

## App 2A: MemeMe v1.0: Meme Editor Rubric

Criteria	Does Not Meet Specifications	Meets Specifications	Exceeds Specifications (Completely Udacious)
General			
<b>Code is at a professional quality level, uses control statements, methods, and commenting appropriately and effectively. Code adheres to Swift naming and style conventions (<a href="https://github.com/github/swift-style-guide">https://github.com/github/swift-style-guide</a>).</b>	<ul style="list-style-type: none"> <li>Control statements (for, while, if, else, switch) are not used appropriately.</li> <li>Code contains unnecessary repetition not contained in a method.</li> <li>Arguments and return values are inappropriate for their method's purpose.</li> <li>Code is not adequately commented.</li> <li>Code does not adhere to Swift style conventions.</li> </ul>	<ul style="list-style-type: none"> <li>Control statements (for, while, if, else, switch) are used appropriately.</li> <li>Potentially repetitive blocks of code are effectively abstracted into reusable methods.</li> <li>Arguments and return values are procedure-appropriate.</li> <li>Code is easy to understand. Any code that may be hard to follow is commented effectively.</li> <li>Code adheres to Swift style conventions.</li> </ul>	<i>Not Applicable</i>
Meme Editor			
<b>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.</b>	<ul style="list-style-type: none"> <li>The Meme Editor view is missing one or both buttons.</li> </ul>	<ul style="list-style-type: none"> <li>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.</li> </ul>	<i>Not Applicable</i>
<b>The app displays the image picker when the Album button is pressed.</b>	<ul style="list-style-type: none"> <li>The app does not display the image picker when the Album button is pressed.</li> </ul>	<ul style="list-style-type: none"> <li>The app displays the image picker when the Album button is pressed.</li> </ul>	<i>Not Applicable</i>
<b>The app displays the camera when the camera button is pressed on a phone.</b>	<ul style="list-style-type: none"> <li>Pressing the Camera button on a phone does not display the camera.</li> </ul>	<ul style="list-style-type: none"> <li>The app displays the camera when the camera button is pressed on a phone.</li> </ul>	<i>Not Applicable</i>

<b>The camera button is disabled on devices without a camera.</b>	<ul style="list-style-type: none"> <li>The Camera button is enabled on the Simulator.</li> </ul>	<ul style="list-style-type: none"> <li>The Camera button is disabled when app is run on devices without a camera, such as the simulator.</li> </ul>	<i>Not Applicable</i>
<b>The image from the camera/photo album is displayed.</b>	<ul style="list-style-type: none"> <li>The chosen image is not displayed.</li> </ul>	<ul style="list-style-type: none"> <li>The chosen image from the camera or the photo album is displayed, perhaps not scaled properly.</li> </ul>	<ul style="list-style-type: none"> <li>The chosen image from the camera or the photo album is displayed and scaled properly with Aspect Fit to fit the device screen.</li> </ul>
<b>Text fields for providing top and bottom text are provided. Font is bold, all caps, white with a black outline, and shrink to fit.</b>	<ul style="list-style-type: none"> <li>Text fields are not present.</li> </ul>	<ul style="list-style-type: none"> <li>Text fields are provided for top and bottom text.</li> </ul>	<ul style="list-style-type: none"> <li>Text fields are provided for top and bottom text in a font and style that are easy to read: bold, all caps, white with a black outline, and shrink to fit.</li> </ul>
<b>The view slides up to allow the bottom text field to be shown while typing.</b>	<ul style="list-style-type: none"> <li>The bottom text field is obscured by the keyboard while editing.</li> </ul>	<ul style="list-style-type: none"> <li>The bottom text field is moved during editing and remains legible while typing.</li> </ul>	<ul style="list-style-type: none"> <li>The entire view slides up during editing, and back after the keyboard is dismissed.</li> </ul>
<b>There is a social share button.</b>	<ul style="list-style-type: none"> <li>The share button is missing or does not use the iOS default "Action" icon.</li> </ul>	<ul style="list-style-type: none"> <li>The app has a social share button that uses the "Action" icon built into iOS.</li> </ul>	<i>Not Applicable</i>
<b>The share button launches the Activity View.</b>	<ul style="list-style-type: none"> <li>The share button does not launch the Activity View.</li> </ul>	<ul style="list-style-type: none"> <li>The share button launches the Activity View.</li> </ul>	<i>Not Applicable</i>
<b>When the share action is complete the meme is saved.</b>	<ul style="list-style-type: none"> <li>The meme is never saved.</li> </ul>	<ul style="list-style-type: none"> <li>The meme is saved before the Activity View Controller is dismissed.</li> </ul>	<i>Not Applicable</i>
<b>Memes are stored using a Meme model.</b>	<ul style="list-style-type: none"> <li>The Meme model is missing.</li> </ul>	<ul style="list-style-type: none"> <li>The Meme model is a struct that includes: <ul style="list-style-type: none"> <li>two strings representing the top and bottom text</li> <li>the original image</li> <li>a memed image combining image and text</li> </ul> </li> </ul>	<i>Not Applicable</i>

Once you have a functioning app, consider adding more features to make your app stand out.

Here are a few suggestions:

- User can choose between different fonts.
- App uses "Impact" font.
- User can crop image
- App displays exceptional graphic design