**Mobile Computing – iOS Spring’23**

**Assignment04**

**30 Points**

**Please follow the following instructions to complete this assignment.**

1. Open Xcode from the launchpad of your Mac.
2. Click on create a new Xcode project. Select the iOS template and click on the App application.
3. Click on next which will prompt you to choose options for the project.
4. Provide product name as **LastnameGalleryApp**, “**edu.nwmissouri**” for organization identifier, “**storyboard**” as interface and “**Swift**” as the language.
5. Click on next and select an appropriate location to save your app and click on create. A project directory will be loaded.
6. From the project navigator click on “Main.storyboard” file, a blank mobile screen will be loaded, where the required fields for an app need to be added.

**View**

Table 1: UI elements configuration

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 2 text fields | To perform keyword searches | searchTF |
| To add a comment | commentTF |
| 1 image view | To display a picture | pictureIV |
| 1 text view | To show a user’s comments about a picture | pictureCommentsTV |
| 5 UIButton elements | Search - To search for pictures in a category | searchBTN |
| search: |
| 👍- To like a picture | likeBTN |
| like: |
| 👎- To dislike a dislike | dislikeBTN |
| dislike: |
| Save - To show previous picture | saveBTN |
| save: |
| Reset - To reset the app | resetBTN |
| reset: |

*Note: Names that are ending with a colon (****:****) are actions.*

1. Open library (cmd+shift+l). Search for “text field”, add (drag and drop) it to the storyboard. Give its placeholder value as *Search keyword*.

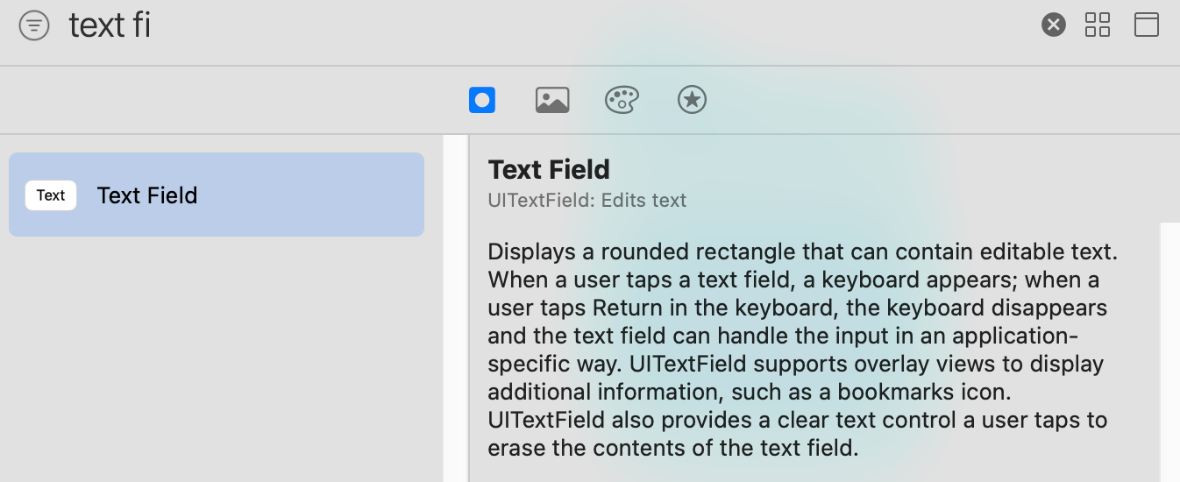


Figure Xcode Library

1. Add a filled button to the storyboard. Add an image to it by using its image attribute in its Attributes Inspector by selecting the *magnifyingglass* icon from the dropdown.
2. Add an Image View to the storyboard. Set its default image property to folder in its Attributes Inspector.
3. Add two normal buttons to the app and give like (👍) and dislike (👎) emojis as their titles, respectively.
   1. In Xcode menu bar, see Edit > Emoji & Symbols to find special Unicode characters.
4. Add a text view to the app. Uncheck its Editable property under Behavior group in its Attributes Inspector. Also, change its alignment to Justified.
5. Add a text field to the app to read user comment.
6. Add 2 normal buttons to save the comment and reset the app, respectively.
7. Now, all the required UI elements for the app are added to the storyboard. Apply auto layout to the app by adding constraints to UI elements as shown in the *Figure 2*.
   1. Hint: Follow the view hierarchy in the *Figure 2*. (i.e., document outline on the left side) to add and organize stack views and embed elements inside them.

