



App Specification: MemeMe 2.0, The Final Product

iOS Developer Nanodegree

[Note that this is an informal app description. It will give you an idea how the app should work, but when you submit your app it will be rated based on the [Rubric](#).]

MemeMe is a meme-generating app that enables a user to attach a caption to a picture from their phone. After adding text to an image chosen from the Photo Album or Camera, the user can share it with friends. MemeMe also temporarily stores sent memes which users can browse in a table or a grid.

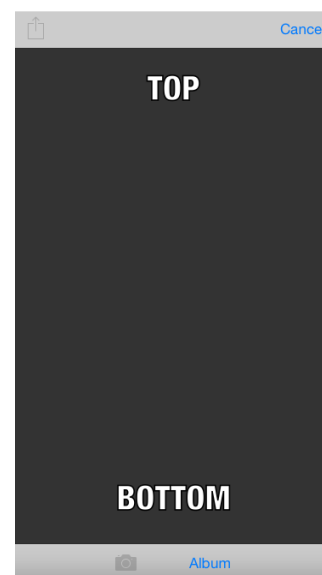
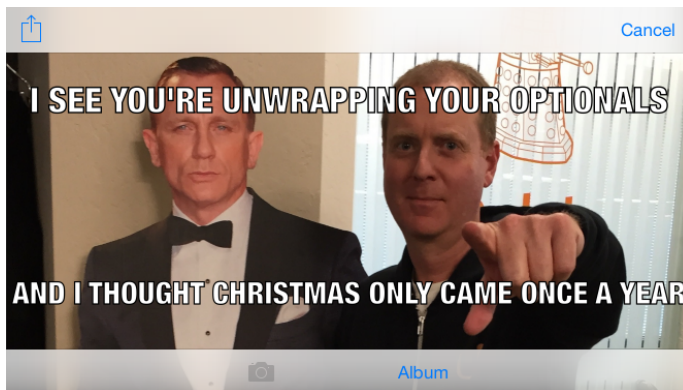
The app has three pages of content:

- **Meme Editor View:** Enables a user to add text to an image and share it.
- **Sent Memes View:** Enables a user to browse sent memes in a table or a grid.
- **Meme Detail View:** Displays an image of a sent meme

The three pages are described in detail below.

Meme Editor View

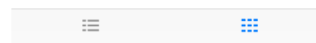
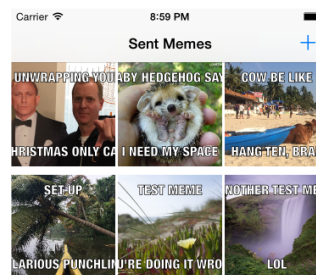
The **Meme Editor View** consists of an image view overlaid by two text fields, one near the top and one near the bottom of the image. This view has a bottom toolbar with two buttons: one for the camera and one for the photo album. The top navigation bar has a share button on the left displaying Apple's stock share icon and a "Cancel" button on the right.

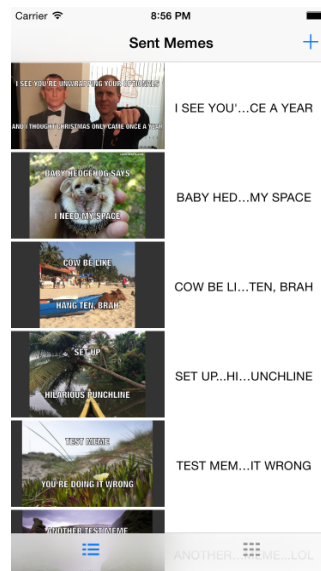


Sent Memes View

The sent memes view displays recently sent memes. It has a bottom toolbar with tabs that allow the user to toggle between viewing sent memes in a table and viewing them in a grid. The top navigation bar has a title that reads "Sent Memes" and an add button in the right corner displaying Apple's stock "Add" icon.

If the user taps the "table" tab on the left of the bottom toolbar, sent memes are displayed in a table. If the user taps on the "collection" tab on the right of the bottom toolbar, sent memes are displayed in a grid. Selecting a meme from the table or collection presents the **Meme Detail View**. Pressing the "add" button brings up the **Meme Editor View**.





Meme Detail View

The **Meme Detail View** displays the selected meme in an image view in the center of the page with the meme's original aspect ratio. The detail view has a back arrow in the top left corner. To the right of the arrow reads the title of the previous view, "Sent Memes."

User Flow

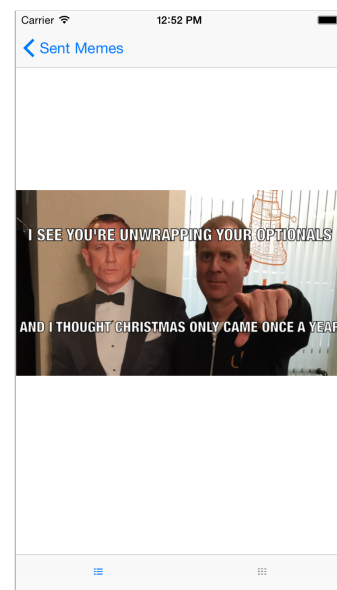
When the user first launches the app the **Sent Memes View** will appear. It will be the root view of the navigation stack. When the user taps the + button in the top right corner the app should push the **Meme Editor View** on top of the **Sent Memes View**.

