Pierwsze kroki z niestandardowymi UIControls

W tym momencie Twoja aplikacja zawiera dane na wszystkich swoich ekranach, ale ekran szczegółów restauracji jest niekompletny. Nie możesz ustawić oceny w gwiazdkach dla restauracji, nie możesz też dodawać zdjęć ani opinii. Do tej pory używałeś standardowych elementów interfejsu użytkownika Apple. Tu utworzysz niestandardową podklasę klasy UIControl, która wyświetla oceny restauracji w postaci gwiazdek. Zmodyfikujesz tę podklasę, aby użytkownicy mogli ustawić ocenę restauracji, dotykając jej. Następnie wdrożysz formularz recenzji, który umożliwi użytkownikom przesyłanie recenzji restauracji. Pod koniec tego rozdziału dowiesz się, jak tworzyć niestandardowe klasy UIControl, obsługiwać zdarzenia dotykowe i implementować formularze recenzji dla własnych aplikacji. Zostaną omówione następujące tematy:

• Tworzenie własnej podklasy UIControl

• Wyświetlanie gwiazdek w niestandardowej podklasie UIControl

• Dodanie obsługi zdarzeń dotykowych

• Wdrożenie metody rozwijania dla przycisku Anuluj

• Tworzenie klasy ReviewFormViewController

Tworzenie własnej podklasy UIControl

Do tej pory używałeś tylko predefiniowanych elementów interfejsu użytkownika Apple, takich jak etykiety i przyciski. Wystarczyło kliknąć przycisk Biblioteka, wyszukać żądany obiekt i przeciągnąć go do scenorysu. Zdarzają się jednak przypadki, w których obiekty dostarczone przez Apple są albo nieodpowiednie, albo nie istnieją. W takich przypadkach będziesz musiał zbudować własny. Przyjrzyjmy się ekranowi szczegółów restauracji, który widziałeś podczas prezentacji aplikacji:

Możesz zobaczyć grupę pięciu gwiazdek tuż nad przyciskiem Dodaj recenzję. Obecnie scena kontrolera widoku szczegółów restauracji w pliku scenorysu RestaurantDetail i scena kontrolera widoku tabeli w pliku scenorysu ReviewForm mają puste obiekty widoku w miejscu, w którym powinny znajdować się gwiazdy. Utworzysz klasę RatingsView, niestandardową podklasę klasy UIControl, której będziesz używać w obu scenach. Klasa UIControl jest podklasą klasy UIView i jest używana jako nadklasa dla klasy RatingsView, ponieważ wystąpienia RatingsView muszą odpowiadać, gdy użytkownik je kliknie.

Instancja RatingsView będzie wyświetlać oceny jako gwiazdki. Użytkownik będzie mógł również wybrać półgwiazdki. Zacznijmy od stworzenia podklasy klasy UIControl. Wykonaj następujące kroki:

1. Kliknij prawym przyciskiem myszy folder Formularz recenzji i wybierz Nowy plik.

2. iOS powinien być już wybrany. Wybierz klasę Cocoa Touch, a następnie kliknij przycisk Dalej.

3. Skonfiguruj plik w następujący sposób:

Class: RatingsView

Subclass: UIControl

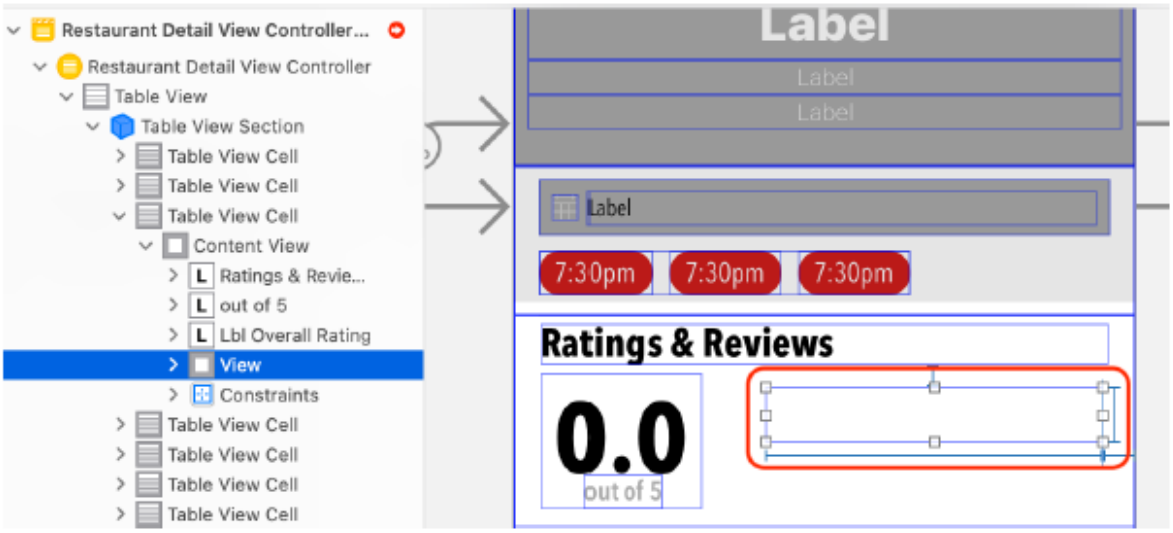
Language: Swift

Click Next.

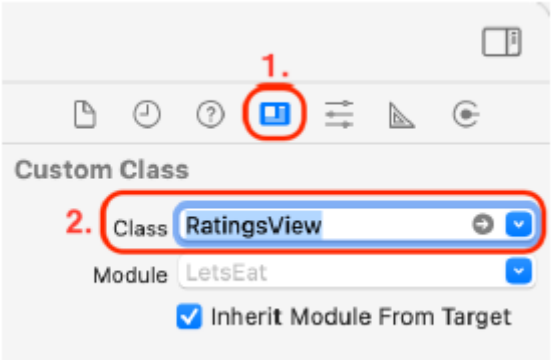
4. Kliknij Utwórz. Plik RatingsView pojawi się w nawigatorze projektu.

