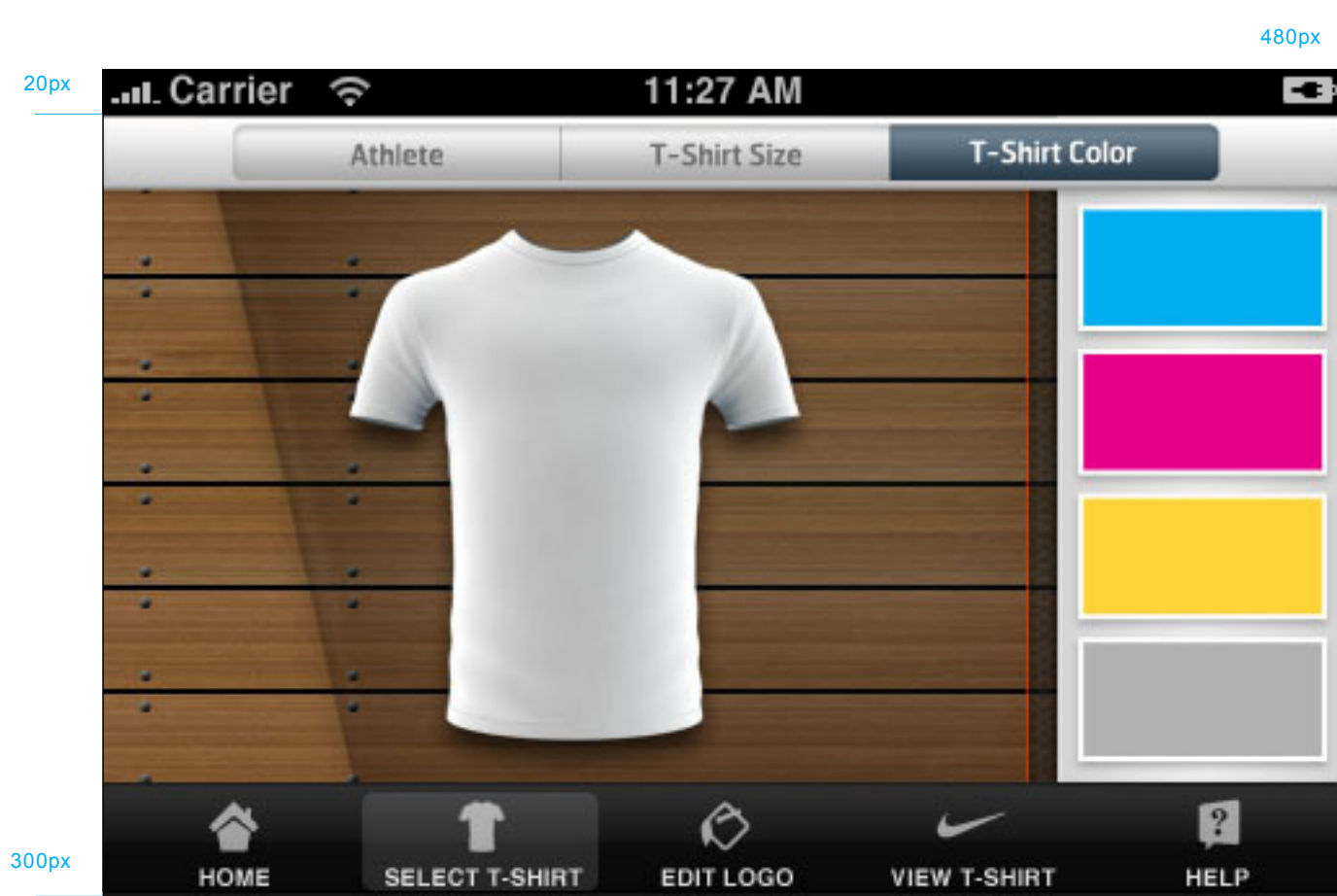
ATHLETE SELECTION TAB—



T-SHIRT SIZE SELECTION TAB—



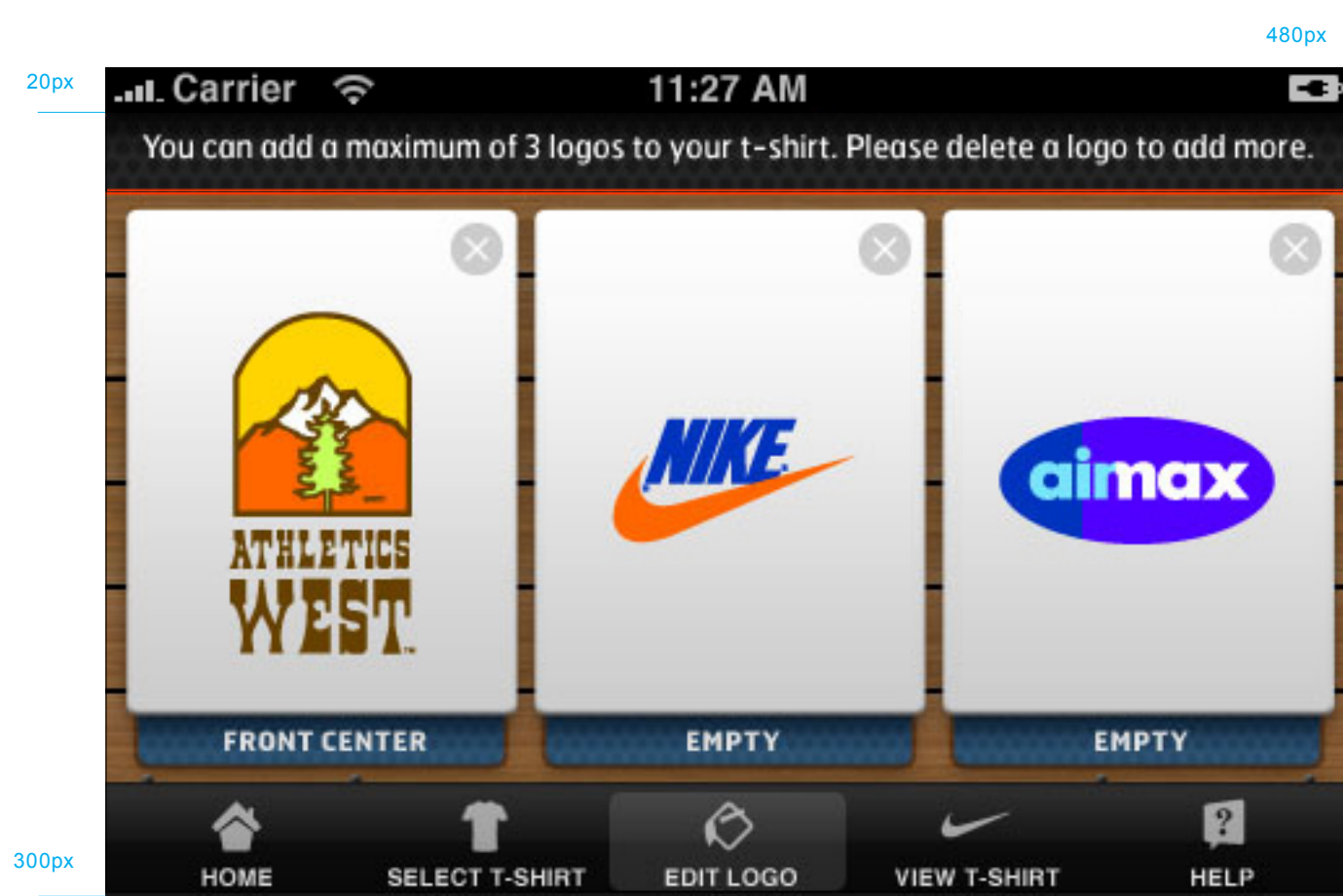
