VRCode
Products App

A Demo Product Inventory iOS App in Objective-C

vrcode.co/ProductsApp

Approach and Design Decisions

The App was started from scratch with a story board for the initial view controller. The very first classes created where for the Product model; categories where added to it in order to break down the methods and allow easy refactoring by keeping the Product properties in the base class.

The database schema was designed next, consisting of only one table for simplicity and speed of development. SQLite is used for the DB. A DatabaseManager class was created to take care of handling all the tasks related to open, creating and closing the DB.

Test cases where then created for the operations that were needed to interact with the DB, and once in place, code was added to the corresponding categories for the Product class. Code was considered ready once all the tests passed.

A static class with a couple of utility methods were created to assist with a couple of functions, namely get the full path for a file in the Documents folder, and make a dictionary out of a JSON string. Tests cases for these were created as well.

From there, all attention was focused on creating view controllers with their corresponding NIB files and wiring them; anything that could be set in the NIB file was done so in order to avoid having to perform those tasks in code.

View controllers used in this App

- **VRCMainViewController** - The initial view controller that contains 2 buttons:
    - Create Product - Displays an ActionSheet that allows to create 1 of 3 sample products (using a JSON Product representation).
    - Show Product - Displays VRCProductsListViewController.

- **VRCProductsListViewController** - Contains a TableView that shows all available products from the database. Tapping on an item opens the VRCProductViewController to display the product details.
  It also includes an Add (+) button that when pressed presents a view controller that allows to create new Products.

- **VRCProductViewController** - Displays the Product details in a grouped TableView. It allows direct editing by tapping on the corresponding field and changing the text/number directly or by presenting a view controller that allows to modify the selected item. Items such as Colors or Stores can be directly added or removed directly from their section. This controller is also used to add new Products.

- **VRCAddColorViewController** - View controller used to add new Colors to a Product.

- **VRCAddStoreViewController** - View controller used to add new Stores (and quantity for that store) to a Product.

- **VRCPhotoViewController** - This view controller is used to display the photo of a product in more detail.

Custom Table View Cells in the App

- **VRCCellLabel** - A default custom table view cell to be expanded for extra functionality in the future.

- **VRCCellTextField** - A table view cell with an embedded and managed Text Field.

- **VRCCellTextView** - A cell with an embedded and managed Text View for long texts.

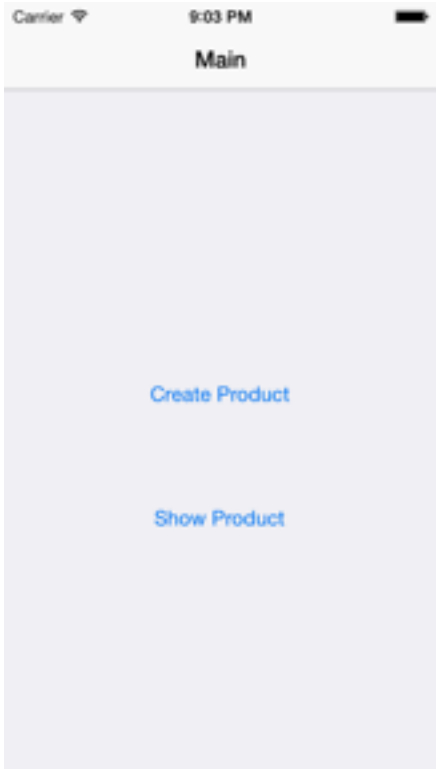
- **VRCCellNumberField** - Inherits from VRCCellTextField, but it's customized for number entry.

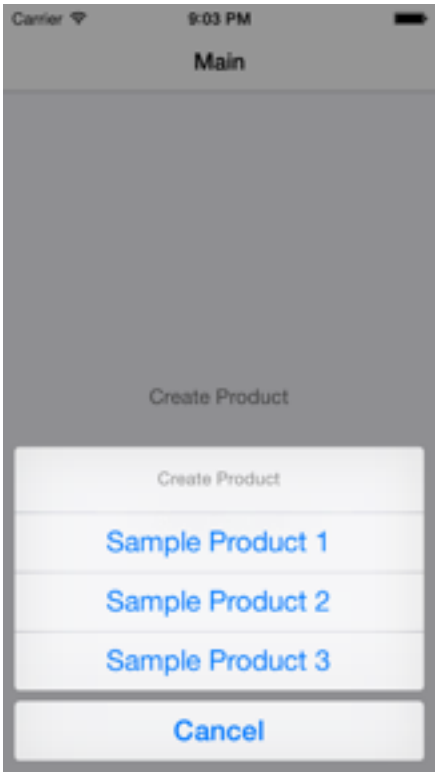
- **VRCCellPictureSelector** - Custom cell that displays images and allows to set it or change it.