Teraz musisz ustawić tożsamość obiektu widoku obok etykiety 0.0 w scenie kontrolera widoku szczegółów restauracji na RatingsView. Wykonaj następujące kroki:

1. Rozwiń folder RestaurantDetail w nawigatorze projektów. Kliknij plik scenorysu RestaurantDetail i wybierz obiekt Widok obok etykiety 0.0, jak pokazano:



2. Kliknij przycisk Inspektor tożsamości. W obszarze Klasa niestandardowa ustaw klasę na RatingsView:



Teraz zmodyfikujmy klasę RatingsView, aby wyświetlała gwiazdki. W następnej sekcji użyjesz w tym celu zasobów graficznych zawartych w pliku Assets.xcassets.

Wyświetlanie gwiazdek w niestandardowej podklasie UIControl

Do tej pory utworzyłeś w swoim projekcie nową podklasę UIControl o nazwie RatingsView. Przypisano również klasę obiektu widoku obok etykiety 0.0 na ekranie Szczegóły restauracji do klasy RatingsView. W dalszej części instancja klasy RatingsView będzie nazywana widokiem ocen (w ten sam sposób instancja klasy UIButton będzie nazywana przyciskiem). W tej sekcji dodasz kod do klasy RatingsView, aby wyświetlić gwiazdki w widoku ocen. Wykonaj następujące kroki:

1. Kliknij plik RatingsView w nawigatorze projektów i usuń cały komentowany kod.

2. Wpisz następujące polecenie po deklaracji klasy RatingsView, aby zadeklarować właściwości klasy:

```
private let filledStarImage = UIImage(named:

"filled-star")

private let halfStarImage = UIImage(named:

"half-star")

private let emptyStarImage = UIImage(named:

"empty-star")

private var totalStars = 5

var rating = 0.0
```

Pierwsze trzy właściwości, filledStarImage, halfStarImage i emptyStarImage, są przypisane do obrazów gwiazd przechowywanych w zasobach pliku xcassets. Właściwość totalStars określa całkowitą liczbę gwiazdek do wylosowania. Właściwość rating służy do przechowywania oceny restauracji. Rodzaje wylosowanych gwiazdek zostaną określone przez wartość rankingu. Na przykład, jeśli ocena wynosi 3,5, widok ocen wyświetli trzy wypełnione gwiazdki, jedną w połowie wypełnioną gwiazdkę i jedną pustą gwiazdkę. Następnie stwórzmy metodę, która narysuje widok ocen na ekranie. Wszystkie podklasy UIView posiadają metodę draw(_:), która odpowiada za rysowanie ich widoków na ekranie. Zastąpisz implementację tej metody nadklasy dla klasy RatingsView. Wykonaj następujące kroki:

1. Dodaj następujący kod w deklaracji klasy po deklaracjach właściwości:

```
override func draw(_ rect: CGRect) {

let context = UIGraphicsGetCurrentContext()

context!.setFillColor(UIColor.systemBackground.

cgColor)

context!.fill(rect)

let ratingsViewWidth = rect.size.width

let availableWidthForStar = ratingsViewWidth /

Double(totalStars)
```

```
let starSidelength = (availableWidthForStar <=
rect.size.height) ? availableWidthForStar :
rect.size.height
for index in 0..<totalStars {
let starOriginX = (availableWidthForStar *
Double(index)) + ((availableWidthForStar -
starSidelength) / 2)
let starOriginY = ((rect.size.height -
starSidelength) / 2)
let frame = CGRect(x: starOriginX,
y: starOriginY, width: starSidelength,
height: starSidelength)
var starToDraw: UIImage!
if (Double(index + 1) <= self.rating) {
starToDraw = filledStarImage
} else if (Double(index + 1) <=
self.rating.rounded()) {
starToDraw = halfStarImage
} else {
starToDraw = emptyStarImage
}
starToDraw.draw(in: frame)
}
}
```

Rozbijmy to:

```
let context = UIGraphicsGetCurrentContext()
```

Tworzy wystąpienie UIGraphicsGetCurrentContext i przypisuje je do kontekstu. Możesz myśleć o tym jako o szkicowniku, w którym będziesz komponował razem elementy interfejsu użytkownika.

```
context!.setFillColor(UIColor.systemBackground. cgColor)
```

Ustawia kolor wypełnienia kontekstu na domyślny systemowy kolor tła.

```
context!.fill(rect)
```

Wypełnia prostokątny obszar określony przez rect kolorem wypełnienia.

let ratingsViewWidth = rect.size.width

let availableWidthForStar = ratingsViewWidth /

Double(totalStars)

let starSidelength = (availableWidthForStar <=

rect.size.height) ? availableWidthForStar :

rect.size.height

Te stwierdzenia określają, jak duża powinna być każda gwiazda. Pierwsza instrukcja pobiera szerokość widoku ocen i przypisuje ją do ratingsViewWidth. Następna instrukcja pobiera szerokość dostępną dla każdej gwiazdy, dzieląc szerokość widoku ocen przez liczbę gwiazdek, które należy narysować. Ta wartość jest przypisana do availableWidthForStar. Przy trzecim stwierdzeniu wyobraź sobie, że każda gwiazda jest zamknięta w prostokącie. Ta instrukcja oblicza, jak długi powinien być każdy bok tego prostokąta, aby zmieścił się w widoku ocen. Jeśli availableWidthForStar jest mniejsze lub równe wysokości widoku ocen, parametr starSideLength jest ustawiany na availableWidthForStar; w przeciwnym razie jest ustawiony na taki sam, jak wysokość widoku ocen.

