Smart Refrigerator Proposal

February 20, 2012

This project will develop a prototype "Smart Refrigerator" system, which will monitor grocery items purchased by the user in order to reduce food waste and facilitate efficient shopping habits.

Steven Strapp - ses6498@rit.edu
Computer Engineering Year 4

Ben Reeves - bpr5171@rit.edu
Computer Engineering Year 5

Dustin Stroup - dxs2857@rit.edu
Computer Engineering Year 5

Contents

1 Overview

1.1 Needs Statement

The New York Times reports that an average American family of four will account for over 120 pounds of food waste per month and that 27% percent of all food available will be lost to waste [1]. In addition, other resources are lost due to inefficient shopping practices; forgetting common items or special trips made for recipe ingredients waste time and fuel. A system is required for shoppers both to ensure their purchases are used before expiration and to assist in planning of grocery shopping trips.

1.2 Objective Statement

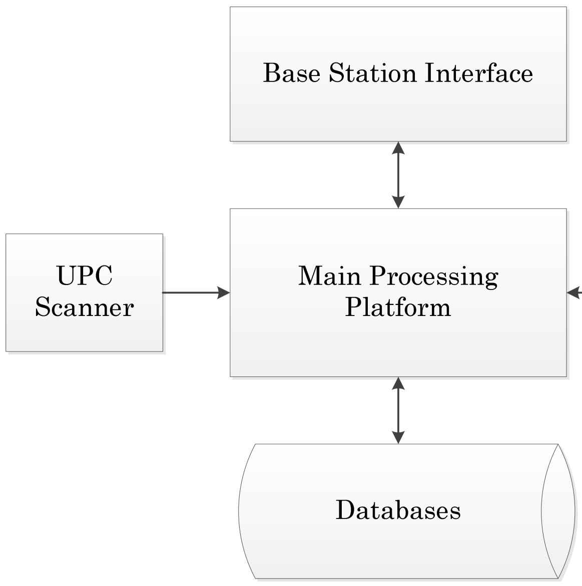
The objective of this project is to design a prototype that will allow a user to track food items in order to reduce waste and improve shopping efficiency. The system will remind the user about items nearing their expiration date and track the frequency of purchased items. From this frequency calculation the system will suggest typical shopping lists. A mobile phone application will provide an interface to the unit to view or create shopping lists and to query inventory.

1.3 Description

A UPC scanner will be used to identify items added or removed from the refrigerator's inventory; a database of UPC codes will translate from the scanned code to an item description. Two databases will be maintained, one linking UPC codes to product descriptions and expiration dates and another to store items currently checked into the refrigerator. A central processing platform on the base station will be used to decode UPC information and to store and interact with the databases. This platform will provide a web interface accessible both via a large convenient display on the main unit and also using a mobile interface. The display on the main unit will allow a user both to check current inventory with expiration dates and to provide additional information when adding or removing items. Both the base station and mobile interfaces can also be used to display and modify suggested shopping lists. The mobile application will interact with the same web interface but will provide a graphical interface optimized for smaller displays. The system will continually estimate the frequency that particular items are purchased and will use this information combined with the expiration dates and purchase dates to suggest shopping lists.

A high level system diagram isolating components is shown in Figure 1.

Figure 1: High Level System Diagram



2 Requirements Specification

2.1 Customer Needs

1. The system should provide an intuitive, easy to use graphical interface.
2. The system should require minimal user input.
3. The system should be able to scan product codes and identify corresponding items quickly.
4. The system should provide secure remote access.
5. The system should report items nearing expiration.
6. The system should provide access to the current inventory.
7. The system should provide a method to create and edit shopping lists.
8. The system should recommend shopping lists which accurately reflect buying habits.
9. The system should function as an add-on to an existing refrigerator or pantry.

