**Mobile Computing – iOS Fall’23**

**Assignment01**

**20 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 platform and click on the App template.
3. Click on next, which will prompt you to choose options for the project.
4. Provide product name as **PlayWithAlphaNumericsApp**, “**edu.northwest**” for organization identifier, “**Storyboard**” as interface and **Swift** as 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 UI components for the app are added.

**The View**

Table 1: UI elements configuration

|  |  |  |
| --- | --- | --- |
| **UI element** | **Purpose** | **Outlet/action name** |
| 8 UILabel elements | To display app header | headerLBL |
| To display “Play with:” | n/a |
| To display “Numbers” | n/a |
| To display “Strings” | n/a |
| To display the “First number:” | n/a |
| To display the “Second number:” | n/a |
| To display the value of the first number | firstNumLBL |
| To display the value of the second number | secondNumLBL |
| 2 UITextField elements | To input the first string | firstStrTF |
| To input the second string | secondStrTF |
| 2 UISwitch elements | For Selecting Numbers | numberSWCH: |
| optionsSWCH: [UISwitch]! |
| For Selecting Strings | stringSWCH: |
| optionsSWCH: [UISwitch]! |
| 2 UIStepper elements | To +/- the first number | firstNumStepper: |
| numbersStepper: [UIStepper]! |
| To +/- the second number | secondNumStepper: |
| numbersStepper: [UIStepper]! |
| 1 UITextView | To display the output | messageTV |
| 3 UIButtons | Button Generate Pattern for displaying the pattern | generatePattern: |
| stringAndPatternBTN: [UIButton]! |
| Button String Operations for displaying the output of strings | manipulateStrings: |
| stringAndPatternBTN: [UIButton]! |
| Button Reset to clear the app | onReset: |

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

1. Open library (cmd+shift+l). Search for “label” and add (drag and drop) it to the storyboard.Graphical user interface, text, application

   Description automatically generated
2. Similarly, add 7 more labels to the storyboard or copy a few more labels.
3. For the header label, set the background color to the matching color in document outline (i.e., R: 0, G: 103, B: 71) and set its title as shown in the below code snippet.



1. Moreover, set its properties as mentioned here.
   1. Color: White Color
   2. Font: System Heavy 26.0
   3. Alignment: Center
2. In the library, search for “text field” and add 2 text fields to the storyboard. Set their placeholder values to “First string value”, and “Second string value”, respectively.
3. Search for “switch” and add two UISwitches to the storyboard.
4. Next, search for “stepper” and add two UISteppers to the storyboard.
5. To display the output, search for “text view” and add a UITextView to the storyboard. Uncheck its Editable property under Behavior group in its Attributes Inspector.
6. Finally, search for “button” and add 3 filled buttons to the storyboard. Name them “Generate pattern”, “Manipulate strings” and “Reset”, respectively. Set the following attributes for them:
   1. Corner Style – “Capsule” (for both filled and normal button)
   2. Foreground – “White Color ” (for normal button)
   3. Fill – “Accent Color” (for normal button)