Załóżmy na przykład, że widok ocen ma szerokość 200 punktów i wysokość 50 punktów. availableWidthForStar byłoby 200/5 = 40. Ponieważ 40 <= 50 daje wartość true, starSideLength zostanie ustawione na 40.

for index in 0..<totalStars {

Ponieważ totalStars jest ustawione na 5, pętla for powtarza się pięć razy.

let starOriginX = (availableWidthForStar *

Double(index)) + ((availableWidthForStar –

starSidelength) / 2

let starOriginY = ((rect.size.height - starSidelength)

/ 2)

let frame = CGRect(x: starOriginX, y: starOriginY,

width: starSidelength, height: starSidelength)

Te instrukcje obliczają początek i rozmiar prostokąta, w którym powinna być narysowana każda gwiazda w widoku ocen. Jest to następnie przypisane do ramki. Wartości początkowe są odsunięte od lewego górnego rogu widoku ocen, a szerokość i wysokość są ustawione na starSidelength. Na przykład, dla pierwszej gwiazdy starOriginX to (40*0.0) + (40-40)/2 = 0. starOriginY to (50 – 40)/2 = 5. ramka byłaby zatem CGRect gdzie x to 0, y to 5, szerokość to 40, a wysokość to 40.

var starToDraw: UIImage!

if (Double(index + 1) <= self.rating) {

starToDraw = filledStarImage

```
} else if (Double(index + 1) <= self.rating.rounded())

{

starToDraw = halfStarImage

} else {

starToDraw = emptyStarImage

}
```

W zależności od wartości właściwości oceny widoku ocen te instrukcje określają, czy gwiazda do narysowania jest wypełniona, wypełniona w połowie czy pusta. Załóżmy na przykład, że ocena wynosi 3,5. Pierwsza gwiazda ma indeks 0. Oznacza to, że Double(0 + 1) <= 3,5 będzie równe 1,0 <= 3,5, co oznacza, że jest prawdziwe. Oznacza to, że pierwsza narysowana gwiazda będzie gwiazdą wypełnioną. To samo dotyczy drugiej i trzeciej gwiazdy.

Czwarta gwiazda ma indeks 3. Oznacza to, że Double(3 + 1) <= 3,5 będzie 4,0 <= 3,5, co oznacza fałsz. Klauzula else oblicza Double(3 + 1) <= 4.0, co daje wartość true, więc czwarta narysowana gwiazda będzie gwiazdą wypełnioną do połowy. Piąta gwiazda ma indeks 4. Oznacza to, że Double(4 + 1) <= 3,5 będzie wynosić 5,0 <= 3,5, co oznacza fałsz. Klauzula else oblicza Double(4 + 1) <= 4.0, co również daje wartość false, więc piąta wylosowana gwiazda będzie pustą gwiazdą.

```
starToDraw.draw(in: frame)
```

Ta instrukcja rysuje gwiazdę w określonej ramce. To cały kod potrzebny do klasy RatingsView. Teraz dodajmy ujście do klasy RestaurantDetailViewController, aby mogła zarządzać tym, co wyświetla widok ocen. Wykonaj następujące kroki:

1. Kliknij plik RestaurantDetailViewController w nawigatorze projektów.

2. Wpisz następujący kod po ścieżce ogólnejRatingLabel:

```
@IBOutlet var ratingsView: RatingsView!
```

Spowoduje to utworzenie gniazdka w klasie RestaurantDetailViewController dla widoku ocen. Masz teraz gniazdko o nazwie ratingsView typu RatingsView, które później połączysz z widokiem ocen w scenorysie.

3. Dodaj metodę przypisywania 3.5 do właściwości ratingu instancji ratingsView. Wpisz następujące polecenie w swoim prywatnym rozszerzeniu po metodzie initialize():

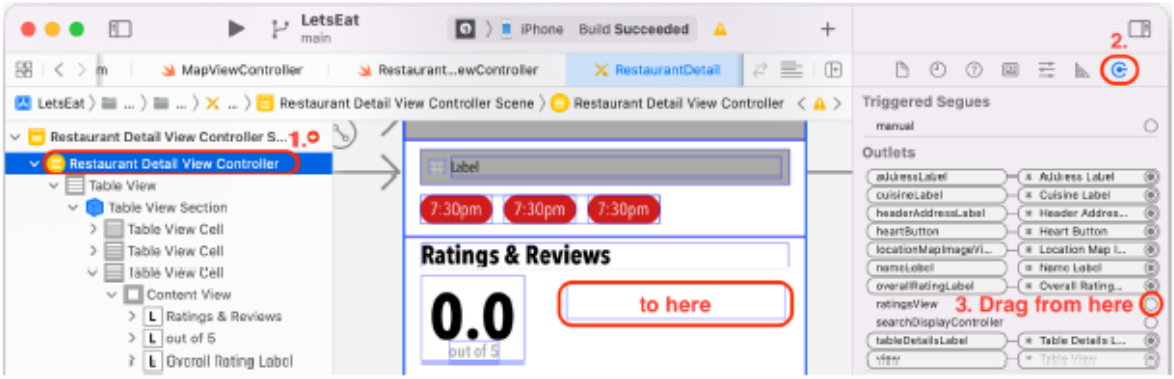
```
func createRating() {

ratingsView.rating = 3.5

}
```

