ITP 342 Homework 2
Interact

Assignment

- Your project will allow your app to interact with the user. Have the user enter information in a text field and tap buttons. A label will update depending on what the user does.
- Open **Xcode**. Your app needs to compile and run using Xcode 9.2.
- Create a new project. For the template, choose **Single View Application**. For the Product Name, enter something like **HW2** in the text field. For the Organization Name, enter your name in the text field. For the Language pull-down, select the **Objective-C** option.
- Add images to your project's Assets.xcassets to use for the app icon image and the image views or buttons. You can use the ones provided, use your own, or get some off the Web. Resize to the correct sizes. You can use Preview to resize.
- Using storyboard, add the interface components:
  - A label to display the question
  - A text field for input from the user with placeholder text
  - 2 buttons (if only text on buttons, then add image views)
  - If you have image views and buttons, then you may want to put each set (image and button) in a view. This is not required.
  - A label for the message using data from the text field and changing depending on what button was tapped
  - A button to reset the input and label
  - A background button to dismiss the keyboard (optionally, you can use the Touch Events)
  - You may add other components if you would like.
- Add constraints. You need to have constraints for the edges for all of the components (pin or align). You may want to use a fixed height for question label, text field, and reset button. You may want to use equal heights for the buttons and message label. You can then edit the constraint and change the multiplier.

- Test your app using the iOS Simulator. All components should be readable for iPhone SE, iPhone 8, and iPhone 8 Plus in portrait orientation only.
- Using the assistant for the ViewController implementation file (.m),
  - Create an IBOutlet for the text field
  - Create an IBOutlet for the message
  - Create an IBAction for the first button
  - Create an IBAction for the second button (optionally, you can use one IBAction for both buttons)
  - Create an IBAction for the text field to dismiss the keyboard
  - Create an IBAction for the background button to dismiss the keyboard (optionally, you can use the Touch Events)
- In the ViewController implementation file (.m),
  - Implement the IBAction methods
  - Make sure to handle no input in the text field
- Test your app!
- Make sure your app name is appropriate and viewable on the Home screen of the Simulator.
- Set your app to only display in Portrait mode.
- You may customize your UI as much as you want.
- Compress your project folder and rename it to *Lastname**Firstname*HW2.zip where *Lastname* is your last/family name and *Firstname* is your first name. Submit the .zip file on Blackboard.

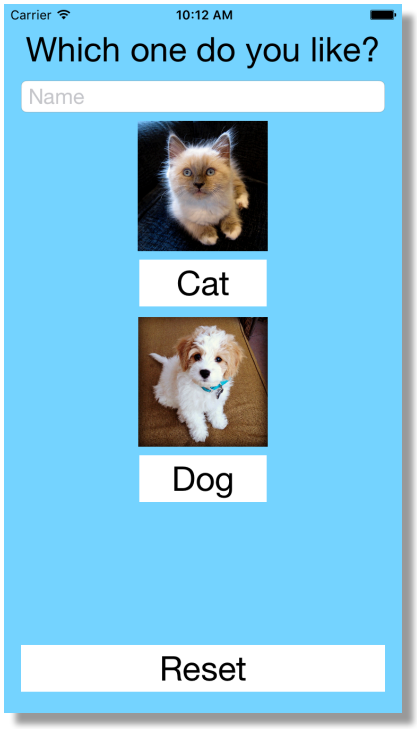
Grading (30 pts)

- Label for question – 1 pt
- Text field for input – 2 pts
- Text field has placeholder text – 1 pt
- Return/Done button on keyboard for text field dismisses keyboard – 2 pt
- Background touch dismisses keyboard – 2 pts
- 2 buttons – 4 pts
- Label for message that changes depending on the button that is pressed. It also includes text from text field and handles empty string in text field. – 6 pts
- Reset button clears text field and message – 2 pts
- App icon (2 sizes) in the assets– 2 pts
- Images are included in the assets – 2 pts
- Project name is something like Choose or Homework2 (not Homework1 or HelloWorld) – 1 pt
- App name is appropriate and viewable – 1 pt
- Using Auto Layout, app looks good in iPhone SE, iPhone 7, and iPhone 7 Plus in portrait orientation – 3 pts
- App only displays in portrait orientation – 1 pt
- Points will be deducted for not having your name in the comment blocks at the top of your .h and .m files.