2.2 Engineering Specifications

Customer Need	Engineering Requirement	Justification
2,3	A. An off-the-shelf UPC scanner should be used to input items.	A UPC scanner can read product codes with a single click.
3	B. An internal UPC code database should be used to associate codes with items.	An internal database will remove delays associated with an internet look-up.
1,4,6	C. The system should be internet enabled and provide a web interface.	By providing a web interface any other internet-connected device can access the system.
4	D. Remote access should be authenticated with user name and password.	User names and passwords are standard for access control.
2,5	E. An internal database will store default recommended expiration estimates for common categories of items.	Inferring expiration dates based on item category helps minimizes user input. It is well known how long some products take to expire.
1,5	F. The user interface will provide a method for updating default expiration estimates.	Default estimates will not account for condition of product on arrival and may need to be updated.
1,5	G. Interface will provide a visual indication to the user when items are within a user-defined margin of expiration.	The goal of the system is to reduce waste due to expiration.
1,6	H. From both the base station and mobile application the user will be able to view an inventory list.	The user needs access to the current inventory in order to use items and shop effectively.
7,8	I. A database will be devoted to storing recommend shopping lists produced by the system.	User may wish to retain generic shopping lists for future use.
8	J. Recommended shopping lists will reflect purchasing history and expiration dates of current inventory.	Recommendation policy must suggest items relevant to the user in order to be useful.
7	K. Custom shopping lists, created either from the base station or the mobile interface, can be added to shopping list database.	Inefficient shopping practices can be prevented by storing shopping lists and the system can not anticipate all required items.
9	L. The system will be self-contained and no modifications will be required to existing appliances.	Similar systems are commercially available but require costly replacement of existing appliances.

3 Concept Selection

3.1 Evaluation of Existing Systems

Many refrigerator systems are current available which offer integrated displays and internet connectivity. LG, Electrolux, and Samsung all offer refrigerators with large LCD displays that provide access to calendar applications, recipes, weather forecasts, and music and photo sharing services. The principle shortcoming of these devices is the elevated price and the need to completely replace existing appliances. As a more affordable alternative, tablet mounts are available for refrigerators as well. However, these systems do not offer tracking of the refrigerator's contents and do not attempt to reduce waste or improve efficiency. LG demonstrated in April 2011 a "Smart Fridge" with goals closer to the proposed system. The sensors and algorithms used were not disclosed but the product objective is similar, tracking user purchases and providing a mobile interface to the refrigerator's contents while shopping [2]. Our system will provide a much more inexpensive alternative and will be more flexible; the system proposed will not be limited to strictly refrigerators and can be used as an add-on to an existing system.

Many patents exist on inventions related to the smart refrigerator system as a whole and its goal to reduce waste, but do not attempt to reduce user input. Patents 2004/0085225 A1 *Methods and Apparatus to Monitor the Inventory of a Food Storage Unit*, 2010/0148958 A1 *Expiration Warning Device of Refrigerator*, and 2011/0109453 A1 *Apparatus for Warning of an Expiration Date* all treat the goals of the overall system but rely on the user to enter expiration dates manually. More advanced systems, as in Patents 7,861,542 B2 *Refrigerator Including Food Product Management System* and 2011/016555 A1 *Refrigerator and Control Method Thereof*, use radio frequency identification (RFID) tags attached to foods to read expiration dates, with user input as a fallback. The prototype designed will improve the simple user-intensive method of the first group but without the added scope of radio frequency identification used in the second group.

3.2 Concepts Considered and Chosen

Many of the system design choices are easily derived from the engineering requirements; a UPC scanner with a standard USB interface is a clear choice for input of product codes and a mobile application is an obvious interface choice for a system catering to an on-the-go shopper. However, the choices of implementation platform and main base station display present more alternatives. Expiration date recognition is also a potential shortcoming of the system; ideally image processing could be employed to read expiration dates. However, the difficulty and computational complexity of applying image processing significantly extends the scope of the project and places additional performance constraints on the processing platform used. An evaluation of different expiration date recognition systems is tabulated in Table 1.

Ease of use is one of the most critical system requirements; a system relying completely on input from the user will not be acceptable to consumers. However, feasibility and limiting processing performance required are important secondary objectives. Accuracy is critical to the goal of reducing waste due to expiration, but there is inherently some variability even in reported expiration dates. Image processing presents too much additional scope and too many additional requirements in exchange for marginal gains. As long as the predictive system learns from user input and anticipates that items will be purchased in different conditions, this scheme should be sufficient. One additional risk posed by the predictive system, the problem of deciphering text descriptions in order to assign an appropriate prediction, has been mitigated by using the

Table 1: Comparison of Expiration Date Systems

	Method			
	User Input of expiration dates	Image to Text Recognition	Predictive Strategy without itemMaster	Predictive Strategy with itemMaster
Ease of Use	- - -	+	+ + +	+ + +
Feasibility	+ + +	- - -	- - -	+ + +
Accuracy	+ +	+ +	+	+
Total	2+	0	4+	7+

ItemMaster UPC database. Many websites, such as the Food and Drug Administration or community based resources like `www.stilltasty.com`, provide "rule of thumb" style predictions for expiration dates. However, the system must associate a product description with a rule of thumb, which after investigation appears to be difficult classification problem. The ItemMaster UPC database provides not only an association between a UPC code and a text description but also provides a GS1 category. There are a modest number of GS1 categories applicable to this system, each of which can be assigned a rule of thumb to initialize the prediction system.