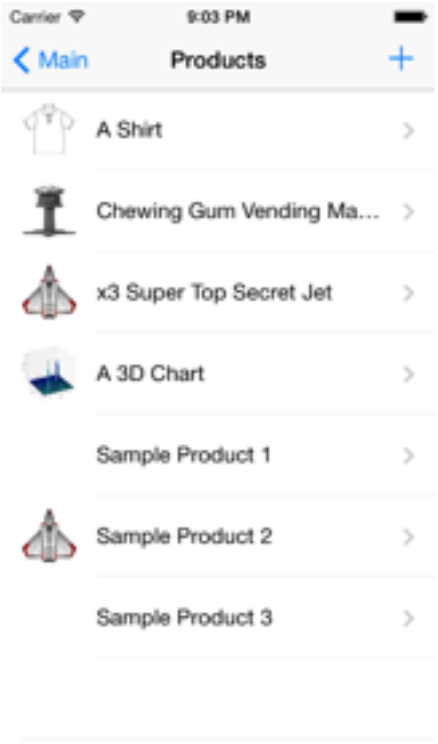
Areas of Improvement

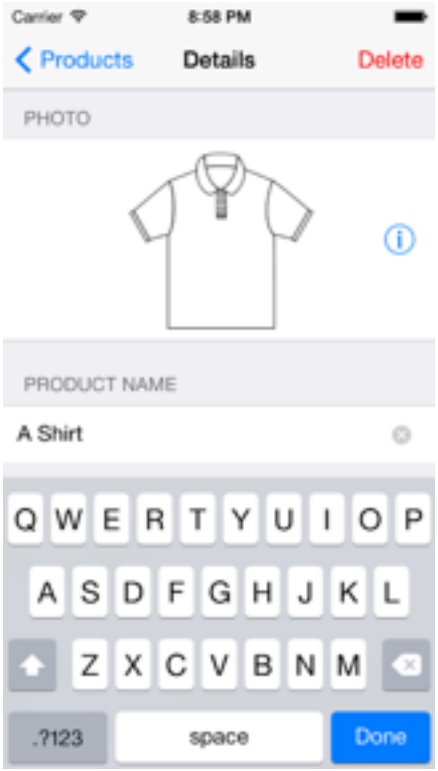
- Using FMDB if multithreading is needed
- Implement localization
- Allow to load only specific fields into properties for improved memory usage
- Atomic field updates, so only changed fields are actually written
- Products List lazy loading to support very large numbers of products
- Add entry validation to number fields
- Implement field value dependency, so for instance, the sale price has to be lower than the regular price
- Implement searching in the Products List
- Normalize DB by putting Colors and Stores in separate tables and adding relations
- Creating more test cases to increase code coverage
- Write UI Automation scripts to allow for easy UI regression testing

