

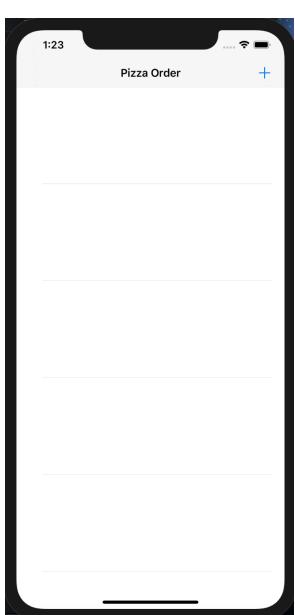
CS 329E: Bulko

Programming Assignment 5

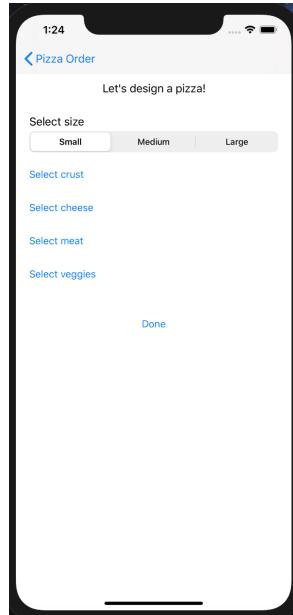
Alert Controllers

1 Problem Definition

In this assignment, you will create a simple application for ordering pizzas! You will use a segmented VC, alerts, and action sheets to allow the user to select options for the pizzas.



(a) The Main View Controller

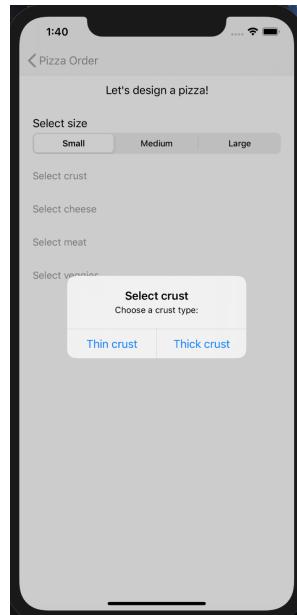


(b) The Pizza Creation View Controller

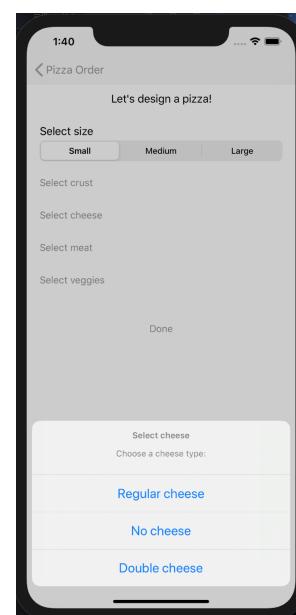
The Main VC consists of a Table View containing all of the pizzas that have been created. When the application first launches, the table will be empty as in figure (a). To create a new pizza, the user must click on the “+” button in the Navigation Bar. This will take the user to the Pizza Creation VC, figure (b). The user then configures a pizza to be ordered.

- The user can select one of three pizza sizes, small, medium, or large, using a segmented VC.
- There are two crust types: Thin Crust or Thick Crust. The user should be presented these two options as buttons in an alert.
- There are three cheese options: Regular Cheese, No Cheese, or Double Cheese. The user should be presented these options in an action sheet. (See Special Note below.)

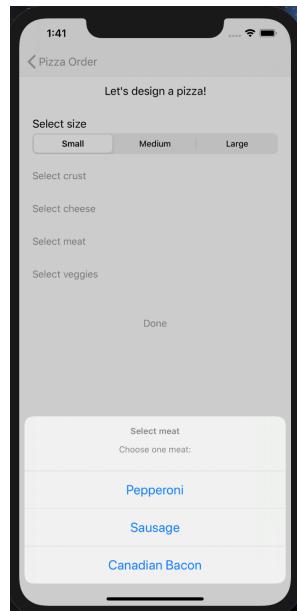
- There are three meat options: Pepperoni, Sausage, or Canadian Bacon. The user should be presented these options in an action sheet. (See Special Note below.)
- There are five veggie options: Mushroom, Onion, Green Olive, Black Olive, or None. The user should be presented these options in an action sheet. (See Special Note below.)
- SPECIAL NOTE: Apple changed the way action sheets behave in iOS 26. If you've migrated to the newer version of Xcode, your action sheets may not appear like they do in the pictures. That's acceptable, and you will not be penalized for it.