The problem of predicting shopping habits will be formulated as a problem of predicting the probability that the user will purchase a product again after N days from the last purchase. A product will be added to the shopping suggestions at the peaks in the probability density function, and this process will reset after every purchase. To evaluate modeling strategies, receipts were retrieved for a three month interval from a single user. An initial attempt was to assume that the large number of factors influencing shopping habits could be approximated as normally distributed. However, for the data tested this approximation was very poor; the data considered were either multi-modal or contained a single mode with outliers. In all cases considered, the distribution was shifted to the point where the most likely suggestion time was actually positioned in an interval not supported by any of the samples. A more advanced approach, a non-parametric distribution estimate, was considered next; this method outperformed the simple normal approximation, but appeared to interpolate more than necessary and was the most computationally complex method considered. A final approach clustered the data points, approximated each cluster with a normal distribution, and summed these distributions. With this strategy each mode can be captured without the influence of outliers. The accuracy of the methods considered were evaluated both qualitatively, by looking at the resulting probability density functions, and also quantitatively, by considering performance on the example sets. Overall, clustering to produce a sum of Guassians appears to be the optimal prediction strategy and the probability metrics used are tabulated in Table 2.

Table 2: Comparison of Distribution Estimate Performance Metrics

		Method		
	Trial	Normal Approximation	Non-Parametric Distribution	Clustering to produce sum of Gaussians
∑ Log Probability Observed Habits (Goal to Maximize)	1	-38.3394	-35.9682	-34.7721
	2	-20.5647	-17.0897	-15.6641
	3	-47.8101	-44.9658	-43.9845
	4	-29.1931	-19.6762	-24.4915
Evaluation		- - -	-	+ + +
∑ Log Probability Habits Not Observed (Goal to Minimize)	1	-36.7898	-38.4187	-50.6578
	2	-188.514	-225.002	-318.926
	3	-62.2909	-63.8609	-69.9759
	4	-29.6667	$-\infty$	-86.0767
Evaluation		- - -	+	+ +
Ease of Computation		+ + +	- - -	-
Total		3-	3-	4+

The choice of the base station main display and processing platform are linked but directed mainly by the processing platform. For example, if a personal computer were used a standard LCD monitor may be appropriate, whereas if a tablet were chosen as the main processing engine the interface would be provided automatically. The most strongly considered option was to use a simple micro-controller or BeagleBoard to handle the processing load and to use a large, relative to the micro-controller, LCD display. Comparisons of different processing platform methods and different user interface choices for the base station are shown in Tables 3 and 4, respectively.

Table 3: Comparison of Main Processing Platforms

	Method			
	Personal Computer	Tablet (Combined UI and Processing)	Micro-controller	Beagleboard-xM
Processing Resources	+ + + +	+ +	+	+ + +
Cost	- - -	+	+ + +	+ + +
Size	- - -	+ +	+ + +	+ + +
Total	2-	5+	7+	9+

Evaluating both the interface choice and the processing platform choice together eliminates the personal computer choice; a personal computer cannot be integrated without significantly increasing the form factor of the system. A personal computer also greatly simplifies the system and strays away from an implementation tailored to this prototype. A tablet based interface was considered a very feasible alternative; however the cost and tailorability of the system are again concerns. A micro-controller based system is more appropriate for a small and specialized solution, with the principle concern being quality of the graphical interface produced compared with the other two methods. However, since the system will provide a general web interface, the mobile application as well as a variety of other possible interfaces can be used to view

Table 4: Comparison of Main User Interface Displays

	Method		
	LCD PC Monitor	Tablet	LCD with BeaglBoard-xM
Integration with Unit	- - -	-	+ + +
Ease of Use	+ + +	+ + +	+ +
Size of Display	+ + +	+ + +	+ +
GUI Quality	+ + +	+ + +	+ + +
Size of Unit	- - -	+ + +	+ + +
Total	3+	12+	13+

the display as well so the weight assigned to a high-quality base station interface is mitigated. Considering both choices together, the Beagleboard-xM with an LCD display to inspect items visually as they are checked in and view inventory appears preferable.

