

May 01, 2007

Intelli-fridge

James Foulis
Northeastern University

Greggory Hagopian
Northeastern University

Michael Moran
Northeastern University

Bret Rouse
Northeastern University

Recommended Citation

Foulis, James; Hagopian, Greggory; Moran, Michael; and Rouse, Bret, "Intelli-fridge" (2007). *Honors Junior/Senior Projects*. Paper 42.
<http://hdl.handle.net/2047/d10001049>

This work is available open access, hosted by Northeastern University.

Intelli-fridge

James Foulis - Electrical and Computer Engineering (Honors Program)

Greggory Hagopian - Electrical and Computer Engineering (Honors program)

Michael Moran - Electrical Engineering

Bret Rouse - Electrical Engineering

Faculty Advisor: Masoud Salehi

May 5, 2007

Northeastern University - College of Engineering
Honors Senior Capstone Design

Abstract

Inventory management has been a term that has long since been used in industrial settings. In its simplest form, it is a process of materials control whereby goods can be tracked in terms of location and quantity. Warehouses commonly deal with large amounts of materials that are of a varying nature and strict records must be kept in some sort of database to ensure on time and accurate shipment. For years, this process was done by hand, whereby items would be periodically counted and records would be updated. But in 1969, a new technology was introduced that would change inventory management forever; this technology was Radio Frequency Identification (RFID). RFID allowed items to be tagged with a transponder that would be readable remotely by a receiver.

But why should RFID be limited in its use to the industrial environment? Why can't an average person keep control of their inventory in a similar way? Why can't a system like this keep track of one thing that exists in constantly changing amounts in every house in the world? Of course, this is referring to the food in one's refrigerator.

Our capstone group has modified a standard refrigerator so that it can keep an inventory of its contents. This refrigerator uses RFID technology similar to that which is currently used by companies around the world to keep track of their own inventory. Food items can be tagged and placed in the refrigerator. The signal from the tag is then received by an antenna, and the signal is then processed by integrated computer software that we designed to interface with a database that we have compiled. We have named our product the Intelli-Fridge.

INTELLI-FRIDGE

Final Report

James Foulis
Greggory Hagopian
Michael Moran
Bret Rouse

Advised by Masoud Salehi

ECEU790: SENIOR CAPSTONE DESIGN II
Department of Electrical and Computer Engineering
314 Dana Research Building
Northeastern University
Boston, MA 02115

May 5, 2007

Abstract

Inventory management has been a term that has long since been used in industrial settings. In its simplest form, it is a process of materials control whereby goods can be tracked in terms of location and quantity. Warehouses commonly deal with large amounts of materials that are of a varying nature and strict records must be kept in some sort of database to ensure on time and accurate shipment. For years, this process was done by hand, whereby items would be periodically counted and records would be updated, but in 1969, a new technology was introduced that would change inventory management forever; this technology was Radio Frequency Identification (RFID). RFID allowed items to be tagged with a transponder that would be readable remotely by a receiver.

But why should RFID be limited in its use to the industrial environment? Why can't an average person keep control of their inventory in a similar way? Why can't a system like this keep track of one thing that exists in constantly changing amounts in every house in the world? Of course, this is referring to the food in one's refrigerator.

Our capstone group has modified a standard refrigerator so that it can keep an inventory of its contents. This refrigerator uses RFID technology similar to that which is currently used by companies around the world to keep track of their own inventory. Food items can be tagged and placed in the refrigerator. The signal from the tag is then received by an antenna, and the signal is then processed by integrated computer software that we designed to interface with a database that we have compiled. We have named our product the Intelli-Fridge.

