Overview
Since food safety issues have become more considerate and critical, it will be very helpful and convenient if the user owns an app that is programmed to inform the user about up-to-date quality and freshness of his/her foods. In this project, we have improved our three tasks...
into three storyboard scenarios, lo-fi prototyping of our interface and a short video prototype to represent the performances and results of each task. Also, tasks are distributed in the interface within three components of our app. For example, the “Danger Zone” component will make sure people receive warning messages when the storage conditions and food status are going bad. Next, the “Food-a-trium” component will mainly keep track of the refrigerator’s temperature, lighting, camera and purchasing history while lastly the “Command Center” component will keep an eye on the newest status of all the refrigerated foods. Moreover, it will display feedback on food safety information such as the purchased date and expiration date based on the receipts.

Task Analyses
This system is designed for the casual everyday user. Currently, kitchen management is largely performed by memory, or at best pen and paper. In order to integrate the system into everyday usage, tasks that the user will have to perform must be largely automated or be simple and quick enough to where the usage of said tasks will not be noted consciously. The tasks which we will be focusing on are simple. The easiest task is the temperature and light warnings. The medium task is history checking. The hardest of the tasks is the management of a food inventory so that users can see what food they possess and estimate expiration dates. These differ from the last iteration because more functionality will be possessed this way, as well as ease of use.

This system is an individual usage item, so communication between users will be non-functional; in fact, for privacy reasons, individual user information should be compartmentalized in order to protect shopping tendencies and patterns from public domain. The data used is entered by the user, or by the devices set up to monitor the status on their food storage. With the exception of entering food in the menu, all data is gathered behind the scenes once installation is complete.

Easy
The warnings system will largely be transparent to the user. When food expires, light thresholds are reached, temperature thresholds are exceeded, etc., the system will register this and send a warning to the user. This warning will be indicated on a device that is equipped with the program, or to an email account of the user's choosing. From there the warning is to be received and dealt with by the user. By notifying the user of a potentially dangerous situation, awareness is raised and it allows the user to react physically. This task is to be performed only as needed and as determined by pre-set thresholds.

Medium
The history functionality of the system is crucial. This task will allow the user to retrieve and use information regarding the food and storage areas which may not have been available before. By merely selecting an area which is of interest to the user, they are able to pull up temperature, light, favorite foods, foods that show a trend of not being eaten in a long period of time, and possibly even an image of their refrigerator, if equipped. This task will be of medium difficulty for the user, as the data is retrieved via multiple levels of menus, but no input is necessary. While learning to use the program, most of the steps will be intuitive, with labeled areas to select and clear ways to go back, with the "back" arrows being universal throughout the application. All data in this section will be aggregate from data entered from either the temperature monitoring device, the light sensor, or the input from the difficult task. Also, it will be coupled with the predictions on expiration dates, usage details from the user input in the difficult task, and time from the program itself. This task should be performed by the user's discretion, and need not
ever be executed by the affect on the other two primary tasks.

**Hard**
The hardest task for the user to use will be the "Command Center" task. This area will be where most input from the user will occur, as well as the basic maintenance of the inventory. Receipts will be recorded via photo and then processed at the server level to extract the text from the pictures and retrieve any pertinent information. This can initially be done by converting the raw IMG file to JPEG and then PDF, then using the Adobe API to extract text as though it was a scanned image via standard OCR systems. While not ideal, it is a start to create our own process of text recognition. Afterwards the store codes can be deciphered on a store by store basis. The resulting item lists are to be propagated and run through the database on the server side to get more information to send to the user, such as projected expiration dates, recipes, and related items. At the client side, usage statistics will be propagated on status change for the items, keeping track of favorite items, items that expire before usage, freshness of individual inventory, etc. Should inventory originate without a receipt, individual items can also be entered. This is also where the user will go to view his inventory and menu items. While viewing the inventory, the user will be able to see the item's freshness, and have different options such as recipes and other items suggested. Ideally this option will be used at least when shopping for food, and regularly enough to maintain the food usage information to prevent false information being provided to the software about food spoilage.

**Sketches**

*Sketch #1 (Yellow) Description*
1. A list of food inventory; user can click to see the detail quantity; user can move their finger and slash to record the foods/drinks they consumed; Pop-up window for expiry date alert.
2. Warning messages for temperature.
3. Tips for store foods; History of temperature, lights, etc.

