cover_logos.tifcover_logos.tifBarcode Scanning System

# Problem & Solution

## Dietary Intake Problem

Fact, astronauts need to take track of their dietary intake to be studied. The current way to track it is using the FFQ (Food Frequency Questionnaire) where they declare the food they have eaten. The main issues with this method are the time it takes to be filled, the difficulty to remember what they have eaten certain times, among others.

This generates the need to create a solution that replaces and improves this method through an iPad application considering the problem key factors and constraints.

## Idea Overview

A barcode scanning system can benefit the data registry process (foods) in certain level. It increases the registry speed by a simple scanning over a package barcode. If they are previously loaded in an internal database of meals and drinks with identified barcodes then the astronaut would have an easy access to a scanner to read these codes and automatically store the information. However, personally I wouldn’t classify this functionality as a major pillar since there are important constraints to consider such as the low lightning conditions and difficult packages (shining, aluminum, reflective surfaces).

I consider *this functionality should be available in the application as an optional registry resource*, a second choice. The application should delegate and make prominent the data registry through the UI and voice command system as first options.

Experienced Source

According to Dr. Pettit – experienced NASA astronaut [interviewed by Clinton Bonner](http://www.youtube.com/watch?v=bmWQRFVSf-8) – there are important conditions we need to consider; the light level, packages surface and there are some that don’t have barcodes. **This may cause problems and frustration**, we don’t want that for the users (we’ll see how to handle this below in “Barcode Scanning System Features”).

## Barcode Sanning System Features

Considering the important conditions, the barcode scanning system should offer certain features in order to successfully perform reading operations. The overall process is quite simple, once the user enables the scanning option the camera view will prompt to work as the scanning device, once the barcode is taken as a picture this is analyzed through image processing algorithms and returns the code that needs to be used. The application can use a SDK that executes this whole process for us, but we need to consider other features for the barcode system overall. In my opinion, this is what the application should be aware with the barcode system:

* **Barcode decoding process**; as mentioned above, the decoding process is quite simple. The iPad camera works as scanning device to obtain the barcode picture. Once the barcode is detected over a package surface the app should launch a loud beep sound to let the user know the barcode has been successfully scanned. There should be a message from the application whether the barcode has a match in the database. The barcodes database could be internal (in-app) but it would be better if it could be managed via server connection since it would make easy the loading process. Let’s say a new barcode has to be loaded, it wouldn’t be very fast if the iPad would need to come to earth to be manually loaded.
* **Food-related types of barcode**; there are several [types of barcodes](http://en.wikipedia.org/wiki/Barcode#Types_of_barcodes) classified by its symbology. *It’s very important to support food-related barcode types*, specifically by the SDK. According to Wikipedia, the barcode types for foods, groceries, drinks, etc, belong to *UPC-A Barcode, UPC-E Barcode, EAN-13, and EAN-8* categories. The SDK needs to provide support for those types of barcodes (mandatory!). Anyway, NASA could create their own barcode for their products, in that case it would be recommended to follow the same categories because they are 1D barcode types and most of the SDKs support them.
* **Hard conditions capability**; it’s a known fact the environment challenge is quite considerable. Low lightning, shiny packages, transparent packages, among others. This goes for the SDK; it must support this kind of conditions.
* **Failure smart response**; from the same hard conditions scenario, the application must provide a mechanism to counter attack this issue. It’s very likeable some scanning may fail and if this happens continuously the application needs to respond back. The application will allow three continuous failures maximum, at the third time a message will be prompted, showing an apology with options to continue, these might be: speak the code (talking), explore the UI and last, try to enter the code manually (written).
* **In-app processing**; there is not access to internet for this application. The framework should be able to provide an off-line performance. All the SDKs I’m explaining below work fine without internet connection.
* **Fixed and auto focus camera support**; there are known issues with several SDKs not working properly on fixed-focus cameras (iPad 2) because they can’t process blurry pictures properly. The new iPad has an auto-focus camera, so it’s not a problem.
* **System bulk pre-load**; the products and barcodes related database should be on-line for easy maintenance. Anyway, a local database could be worked too. A system module should be enabled to allow scanning and storing products data. It’s quite simple, with a written or scanned interface the user could load each product, label it and sets it ready to match against the astronauts interactions.

# Barcode Scanning Frameworks-iOS Review

There is a wide variety of frameworks for barcode scanning, open source and proprietary. In this study I’m focusing the most suitable ones for the proposed behavior of barcode the scanning system.