T-SHIRT COLOR SELECTION TAB—



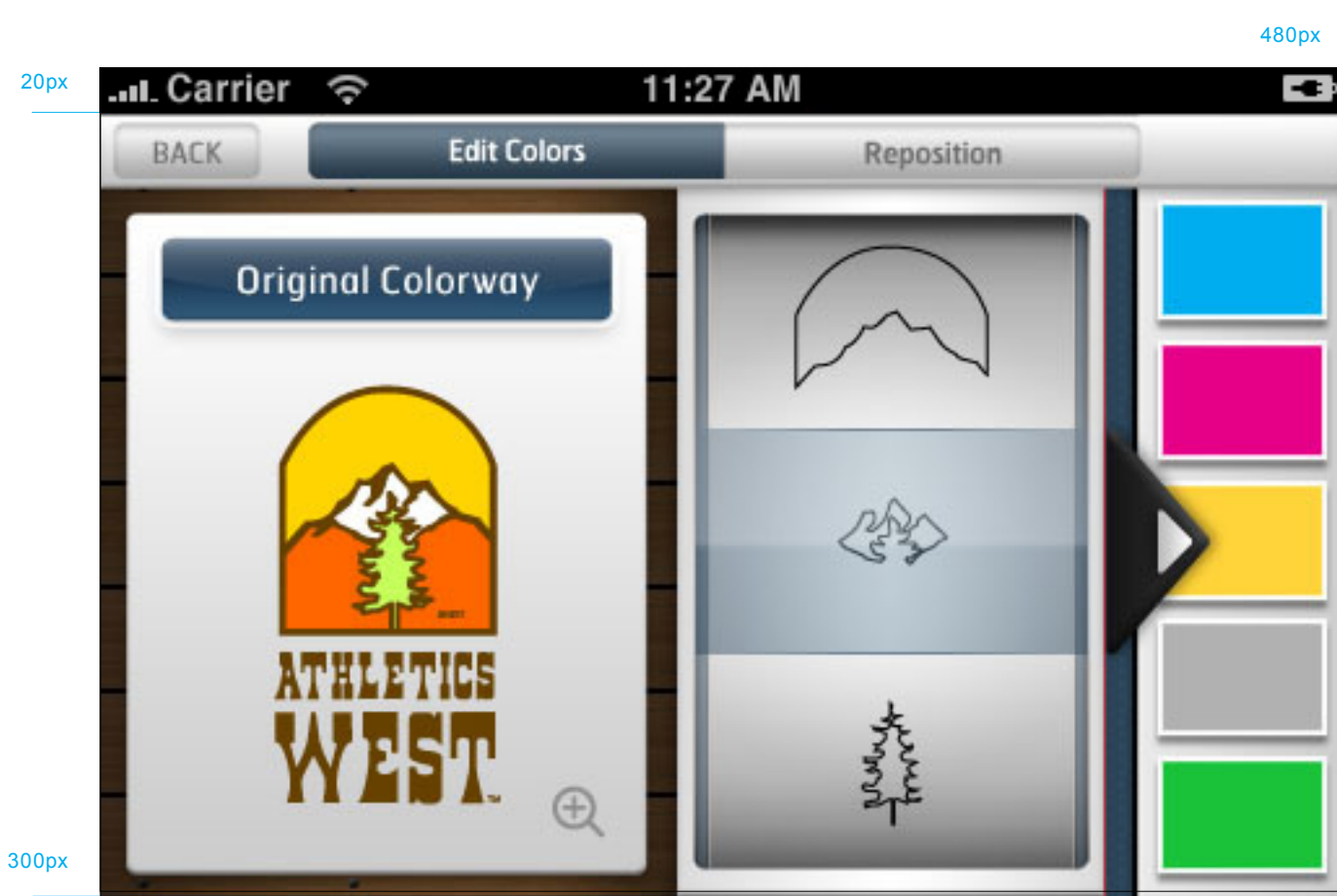
LOGO STATUS SCREEN (POSITIONS AVAILABLE)—