Introduction.....	1
Problem Statement	2
Background.....	4
RFID.....	4
Barcodes.....	6
Universal Product Codes	8
Design and Implementation	9
Hardware.....	9
Mini Fridge	9
Computer	9
LCD Touchscreen.....	9
Bar Code Scanner.....	9
RFID Reader.....	10
Software	11
Operating System.....	11
Programming Language.....	11
Database.....	11
Visual Basic Program.....	12
Add Items	12
Remove Items.....	13
Browse Refrigerator	13
Shopping List	13
Alerts	14
Web Implementation.....	15
Welcome Page.....	15
Browse Refrigerator.....	15
Shopping List	16
Alerts	16
Cost Analysis.....	17
Conclusion	20
Sources.....	21
Appendix A: Visual Basic Source Code.....	23
colors.bas	23
modParseMessage.bas	24
frmMainMenu.frm	26
frmAddItems.frm.....	29
frmBarcodeEntry.frm	35
frmNoUPC.frm	37
frmNewItem.frm	41
frmChooseType.frm	44
frmNewType.frm.....	46
frmCalender.frm	49
frmShoppingList.frm.....	51
frmAddList.frm	53
frmViewFridge.frm	57

frmRemoveItems.frm.....	60
frmAlerts.frm.....	63
Appendix B: Web Source Code.....	68
index.html.....	68
browse.html.....	70
shoplist.html.....	73
printlist.html.....	77
alerts.html.....	78

Introduction

Inventory management is a term widely used in industrial settings to describe a process by which a company can keep track of what it has in stock for purposes including preparing shipments, asset calculation, supply chain management and several others. Depending on the system that is in place, a company can utilize this information in order to predict its needs and avoid problems such as having to halt shipment because it needs to wait for a product to be delivered. Some sort of inventory management is essential to the success and productivity of every major company.

In our daily lives, we encounter problems of involving inventory management all the time. We wonder if we left the milk out when we left for work this morning. We worry about having food in the refrigerator to make dinner for our family at night. We plan a trip to the grocery store after work, but realize later that we forgot the shopping list on the kitchen table. The parallelism between these problems and the problems that large companies face regarding supply chain management should not be overlooked.

While these problems will not cause the collapse of a company, they will certainly affect our daily lives. Therefore we have proposed a system of keeping track of food items in a refrigerator that utilizes techniques already put into practice by industry. By making use of Radio Frequency Identification (RFID) and database manipulation, combined with an easy-to-use graphical user interface and web server, the Intelli-Fridge will solve these problems of refrigerator inventory management.

Problem Statement

The Intelli-fridge doesn't exactly solve a problem. It was created simply to make life easier for the consumer and to enhance their grocery shopping experience. Grocery shopping is something everyone has to do, and most people don't think there is a way that it can be improved. Some of us keep a running list of what we need. Some of us just try to remember it when we go to the supermarket. When more people live in the household, it is more and more difficult to keep track of the refrigerator's inventory and shopping can be difficult.

Time is becoming a scarcity in our busy lives. Between going to work, yoga class, the gym, picking your kid up from his dungeons & dragons guild meeting, and getting to the wine tasting party on time, who has time to look in the fridge? This can lead to little frustrating problems. Today's consumer expects more and more from the products that they buy. In the near distant future, all household appliances will likely have some form of artificial intelligence, whether they have just occupancy sensing technology (lights for example are already offered with this function) or complete computing ability with web access. Computers are becoming cheaper and easier to produce every year. By 2020, microprocessor production costs will be less than a penny. In this case, companies will certainly add intelligence to their products to add to the marketability. While costs are still a bit high, the large expensive appliances must be the first to change. The Intelli-fridge is the first step towards the "smart house".

There are several easy to imagine problems that the Intelli-fridge can solve. For instance, you have a half hour to prepare dinner for your family, and you can't find the salad dressing. You knew it was there yesterday, but no matter how much you scour thru the fridge, it just isn't there. This wasted ten precious minutes that you could've used to go to the convenience store in. Having a way to see the real time inventory of the fridge in the first place would've saved ten minutes and made life a lot easier.

Another example is this. You are at work. It's close to five o'clock and you will leave soon. Since you are starving, it makes you remember that there is almost nothing in your fridge. You didn't bring the shopping list that you keep next to the fridge because you were in such a hurry to get to work this morning. Your buddies are coming over for a poker game at eight o'clock so you don't really have time to go home and grab the list, go back to the supermarket, and return home. You will end up going straight to the supermarket and guessing what to buy, certainly forgetting some very important items. Your buddies are going to be so angry because you forget to buy those tortilla chips. Being able to check the inventory of your fridge from the internet at work would save a lot of time and embarrassment with the boys. Wouldn't it be great if you could view, edit, and print a shopping list from anywhere with web access? The Intelli-fridge makes this possible.