4. Zmodyfikuj metodę initalize() tak, aby wywołała metodę createRating():

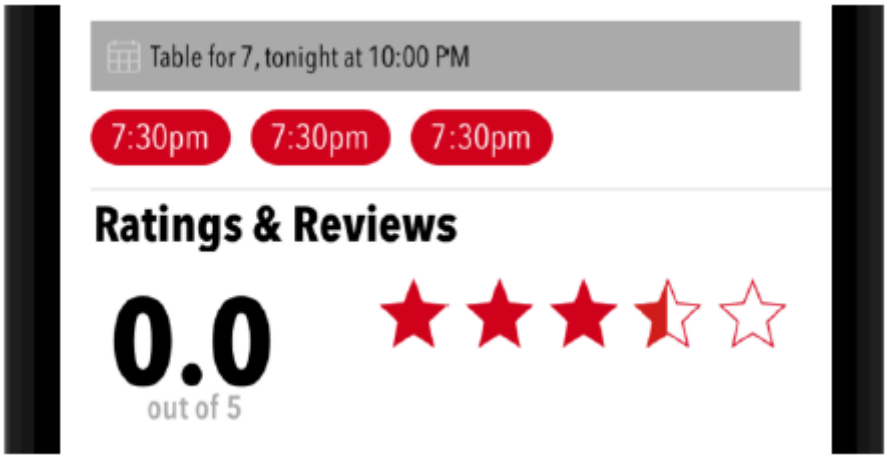
```
func initialize() {

setupLabels()

createMap()

createRating()
```

}

5. Otwórz plik scenorysu RestaurantDetail i wybierz opcję Kontroler widoku szczegółów restauracji w zarysie dokumentu. Kliknij w przycisk Inspektora połączeń. Przeciągnij z gniazdka ratingsView do widoku ocen:



Zbuduj i uruchom swój projekt i przejdź do ekranu szczegółów restauracji dla dowolnej restauracji. Widok ocen powinien zawierać trzy i pół gwiazdki:



Utworzyłeś i zaimplementowałeś widok ocen na ekranie szczegółów restauracji. Wygląda świetnie, ale w tej chwili widok ocen nie reaguje po dotknięciu. W następnej sekcji sprawisz, że będzie reagował na zdarzenia dotykowe, aby użytkownik mógł wybrać ocenę.

Dodanie obsługi zdarzeń dotykowych

Obecnie klasa RestaurantDetailViewController ma gniazdo ratingsView połączone z widokiem ocen na ekranie szczegółów restauracji. Wyświetla ocenę wynoszącą trzy i pół gwiazdki, ale nie możesz zmienić oceny. Musisz obsługiwać zdarzenia dotykowe, aby widok ocen reagował na dotknięcia. Aby obsługiwać zdarzenia dotyku, zmodyfikuj klasę RatingsView, aby śledzić dotknięcia użytkownika na ekranie i używać ich do określania oceny. Wykonaj następujące kroki:

1. Kliknij plik RatingsView w nawigatorze projektów i dodaj następującą właściwość po metodzie draw(_:):

override var canBecomeFirstResponder: Bool {

true

}

canBecomeFirstResponder to właściwość UIControl, która określa, czy obiekt może stać się pierwszym obiektem odpowiadającym. Widok ocen musi stać się pierwszym respondentem, który zareaguje na zdarzenia dotykowe. Ta metoda domyślnie zwraca wartość false, ponieważ nie wszystkie elementy interfejsu użytkownika muszą reagować na dotknięcia. Zastąpisz tę metodę, aby zwracała wartość true, aby widok ocen mógł stać się pierwszą odpowiedzią.

2. Aby śledzić dotknięcia użytkownika na ekranie, dodaj następujący kod po właśnie dodanej właściwości canBecomeFirstResponder:

```
override func beginTracking(_ touch: UITouch, with

event: UIEvent?) -> Bool {

guard self.isEnabled else {

return false

}

super.beginTracking(touch, with: event)

handle(with: touch)

return true

}
```

Rozbijmy to:

override func beginTracking(_ touch: UITouch, with event:

UIEvent?) -> Bool {

Ta metoda jest jedną z metod zadeklarowanych w klasie UIControl. Jest wywoływana, gdy dotyk użytkownika znajduje się w granicach wystąpienia UIControl. Lokalizacja, rozmiar, ruch i siła dotyku na ekranie są przechowywane w instancji UITouch. Ta metoda musi zwrócić wartość true, jeśli chcesz śledzić dotknięcia użytkownika. Zastępujesz tę metodę, aby móc zdefiniować niestandardowe zachowanie, gdy użytkownik dotknie widoku ocen.

guard self.isEnabled else {

return false

}

Właściwość isEnabled jest sprawdzana w tej instrukcji strażnika, aby sprawdzić, czy widok ocen jest włączony. Jeśli widok ocen nie jest włączony, dotknięcia użytkownika nie będą śledzone.

super.beginTracking(touch, with: event)

Wywołuje implementację nadklasy tej metody. To zajmie się inicjalizacją wymaganą przez klasę nadrzędną.

handle(with: touch)

Przekażesz instancję UITouch do tej metody, która będzie wykonywana przy każdym dotknięciu. W następnym kroku zadeklarujesz i zdefiniujesz tę metodę. return true Śledzi dotknięcia użytkownika, gdy włączony jest widok ocen.