*Sketch #2 (White) Description*
1. Temperature recognition function
   This command will sense the refrigerator's temperature via wi-fi in order to let the user know if the temperature of his/her refrigerator is right. If the temperature reaches out of the normal scale, it will warn the user with a warning message.
2. Light sensor function
   This command is built for checking the light effects on the certain food. Also, it will show a light analysis table when the user receives a warning message if his/her food is turning bad because of the light.
3. Scanning receipt function
   This command will let user scan the receipt to store the current information (name, image, purchased date and expiration date) of the foods he/she has purchased. It will allow the user to check the food's status any time after successfully scan the receipt.

*Sketch #3 (Graph Paper) Description*
1. Alarms system: The alarms/warnings will be handled here. Basically, anything that could go wrong would send a message to the user via e-mail or text, and that would just be a notification to read and not have any other function. The user dismisses these on their leisure.
2. History function: This is selected from the main menu, and from here it will give you four basic options. Each one is essentially a graphical representation of the data requested; in order to get to each section, the user will flick a motion sensitive device such as a smartphone or tablet towards the menu option they prefer. It will present the information appropriately, whether a chart or a graph, etc.

3. Main information channel: Upon selecting the inventory from the main menu, the view finder approaches. It is wheel shaped, akin to an old ipod trackwheel, only on a smart phone screen. You cycle through options and select them. You can add, delete, or select viewing options from this track wheel, or go back to the main menu. I have shown two options from the track wheel, but the trackwheel can have more options, such as a menu button added to it for each inventory the user wishes to maintain, custom views, and so forth.
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>potatoes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Milk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>chicken</td>
<td></td>
</tr>
</tbody>
</table>

**Message:** You still have:

- 3 potatoes
- Milk
- Chicken

**Alert:**

3 potatoes are going bad! You have 3 days still.
Alert: Your Potatoes
Probable Bad Conditions: Green

dump it.

this one is not needed

Alert: Notice your refrigerator temperature too high.

instead

Temperature
70°C

Use simpler pop-up windows with images & quick for people to understand what's happening.
red potato
keep 1 weeks
on refrigerator
under 10°C
give tops
on storing foods

inventory
receipt
Brace
Search
Monitor

days expired
after scanning receipts

1. potato x 10
2. Catfish x 1
3. Milk x 1

<- users can
Temperature: go back
view listing
10/01/2010
Light:

- -
xx:
add a list to show the

Slash to dismiss it?

Can set the time for pop up windows
eggs

meat

give hints on how to stoke

potatoes (0-10°C)
Light sensor sending alerts

1

2

3 Click MORE OR

5 Click HINT

Light Analysis

Analysing...
Loading...

Light Analysis's

Analysis Result

Heat Map

Light scale

Your food has been exposed under extreme high light scale in two hours.
Your food is in danger!

Light Analysis's

MORE

Light Analysis's

HINT

0 Remove your food
0 Remove the light
0 Check the food's status
0 Analyse again
1. After taking a pic.

2. Food Inventory Database.

- Select Item
- Include Item into Inventory Processing...

3. Click Cooking? OR 3. Click Details.

Food Inventory

| Item Name | Exp. Date | Alley | Expected
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Recommended Entrees or Sides:
- direction

Suggestions:
- Proper way to store
- Proper way to cook
- 

Food Inventory Details

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Expiry Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Directions:
- 

Comments:
- 


Task 1: Alerts

1. => |
2. => |
3. => |

Beep

Dismiss message

Task 2: History

M

Lights

Shakes in the direction to access W/Click

or

Count Red

Total
Interface Narrative
We decided to create an interface with three main components which are “The Danger Zone” (Warnings/Alerts), “Food-a-trium” (User’s Food History), and “Inventory” (Food Database).

In “The Danger Zone” component, it will keep track of all the food items in the refrigerator that are both about to expire/spoil and items that are already expired/spoiled. This component will send pop up windows to the user in order to notify them about their food spoilage. The pop up windows will include the name, image, and expiration date of the food item. Warnings will be sent if the refrigerator’s temperature and lighting source drastically changes. Most likely, this will happen if there’s a power out or the refrigerator’s door was left open. It will have an option for the user to view recipes that use these foods. This component is necessary to keep the user’s informed about any problems with their foods.

The “Food-a-trium” component will show the status and history of the user’s foods. There will be graphs to show the changes in temperature and lighting sources, and brief messages on how it may affect the user’s foods. If the user sees a negative pattern on the graphs, then they can take necessary steps to fix the temperature or lighting on their foods. The camera view in the Food-a-trium will show the users a heat map in which the expired or soon to be expired items marked in red, neutral items marked in yellow, and new/fresh foods marked in green. The colors will be transparent, so the user can still identify their foods. The camera will give aid “The Danger Zone” by tracking expired foods and also give the user a better representation of their foods. The history will store any foods that are bought regularly under the “Favs” section, and foods that are rarely bought under the “Villians” section.