In the **Meme Editor View**, when the user clicks on the "Album" button, an Image Picker is presented, making it possible to choose an image from the Photo Album. If there is a camera available on the device, pressing the camera button launches the camera, and a newly snapped photo can be chosen for the meme. If a camera is not available on the device, the camera button is disabled.

After an image is chosen, the image picker is dismissed, allowing text to be entered into the top and bottom text fields of the editor. When a user clicks inside one of the text fields, the default text disappears and the keyboard slides up. When the user finishes entering text and presses return, the keyboard is dismissed and the new meme is displayed.

When the user presses the share button, Apple's stock Activity View appears, displaying several options for sharing the meme. After an option is chosen, the Activity View is dismissed and the **Sent Memes View** appears. The **Sent Memes View** also appears upon pressing the "Cancel" button.

Selecting a meme from the table or collection presents the **Meme Detail View**. Clicking on the back arrow of the **Meme Detail View** presents the previous **Sent Memes View**, either the table or collection.



PROJECT SPECIFICATION

MemeMe 2.0: The Final Product

General

CRITERIA	MEETS SPECIFICATIONS
All three pages of content are present.	The app has three pages of content: Meme Editor, Sent Memes, and Meme Detail.
The Sent Memes page is displayed first.	The Sent Memes page is displayed when the app launches.
Code uses control flow effectively.	Control statements (<code>for</code> , <code>while</code> , <code>if</code> , <code>else</code> , <code>switch</code>) are used appropriately.
Code is properly abstracted.	Potentially repetitive blocks of code are effectively abstracted into reusable methods.
Code inputs and outputs are appropriate.	Arguments and return values are procedure-appropriate.
Code is readable by others.	Code is easy to understand. Any code that may be hard to follow is commented effectively.
Code follows appropriate style.	Code adheres to Swift naming and style conventions .
App content is properly displayed in both portrait and landscape mode.	User interface elements, images, and generated memes are properly displayed and function as expected in portrait as well as landscape orientations.

Meme Editor

CRITERIA	MEETS SPECIFICATIONS
Buttons on the MemeEditor toolbar are correct.	The Meme Editor view has a toolbar with two buttons: one that allows user to open the Photo Album to select an image, one that launches the camera.

CRITERIA	MEETS SPECIFICATIONS
The Album button works correctly.	The app displays the image picker when the Album button is pressed.
The Camera button is enabled correctly.	The app displays the camera when the Camera button is pressed on a phone.
The Camera button is disabled correctly.	The Camera button is disabled when app is run on devices without a camera, such as the simulator.
Images are displayed properly.	The chosen image from the camera or the photo album is displayed and scaled properly with <code>AspectFit</code> to fit the device screen.
The necessary text fields are provided.	Text fields are provided for top and bottom text.
Font in text fields is easy to read.	The font and style used to display the meme text is easy to read: bold, all caps, white with a black outline, and shrink to fit.
The text fields are properly shown while typing.	The text field that the user is currently editing remains fully visible with the keyboard on screen. For the bottom text field this is achieved by moving the entire view up to keep the text field on screen, then back after the keyboard is dismissed.
The app has a cancel button that navigates to a view displaying sent memes.	The Meme Editor uses the Cancel button to return to the Sent Memes View.
There is a social share button.	The app has a social share button that uses the “Action” icon built into iOS.
The share button works as intended.	The share button launches the Activity View.
When the share action is complete the meme is saved.	The meme is saved before the Activity View Controller is dismissed.

CRITERIA	MEETS SPECIFICATIONS
Memes are stored using a <code>Meme model</code> .	<p>The <code>Meme model</code> is a <code>struct</code> that includes:</p> <ul style="list-style-type: none"> two <code>strings</code> representing the top and bottom text the original image a memed image combining image and text

Sent Memes

CRITERIA	MEETS SPECIFICATIONS
The tab view is correctly implemented.	The app shows all sent memes in a tab view, with table and collection tabs.
The table and collection views properly display the memes.	Both a table and a collection of memes are displayed.
Meme Detail View is displayed upon selection.	Upon selection, an appropriately sized and scaled Meme Detail view displays the meme.
Meme Detail View slides in from right to left.	The detail view slides in from right to left, and a back arrow in the top left corner leads back to the table/collection.

Suggestions to Make Your Project Stand Out!

- Have your app use the `Impact` font by default.
- Let the user choose between different fonts for the meme text.
- Allow the user to crop the image.
- Use your creativity to improve the UI so your app displays exceptional graphic design.
- Add a "swipe to delete" function to your table view to allow users to delete a meme.
- Add an `Edit` button to your Meme Detail View that launches the Meme Editor to allow the user to edit an already existing meme.