LOGO STATUS SCREEN (FULL)—



LOGO COLORIZATION—



LOGO ZOOM—



REPOSITION LANDING—

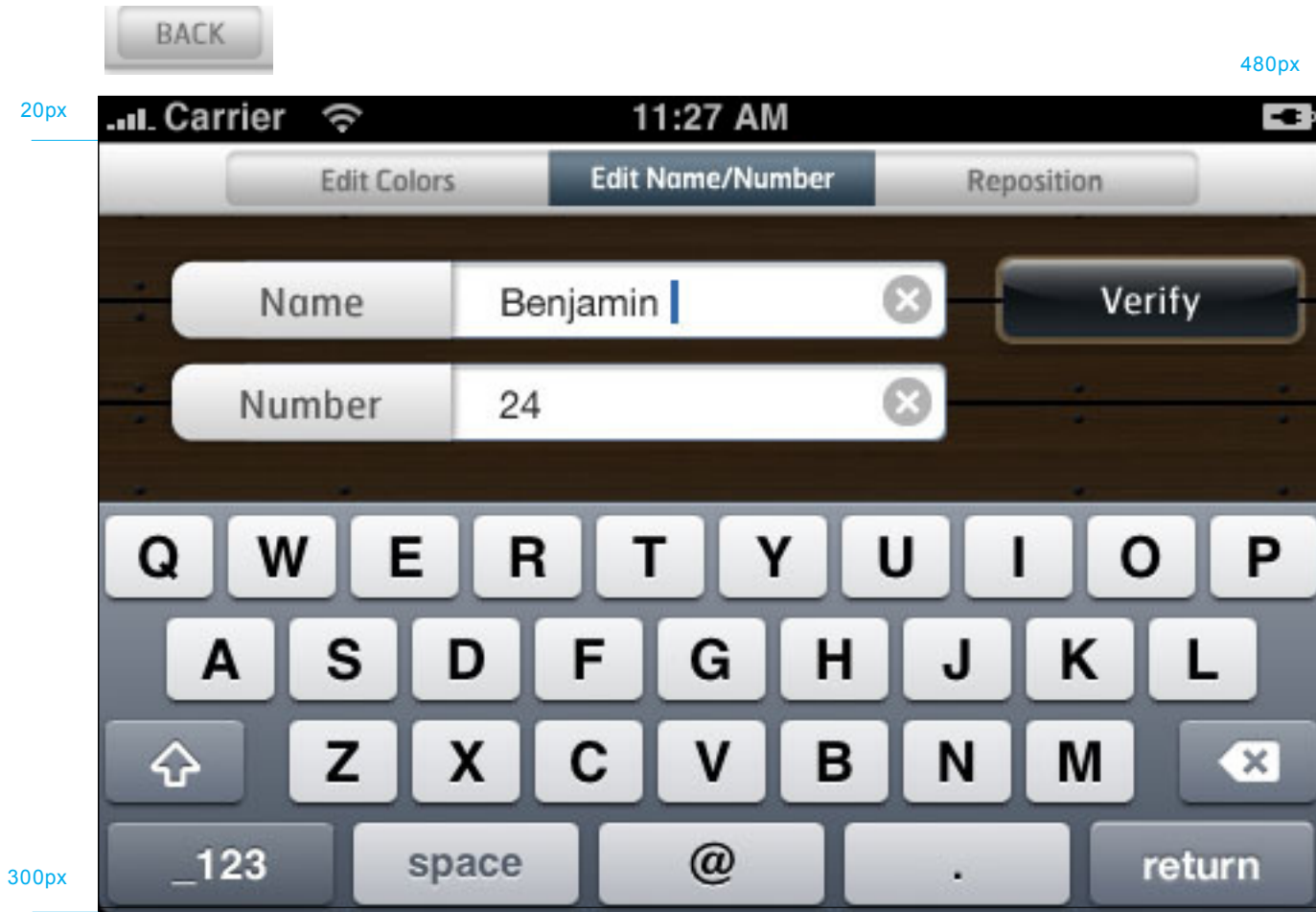


Need to add Back button.

The first time a user arrives at the position screen, T-shirt does a quick 180 to demo spin functionality.