After a detailed review I think the most suitable SDK for a reliable performance is provided by **Visionsmarts Barcode Reader SDK** (private). However, if you consider making this scanning functionality a second resource (not a major pillar of the app) as I propose then I would definitively go for an efficient open source library such as **ZXing**. Let’s study *the why* of these particular picks.

features.tif

*Key Features Comparison*

## Visionsmarts Barcode Reader SDK

I’m going to quote myself “these guys are pro”. They offer the whole package for a fully reliable scanning application. They work with very advanced image processing techniques that allow processing hard condition barcodes, like blurry (fixing the fixed-focus iPad2 camera issue), low lightning, high contrast, etc. On the negative side, their license is very expensive.

In Details:

* Live scan, similar to a checkout scanner.
* Reads UPC-A/E and EAN-13/8 bar codes (even blurry) on iPhone, iPod Touch, iPad.
* White label license. You have complete control of the user interface.
* Extremely easy to integrate, there is no parameter to tune.
* Fully compatible with iOS 4 and higher.
* License cost must be quoted. *I emailed them and gave an estimate of 2000€/year for up to 200 iOS devices but I didn’t provide the amount of devices NASA needs, so another quote could be requested.*
* [Website](http://www.visionsmarts.com/products/vs-barcode-reader.html).
* [Sample application](http://www.pic2shop.com/).

## ZXing

It’s a well documented open source library for multiple platforms, including recent iOS versions. Multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages. It uses the built-in camera on mobile phones to scan and decode barcodes on the device, without communicating with a server.

From all the open source libraries this is the most efficient. It has an acceptable performance under low lightning environment

In Details:

* Supports food-related barcode types.
* Supports iOS 4 and higher.
* Free to use (Apache Lincese).
* [Website](http://code.google.com/p/zxing/).
* [Tutorial](http://basheerad.blogspot.com/2012/04/integrating-zxing-qr-code-reader-in.html).

## Scandit Barcode Scanner SDK

This is my second favorite SDK from paid licenses. They offer a demo on their website very easy to install and test. The results are quite acceptable (not reliable though) for the hard conditions. It’s well documented and supports iOS 4 and higher. It’s important to highlight that it doesn’t have issues with fixed-focus cameras (iPad2).

In Details:

* Supports food-related barcode types.
* Fixed focus camera support.
* Free and paid options.
* Supports iOS 4 and higher.
* [Website](http://www.scandit.com/barcode-scanner-sdk/).
* [Tutorial](http://www.scandit.com/support/getting-started-with-scandit-sdk-for-ios/).

## ZBar SDK

Open source library with an acceptable performance, some reports say it’s a little buggy. It has issues to support iOS5 and higher versions (needs too many tricks). It uses the camera feed as a regular shop scanner.

In Details:

* Scans EAN/UPC codes.
* Doesn't play well with fixed focus cameras.
* Free to use.
* [Website](http://zbar.sourceforge.net/iphone/index.html).
* [Tutorial](http://zbar.sourceforge.net/iphone/sdkdoc/tutorial.html).
* Fix for higher iOS4 versions: <http://www.federicocappelli.net/2012/10/05/zbar-library-for-iphone-5-armv7s/>

## Other Paid Frameworks

Since there a lot of frameworks under license that perform in an acceptable way I won’t keep them out so I’ll just group them quickly here, in the expensive / middle-reliability level / average performance for hard conditions group.

[RedLaser SDK](http://redlaser.com/developers/) (an ebay company) »

* It’s technically well gifted (reliable).
* It’s very expensive. [See pricing](http://redlaser.com/developers/pricing/).
* Supports iOS 4 and higher.
* Scans EAN/UPC codes.
* [Tutorial](http://redlaser.com/developers/documentation/ios/using-the-sdk/).

[Accusoft Barcode Xpress Mobile SDK](http://www.accusoft.com/barcodemobile.htm) »

* Well documented.
* Pricing must be quoted.
* Supports iOS 4 and higher.
* Scans EAN/UPC codes.
* [Tutorial](http://www.accusoft.com/barcodemobiledemo.htm).

[Softek Barcode Reader SDK](http://www.bardecode.com/en/products/200-barcode-reader-sdk-for-the-iphone.html) »

* Well documented
* Provides easy way to integrate to projects.
* Provides some methods for image processing.
* Supports iOS 4 and 5 only.
* Free edition license for non-business purposesPricing must be quoted.
* Scans EAN/UPC codes.

# Hard Conditions Study

Since this topic was an important concern for me during the research and tests I had to go beyond reading technical documentation and ask for support directly to the SDK companies. The tests I ran with their applications or tutorials couldn’t be applied on the proper surfaces, I found some similar packages but in order to provide a 100% accurate opinion I should test with the proper packages. Anyway, this information may be useful.

Alternative “Hard” Solution

Due to these scenarios and after tests is why I consider the barcode scanning system shouldn’t be a major pillar of the application. It should work as a secondary resource for accelerating the data registry process. One way to work this issue efficiently would be to integrate the SDK with image processing libraries and apply real studies on the final packages, so we can create methods to adjust the barcode picture contrast, gray scale and other techniques that help to read the barcode properly. The SDK would have to be open source and well documented.

