**Mobile Computing – iOS Fall’23**

**Sec03 Exam01**

**50 Points**

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

1. Launch Xcode and clone the exam GitHub repo.
2. Finish the app as per the requirements described below. Use appropriate data structures wherever required.

**View**

Table 1: UI elements configuration

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 2 UILabel elements | To display | n/a |
| To display the error messages and output | infoLBL |
| 1 UITextView | To display the items lists and persons in split | messageTV |
| 1 UIImageView element | To display the image | expenseSplitIMG |
| 5 UIButton elements | Title is Split Equally | calculateSplit: |
| Title is Clear All | clearsplit: |
| Add image “plus.circle” as title | addRemoveBTNS |
| addItem: |
| Add image “person.crop.circle.badge.plus” as title | addRemoveBTNS |
| addPerson: |
| Add image “person.crop.circle.badge.minus” as title | addRemoveBTNS |
| removePerson: |
| 3 UITextField elements | Placeholder is Item | itemTF |
| Placeholder is Amount | amountTF |
| Placeholder is Person | personTF |

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

1. Use a label to display the app title. Height of the label is 0.1 percent of its containing view.
2. Import all the provided images into Assets.xcassets folder.
3. Make Content Mode of image view to Scale To Fill and set its corner radius to 10.0 points. Set “expense-splitter.png” (given to you) as default image to it. Its height is 0.25 percent of its containing view.
4. Provide placeholders to the item name and amount text fields and stack them horizontally. Set the stack alignment property to fill, distribution to fill equally and use standard spacing.
5. Use a normal button to add item. Set the image to it as shown in the sample output.
6. Stack the above created horizontal stack and the button into one. Set the axis of the stack as horizontal. Set the stack alignment property to fill, distribution to fill and use standard spacing. The text fields stack must occupy 0.7 percent width of the new stack. Height of the newly created stack view is 0.1 of its containing view.
7. Provide placeholder to the person’s name text field. Use 2 normal buttons to add and remove a person, respectively. Set the images for the buttons as shown in the sample output. Stack the buttons horizontally with stack alignment property as fill, distribution as fill equally and standard spacing between them.
8. Stack the buttons stack view and text field into one with horizontal alignment. Set the alignment property to fill, distribution to fill and spacing to standard. The text field must occupy 0.7 percent width of the stack. Height of the newly created stack view is 0.1 of its containing view.
9. Set the text alignment to Justify. Use it to show app’s data.
10. Use 2 buttons to split expenses equally and clear the app’s data. Name them as per *Table 1*. Stack the buttons horizontally. Set the stack alignment property to fill, distribution to fill equally and use standard spacing for the stack. Height of the stack view is 0.08 percent of its containing view.
    1. Set the below properties to the buttons.
       1. Background color: System Purple Color
       2. Foreground: White
11. Use a label to display total and split amounts. Set its alignment to Center. Height of the label is 0.1 percent of its containing view.
12. Set the following properties to all the add, remove buttons:
    1. Border Width: 1.0
    2. Corner Radius: 7.0
    3. Border Color: System Gray
13. Now, apply auto layout to the app by adding constraints to the UI elements.

**Controller:**

1. Create outlets and actions as specified in *Table 1*.
2. Set the default message "Add item(s) and person(s) to calculate the split." using the infoLBL. If no input is provided for the item name and amount, and person name, display appropriate alert messages using the infoLBL.
3. Once the user enters an item name and its amount, and taps on 􀁌, then it must be added to items list. Update the list if the item is already present in the list. Display them in the text view as shown in the sample output. (Display error messages if the user taps the 􀁌 with empty text fields (i.e., item name and item amount) using the infoLBL).
4. Similarly, when the user taps on 􀉯 button, the name must be added to the persons list. If the person is already in the list, display appropriate validation message using the infoLBL. Display the list in the text view as shown in the sample output. (Display the error messages if the user tries to add a person with an empty value to person name text field). When the user clicks on 􀉱, remove the name from the list. (Display appropriate validation message if the name is not in the list).
5. When user taps on the Split Equally button, split the total amount (i.e., sum of all item amounts) among the people equally. Display the final split amount using the infoLBL.
6. When the Clear All button is tapped, the app should return to its initial state.