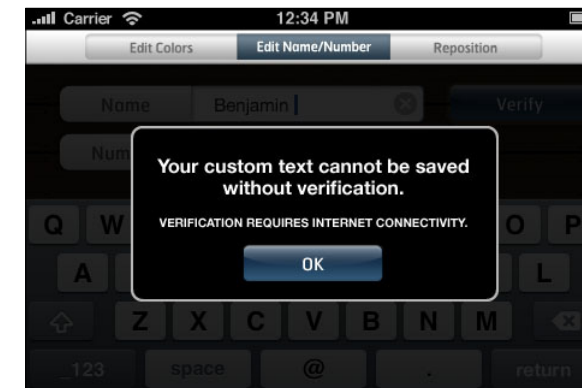
Once the logo is placed, the targets disappear. If the user taps the shirt or logo again, the targets re-appear.

CUSTOM NAME / NUMBER—

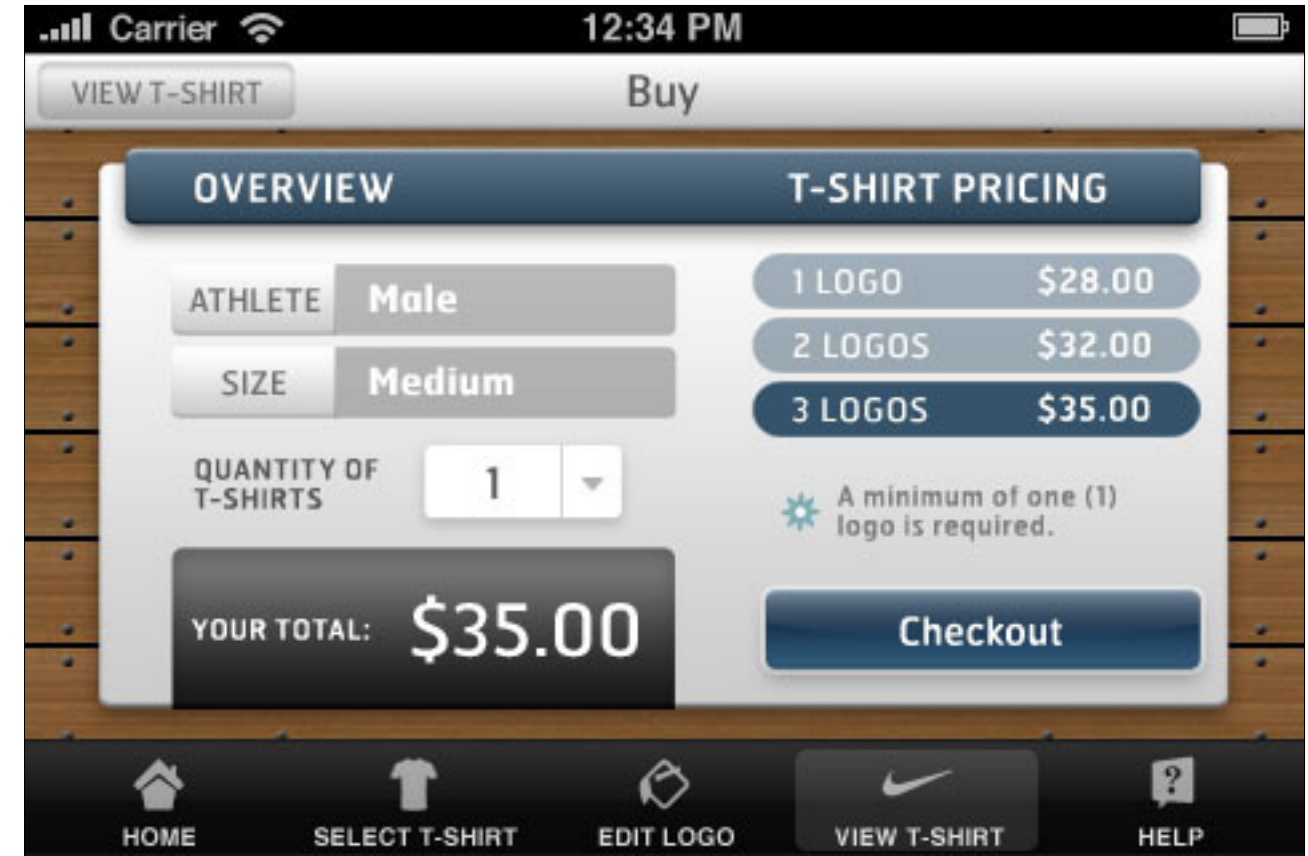


Need to add Back button.