The technology used by the Intelli-fridge can be incorporated into other markets aside from the consumer/home appliance market. Being able to keep real time inventory inside

a small enclosure is a necessity for many other applications. Medical and pharmaceutical storage facilities are a good example. Keeping a log of what is inside/outside of the storage container is time consuming and can easily fail due to human error. The RFIID solution would be best to solve this problem. Other parties that may be interested in this technology are those who sell items from small enclosures. Vending machine companies could keep track of what's been sold, what sells more/less often, and when the machine needs to be refilled if they had RFIID to keep them updated remotely. Hotel fridges and mini-bars could also use the technology to bill the room automatically when items are removed from the fridge.

Although none of the problems solved by the Intelli-fridge technology, it does the world a service by making life easier and creating more free time for the busy working family. Improving the quality of life for our customers is our greatest concern and we are confident that Intelli-fridge does just that.

Background

RFID

The key piece of technology that was necessary to implement this project was a fairly old technology, but only recently has been improved to the level to which it has seen common use. This technology is Radio Frequency Identification (RFID). RFID was first developed in 1969 by Mario Cardullo. During this time, he was working on a new system that would track railroad cars. Originally, the design called for using a modified type of barcode that could be read by a sensor as the car passed by it. It was quickly realized that the barcodes were easily made unreadable by dirt, vandalism, or simply wear from the elements of nature. Over time, he and his colleagues developed a passive transponder that was capable of recording data to memory. This developed in to what has become the model for other future RFID tags. One of Cardullo's original ideas was that this type of tag could be used as a method of electronically collecting tolls on highways. His vision remains alive to this day with implementation of FastLane, and other similar systems around the country. [1]

Today, two primary types of tags exist, passive and active. Active tags use an internal battery which currently has a limited lifetime. As a result, these tags have longer ranges, greater data storage abilities, and faster transmission rates. Unfortunately, these tags are larger and more expensive. Passive tags, on the other hand use power that is generated by the transceiver, which gives the tag virtually unlimited life. These tags are smaller and cheaper than active tags. Their downside is that they have a smaller range and slower transmission rates. [2, pp48]

One type of passive tag called, the Trovan tag, uses inductive coupling. When the tag is scanned the circuit is closed and becomes active. This signal stored on the tag is then sent back to the reader. And read in about one tenth of a second. The Trovan tag has a capacity of thirty-nine bits of data. 2^{39} =five hundred fifty billion individual combinations. These tags work on low frequencies, one hundred twenty-eight kilohertz, so as a result, these tags have relatively low range. The positive side of that is that low frequency waves are absorbed less by nonmetallic objects allowing the tag to still be read even when it is buried. These tags are about the size of a grain of rice. Another downside of these tags is that only the tag closest to the reader is read, all others are neglected. [3, pp 3-4]

Another type of tag is the Tiris Tag-it system developed by Texas Instruments. These tags are virtually flat, like a barcode. They work on higher frequencies, providing a greater range. They also have a capacity of two hundred fifty-six bits of data storage. The first thought about the increased data capacity is that it will take longer to read and process. Fortunately the transfer rate on these tags is much higher than the Trovan, and the larger amount of information is read in virtually the same amount of time. [3, pp 4-5]

Barcodes



Figure 1. A Standard Barcode [5]

Barcodes are truly remarkable pieces of technology and are capable of performing the simple task of item identification at an incredible rate. How these little tags work, however, is not as simple as it may seem.

Although all barcodes may look similar, different systems of writing barcodes are used for different applications. The basic idea of a barcode, however, remains the same for all of

these systems. A scanner emits a light at the barcode like the one shown in Figure 1. A barcode simply a sequence of dark bars and white space. When the light hits the dark bars it is simply absorbed into the label. When the light hits the white space, however, it is reflected back at the scanner. The scanner then interprets this data into the information that has been associated with this pattern [4].

