**IOS Assignment**

**Aim :** In this assignment we are going to create an app which contains table view and multiple view controller. When we click on table cell the appropriate data should be displayed on the final screen.

1. Open Xcode in your Mac, Select App and click on next and give the name as LastName\_Tableview to the project and make sure to select the user interface as storyboard, and language as Swift. Click on next choose a folder from your mac to store the project. Give the app Display name as LastName\_Tableview.

A screenshot of a computer

Description automatically generated

2. After creating the project lets create a design for your first screen. Click on main.storyboard which will be on the left in the folder section.

3. Add a navigation controller to the main.storyboard as we are dealing with multiple view controllers, we need a navigation controller to redirect to other screens. To add it click on view controller screen then find editor on top of your mac click on Editor click Embed In and click on Navigation Controller and add it beside view controller in main.storyboard.

4. Our first screen contains a label at the top and a table view below the label. Now let’s begin by adding a label, search for label in the library in main.storyboard . Drag and drop to the screen and give constraints top = 30, left = 20, right = 20, height = 40 and also make the text alignment to center. Give the name of the label in attribute inspector of your choice based on the topic you choose.

5. Now, search for table view in library and drag drop the UITableView on the Storyboard below the title label and give constraints top = 0, left = 0, right = 0, bottom = 0.

6. Now go to the attribute inspector of table view and select the prototype cell to 1. Now give the identifier to the cell as “cell”. Now while making a connection to the View Controller, give the name for the outlet as tableView.

Graphical user interface

Description automatically generated

7. Now we are going to add an imageView and a label to the tableView cell. Click on the cell which you see on the table view of your first screen. You can see a small square box in the middle of the cell drag it down to increase the size of the cell.

Graphical user interface, application

Description automatically generated

8. For the For UITableView Cell add a UIImageView. Look for UIImageView in the library and drag drop to the table cell and give constraints as height = 100, width = 100, left = 12, make vertical center in container. Change the content mode in attribute inspector to aspect fill.

9. Now we add a label to the table cell beside of the UIImageView that is just created on above step. Look for UILabel in the library and drag drop to UITableView cell and give constraints left = 12, right = 12 and height 30, make vertical center in container. The final design should look like the below reference.

Graphical user interface, application

Description automatically generated

10. As we are dealing with adding extra features to your table view cell, we need to create a file for the tableViewCell.

11. Now right click on the project, create a new file, select Cocoa Touch Class, and select next. Now give the class name as CellTableViewCell and give the subclass as UITableViewCell, language as swift and click next.

A screenshot of a computer

Description automatically generated

12. Now the file has been successfully created. Click on the tableview cell Now in the identity inspector of the cell, give the name of the class as “CellTableViewCell” and click return.

13. Now we need to add connections to the UIImageView and label of the cell. Drag the UIImageView to the CellTableViewCell file at the top and give the name as “viewImage” and click connect.

14. Similarly do it for label Drag the label to the CellTableViewCell file below the “viewImage” and give the name as “labelDisplay” and click connect.

Graphical user interface, text

Description automatically generated

15. Next we need to add a new View Controller to display contents in next screen. Now from the library in the Main.StoryBoard add a new View Controller. Now right click on the project, create a new file, select Cocoa Touch Class, and select next. Now give the class name as DetailViewController and give the subclass as UIViewController, language as swift and click next.

A picture containing text, screenshot, monitor

Description automatically generated

16. Now the file has been successfully created. Click on the newly added viewcontroller. Now in the identity inspector of the viewcontroller, give the name of the class as “DetailViewController”.

Graphical user interface

Description automatically generated

17. Create a segue from the table view cell of ViewController by clicking on tableViewCell and drag dropping it to the display screen of DetailViewController and name the identifier as “detailSegue”.

Graphical user interface, application

Description automatically generated

18. Our second screen contains a label at the top and a UIImageview below the label. Now let’s begin by adding a label, search for label in the library in main.storyboard. Drop it on the screen and give constraints top = 30, left = 20, right = 20, height = 40 and also make the text alignment to center. Now while making a connection to the DetailViewController, give the name for the outlet as “displayLabel”.

19. Now, take UIImageView from library and drag drop to DetailViewController and give the constraints top = 0, left = 20, right = 20, height = 170. You can change the height according to your need. . Now while making a connection to the UIImageView in DetailViewController, give the name for the outlet as “displayImage”.

20. Implement UIButton at the bottom by searching for button in library and drag drop to DetailViewController and give constraints top = 30, left = 20, right = 20 and height = 40. Now while making a connection to the button in DetailViewController, give the name for the action outlet as “onClickAction”. When button is clicked respective content should be displayed in text view.

21. Add a text view below the button by searching for textView in library and drag drop to DetailViewController and give constraints top = 30, left = 20, right = 20 and height = 50. Now while making a connection to the textView in DetailViewController, give the name for the outlet as “displayText”.

22. Now we are done with the design part. Final design of DetailViewController should look as below

A screenshot of a phone

Description automatically generated with medium confidence

23. Now the coding part begins, Open ViewController of your first screen and create an array of names and it should contain atleast 5 names.

24. Download images from browser of any 5 items which you have chosen for above names. Add the images to assests folder in Xcode.

25. Create an array of images for above downloaded pictures and give the names in array which matches with above images names.

26. Finally create an array of topic for above items. You are done with creating names, images, and topics.

27. Now let’s populate the UITableView cell with image and name for UIImage and label respectively.

28. When you populate an image to UITableView cell the image size will be predefined, and you can see the image clumsy colliding with below cells when you run the app.

29. To avoid above problem use float value as 120 and the implementation is given below

func tableView(\_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {

120

}

30. Use Prepare method to send the data from ViewController to DetailViewController you need to send names, images, and text using segue.

31. Declare 3 variables to capture received data in DetailViewController and assign them to appropriate outlets.

32. Finally when a cell is clicked in table view of ViewController Name, Image, and text should be populated and for the text to be displayed write code in action. Test for different by clicking on different cells in tableView. Below are some references

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**Please submit your app as compressed file, your compressed files should contain**

**LastName\_Tableview folder and LastName\_Tableview.xcodeproj file. Please check**

**your submission by downloading the submitted file and rechecking in xcode.**