Stack Views and Segues-08

In this assignment you are going to continue working on Homeowner to create an interface for displaying the details of a specific Item. The interface that you create will consist of multiple nested stack views, both vertical and horizontal.

1. Ensure you commit and push, then submit your commit ID for Lab 8 before moving on with this lab. Also, ensure you create a new repository for this lab, remove your remote for lab 8, add a new remote for lab 9, then do a push (also don't forget to add collaborators).

2. Open your Homeowner project and then open Main.storyboard.

3. Drag a new View Controller from the object library onto the canvas.

4. Drag a Vertical Stack View from the object library onto the view for the View Controller.

5. Add constraints to the stack view to pin it to the leading and trailing margins, and pin the top and bottom edges to be 8 points from the top and bottom layout guides.

6. Now drag four instances of UILabel from the object library onto the stack view. From top to bottom, give these labels the text "Name," "Serial," "Value," and "Date Created".

7. Select the Date Created label and open its size inspector.

8. Find the Vertical Content Hugging Priority and lower it to 249. Now the other three labels have a higher content hugging priority, so they will all hug to their intrinsic content height. The Date Created label will stretch to fill in the remaining space.

9. Select the stack view, either on the canvas or using the document outline. Open its attributes inspector and find the section at the top labeled Stack View.

10. One of the properties that determines how the content is laid out is the Distribution property. Currently it is set to Fill, which lets the views lay out their content based on their intrinsic content size. Change the value to Fill Equally. This will resize the labels so that they all have the same height, ignoring the intrinsic content size.

11. Change the Distribution of the stack view back to Fill. This is the value you will want going forward in this assignment (we only changed it to Fill Equally so you can see how it works).

12. Select the Name label on the canvas. Click the second icon from the left in the Auto Layout constraints menu: The Down Arrow icon is shown.. This will embed the selected view in a stack view.

13. Select the new stack view and open its attributes inspector. The stack view is currently a vertical stack view, but you want it to be a horizontal stack view. Change the Axis to Horizontal.

14. Now drag a Text Field from the object library to the right of the Name label. Because labels, by default, have a greater content hugging priority than text fields, the label hugs to its intrinsic content width and the text field stretches.

15. Open the size inspector for the text field and set its Horizontal Content Compression Resistance Priority to 749. This will ensure that the text field's text will be truncated if necessary, rather than the label.

16. Select the horizontal stack view and open its attributes inspector. Change the Spacing to be 8 points. Notice that the text field shrinks to accommodate the spacing, because it is less resistant to compression than the label.

17. Repeat these steps for the Serial and Value labels:
    a) Select the label and click the Stack view spacing icon.
    b) Change the stack view to be a horizontal stack view.
    c) Drag a text field onto the horizontal stack view and change its horizontal content compression resistance priority to be 749.
    d) Update the stack view to have a spacing of 8 points.

18. Select the vertical stack view, open its attributes inspector, and update the Spacing to be 8 points. Then select the Date Created label, open its attributes inspector, and change the Alignment to be centered.

19. Control-drag from the Name text field to the Serial text field and select Leading. Then do the same for the Serial text field and the Value text field.

20. In Main.storyboard, select the ItemCell prototype cell on the Items View Controller. Control-drag from the cell to the new view controller that you set up in the previous section. (Make sure you are Control-dragging from the cell and not the table view!).

21. A black panel will appear that lists the possible styles for this segue. Select Show from the Selection Segue section of the pop up.

22. Build and run the application. Tap a cell and the new view controller will slide up from the bottom of the screen. (Sliding up from the bottom is the default behavior when presenting a view controller modally.)

23. Create a new Swift file and name it DetailViewController. Open DetailViewController.swift and declare a new UIViewController subclass named DetailViewController.

24. With DetailViewController.swift open, Option-click on Main.storyboard in the project navigator. This will open the file in the assistant editor right next to DetailViewController.swift.

25. Before you connect the outlets, you need to tell the detail interface that it should be associated with the DetailViewController. Select the View Controller on the canvas and open its identity inspector. Change the Class to be DetailViewController.

26. In DetailViewController.swift, create and connect 4 outlets for your name field, serial number field, value field, and date label.

27. DetailViewController will hold on to a reference to the Item that is being displayed. When its view is loaded, you will set the text on each text field to the appropriate value from the Item instance. In DetailViewController.swift, add a property for an Item instance and override viewWillAppear(_:) to set up the interface.

28. Add an instance of NumberFormatter and DateFormatter to the DetailViewController. Use these formatters in viewWillAppear(_:) to format the valueInDollars and dateCreated.

29. Open Main.storyboard again. Select the show segue by clicking on the arrow between the two view controllers and open the attributes inspector. For the identifier, enter showItem.

30. With your segue identified, you can now pass your Item instances around. Open ItemsViewController.swift and implement prepare(for:sender:).

Final App