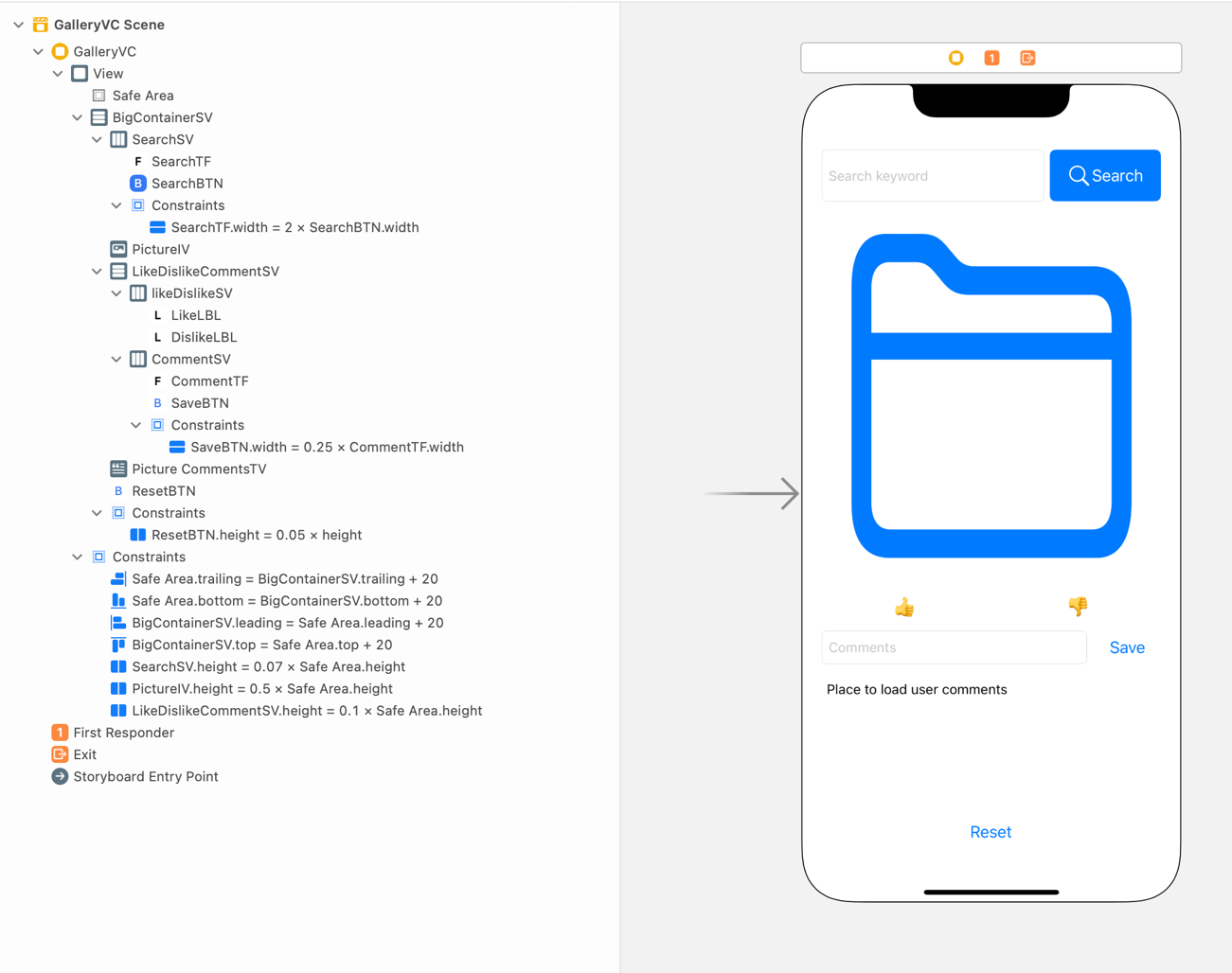


Figure The View

Table 1 GalleryVC’s properties for stack views

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stack View** | **Axis** | **Alignment** | **Distribution** | **Spacing** |
| BigContainerSV | Vertical | Fill | Fill | Standard |
| SearchSV | Horizontal | Fill | Fill | Standard |
| LikeDislikeCommentSV | Vertical | Fill | Fill Equally | Standard |
| LikeDislikeSV | Horizontal | Fill | Fill Equally | Standard |
| CommentSV | Horizontal | Fill | Fill | Standard |

**Model**

1. Create a new Model.swift file, which will serve as our app’s model. Define the following types in it.



Figure The Model

1. AppConstants will contain different categories’ keywords and pictures names. A sample for category cars is shown above. Similarly, add at least 5 appropriate categories of your choice.
2. The Picture has name, like count, dislike count and comments. Name is the title of the image that put in your assets. At first, the like and dislike counts for a picture are randomly generated from 0 to 99, respectively. Comments will be an empty array.
3. The Category is the class that is used to create different categories like cars, bikes, etc. using its initializer. The initializer will initialize the keywords and all the pictures in a category.
4. Provide a set of 5 different keywords, for instance, actor, movie, theatre, film, and artist. Provide at least 5 pictures with their individual details for each category. Add all the pictures to your app’s Assets folder in Xcode.
   1. Note: Provide high resolution files for each picture that you pick like @2x and @3x versions of that picture.

**Controller**

1. Create a Cocoa Touch Class “GalleryVC” that is a sub class of UIViewController and assign it as class to Main.storyboard file.
2. Create outlet/action items as specified in *Table 1*.
3. This application is about displaying pictures based on the keyword a user inputs. For example, if the search keyword is actor**,** all the pictures that are related to the actor category and their individual like and dislike counts should be displayed to the user one at a time. Moreover, comments for each picture in the actor category should also be displayed using the pictureCommentsTV.
   1. If a picture has no comments, display “No comments yet.”