The “The Command Center” component will hold all the user’s refrigerated foods in a database. The user will be able to add, remove and search the database for food items. The user can either manually enter their items in the database or scan their receipts. Sometimes, the user will not receive a receipt after a food purchase from places such as food trucks or bake sales, so they will have to manually enter the item in the database. The database will include the name, image, purchased date, and expiration date of the food items. Additionally, it will send a notification to the users after items have been inserted or removed and keep a history of recent searched items. These features will keep the user up-to-date on the latest items and actions in their database.

Storyboards’ Narratives
In the first scenario, Roberto goes to a nearby grocery store called “Shoppers”. He begins picking up some items for his apartment. He picks up some eggs, milk, cheese, orange juice, lettuce, broccoli, and green beans. Then, he walks over to the checkout line and after the cashier gives him the receipt. He takes out his smartphone and takes a picture of his receipt. The “The Command Center” notifies him that his food database has been updated with new items. He carefully packs his items in his car and drives home. After arriving home, he packs all his foods in the refrigerator.

Secondly, Mark is going on a vacation for the first time in three years. On the plane, Mark decides to check on his refrigerated foods. He opens up “The Danger Zone” on his iphone and sees all his foods are in good condition. There are no current warnings, alerts, or red images on his camera view. He finally arrives to Bahamas and he’s on his way to the hotel. Mark quickly unpacks his luggages, so he can take a tour around the his luxurious hotel and resort. Later on, he stops at a bar and checks on “The Danger Zone” again. He sees a few warnings about upcoming expired foods, but he just ignores it. Towards the end of his two weeks vacation trip, he is relaxing on the beach and his iphone sends a bunch of pop up messages about several
food items expiring. Finally, Mark gets back to his house, and throws out all the bad foods in the refrigerator.

Lastly, Ashley is on her way to a long day of work. Later on, she is working diligently and consistently on updating her company’s database for approximately five hours at that time. She decides to take a little break to check on her foods at home and to look up a recipe for her dinner. She opens up “Food-a-trium” to view the temperature and lighting source on her refrigerated foods, and the camera history to find a couple of her foods are expiring soon. She then opens her favorite foods in the “Favs” section and recipes, but she quickly goes back to work when her manager sees her on the phone.

**Storyboards’ Visualizations**
Roberto is shopping
Database

[Sketch of a database interface on the left and a floor plan of a building on the right.]

(Hand will be updated later)
Checking his phone on the plane.

Hotel checking phone
Temperature

1.

Everything is fine.

2.

Normal scale.

AT Beach

He dismissed it because he is too busy plus with HOT girl!
Drive to Work

Working hard!

Checking on the fund history on cell phone
Details on the feed history

Details on the feed history by monitoring on camera.
Video
http://dl.dropbox.com/u/11233286/PureFoods.avi

Video’s Report (ups and downs about the video process)
Overall, the video-making process was a very fun and rewarding experience even though we experience several challenges along the way. At first, we thought carefully about how we would integrate both the real life scenes and the paper interface in the video. We thought about using post-it cards or index cards to sketch our interface, but then we decided notebook paper would work better. The camera would have a bigger and clearer vision with the notebook paper. We traveled to several different places to tape our scenes such as LaPlata Volleyball Field, Stamp Convenience Store, Mengyi’s Suite/Dorm, and Engineering and Physical Sciences Library. For each scene, we taped it twice in order to improve on the first videotaping. We also practiced each scenario before actually videotaping them. Sometimes, we would be positioned in wrong places suitable for the camera view, so we had to revise our positions and environment. At one point, our camera died so we went to eat dinner. We left the camera to charge in order to record some more scenes. First, we taped all the acting scenes, then all the interface scenes. Afterwards, we did the video editing process in order to combine both the interface and acting scenes together. During editing, it was interesting to see everything mesh together to weave the story that the storyboards attempted to tell. Some difficulties were met when taking the raw footage to the editing floor, as the point and shoot camera stored the data as quicktime .mov formatted files. Luckily everyone worked in Windows or Linux, so no one had any editors that could not work with the format. After converting to AVI, the files were manipulated in a way to show the story that was envisioned in the storyboards. After the final product was completed, we ran it through a compression program to convert from AVI to AVI (divx format) in order to help file transfer, as 900Mb was a bit excessive for a 3:46 min short film. Trimmed to a slim 62Mb, it was uploaded to dropbox and the rest was history.

Appendix
Lo-fi Prototype Sketch