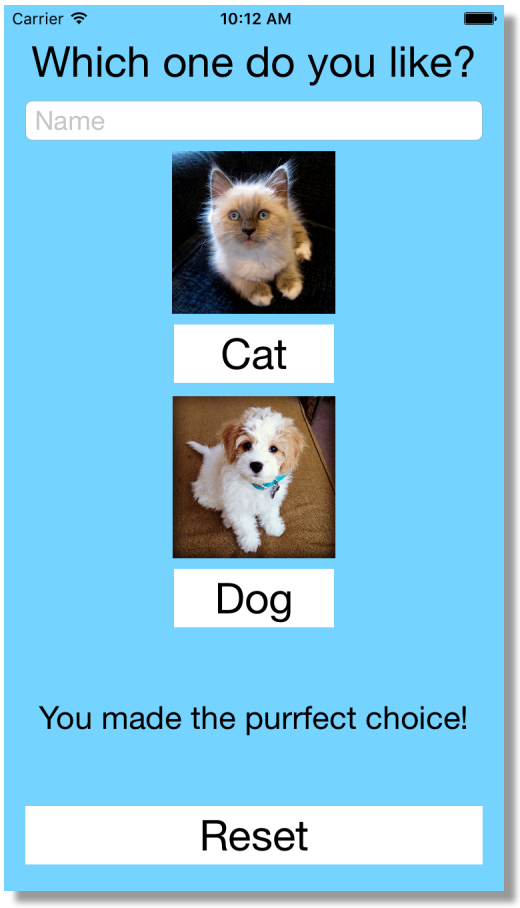
Other Info

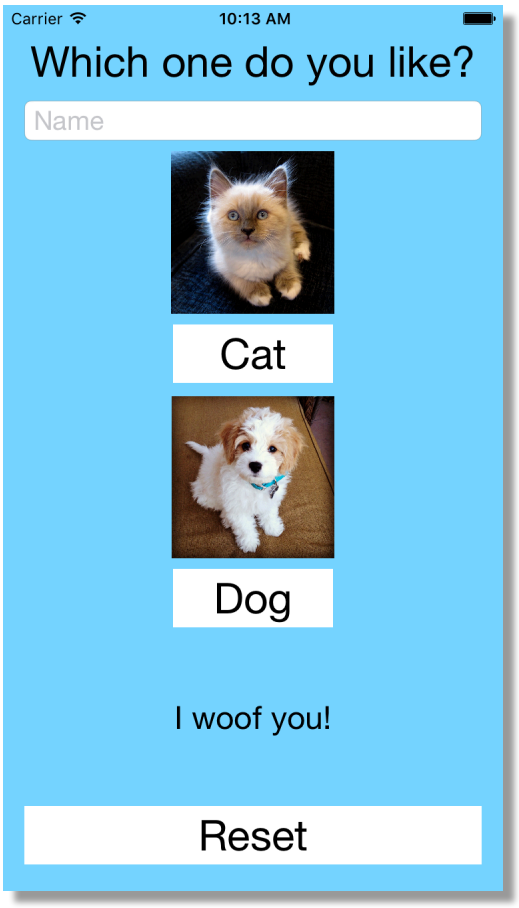
- Images for your app icon should be 120 x 120 pixels (@2x) and 180 x 180 pixels (@2x).
- For each image, include 3 files: *image*.png, *image*@2x.png, and *image*@3x.png where *image* is an appropriate name for your image. Then add them to your project's Assets.xcassets as one image set.