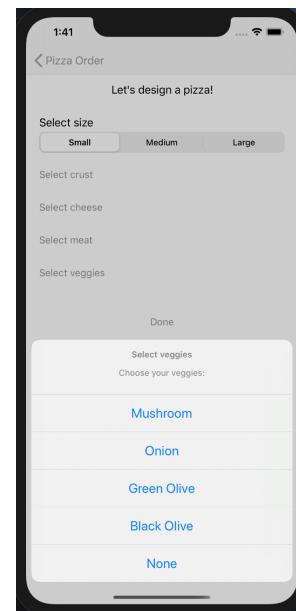
(c) Selecting a crust option



(d) Selecting a cheese option



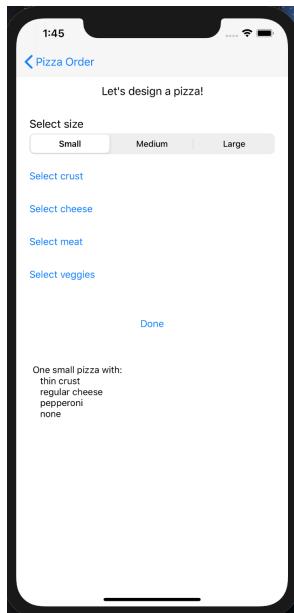
(e) Selecting a meat option



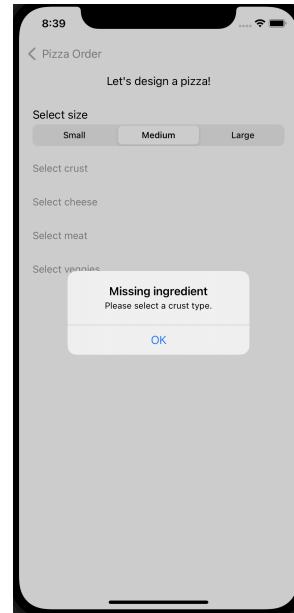
(f) Selecting a veggie option

When one of each option has been selected, the user should click the “Done” button. This should display a summary of the pizza created below the button as shown in figure (g) below. If the user clicks “Done” multiple times without leaving this screen, multiple pizzas are created. (This is how someone might order three identical pizzas for a party.) The summary at the bottom of the screen should only show the one newest pizza.

If the user failed to make a selection for any of the options, an error message resembling (h) below should appear. (If the user missed two or more selections, just choose any one of the error messages to display.) Note that since the segmented controller comes up with “Small” preselected, a small pizza is the default, and you should never see an error for a missing pizza size.

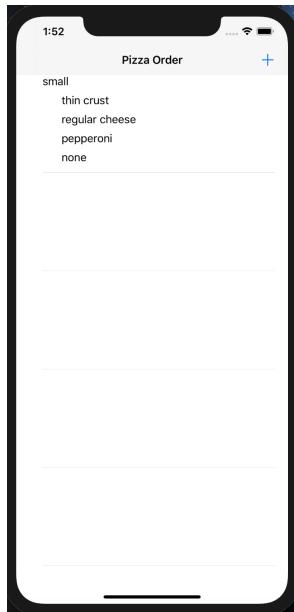


(g) Clicking the Done button
on a complete order

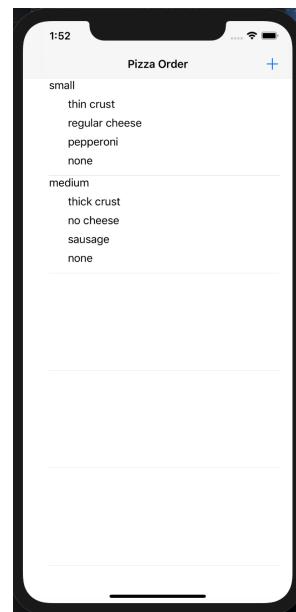


(h) Clicking the Done button
on an incomplete order

If the user now clicks on the Pizza Order nav button, the app should return to the Main VC, and the new pizza should appear at the bottom of the Table VC.



(i) After one pizza has been created



(j) After a second pizza has been created

2 Hints to Get Started

- Create a class called Pizza that includes properties pSize, crust, cheese, meat, and veggies. Then create a variable `pizzaList : [Pizza] = []`. This will contain the pizzas you create.
- Use delegates and protocols to update the Table View.
- For this assignment, you are not required to write code that allows the user to select a pizza from the table. The only action available from the Main VC is the “+” button to create a new pizza.
- To set a title for a VC, set `self.title` to the desired title in that VC’s `viewDidLoad()` method. To set the string that appears in a subsequent VC’s back button, set `self.navigationItem.backButtonTitle` to the string in the **first** VC’s `viewDidLoad()` method.

3 Grading criteria

1. You have UI components as defined. (20%)
2. The Main VC works as defined. (20%)
3. The Pizza Creation VC works as defined. (20%)
4. The alerts and action sheets work as defined. (20%)
5. Error handling for missing ingredients works as expected. (20%)
6. **If the app does not build and run, ZERO points will be given.**
7. The Coding Standard is followed. One point deducted for each violation.

4 General criteria

1. Set the simulator to the newest model supported by the version of Xcode you have installed.
2. I will be looking for good documentation, descriptive variable names, clean logical structure, and adherence to all coding conventions expected of an experienced programmer, as well as those outlined in the Coding Standard document. There will be penalties for failure to meet these standards.
3. Your code must compile and run before submission.
4. Xcode will automatically generate standard headers to your .swift files. Add two lines to each Swift file so that the header includes the following:

```
// Project: LastnameFirstname-HW5  
// EID: xxxxxx  
// Course: CS329E
```