Depending on which type of barcoding algorithm is utilized, the language in which the barcode is written changes. One language of writing barcodes that is fairly simple to understand is called “Two of five Unidirectional” [4]. In this language, each digit is represented by five bars, two of which are wide. Each of the five bars is assigned a value; 1,2,4,7,and Po. Different combinations of these values can be used to produce the desired digit. Figure 2 shows how each decimal number (0-9) can be represented in this system. ST and SP represent the beginning and the end of each code, respectively.

	1	2	4	7	Po
1	■	■	■	■	■
2	■	■	■	■	■
3	■	■	■	■	■
4	■	■	■	■	■
5	■	■	■	■	■
6	■	■	■	■	■
7	■	■	■	■	■
8	■	■	■	■	■
9	■	■	■	■	■
0	■	■	■	■	■
ST		■	■	■	■
SP	■	■	■	■	■

Figure 2. Truth table for two of five symbology [4]

This language of coding barcodes is very effective because of its accuracy because it is very simple and straight-forward. The fact that it is a simple combination of two different sized black stripes makes it very easy for the scanner to process. There are, however, two disadvantages of this system. Since each digit has five bars, they tend to get very lengthy.

Also, this type of code can only represent numbers, not letters. Other systems are available that are capable of encoding letters. Also, other systems utilize logical optimization techniques and they are able to code more information into smaller codes [4].

Universal Product Codes

Universal Product Codes (UPC) were designed in 1973 by George Laurer. The system was designed as a means of automated checkout for grocery stores. Every item that is sold is assigned a unique code. The current code that is in place is a sequence of twelve decimal digits which is usually printed both as a twelve digit decimal number, so that humans can read them, and as a barcode for quick scanning. Each digit, 0-9, could be represented as a binary sequence of length four, however the current system encodes the digits into seven bits, or seven black and white stripes on the barcode, in order to allow each code to be more unique and prevent misreads by scanners [6]. Scanning the UPC is a key element in the Intelli-fridge in order to acquire the product information that will be associated with an RFID tags in the database.

Design and Implementation

Hardware

We had several criteria when choosing what pieces of hardware we were going to use. First, and most important, it had to serve its purpose. Second, it had to be easy to integrate. Third, the cost had to be low. These three criteria were considered for every piece of hardware we ended up using. The various pieces of hardware will be listed and described in greater detail in this section.

Mini Fridge

We decided early on that the refrigerator was going to be a small refrigerator in order to save space in the capstone studio as well as simplify moving it to the competition when that time came. We approached the company that provides Micro-Fridges to campus about a non-functional unit, and were lucky enough to have one donated. The refrigerator had separate doors for the refrigerator and freezer. That fact made the decision to place the computer in the freezer almost instantaneous.

Computer

In our proposal we had planned on using a mini-itx computer because of its form factor, but during implementation we decided to go with a standard atx computer (along with video card, NIC, Hard Drive, etc.) because a group member was willing to volunteer his computer to the project. The computer we ended up using was significantly more powerful than necessary, but the price of this system drove us to decide to use it.

LCD Touchscreen

Early on the decision had been made that a touchscreen LCD would be the preferred method of input. We chose a VGA model in order to simplify integration. USB was the obvious choice for interfacing with the touchscreen because the computer had a number of free USB ports and the majority of the touchscreen LCDs available today are usb devices.

Bar Code Scanner

Like the touchscreen, we chose a PS/2 bar code scanner for ease of integration. PS/2 has been the standard keyboard interface for a number of years, and a ps/2 bar code scanner simply emulates a keyboard. This emulation made programming for the scanner easy. It also allowed us to use a keyboard to test functionality.