4 Design

Consideration of the these concerns, as well as the high level system diagram presented in Figure 1, clarifies the separation of tasks while implementing the project. One group of tasks will contain the mobile interface and also development of an Interface Control Document (ICD) to enumerate the commands provided over the web interface. A second task group will consist of developing the internal databases, the expiration date warning system, and shopping list suggestion algorithm. The final group of tasks will consist of interfacing the processing platform with the scanner, Ethernet interface, and with the main base station interface. The final task group will also contain development of the base station interface.

Further specification of the web interface cannot be solidified without additional investigation into Android application development, and a more detailed system design can not be accomplished without knowledge of the processing platform. The system architecture will change significantly depending on whether a BeagleBoard can be procured. Development with a BeagleBoard will occur on top of an existing Linux distribution, whereas a microcontroller based system would be designed from the ground up.

4.1 User Interface Design

Some initial layouts for the graphical user interface have been designed and are shown in Figures 2, 3, 4, and 5. When designing the interface layouts the possibility of a touch screen interface was considered; all buttons and tabs are intentionally large and easy to click. The Product Entry tab will be the default, and will provide feedback to the user while scanning items. The "Check In" and "Check Out" buttons will function as radio buttons to indicate whether the next scanned item will be interpreted as a new purchase or an item being removed from the current inventory. The shopping list tab will provide a straight-forward view of past shopping lists, organized by ascending creation dates. The "Suggested List" button will produce a new recommended shopping list. The current inventory tab will simply list items currently checked into the refrigerator and provide a reset function to clear the current inventory. As shown in

Figure 5, expiration warnings will be presented as pop-up windows. To prevent these warnings from becoming an annoyance to the user the system will attempt to group together multiple alerts.

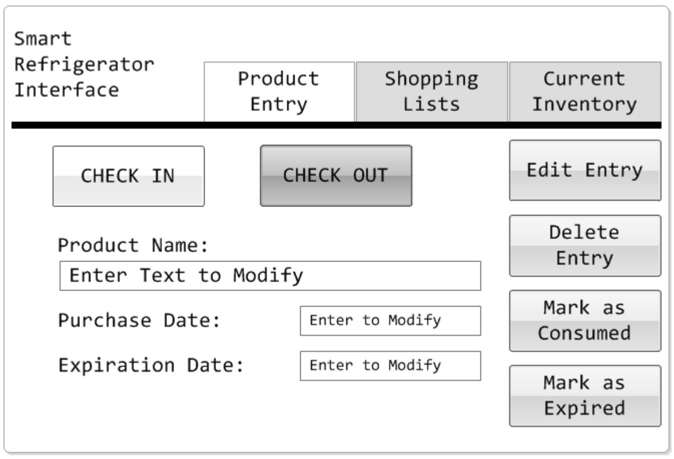
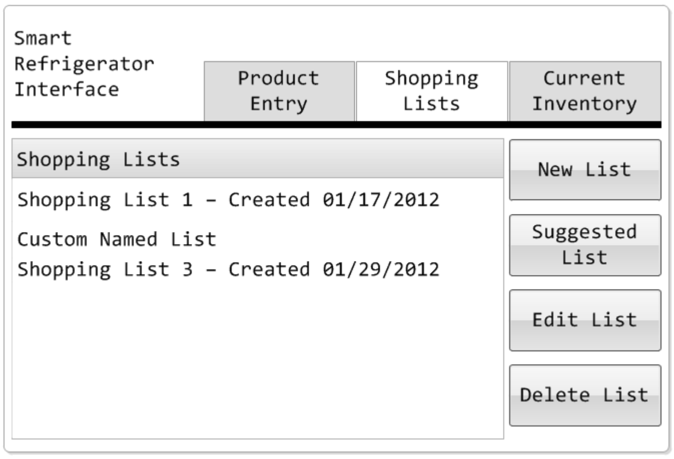
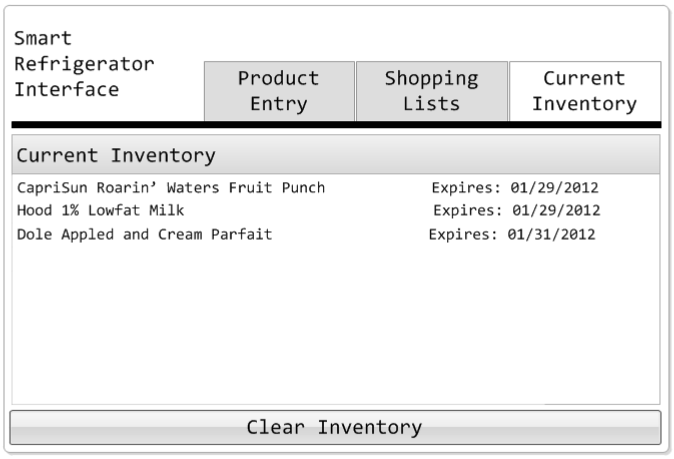