3. Zobaczysz błąd, ponieważ nie zaimplementowałeś jeszcze handle(with:), więc utwórz prywatne rozszerzenie dla RatingsView po całym kodzie w pliku i wpisz do niego następujący kod:

```
private extension RatingsView {

func handle(with touch: UITouch) {

let starRectWidth = self.bounds.size.width /

Double(totalStars)

let location = touch.location(in: self)

var value = location.x / starRectWidth

if (value + 0.5) < value.rounded(.up) {

value = floor(value) + 0.5

} else {

value = value.rounded(.up)

}

updateRating(with: value)

}

}
```

handle(with:) obliczy wartość oceny na podstawie lokalizacji dotyku użytkownika. Jako parametr przyjmuje instancję UITouch. Najpierw do starRectWidth przypisywana jest szerokość widoku ocen podzielona przez 5. Następnie lokalizacja instancji UITouch w widoku ocen jest przypisywana do lokalizacji. Następnie wartość jest przypisywana do pozycji x lokalizacji podzielonej przez starRectWidth. Oznacza to, że wartość będzie zawierała zakres wartości od 0 do 5. Następnie instrukcja if oblicza ocenę odpowiadającą pozycji dotyku i wywołuje updateRating(with:), przekazując do niej wartość. Wdrożysz updateRating(with:) w następnym kroku. Aby zrozumieć, jak działa instrukcja if, załóżmy, że szerokość widoku ocen wynosi 200. starRectWidth zostanie ustawione na 200/5 = 40. Załóżmy, że użytkownik dotknął ekranu w pozycji x = 130 , y = 17, co odpowiada punktowi między trzecią a czwartą gwiazdą. wartość zostanie przypisana 130/40 = 3,25. Tak więc instrukcja if wyliczyłaby (3,25 + 0,5 < 3,25.rounded(.up), co daje (3,75 < 4,0), co zwraca prawdę, zatem wartość zostanie ustawiona na floor(3,25) + 0,5, co stanie się 3,0 + 0,5 , czyli 3.5. Czyli updateRating(with:) otrzyma wartość 3.5

4. Zobaczysz błąd, ponieważ nie zaimplementowałeś jeszcze updateRating(with:), więc wpisz następujący kod do prywatnego rozszerzenia po uchwycie (z:) metoda:

```
func updateRating(with newValue: Double) {

if (self.rating != newValue && newValue >= 0 &&

newValue <= Double(totalStars)) {

self.rating = newValue

}
```

}

updateRating(with:) sprawdza, czy wartość nie jest równa aktualnej ocenie i wynosi od 0 do 5. Jeśli tak, wartość jest przypisywana do oceny. Kontynuując poprzedni przykład, ponieważ 3,5 wynosi od 0 do 5, zostanie przypisane do oceny, jeśli nie jest równe bieżącej wartości oceny.

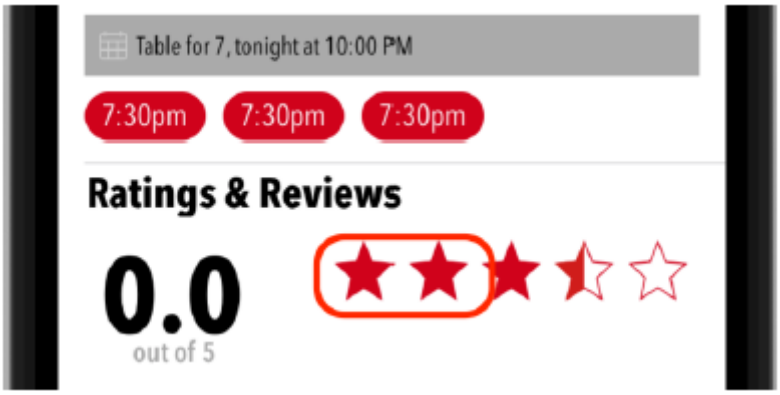
5. Widok ocen będzie musiał zostać przerysowany po zmianie oceny, aby wyświetlać prawidłowy stan gwiazdek. Zmodyfikuj deklarację właściwości oceny w następujący sposób:

```
var rating = 0.0 {

didSet {

setNeedsDisplay()

}

}
```

W tym miejscu zdefiniowano obserwatora właściwości, który monitoruje zmiany wartości właściwości oceny. Za każdym razem, gdy zmienia się ocena, wywoływana jest funkcja setNeedsDisplay() i widok ocen jest rysowany od nowa. Ponieważ ekran jest przerysowywany tylko w przypadku zmiany oceny, występuje niewielka korzyść w zakresie wydajności. Dodałeś cały kod, który jest niezbędny, aby widok ocen reagował na dotknięcia. Teraz musisz zaktualizować klasę RestaurantDetailViewController, aby ustawić właściwość isEnabled dla widoku ocen. Kliknij plik RestaurantDetailViewController w nawigatorze projektów i zmodyfikuj metodę createRating() w następujący sposób:

```
func createRating() {

ratingsView.rating = 3.5

ratingsView.isEnabled = true

}
```

Ustawienie właściwości isEnabled na true pozwala widokowi ocen stać się pierwszym odpowiadającym i rozpocząć śledzenie dotknięć, co spowoduje wyzwolenie handle(with:) w celu obliczenia oceny na podstawie pozycji dotknięcia, co z kolei wywołuje funkcję updateRating(with:) do zaktualizować widok ocen. Zbuduj i uruchom swój projekt. Stuknięcie w widok ocen zmienia teraz ocenę w zależności od tego, gdzie stuknąłeś. Stuknij między pierwszą a drugą gwiazdką, jak pokazano:

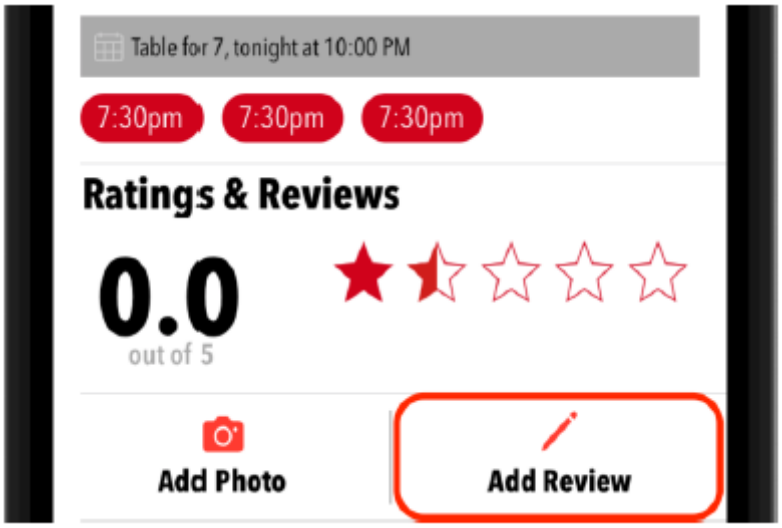


Ocena zmieni się na półtorej gwiazdki. Ostatecznie obliczysz ogólną ocenę, sumując wszystkie oceny przesłane przez użytkowników na ekranie Formularz recenzji. Po dotknięciu przycisku Dodaj recenzję