RFID Reader

Originally, we'd looked into getting a high frequency (13.56MHz) RFID reader/writer from Texas Instruments, but were unable to afford one along with an antenna on our budget. We later found a Philips device which fit our budget and needs almost exactly. The device was a high frequency serial reader/writer and came with a library of functions for us to use. Unfortunately, two weeks before the capstone presentation, we lost our reader to a grounding problem. We no longer had the time or budget to order a replacement so we switched to a low frequency (134kHz) TI reader that happened to be in the Capstone studio. The switch from high frequency to low frequency forced us to only be able to recognize one object in the refrigerator at a time, but otherwise worked as well, or better than, the high frequency option.

Software

The criteria for choosing the software packages to use was the same as the criteria for choosing hardware. It had to serve the purpose well, had to be easy to integrate, and had to fit our budget. Because of the large amount of money we spent on hardware, we had virtually no money left to purchase software, so we were limited to packages the school had licenses for.

Operating System

We chose Windows XP because a group member had an extra license of it and we were all most comfortable developing for Windows XP. A final product would likely use either some version of Windows Embedded (either XP Embedded or CE) or Linux in order to save on licensing fees.

Programming Language

We went with Visual Basic 6 for our programming language because we, as a group, were most comfortable making GUI applications with it.

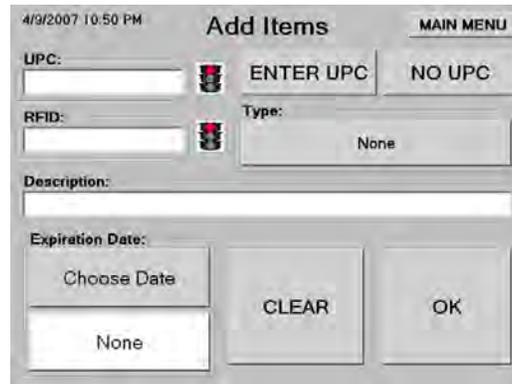
Database

We decided to use a Microsoft Access database for our database needs because it includes a simple GUI for doing data entry for testing, and has a user friendly system for database design and viewing relationships. We designed our database to conform to the rules of third normal form.

Visual Basic Program

The main menu displayed the five options to the user, *Add Items*, *Remove Items*, *Browse Refrigerator*, *Shopping List*, and *Alerts*.

Add Items



The screenshot shows a software interface titled "Add Items" with a timestamp "4/9/2007 10:50 PM" and a "MAIN MENU" button in the top right. The interface includes several input fields and buttons: a "UPC:" field with a barcode scanner icon and "ENTER UPC" and "NO UPC" buttons; an "RFID:" field with a barcode scanner icon and a "Type:" dropdown menu currently set to "None"; a "Description:" text area; and an "Expiration Date:" section with a "Choose Date" button, a "None" dropdown, and "CLEAR" and "OK" buttons.

Figure 3: Add Items menu

When the user comes to this menu, the barcode scanner is automatically activated so the user can begin scanning the item and the tag. Once an item is scanned the user can set the type and the expiration date. If there is no record of the scanned UPC in the database, the user is prompted to enter the new information in and the program will store that UPC. Once both the item and tag are scanned, the user may press OK or scan another item to enter the current item into the database. If there is no UPC, the user can press the “No UPC” button to be brought to a menu of common items that have no UPCs, such as produce. The user also has the option to manually enter the UPC via the “Enter UPC” button.

Remove Items



Figure 4: Remove Items menu

From this menu, the user is presented with the option of removing any item that is currently outside of the refrigerator from the active database. The user may also add any item to the shopping list from this menu.

Browse Refrigerator



Figure 5: Refrigerator Viewer menu

Here the user is presented with a list of items currently in the refrigerator, and is given the option to add any item to the shopping list.

Shopping List

This menu shows the user a list of items in the shopping list, and allows the user to add items, or edit/remove items currently in the list.

Alerts

Alert Type	Size	Description	Exp. Date
Expires Soon	20 oz (49)	GATORADE ORANGE	4/11/2007
Out of Fridge	20 fl. oz.	Dr Pepper	
Out of Fridge	N/A	Cheese (other)	4/8/2007
Out of Fridge	20 oz (49)	GATORADE ORANGE	4/11/2007
Out of Fridge	N/A	Ice Cream (generic)	4/8/2006

Figure 6: Alerts menu

Here the user is shown a list of alerts for current items that are active in the database and either currently outside the fridge, expired or soon to expire. The user can add any item to the shopping list from this menu and can also choose which alerts to display via the radio buttons above the list.