If user does not verify before leaving this screen, we present an alert.



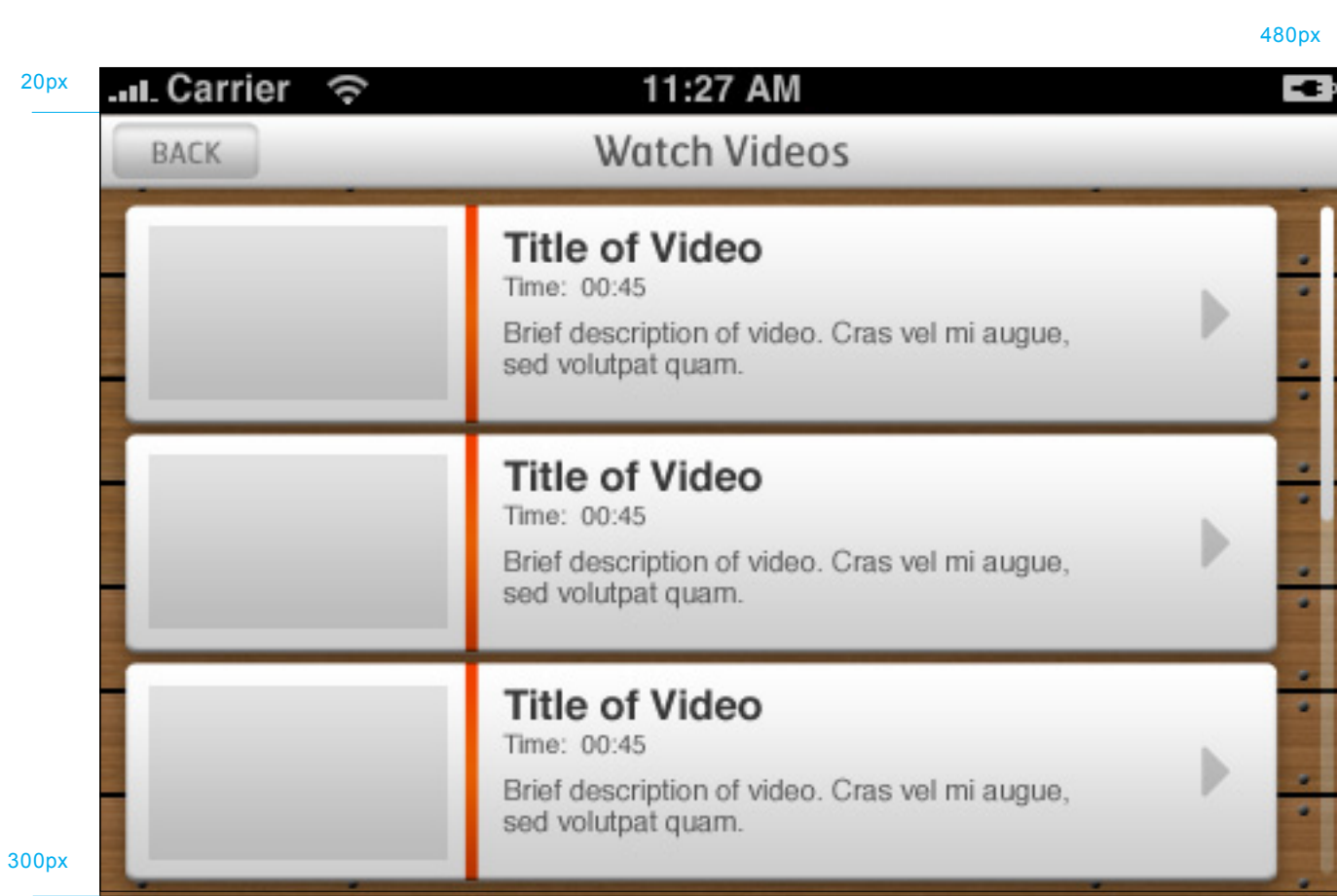
[VIEW T-SHIRT / BUY](#)