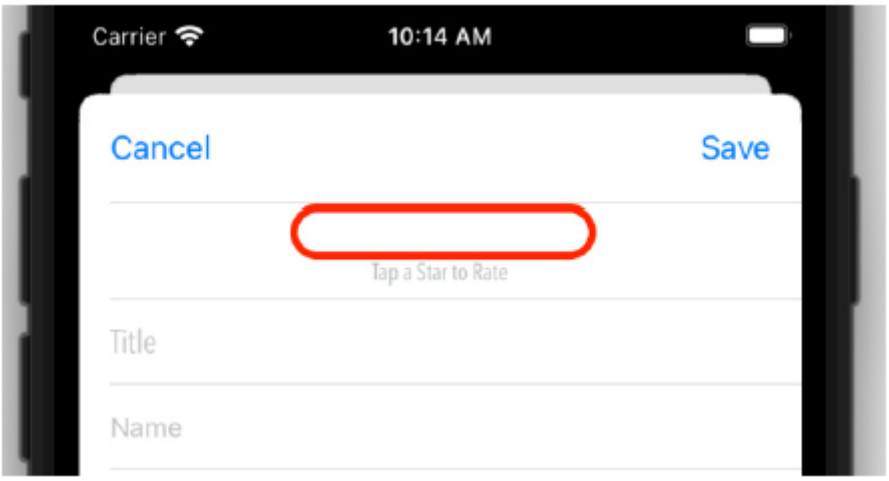
zostanie wyświetlony ekran Formularz recenzji, ale nie można go odrzucić ani ustawić oceny. W następnej sekcji skonfigurujesz przycisk Anuluj, aby po dotknięciu zamykał ekran formularza recenzji.

Implementacja metody rozwijania dla przycisku Anuluj

Rzućmy okiem na ekran formularza recenzji. Przejście między przyciskiem Dodaj recenzję a ekranem Formularza recenzji zostało już dla Ciebie wykonane. Zbuduj i uruchom swój projekt, przejdź do ekranu Szczegóły restauracji i naciśnij przycisk Dodaj recenzję:



Zostanie wyświetlony ekran Formularz recenzji (zwróć uwagę, że górna komórka widoku tabeli ma puste miejsce w miejscu, w którym powinien znajdować się widok ocen, który dodasz później):



Gdy na ekranie pojawi się ekran formularza recenzji, nie można go odrzucić, ponieważ akcje przycisków Zapisz i Anuluj nie zostały skonfigurowane. Podobnie jak w przypadku ekranu Lokalizacje, musisz zaimplementować metodę rozwijania, aby zamknąć ekran Formularz recenzji. Wykonaj następujące kroki:

1. Kliknij plik RestaurantDetailViewController w nawigatorze projektów.

2. Zaimplementuj metodę unwind w następujący sposób w rozszerzeniu prywatnym, przed metodą createRating():

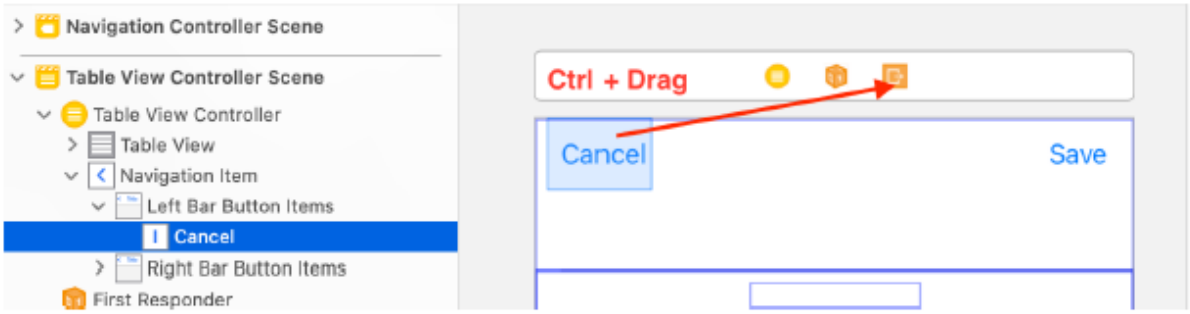
@IBAction func unwindReviewCancel(segue:

UIStoryboardSegue) {

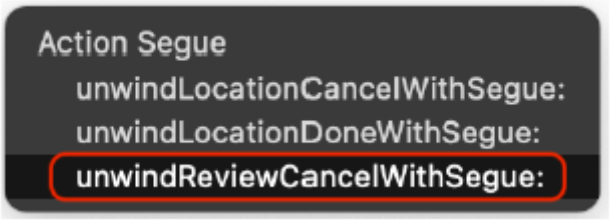
}

Ta metoda zostanie wywołana, gdy ekran formularza recenzji przejdzie do ekranu szczegółów restauracji.