Web Implementation

In order to provide remote access to the database we created a web interface for the user. The pages were programmed in PHP, interfacing with the Microsoft Access database; and the design was a freeware CSS template created by Andreas Viklund. The site consists of 5 main pages to provide all functionality that a remote user would need.

Welcome Page

This is the first page that is displayed when a user accesses the website. It displays a picture of the fridge and a menu listing the possible actions available to the user.

Browse Refrigerator

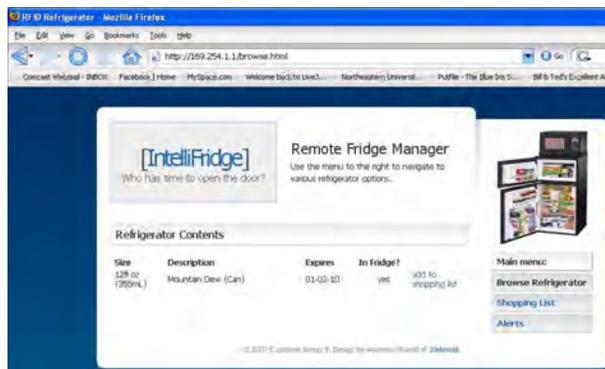


Figure 7: Browse Refrigerator page

The main functionality of the system, the browse function. On this page the user is presented with a list of items currently inside the refrigerator. Associated with each item is a hyperlink to add that item to the shopping list.

Shopping List



Figure 8. Shopping List page

When you load the shopping list page, it presents you with a list of items on the list. From this page there are four submenu items available: *View List*, *Print List*, *Add Items to List*, *Edit List*. *Print List* opens a new page with no background and simple text formatting that is compatible with printing. *Add Items to List* loads a simple web form to add items to the shopping list. And *Edit List* loads a form where you can edit each field of the items in the shopping list, as well as remove them.

Alerts

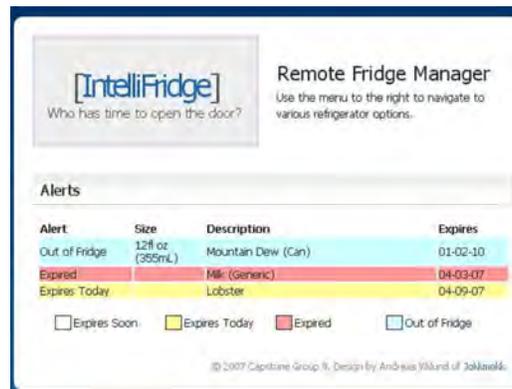


Figure 9. Alerts page

This page will show the user a complete color-coded list of alerts (Out of Fridge, Expires Soon, Expires Today, and Expired). Since the user is logged in remotely there is no action to be taken.

Cost Analysis

In order to keep our costs at a minimum, as well as to cope with issues that we had during design, a lot of the items that were utilized in our prototype are different than the items that we would actually be using in production. The following will outline what we did use as well as what we would have used in production and why.

The prototype is currently using an Abit NF7 motherboard with a 2.2 GHz AMD 3000 processor with two gigabytes of RAM, a thirty gigabyte hard drive and a nvidia gforce 256mb video card. This system is extremely powerful and far exceeds the requirements of the program that we are running on it. We used this system, however because the entire system was generously loaned to our group by one of our members, at no cost. In production, we would be using a scaled down computer consisting of an EPIA 5000 (fanless) Mini-IXT motherboard, or possibly something even smaller. This motherboard was chosen because of its simple design including all necessary ports, onboard video, onboard Ethernet, and fanless operation. This board would cost only about one hundred eighteen dollars. Additional components including 512 MB of RAM and a 10 GB hard drive would add about another thirty and twenty-five dollars respectively. A power supply was also contributed by a member of the group but a supply that is capable of powering all of our components would not cost more than thirty dollars.