Figure 2: Product Entry Tab Layout



Shopping List 1 - Created 01/17/2012
Custom Named List
Shopping List 3 - Created 01/29/2012

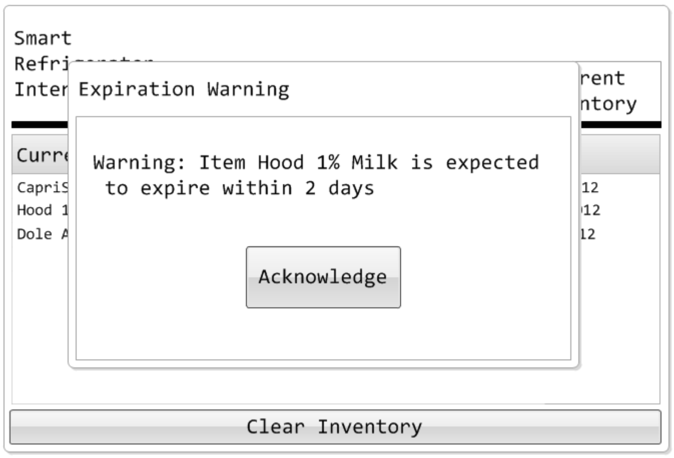
Figure 3: Shopping List Tab Layout



Current Inventory

```
CapriSun Roarin' Waters Fruit Punch        Expires: 01/29/2012
Hood 1% Lowfat Milk                        Expires: 01/29/2012
Dole Appled and Cream Parfait              Expires: 01/31/2012
```

Figure 4: Current Inventory Tab Layout



Warning: Item Hood 1% Milk is expected to expire within 2 days

Figure 5: Expiration Warning Pop-Up Layout

5 Considerations

The Smart Refrigerator system proposed is a step toward promoting sustainability and good stewardship of natural resources. Both the New York Times article mentioned in the statement of needs, and other reports [1, 3], indicate that approximately 27% of all food available for consumption is lost to waste. A study published by the UN Food and Agriculture Organization declares the global percentage is even higher, totalling 1.3 billion tons or 33% overall [4]. The system designed will increase awareness about expiring items, with the goal of reducing these figures. Hugh Collins from AOL News speculates that food waste is dismissed subconsciously; many foods are cheap, and the average consumer does not think about the cost of these small wastes aggregated [3]. By providing reminders to the user the Smart Refrigerator can remedy this source of waste by keeping the user aware of all their purchases. Expiration of food products themselves is not the only source of waste involved with grocery shopping. Making unnecessarily frequent trips to a store for forgotten or unexpected items also wastes resources. The shopping list recommendations provided by the Smart Refrigerator will hopefully mitigate waste here as well. A final consideration of the system is health and safety, by providing reminders about expiring products the risk of eating expired products will hopefully be decreased.

6 Cost Estimates

We have submitted a proposal to the ARM student design contest requesting a BeagleBoard-xM and power adapter. If our proposal is accepted we will be able to obtain these parts at no cost. We also already own many of the principle system components; the dorm room refrigerator, android smart phone, and LCD display will not need to be purchased.

Table 5: Cost Table

Part	Retail Cost	Our Cost
BeagleBoard-xM	$149	$0
BeagleBoard-xM Power Adapter	$14.87	$0
Dorm Room Refrigerator	$100	$0
Android Smart Phone	$100	$0
LCD Display	$80	$0
UPC Barcode Scanner	$35	$35
USB Keyboard/Keypad	$10	$10
Total Cost	$488.87	$45

7 Testing Strategy

Testing of the Smart Refrigerator will be divided into unit testing of the various subsystem and then top-level integration testing once the sub-systems have been connected. Some components used within the system, such as the Angstrom operating system and SQL database implementation, have undergone extensive test prior to use in our system will only be tested to ensure proper configuration. The principle subsystems tested will be the base station user interface, mobile user interface and network interface, expiration date and shopping list prediction algorithms, and integration with the BeagleBoard.

7.1 Base Station User Interface Testing

The main testing focus will be on the user application, both the software running on the base station as well as the web and Android interfaces. Unit testing will be performed during development of each component, as well as integration testing of the final application. This subsection will focus on top-level testing of the base station user interface as a module, with tests particularly directed at the engineering specifications and user requirements. Tests directly motivated by the requirements specification and engineering specifications are listed below and a test procedure is tabulated in Table 6.

