iOS Programming Test
Nenno’s Pizza

The goal is to implement an application that allows the user to order pizzas and drinks.

Remote service calls will provide information about available goods in JSON format. The calls to fetch goods for each category are listed below:

Ingredients:

https://api.myjson.com/bins/ozt3z

(Backup http://next.json-generator.com/api/json/get/EkTFDCdsG)

Provides an array of *Ingredient* objects, defined as:

```
[{
     "id"            : Long    // unique identifier
     "name"          : String  // display name
     "price"         : Double  // price
}]
```

Drinks:

https://api.myjson.com/bins/150da7

(Backup http://next.json-generator.com/api/json/get/N1mnOA_oz)

Provides an array of *Drink* objects, defined as:

```
[{
     "id"            : Long    // unique identifier
     "name"          : String  // display name
     "price"         : Double  // price
}]
```

Pizzas:

https://api.myjson.com/bins/dokm7

(Backup http://next.json-generator.com/api/json/get/NybeIGcjz)

Provides information about available pizza items, defined as:

```
{
     "pizzas"        : [PIZZA] // array of Pizza objects
     "basePrice"    : Double  // the base price of a pizza
}
```

Where a *Pizza* object is defined as:

```
{
     "name"          : String            // Display name
     "ingredients"   : [INGREDIENT ID]   // ingredients
     "imageUrl"      : String            // Optional image url
}
```

Where `INGREDIENT ID` is the unique identifier of an ingredient in the list of available ingredients. All properties are mandatory except `imageUrl` which is optional.

Please note that the json responses can also be found in the `responses` folder, you can use those during development for unit testing or as a backup in case the service is temporarily unavailable.

The final price of a pizza will be the sum of the base price and the price of each ingredient.

Once pizzas and drinks are added to the shopping cart, the user can proceed to checkout, optionally removing items from the list before completing the order.

To checkout the app will use the service call:

http://httpbin.org/post

Providing a JSON of the form:

```
{
     "pizzas"  : [PIZZA] // array of pizza objects
     "drinks"  : [DRINK ID] // array of drink ids
}
```

Requirements:

- A main task is described below, together with secondary tasks. You don’t have to complete all the tasks, it’s better to provide less functionalities but with clean and well organized code

- A suggested UI, complete with details and UX flow, is provided at https://invis.io/TJAY937WA. Assets referenced by the UI design are provided in the *icons* folder, as both png and svg files. UI specs can be found in *specs/specs.html*. Follow the specified design as closely as possible. It is recommended that you view the InVision prototype in a desktop browser
- A Test Driven Development (TDD) approach is strongly advised and will yield extra points!
- You can use any framework and dependency manager (e.g. CocoaPods, Swift Packet Manager etc) you deem appropriate, although CocoaPods is preferred
- The recommended language is Swift 4 but Objective-C will be accepted as well.

Main Task

- Fetch the menu via the *pizzas*, *ingredients* and *drinks* apis and display pizzas (with their ingredients and price) and drinks as indicated in the design
- Select pizzas and drinks and add them to the shopping cart (note about the design: the drop down “ADDED TO CART” bar should stay on screen 3 seconds and then disappear)
- Checkout

Additional Tasks

1. Edit a single pizza by adding and removing ingredients
2. Allow creating a custom pizza from scratch
3. Remove items from the cart
4. Persist the current cart between app launches.