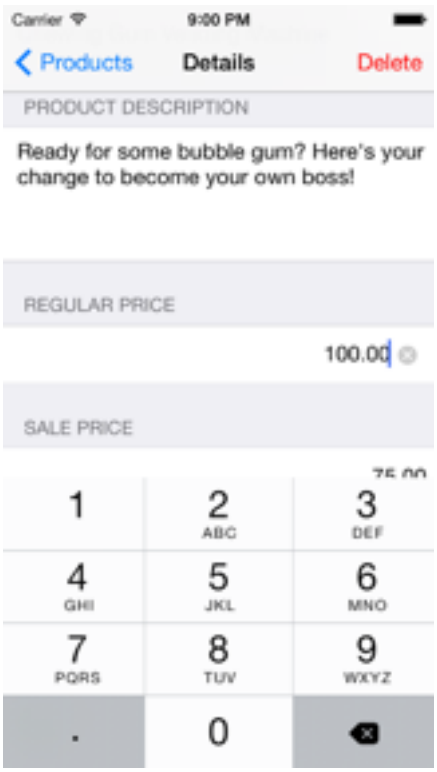
Screenshots

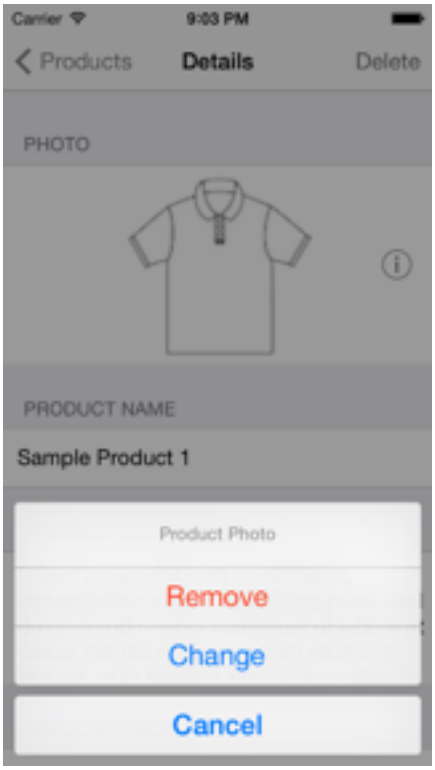




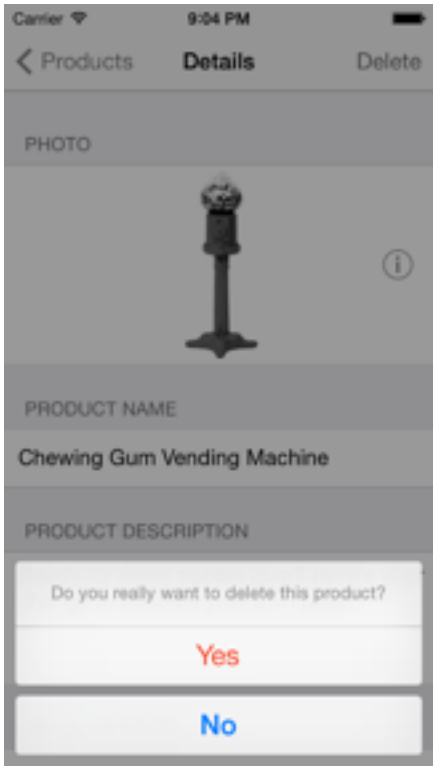




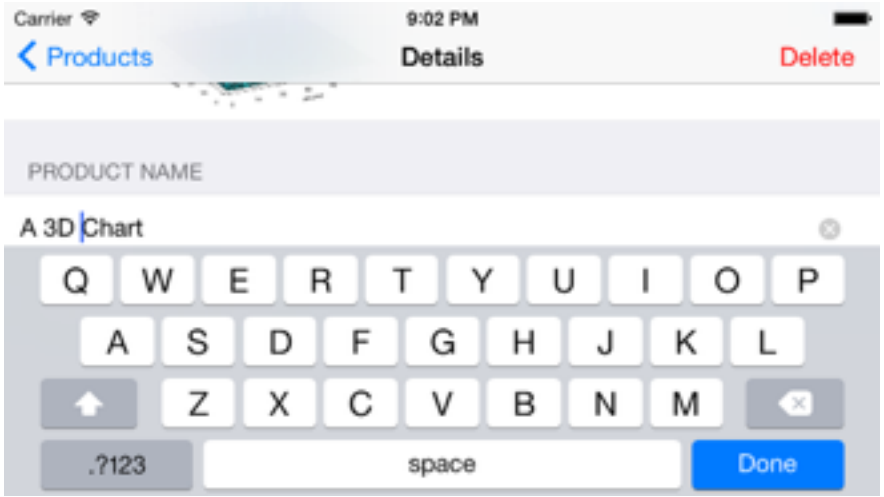



Carrier
9:03 PM
Products
Details
Delete
PHOTO
PRODUCT NAME
Sample Product 1
Product Photo
Remove
Change
Cancel


Carrier
9:03 PM
Products
Details
Delete
PHOTO
PRODUCT NAME
Sample Product 1
PRODUCT DESCRIPTION
Do you really want to remove the Product photo?
Yes
No


Carrier
9:04 PM
Products
Details
Delete
PHOTO
PRODUCT NAME
Chewing Gum Vending Machine
PRODUCT DESCRIPTION
Do you really want to delete this product?
Yes
No


Carrier
9:04 PM
Details
Photo


Carrier
9:02 PM
Products
Details
Delete
PRODUCT NAME
A 3D Chart
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
.?123
space
Done