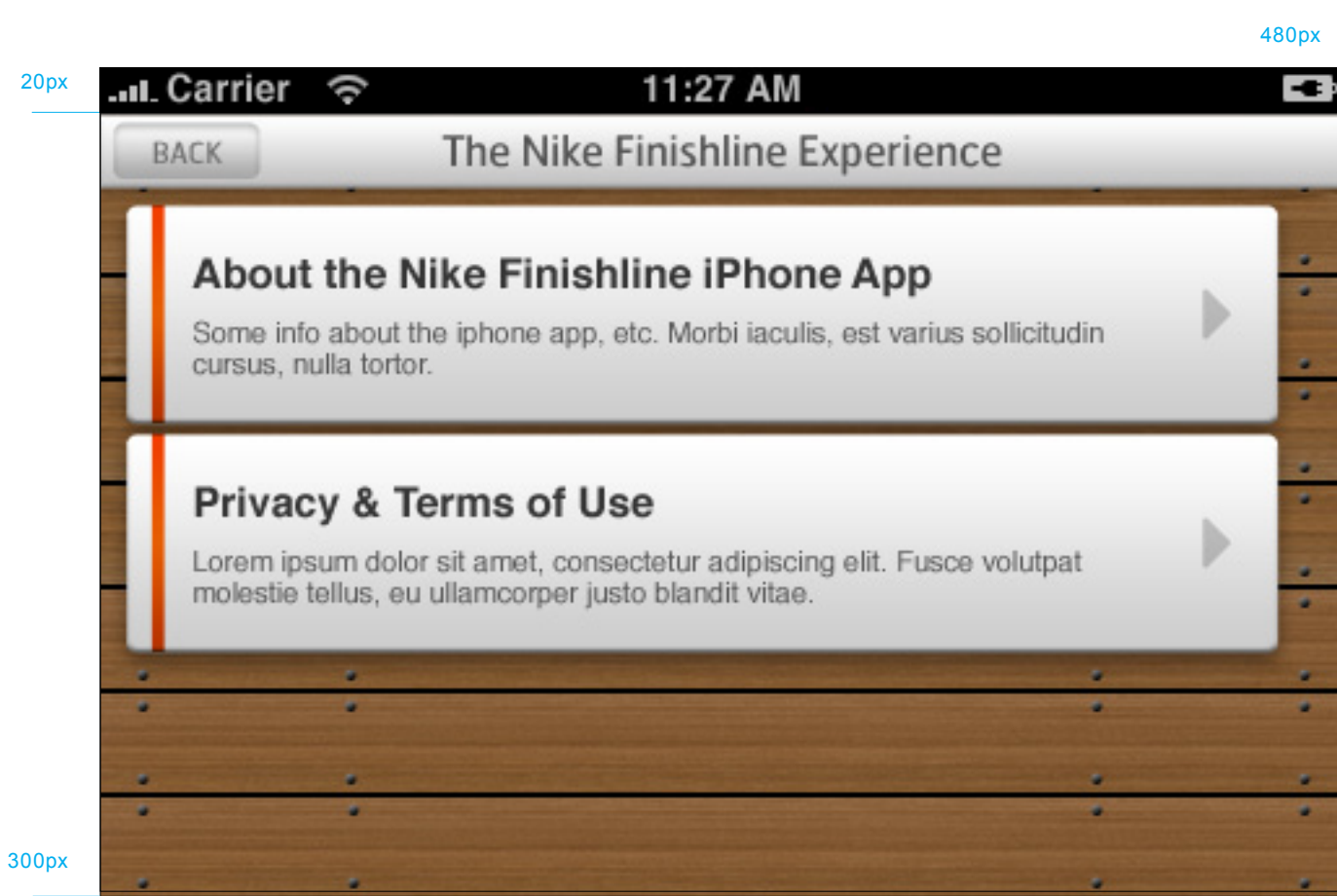
HELP CARD(S) —



VIDEO MENU (ACCESSED FROM HOME)—



ABOUT MENU (ACCESSED FROM HOME)—



ASSOCIATED APPLICATION (ACCESSED FROM HOME)—

