

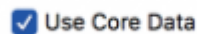
TEMAT: Składowanie danych i zasoby sieciowe.

TREŚĆ ZADANIA

Cel: implementacja składowania danych oraz wykorzystania zasobów sieciowych w aplikacji iOS.

Składowanie danych (1 pkt)

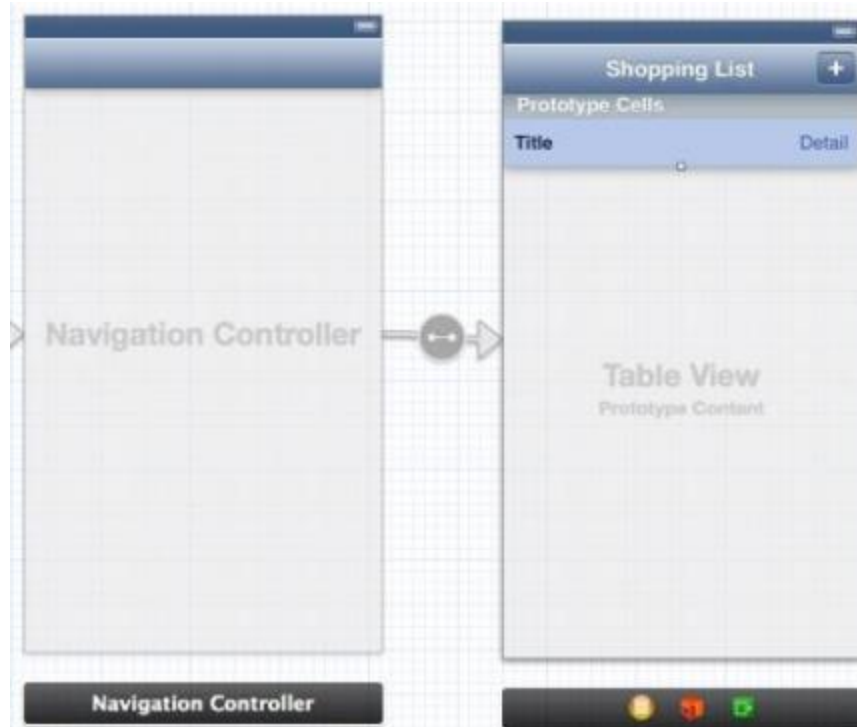
1. Framework *Core Data* odpowiada za obsługę bazy danych (przechowywanie danych i zarządzanie nimi) w sposób zorientowany obiektowo. Jest to jedna z kilku dostępnych opcji składowania danych na platformie iOS.
2. W ramach zadania należy stworzyć aplikację do zarządzania listami zakupów. Powinna zapewniać możliwość wykonywania podstawowych operacji CRUD (create, read, update, delete).
3. W tym celu utwórz nowy projekt na bazie szablonu *Single View Application*. Wybierz opcje *Use Core Data* oraz *Use Automatic Reference Counting*.



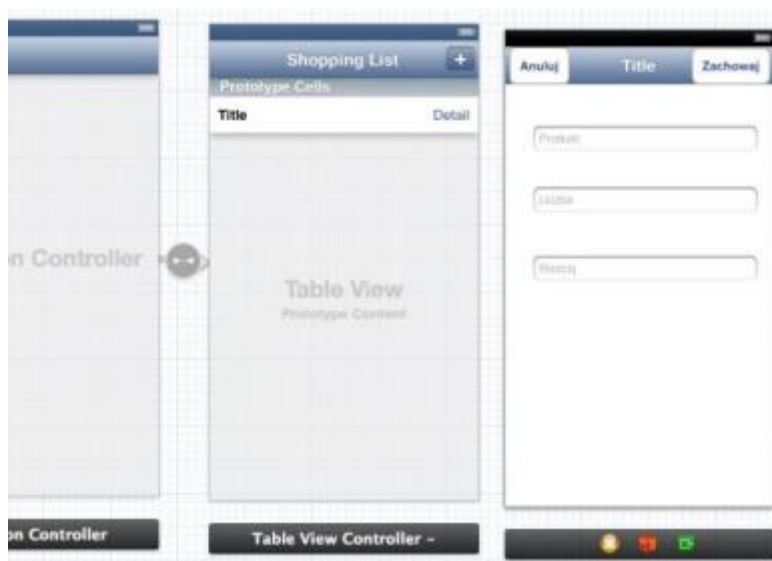
4. Dzięki zaznaczeniu opcji *Use Core Data* w drzewie projektu powinien pojawić się plik *<nazwaProjektu>.xcdatamodel*. Po otwarciu tego pliku możliwe jest korzystanie z edytora modelu danych (*Core Data Model Editor*). W tym miejscu należy zdefiniować model obiektu. Dodaj więc jedną encję (*Add Entity*), która będzie reprezentowała elementy dodawane do listy zakupów. Dodaj do niej następujące atrybuty (przycisk + w sekcji *Attributes*):
 - Product
 - Number
 - Type
5. Atrybuty powinny być typu *String*.