3. Otwórz plik scenorysu ReviewForm i naciśnij Ctrl + przeciągnij z przycisku Anuluj do ikony Zakończ w doku sceny, jak pokazano:



4. Wybierz unwindReviewCancelWithSegue z menu podręcznego:



Zbuduj i uruchom swój projekt. Możesz teraz zamknąć ekran formularza recenzji, dotykając przycisku Anuluj. Następnie spójrzmy na przycisk Zapisz. Utworzysz kontroler widoku dla ekranu Formularz recenzji, aby przetwarzać dane w polach ekranu Formularz recenzji po dotknięciu przycisku Zapisz. Zrobisz to w następnej sekcji.

Tworzenie klasy ReviewFormViewController

Aby przetworzyć dane wejściowe użytkownika, utworzysz klasę ReviewFormViewController, która będzie kontrolerem widoku dla ekranu formularza recenzji. Na razie skonfigurujesz tę klasę, aby pobierała wszystkie wartości z pól ekranu formularza recenzji i drukowała je w obszarze debugowania. Wykonaj następujące kroki:

1. Kliknij prawym przyciskiem myszy folder ReviewForm i wybierz Nowy plik.

2. iOS powinien być już wybrany. Wybierz klasę Cocoa Touch, a następnie kliknij przycisk Dalej.

3. Skonfiguruj plik w następujący sposób:

Class: ReviewFormViewController

Subclass: UITableViewController

Also create XIB: Unchecked

Language: Swift

Click Next.

4. Kliknij Utwórz. Plik ReviewFormViewController pojawi się w nawigatorze projektów.

5. Usuń wszystko po metodzie viewDidLoad() i cały zakomentowany kod. Dodaj następujące punkty sprzedaży po deklaracji klasy. Odpowiadają one polom na ekranie Formularza recenzji:

@IBOutlet var ratingsView: RatingsView!

@IBOutlet var titleTextField: UITextField!

@IBOutlet var nameTextField: UITextField!

@IBOutlet var reviewTextView: UITextView!

6. Musisz również skonfigurować akcję dla przycisku Zapisz. Dodaj następujący kod po metodzie viewDidLoad():

@IBAction func onSaveTapped(_ sender: Any) {

print(ratingsView.rating)

print(titleTextField.text as Any)

print(nameTextField.text as Any)

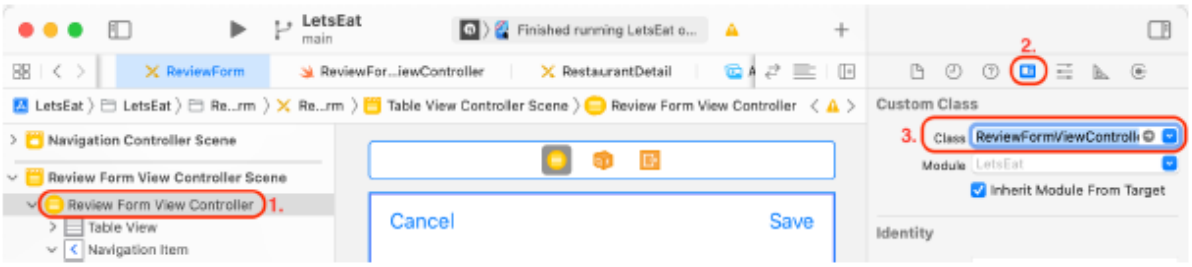
print(reviewTextView.text as Any)

dismiss(animated: true, completion: nil)

}

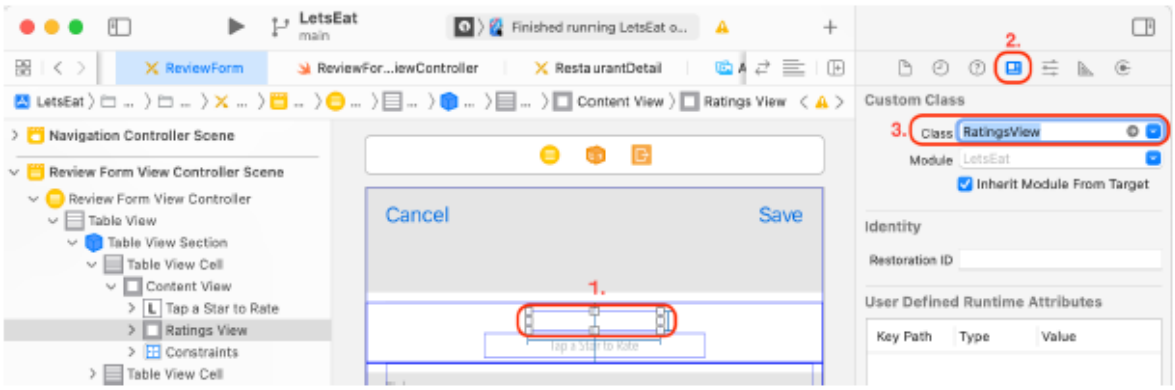
Ta metoda wyświetla zawartość pól ekranu formularza recenzji w obszarze debugowania i odrzuca ją. Teraz połączmy gniazda w klasie ReviewFormViewController z elementami interfejsu użytkownika w scenie kontrolera widoku tabeli w pliku scenorysu ReviewForm w następujący sposób:

1. Kliknij plik scenorysu ReviewForm w nawigatorze projektów i kliknij ikonę Kontroler widoku tabeli wewnątrz sceny Kontroler widoku tabeli. Kliknij w przycisk Inspektora tożsamości. W obszarze Klasa niestandardowa ustaw klasę na ReviewFormViewController:

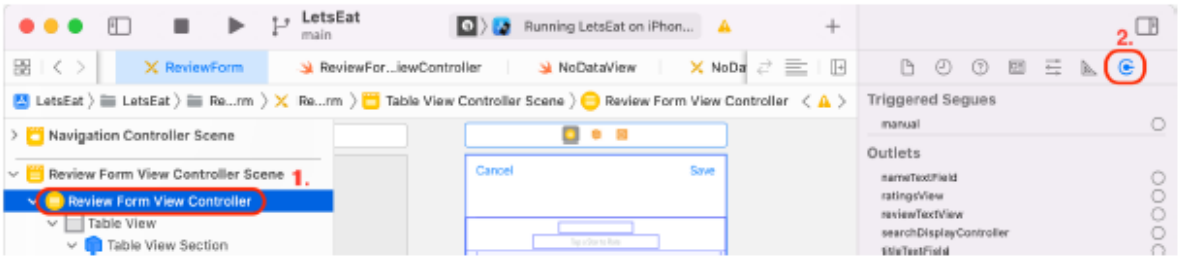


Zauważ, że nazwa Scena kontrolera widoku tabeli zmieni się na Scena kontrolera widoku formularza przeglądania.