- The user interface is required to be easy to use and intuitive; in order to verify this someone not involved in the project should contribute to top-level testing of this sub-system. This also can be tested quantitatively, tests should be performed to ensure the most used items are presented on the default tab and are the most frequently used controls are the most accessible.

- The user interface will provide access to the current inventory, which will be stored using an SQL database. The principle test effort at this step will be verifying integration of the display with the database, not verifying the storage of items themselves.
- The user interface will provide both read and write access to shopping lists, also stored using an SQL database. Testing of this feature will again focus on the ability of the interface to query and modify database entries, not on the database implementation itself.
- The user interface must provide a method to update expiration estimates. Testing of this subsystem will not verify that the update is reasonable or correct but simply verify that this user interface action triggers an update from the expiration prediction subsystem.
- To achieve the principle goal of the system, the user interface must provide a notification of items about to expire. Testing of this subsystem will not verify that the expiration estimate is reasonable or correct, but simply that if triggered by the expiration prediction subsystem the user interface will display an indication.

7.2 Mobile User Interface and Network Interface Testing

The web and mobile interfaces will have their own set of tests, focused on basic functionality and interoperability on various platforms. The web interface will be tested on the most popular browsers (Google Chrome, Firefox, and Internet Explorer), as well as some of the most popular mobile platforms (Android, WebOS, and iOS). The Android interface will need to be tested on various versions of the operating system. At a minimum, major versions between 2.1 and 4.0 will be tested.

Table 6: Base Station User Interface Test Cases

Test Writer:Steven Strapp			
Test Case Name:	Base Station Interface Top-Level Unit Tests	Test ID #:	Base-GUI-01
Description:	Verify that the base station user interface meets the requirement and engineering specifications. Some, such as usability will be evaluated qualitatively and are difficult to outline in this way.	Type:	White Box
Tester Information			
Name of Tester:		Date:	
Hardware Ver:		Time:	
Setup:	User interface subsystem should be entirely integrated with prediction subsystems and SQL databases. System should begin without shopping lists or inventory.		

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Enter fake product code	Switch to inventory tab, entered product should be shown. Inventory should be otherwise empty.				
2	Wait for fake product to nearly expire	Interface should display a notification indicating expiring item.				
3	Use interface to indicate product has not yet expired	Verify that prediction sub-system is triggered to update its estimate.				
4	Create fake shopping list	Verify that list becomes accessible through base station and Android interface				
5	Modify items on fake shopping list	Verify that changes are retained and visible through base station or Android interface				

Table 7: Mobile App Tests

Test Writer:Ben Reeves			
Test Case Name:	Downloading large database updates over an intermittent network connection	Test ID #:	MobApp-01
Description:	Ensure that the database is correctly downloaded even if the device's network connection is interrupted. This could be due to loss of service, a disabled network adapter, or the device powering down.	Type:	White Box
Tester Information			
Name of Tester:		Date:	
Hardware Ver:		Time:	

Setup:	System should have a fresh install of the application and no previous copies of the database downloaded.					
Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Initiate download update of the database	System should connect to the server and begin downloading.				
2	Sever device's network connection	System should pause the download upon sensing the interrupted connection.				
3	Reconnect device to the network	System should resume download of the database				
4	Allow update to complete	System should download the remaining portion of the database				

Table 8: UI Usability Test

Test Writer:Ben Reeves							
Test Case Name:		UI Usability Test				Test ID #:	UI-01
Description:		Ensure that the both the web and mobile versions of the User Interface are accessible and intuitive.				Type:	White Box
Tester Information							
Name of Tester:						Date:	
Hardware Ver:						Time:	
Setup:		System should be representative of one which is in active use; that is, its database should contain both shopping lists and grocery items associated with them.					
Step	Action	Expected Result	Pass	Fail	N/A	Comments	
1	System is given to a user unfamiliar with its operation and submitted to stress testing	User should experience little difficulty navigating the application and experience no bugs, freezes, or crashes.					

Table 9: UI Interoperability Test

Test Writer:Ben Reeves						
Test Case Name:		UI Interoperability Test			Test ID #:	UI-02
Description:		Ensure that the both the web and mobile versions of the User Interface are fully compatible with popular browsers.			Type:	White Box
Tester Information						
Name of Tester:					Date:	
Hardware Ver:					Time:	
Setup:		System should be representative of one which is in active use; that is, its database should contain both shopping lists and grocery items associated with them.				

