WhatsInMyRefrigerator

Maulik Zota, Saurabh Mehta

Category: Productivity **Target Audience**: Everyone

About:

WhatsInMyRefrigerator is a handy organizer that allows user to create a list of refrigerator items and notify the user about their items.

Features:

- Signup/login
- Multi fridge creation
- Adding fridge items
- Updating/Deleting fridge or fridge items
- Barcode / QRCode scanner
- Adding items to the favorites list
- Push notifications
- Nearby grocery store

Wish List of Features: (next version)

- Sharing specific inventory with the other users
- Add notifications to the specific items
- Built-in set of categories and on selection to auto set the notification based on the category of the food item (by knowing their self life)
- Highlighting the items based on expired, about to expire and more than 1 week from the date of expiration.
- Improved design

A level work:

- We spent an incredible amount of time in making this app work as proposed. We focused on the core features of this app, which is to scan an item and auto-populate the product title, storing and retrieving the data from MySQL and setting the notifications.
- After an intense research and implementation, we found out ZDBar barcode scanner was adding a lot of bugs with its basic barcode scanner. We then realized that with the release of iOS7, apple included the ability to scan barcodes through its AV foundation framework. We successfully implemented it and also retrieved the product title through an API call to outpan.com.
- Data is being loaded and retrieved from the MySQL DBMS.
- We added a very simple design in order to make the user focus on its core functionality.

- Majority of the methods in the MiddleLayer class and cameraScanner class are reusable.
- Login session is stored and hence user is not required to type in login credentials anywhere except in the login page.

Technical Specification: (important methods)

1. **Table View Controllers:**
    I. InventoryTVC:
        i. clickedButtonAtIndex: as soon as the user create a fridge repository, database call is made to insert and retrieve the data
        ii. didSelectRowAtIndexPath: when “logged in” user selects the fridge, the control is passed to the InventoryItemTVC and lists all the fridge items
    II. InventoryItemTVC
        i. AddingItem: loads a menu with three options to choose from, when adding an inventory items
    III. FavoriteTVC
        i. cellForRowAtIndexPath: configuring the table cell
        ii. didSelectRowAtIndexPath: when one of the table cell is selected, the item is added to the InventoryItemTVC using pushViewController

2. **View Controllers:**
    I. SubmitVC
        i. username: validation for username
        ii. checkpwd: validation for password
        iii. emailed: validation for email
        iv. submit: on submit check for username, password and email in the database and create/read the user
    II. LoginVC
        i. login: call to the MiddleLayer model to load the inventory items in the NSDictionary and make a call to the login.php file to verify the username and password
        ii.
    III. productVC
        i. submit: when pressed, handles all the validation of the text fields and makes the necessary call to the database to add/update the inventory items
        ii. setNotification: setting the notification based on the time selected by the user
        iii. updateTime: sets/updates the text field when the user sets/updates the UIDatepicker

3. **Model**

I. MiddleLayer
    i. downloadItems: makes a call with url was the parameter to retrieve the data in the JSON format
II. cameraScanner
    i. productInfo: makes the API call to retrieve the product information and parses the JSON format to retrieve the product title
    ii. viewDidload: toggles the flashlight, adds two sub views: green rectangle when barcode is detected and a label which shows the output of the scanned item
    iii. didOuputMetadataObjects: detects the barcode displays green rectangle when the barcode is scanned and the retrieved barcode is displayed in the label
    iv. flashlight: to toggle flashlight

4. **Working:**
    - This app works in conjunction with database calls using PHP and MySQL. Whenever a UIViewController is loaded, a database call is made to check for the session user, any items if exist and accordingly the UIView is populated.
    - In order to add an item, user is given the option to choose from: Scan, Favorites and manual entry.
    - Upon selecting scan, user can scan a food item (preferably a barcode scan) and when the camera captures the 12 digit barcode it is printed in the label on the same view to ensure a successful scan.
    - An API call is made to outpan.com to retrieve the product details. The response form this call is in the JSON format that is then parsed from within the app and only the title is selected. This data is then passed on to the productVC to display the scanned product title to the user.
    - User can then enter or edit the rest of the text-field along with the expiration date.
    - As soon as the user hits submit, couple of methods: Database call methods and add notification methods are triggered.

References/Tutorials:

- Scanning Barcode Tutorial
    - http://www.infragistics.com/community/blogs/torrey-betts/archive/2013/10/10/scanning-barcodes-with-ios-7-objective-c.aspx
    - http://zbar.sourceforge.net/iphone/sdkdoc/tutorial.html

- MySQL Database Tutorial
    - How to make an iPhone app connect to a MySQL database

        - http://codewithchris.com/iphone-app-connect-to-mysql-database/
    - https://www.youtube.com/watch?v=Q4PrDYhVJdY

- UIAlertViewController
    - http://www.appcoda.com/uialertcontroller-swift-closures-enum/

- Others:
    - appcoda.com
    - stackoverflow.com