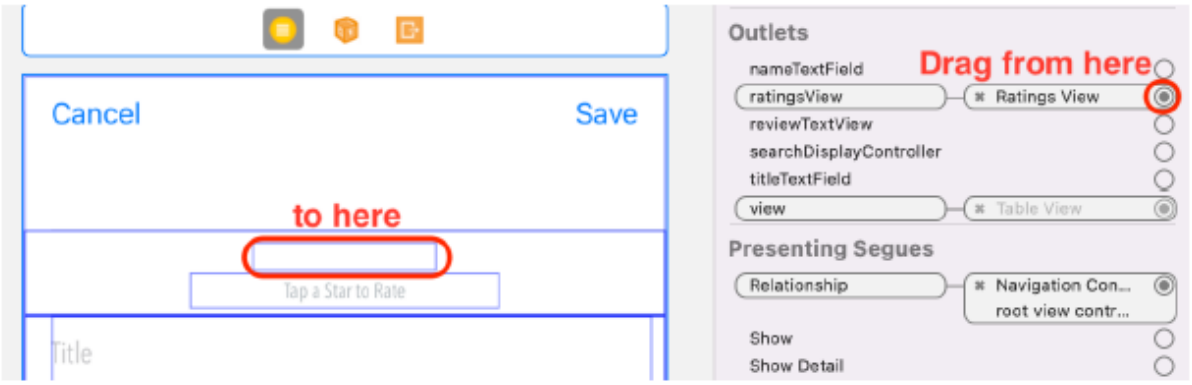
2. Kliknij widok wewnątrz widoku zawartości pierwszej komórki widoku tabeli, jak pokazano, kliknij przycisk Inspektor tożsamości i w obszarze Klasa niestandardowa ustaw opcję Klasa na Widok ocen. Nazwa widoku zmieni się na Widok ocen:



3. Następnie połączysz gniazda. Kliknij ikonę Kontroler widoku formularza recenzji w konspekcie dokumentu, a następnie kliknij przycisk Inspektora połączeń:



4. Połącz gniazdo ratingsView z widokiem ocen:



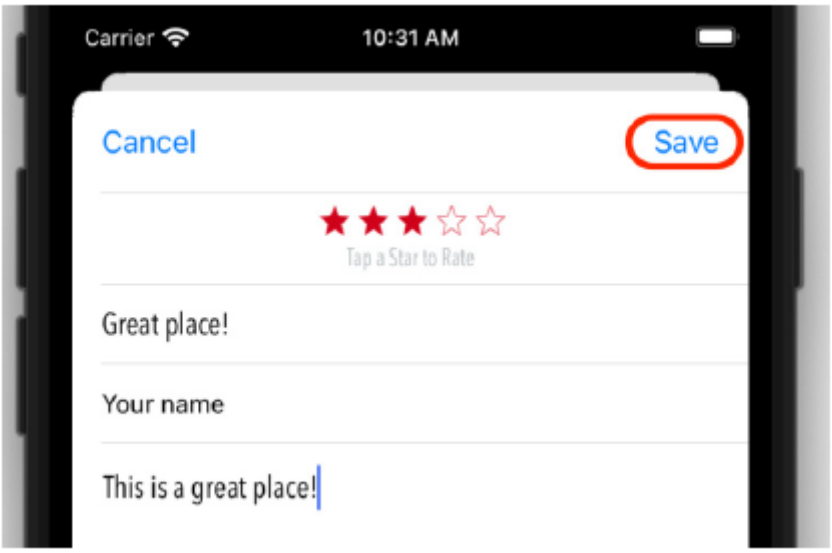
5. Podłącz wyjście titleTextField do pierwszego pola tekstowego:

6. Podłącz wyjście nameTextField do drugiego pola tekstowego:

7. Podłącz wyjście reviewTextView do widoku tekstu:

8. Na koniec połącz akcję onSaveTapped: z przyciskiem Zapisz:

Zbuduj i uruchom swoją aplikację. Przejdź do ekranu Formularz recenzji, ustaw ocenę, dodaj przykładowy tekst do pól i naciśnij przycisk Zapisz:



Zobaczysz, że wprowadzone dane pojawią się w obszarze Debugowanie:

Gratulacje! Ekran formularza recenzji może teraz akceptować dane wprowadzone przez użytkownika.

Podsumowanie

Utworzyłeś od podstaw nową niestandardową podklasę UIControl, RatingsView, i dodałeś ją do ekranów Szczegóły restauracji i Formularz recenzji. Skonfigurowano go tak, aby reagował na dotknięcia, dzięki czemu użytkownik może ustawić ocenę restauracji na ekranie formularza recenzji. Na koniec zaimplementowano klasę ReviewFormViewController, kontroler widoku dla ekranu formularza recenzji i skonfigurowano akcje przycisków Anuluj i Zapisz, aby użytkownik mógł odrzucić ekran formularza recenzji lub przesłać recenzję.