Example with Image Views and Buttons



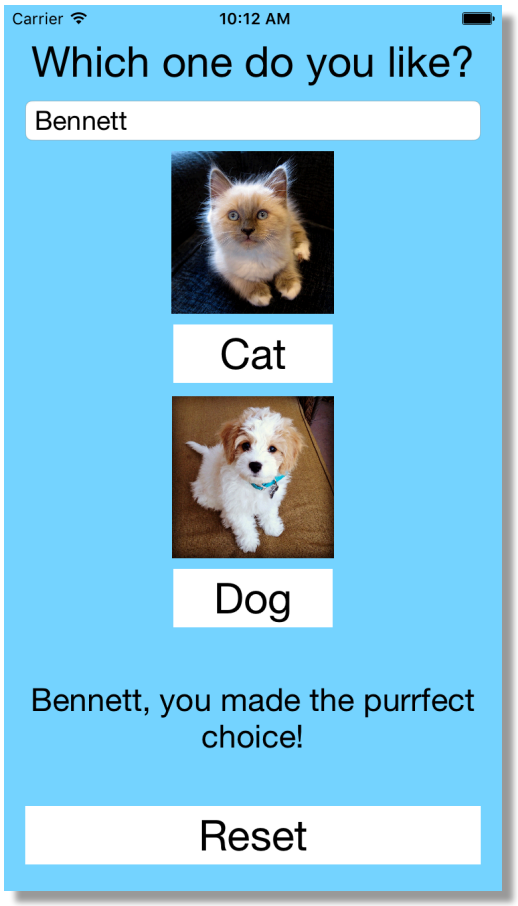
Buttons touched with no name entered

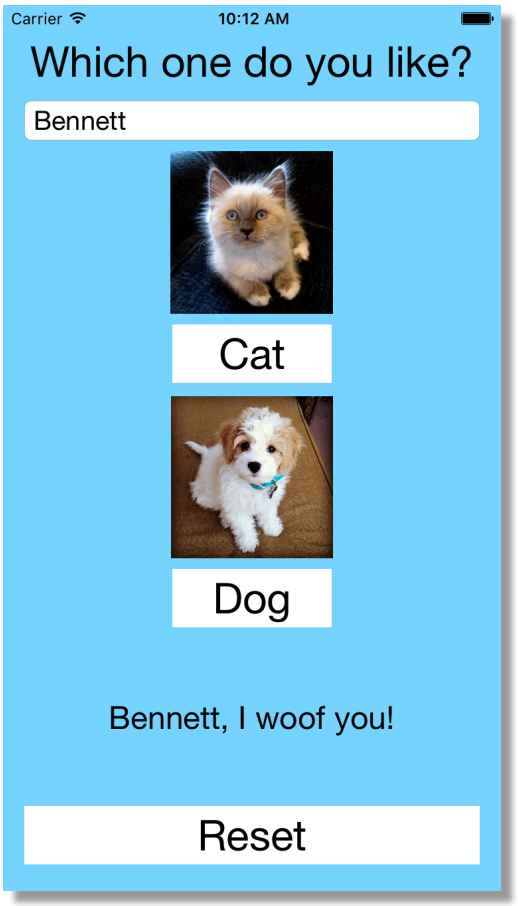




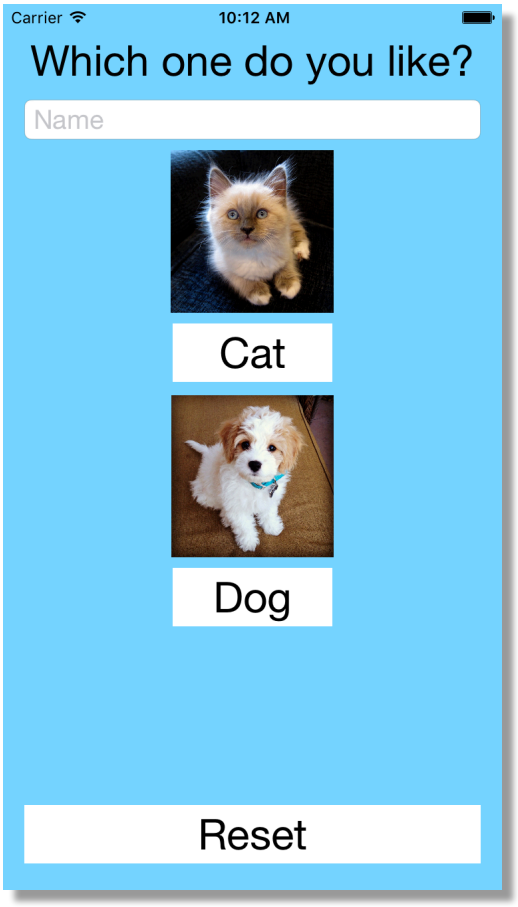
Keyboard with Done button

Buttons touched with a name entered

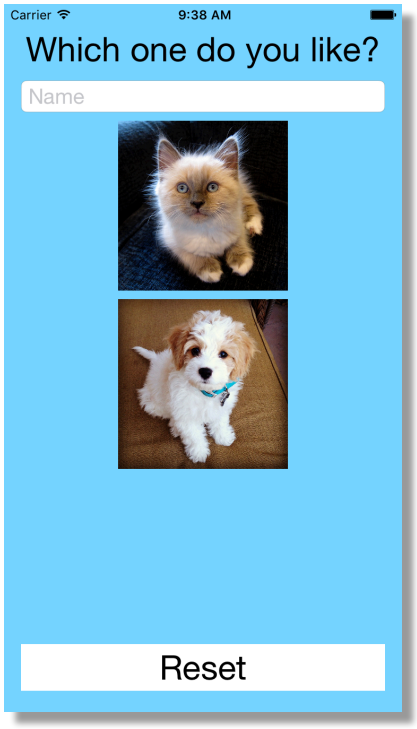




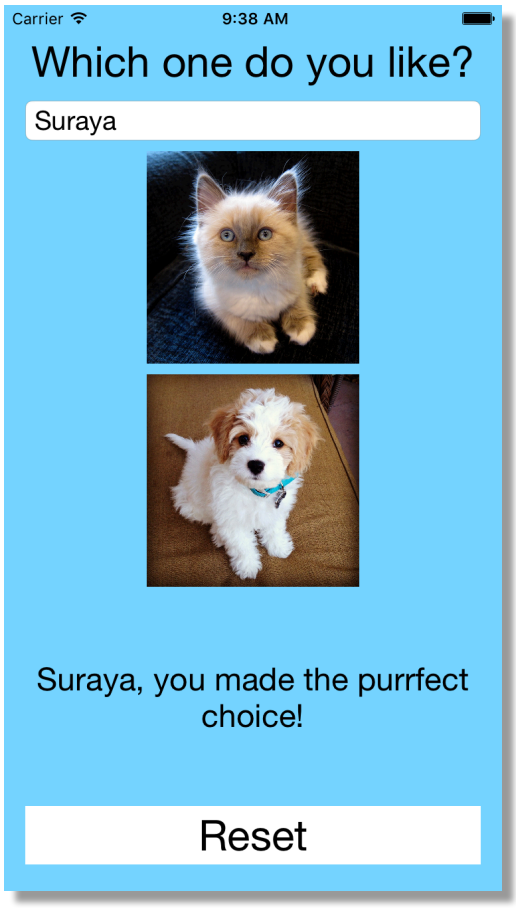
Reset button



Example with Buttons that use images



Cat Button touched



Dog Button touched