## VisionSmarts Email Support

|  |  |
| --- | --- |
| Gmail | **Luis Millán <millanluiseduardo@gmail.com>** |

|  |
| --- |
| **Re: VisionSmarts.com contact form** |

|  |  |
| --- | --- |
| **Benoit Maison**<benoit.maison@visionsmarts.com> | Fri, Feb 22, 2013 at 11:23 AM |
| To: millanluiseduardo@gmail.com | |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Mr Millan,  Our reader is designed to work in difficult lighting conditions, but the best would be to test on actual barcodes. You can use "pic2shop" or "pic2shop pro" for that.  Which company are you representing?  Unfortunately, we cannot license to individuals.  Regards,  Benoit Maison Vision Smarts SPRL [+32 484 430 936](tel:%2B32%20484%20430%20936) [www.visionsmarts.com](http://www.visionsmarts.com/) Message: My company is planning to implement an iPad business application enabled with a barcode reader feature. There is a concern though, the environment where this application is going to be used has very low lightning and most of the barcodes are printed over high-bright aluminum surfaces, something like this <http://bit.ly/WWPSq3>. Do you think your image processing capabilities can work this issue out? I'd like to know about that and also to know the licensing costs of course. Thanks in advance for your reply, Luis Scandit Email Support  |  |  | | --- | --- | | Gmail | **Luis Millán <millanluiseduardo@gmail.com>** |  |  | | --- | | **Technical details** |  |  |  | | --- | --- | | **Scandit**<info@scandit.com> | Fri, Feb 22, 2013 at 5:41 AM | | To: Luis Millán <millanluiseduardo@gmail.com> | | | |  | | --- | | Hi Luis,  Thanks for your message and your interest in Scandit.  Your set up (low lighting conditions and long codes) sound relatively difficult to deal with to be honest and while our Barcode Scanner SDK is certainly leading edge I’m not quite sure if we’ll be able to read those codes reliably. The only think I can suggest at this point that you sign up for a free Enterprise Trial account at [www.scandit.com/pricing](http://www.scandit.com/pricing) and try it for yourself. The downloadable package also contains a simple demo app that you can build to test the scan performance.  I look forward to hearing your feedback.  Thanks, Samuel  --  Samuel Mueller  CEO and co-founder  Scandit  **From:** Luis Millán [mailto:[millanluiseduardo@gmail.com](mailto:millanluiseduardo@gmail.com)]  **Sent:** Friday, February 22, 2013 4:23 AM **To:** [info@scandit.com](mailto:info@scandit.com) **Subject:** Technical details  Hey there,    I'm planning to implement an iPad business application which is going to use in a low lightning environment and with products that have barcodes printed over difficult surfaces, like aluminum, food packages, something like this <http://bit.ly/WWPSq3>.  I'd like to know if your library offers image processing methods or any other tools that could help to ease the reading under the mentioned environment.  Thanks in advance for your reply,  Luis | | |  Softek Email Support  |  |  | | --- | --- | | Gmail | **Luis Millán <millanluiseduardo@gmail.com>** |  |  | | --- | | **Technical details** |  |  |  | | --- | --- | | **Andy Beezer**<andy.beezer@softeksoftware.co.uk> | Fri, Feb 22, 2013 at 5:43 AM | | To: Luis Millán <millanluiseduardo@gmail.com> | | | |  | | --- | | Hi Luis,  Thanks for your enquiry. I don't think this will have much chance of success in low light - at least not with our sdk. We require fairly good definition between the black bars and the background and I would think that the reflections from the shiny surfaces will probably cause further issues. Sorry we can't help in this case.  Best regards  Andy Beezer Softek Software  Phone or fax [+44 (0) 1296 663633](tel:%2B44%20%280%29%201296%20663633) [http://www.bardecode.com](http://www.bardecode.com/)  Softek Software Ltd Company Registration Number (England): 3560315 Registered Office: Ladysmith Road, Ivinghoe, Leighton Buzzard, LU7 9EE, United Kingdom  On 22 February 2013 03:13, Luis Millán <[millanluiseduardo@gmail.com](mailto:millanluiseduardo@gmail.com)> wrote: > Hey there, > > I'm planning to implement an iPad business application which is going to use > in a low lightning environment and with products that have barcodes printed > over difficult surfaces, like aluminum, food packages, something like this > <http://bit.ly/WWPSq3>. > > I'd like to know if your library offers image processing methods or any > other tools that could help to ease the reading under the mentioned > environment. > > Thanks in advance for your reply, > Luis | | | | | |