4. The user can like or dislike a picture multiple time. If it’s a like, user double taps on the image that increments the like count for that picture by 1 and the like button’s title is updated in place. In case of dislike, user triple taps the image that increments the dislike count.
5. User can navigate back and forth using swipe gestures left swipe for previous image and right swipe for next image. The pictureIV, likeBTN, dislikeBTN, and pictureCommentsTV should be updated w.r.t the action performed.
6. If the user taps on the Reset button, app will be getting back to its initial state.

**Functionality**

1. At first, all the buttons and text fields except **searchTF** should be in disabled mode.
2. Enable the Search button only if some text is entered in the **searchTF.** Otherwise, disable the Search button.
   1. Make use of **Editing Changed** event for the **searchTF** to add an action in which you may write code to enable/disable the Search button**.**
3. Whenever the user enters a search keyword and clicks on the Search button, if the search keyword matches with any category or any keyword in anyone of the categories, display only that category related pictures one at a time along with that picture’s like and dislike counts, and comments.
   1. Use Array class’ contains(\_:)method to check whether an array contains a particular element are not.
4. Enable all other UI elements only if there is a match for the search keyword.
5. If no matches found for the search keyword,
   1. Display “Sorry! No pictures found for the search. Try again with a different search keyword.in the pictureCommentsTV.
   2. Display “folder” system image in the pictureIV. Moreover, the likeBTN, dislikeBTN, commentTF, saveBTN, and resetBTN buttons should be in disabled mode.
6. User can like or dislike a picture. If it’s a like, increment and save the like count for that picture and update the title of the likeBTN in place. Similarly, for dislike as well.
   1. Hint: Use UIButton’s setTitle(\_:for:) method to change a button’s title.
7. When the user swipes left or right, refresh the app’s UI by updating the picture (i.e., pictureIV) and its likes and dislikes count, and comments.
8. The Reset button will simply reset the app to its initial launch state.

**Sample Output (Video Link):**

<https://app.vidgrid.com/view/tfnwTeWjwJ3D>