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Interface is accessed via Mozilla Firefox and subjected to stress testing	Interface is displayed properly, no artifacts or misplaced elements apparent.				
2	Interface is accessed via Google Chrome and subjected to stress testing	Interface is displayed properly, no artifacts or misplaced elements apparent.				
3	Interface is accessed via Microsoft Internet Explorer and subjected to stress testing	Interface is displayed properly, no artifacts or misplaced elements apparent.				
4	Interface is accessed via Android 2.1 and subjected to stress testing	Interface is displayed properly, no artifacts or misplaced elements apparent.				
5	Interface is accessed via Android 4.0 and subjected to stress testing	Interface is displayed properly, no artifacts or misplaced elements apparent.				

7.3 Shopping List and Expiration Prediction Test

Testing of the expiration prediction and shopping list prediction subsystems will be difficult if the system's timing cannot be accelerated; testing should occur over a few minutes not a series of days. For expiration date testing a special set of UPC codes can be added with a fabricated GS1 category so they expire very quickly. The intelligence of the system can then be tested by providing feedback that these imaginary products expired more or less quickly than expected and evaluating the updated predictions. Similarly, the recommendation system will normally discretize purchase dates into intervals of days. A special mode should be added to this subsystem which will consider purchase intervals in the range of seconds; with this accelerated mode new products can be purchased every few minutes and the prediction algorithm can be verified quickly. A test sets are shown for this subsystem in Tables 10 and 11, below.

Table 10: Expiration Date Prediction Test Cases

Test Writer:Steven Strapp			
Test Case Name:	Expiration Date Prediction System Unit Tests	Test ID #:	Pred-01
Description:	Verify that expiration data prediction system makes recommendations within an acceptable margin of true expiration. Simulates expiration of products.	Type:	White Box
Tester Information			
Name of Tester:		Date:	
Hardware Ver:		Time:	
Setup:	Develop fake product codes for quick testing of expiration. System should have no previous expiration date history		

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Enter fake product code	Expiration date should be initialized with recommended "rule of thumb" value.				
2	Provide feedback the product expired before/after recommendation	Re-scan product and shelf-life estimate should decrease/increase.				
3	Enter fake product code and allow to nearly expire	Prior to expiration system indicates to the user product is nearing end of shelf-life.				

Table 11: Shopping List Prediction Test Cases

Test Writer:Steven Strapp						
Test Case Name:		Shopping List Prediction System Unit Tests			Test ID #:	Pred-02
Description:		Verify that shopping list recommendations are helpful, intuitive and reflect previous purchasing habits.			Type:	White Box
Tester Information						
Name of Tester:					Date:	
Hardware Ver:					Time:	
Setup:		System should be placed in time accelerated mode to facilitate quick testing. System should have no previous shopping history.				

Step	Action	Expected Result	Pass	Fail	N/A	Comments
1	Enter fake products indicative of uni-modal shopping habit	System should recommend purchase again on this mode.				
2	Add outlier shopping habits	System should continue to recommend purchase again after mode value.				
3	Enter various items with different buying habits	System should recommend products with highest probabilities.				
4	Begin with uni-modal habit only and add significant variation	System should attempt to track variation in habits.				
5	Enter fake products indicative of bi-modal shopping habits	System should recommend purchase again on each mode.				

7.4 Integration with BeagleBoard

Preliminary testing will focus on the BeagleBoard itself and its ability to interact with the desired peripherals. The system will require an LCD screen, a USB barcode scanner, a network connection, a keypad, and temperature/humidity sensor. Basic functionality of these components will be tested thoroughly during development, as well as during final system testing.

The SQL database used to store all data for the system will be tested once the core of the user application has been coded. Test scripts will be written to populate the databases with fake data in order to ensure that the database is configured as desired, and to verify that the user application is properly communicating with the database alongside the web interface.

It is difficult to outline exactly what testing will be required for the processing platform, since it is unclear what compatibility issues will arise that would not be presented by a conventional platform, where ideally the system would be entirely "plug and play". However, listed below is a baseline sequence of tests.