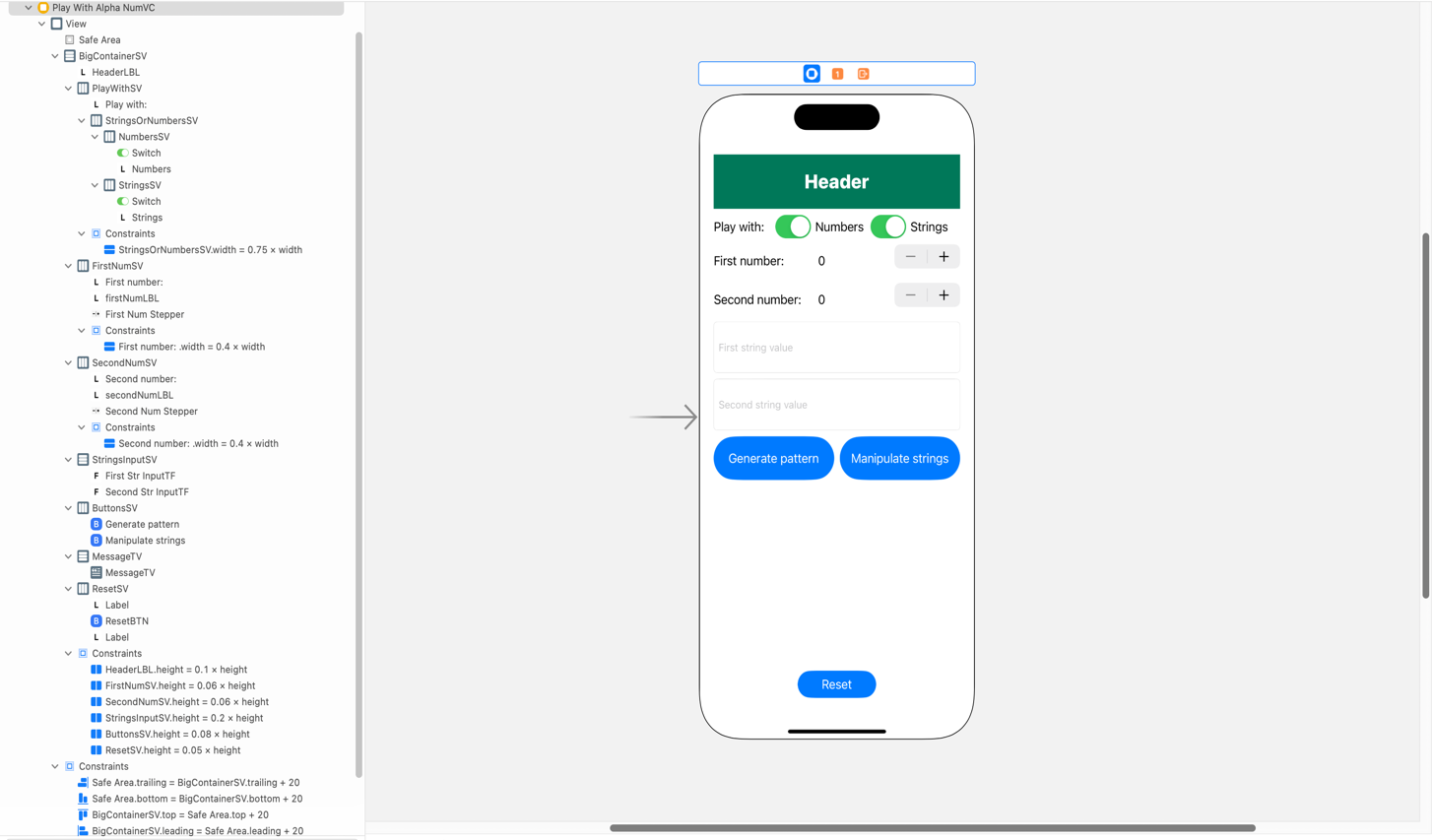


Figure Auto Layout

1. 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 Figure 1.
   1. Hint: Follow the view hierarchy in the Figures 1 (i.e., document outline on the left side) to add and organize stack views and embed elements inside them.

Table 1 PlaywithAlphaNumVC’s properties for stack views

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stack View** | **Axis** | **Alignment** | **Distribution** | **Spacing** |
| BigContainerSV | Vertical | Fill | Fill | Standard |
| playwithSV, firstNumSV, secondNumSV | Horizontal | Fill | Fill | Standard |
| stringsOrNumbersSV,  buttonsSV, resetSV | Horizontal | Fill | Fill Equally | Standard |
| numbersSV, stringsSV | Horizontal | Fill | Fill | 8 |
| stringsInputSV, messageTV | Vertical | Fill | Fill Equally | Standard |

**The Controller**

1. Create a Cocoa Touch Class “PlayWithAlphaNumVC” that is a sub class of UIViewController and assign it as class to the view controller in Main.storyboard file.
2. Create outlets and actions as specified in Table 1.
3. Initially,
   1. All the UI elements (UISteppers, UITextFields, UIButtons) are disabled except for the UISwitches and the Reset button.
   2. The messageTV is set to empty string.
4. When the Numbers switch is “on”, enable the UI elements related to numbers (UISteppers and Generate pattern button).
5. Similarly, when the Strings switch is “on”, enable the UI elements related to strings (UITextFields and Manipulate strings button).
6. On tapping “Generate pattern”, display the pattern as shown in the sample output in the text view (for pattern, consider the first number as rows and second number as columns). If first number and second number are zero, show appropriate alert message. For example, if first number is zero, display “First number is invalid. Please provide a value greater than zero.” using messageTV.
7. When the button “Manipulate strings” is tapped, perform the string operations as shown in the sample output, and display the output in the text view. In case the text fields are empty, show appropriate alert message. For example, if first string value is empty, display “First string value is invalid. Please provide at least one character.” using messageTV.
8. The reset action will set all the UI elements to their initial state.

**Note**

1. The value of switch is a boolean. If the switch is on, the values will be true and vice versa, to access the values of the switch use its isOn property.
2. Assign the value of stepper the label using value property. The stepper value is of Float type (concert it into Integer type).
3. To enable and disable the UI elements use isEnabled property. This property accepts Boolean values.

**Submission:** Push your entire Xcode project to the GitHub and submit your private repository link on Canvas via Text Entry.