6. Kolejnym krokiem jest utworzenie interfejsu użytkownika. W pliku *Storyboard* należy dodać do kontrolera aplikacji przycisk z identyfikatorem *Add*. Przycisk powinien znajdować się w prawym górnym rogu widoku aplikacji.
7. Następnie należy umieścić widok tabel (*TableView*), a także komórkę (*TableViewCell*). Zaznacz opcję *prototype cell* i staw jej styl na *Right Detail*.



8. Dodaj kolejny kontroler widoku i dodaj w górnej części jego widoku pasek nawigacji. W pasku tym umieść dwa przyciski: *Cancel* oraz *Save*.
9. W zawartości widoku nowego kontrolera dodaj trzy pola tekstowe nazwane odpowiednio: *product*, *number* i *type*.
10. Utworzone okno powinno być uruchamiane po wciśnięciu przycisku + w pierwszym kontrolerze.



11. W tym celu przytrzymaj przycisk Ctrl i dokonaj przeciągnięcia od przycisku + do nowo utworzonego widoku. W menu, które się pojawi, wybierz opcję *modal*.

12. Od tego momentu uruchomienie aplikacji i wciśnięcie przycisku + powinno powodować wyświetlenie nowego okna z trzema polami tekstowymi.
13. Do obsługi kolejnych widoków niezbędne jest zdefiniowanie nowych klas i dodanie ich do projektu. W tym celu na katalogu *ShoppingList* należy wybrać z menu otwieranego po kliknięciu prawym przyciskiem myszy *New File*. Trzeba zaznaczyć opcję *Objective-C class*. Klasę należy nazwać *ShoppingViewController*. Jako podklasę należy wybrać *UITableViewController*. Utworzoną klasę kontrolera trzeba skojarzyć z kontrolerem widoku widocznym w *storyboard*.
14. Analogiczne kroki należy wykonać dla jeszcze jednej klasy *DetailShoppingViewController*. Powinna zostać skojarzona z kontrolerem widoku szczegółów. Jako jej klasę nadrzędną (po której dziedziczy) należy ustawić *UIViewController*.
15. Po utworzeniu klas przypisanych do widoków możliwe jest powiązanie pól tekstowych z odpowiednim kontrolerem (*DetailShoppingViewController*).

Zasoby sieciowe (1 pkt)

1. Stwórz widok odpowiadający za konwersję walut.
2. Dodaj do widoku przycisk oraz etykietę. Stwórz odpowiednie pola w klasie kontrolera, aby umożliwić ich obsługę.
3. Wykonanie zapytania ubiegającego się o aktualne wartości walut jest możliwe poprzez wykorzystanie usługi udostępnianej przez fixer.io, np.:
<http://api.fixer.io/latest?base=USD&symbols=PLN>
4. W metodzie obsługującej zdarzenie kliknięcia przycisku dodaj kod konstruuujący odpowiedni adres URL:

```
NSString *fromCurrency=@"USD";
NSString *toCurrency=@"PLN";
NSString *stringUrl = [NSString
stringWithFormat:@"http://api.fixer.io/latest?base=%@&symbols=%@"
,fromCurrency, toCurrency];
```

5. Wykonaj połączenie *NSURLConnection*:

```
NSURL *url = [NSURL URLWithString:stringUrl];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
[NSURLConnection sendAsynchronousRequest:request
queue:[NSOperationQueue mainQueue]
completionHandler:^(NSURLResponse *response NSData *data, NSError
*connectionError)
{
    if (data.length > 0 && connectionError == nil)
    {
        NSDictionary *dicYourResponse = [NSJSONSerialization
JSONObjectWithData:data options:0 error:NULl];

    }
}];
```

6. Połączenie wykonywane jest asynchronicznie. Dlatego definiowana jest metoda do obsługi odpowiedzi z usługi. W tej metodzie (w obrębie bloku warunkowego if) dodaj obsługę odpowiedzi - ustaw odpowiednią wartość ze zwróconego słownika pod kluczem

"rates". Tam będzie zagnieżdżony słownik, w którym trzeba wybrać wartość pod kluczem "PLN".

7. Zmodyfikuj program tak by wydobywać informacje na temat wybranych kursów walut (dwa dropdown boxy i przycisk).