- Verify that the BeagleBoard, with power adapter, can power all peripheral devices reliably. No sporadic failures occur, this will be performed as an endurance test.
- Verify that MAC address of Ethernet interface can be statically assigned and the BeagleBoard can be pinged reliably; this will be performed as an endurance test, cycling power or disconnecting the board multiple times.
- Verify that the BeagleBoard can reliably interface with the USB scanner and USB keypad, these tests should be performed by writing to a text editor or another program external to the user interface to isolate failures.
- Verify that the BeagleBoard's consistently receives accurate temperature and humidity measurements from the sensor, via the general purpose input/output pins. The measurements should be verified with an external sensor.
- Verify that the touchscreen display accurately records users clicks and controls the pointer; tested outside of the user interface to isolate failures.
- Verify that touchscreen accurately displays the graphical user interface without artifacts or distortion consistently, and ensure all controls on the display are accessible.

8 Risks

8.1 Potential Difficulties

There are a few things that could present difficulties as this project is implemented. The first is determining an accurate method to predict when a user is likely to purchase a food item. This requires statistical analysis of every food item which, in turn, requires a non-trivial amount of processing power. This brings us to the next challenge: finding an appropriate processing platform. It must be small, but have enough memory to store the product database, enough processing power to host the web service and calculate the product statistics, enough peripheral ports for the scanner and a small keypad, and have the ability to drive a display. The display is a third challenge, as it must be large enough to be useful, without increasing the power draw of the device or the overall cost.

8.2 Sources of Failure

The three challenges listed above are also possible sources of failure. If the statistical analysis cannot be run on our platform, whether by processing power or other limitations, then there will be no way to know when to suggest that the user add an item to a grocery list. If the analysis is done improperly, the suggestions will not aid the user or improve shopping habits. If there is no feasible display to use with the system, then there would be no way for the user to receive any feedback when items were scanned. If the bar code was misread, then the wrong item would be added to the database and the user would have no way of knowing. Finally, if there were not platform powerful enough or robust enough for the system, then the system would not be able to perform as required, and features and functionality would be severely limited.

8.3 Contingency Plans

It is possible to limit or even eliminate the risks of such failures. The implementation of a web interface will allow a user to access the database and see what items have been scanned, and make sure that no items were misread. Via this interface, the user will also be able to set default expiration dates and other information about the products they have purchased. That way, in case the display or statistical analysis are unavailable, some related but basic functionality will still be available to the user. To ensure that the processing platform is robust enough, various requirements were set which defined the number of peripheral ports, the processor power, and display capabilities.

8.4 Analysis

In order to determine the best course of action in each case, analysis was done on each risk point. A number of display methods were investigated and compared, as well as a number of different processing platforms. Each comparison showcased the benefits and drawbacks of each item, and the best solution was determined in a way that was as empirical as possible. To determine the best method of statistical analysis, actual shopping histories were collected and analyzed in a variety of different ways to find a model that matched, as closely as possible, the real world shopping habits of the user. With such methods implemented, the device will be able to provide item suggestions accurately.

9 Milestones

Table 12: Table of Milestones

Milestone	**Scheduled Completion Date**	**Assigned**
Board Procurement	February 10, 2012	Steven Strapp
Operating System running on board	February 17, 2012	Ben Reeves
Peripherals properly interfacing with board	February 24, 2012	Dustin Stroup
Basic UI, suitable for debugging	March 2, 2012	Ben Reeves
Database I/O, proper reading from scanner	March 2, 2012	Steven Strapp
Web interface operational	March 2, 2012	Dustin Stroup
User profiling and Statistical Analysis	March 9, 2012	Steven Strapp
Shopping lists, item modification, basic settings	March 9, 2012	Ben Reeves
Mobile application functional	March 16, 2012	Dustin Stroup
Integration and System Testing	March 30, 2012	Ben Reeves

References

[1] Martin, Andrew. "One Country's Table Scraps, Another Country's Meal." New York Times. N.p., 18 May 2008. Web. 26 Jan 2012. http://www.nytimes.com/2008/05/18/weekinreview/18martin.html?pagewanted=all.

[2] Ridden, Paul. "LG launches first Smart-Grid appliance: the Smart Fridge." gizmag. N.p., 27 Apr 2011. Web. 29 Jan 2012. <http://www.gizmag.com/lg-smart-fridge/18502/>.

[3] Collins, Hugh. "Study: US Food Waste Is a Huge Energy Drain." AolNews. N.p., 02 Oct 2010. Web. 29 Jan 2012. <http://www.aolnews.com/2010/10/02/study-american-food-waste-is-a-huge-energy-drain/>.

[4] Ramaker, Rob. "Food waste is hard to combat." Resource. N.p., 26 Jan 2012. Web. 29 Jan 2012. <http://resource.wur.nl/en/wetenschap/detail/food_waste_is_hard_to_combat/>.