The screen that was chosen is a TFT eight inch LCD touch screen display. It was chosen for its appropriate compromise between size and cost. The screen has a VGA input for the computer as well as auxiliary inputs that are currently unused. The screen supports a resolution of 640x480 which worked out very well with the GUI that we designed. This screen cost us about one hundred fifty-five dollars.

The barcode scanner that was utilized is a V-Line Shooter Keyboard Wedge Scanner. It was chosen for its ability price as well as its ability to be programmed to read different barcode types and to return them in whatever format that we chose. We were also easily able to manipulate its circuitry to be trigger a scan using the I/O pins of the RFID reader. Being a LED barcode scanner however created range issues with its use. In production the Intelli-fridge would likely utilize a higher quality, laser barcode scanner. This scanner cost us about sixty dollars.

The RFID system that was originally chosen was a Philip (NXP) SLRM900 13.56 MHz I-Code reader. This reader was chosen for its low cost of only one hundred dollars and its high frequency nature that would allow us to read several high frequency tags at a given time. This reader can be seen in Figure 10 below.



Figure 10. Philip (NXP) SLRM900 13.56 MHz I-Code reader

An SA70 Standard Antenna was purchased from Frosch Electronics. This antenna was chosen for its design that included a PCB that tuned the antenna to 13.56 MHz and its ability to work at up to seventy centimeters with our reader. This antenna was purchased for about two hundred dollars. Unfortunately, due to the malfunction of the purchased reader with no time to order a replacement, we were no longer able to use it or the antenna in our prototype. Instead we were resourceful and used a Texas Instruments S2000 134 KHz reader that was in the capstone lab from a previous year. Because of it being a LF RFID reader, it only allowed us to read a single tag at a given time, but it was enough for us to show a proof of our concept in a scaled down fashion. This antenna can be seen in Figure 11 below.



Figure 11. Frosch Electronics SA70 Standard Antenna

The RFID tags that were used were RI-TRP-WFOB-01 and RI-TRP-RFOB-01 for the low frequency implementation of our design. They were chosen for their small design with a key ring attached for us to easily label them. In a high frequency implementation we could have used a variety of tags that are available in many shapes and sizes from Texas Instruments. Most tags, regardless of frequency, that are offered through TI cost about seventy cents apiece when ordered in bulk. For our prototype, however, TI was gracious enough to provide us with samples of all of their tags at no cost. The label that is attached to each low frequency tag is a barcode that was created using a free version of Barcode Magic software. These labels were printed and laminated so that they could be easily attached to the key ring. The cost for each of these to be made came to about thirty cents apiece. These tags can be seen in Figure 12 below.



Figure 12. RI-TRP-WFOB-01 LF RFID tag and barcode label

The final element in the Intelli-Fridge is, of course, a refrigerator. For our prototype, we were able to acquire a mini-fridge for free through Micro Fridge, a company that rents refrigerators to the freshman dorms at Northeastern University. In production, however, the cost of the refrigerator could vary from a few hundred to a few thousand dollars depending on the model of refrigerator that was being integrated.

Overall we believe our total production cost to be about seven hundred fifty dollars, plus the cost of the refrigerator.

Conclusion

In conclusion, although we were unable to implement everything in our project proposal, we believe our final project was a sufficient proof-of-concept and that with proper implementation of HF RFID and the integration of RFID into supermarkets our project would become a very real possibility for the consumer market.

Sources

- [1] M. Cardullo. n.d. [online]. Genesis of the Versatile RFID Tag. Available: <http://www.rfidjournal.com/article/articleview/392/1/2/> June 11, 2006 [date accessed].
- [2] H.E. Davis and M.S. Luehlring, "Radio Frequency Identification: The Wave of the Future," *Journal of Accountancy*, vol. 198, no. 5, 2004, pp. 43-49.
- [3] R. Want, and D.M. Russell, "Ubiquitous Electronic Tagging," *IEEE Distributed Systems Online*, vol. 1, no. 2, 2000, pp. 1-12.
- [4] T. Sriram, et al., "Applications of Barcode Technology in Automated Storage and Retrieval Systems," *Industrial Electronics, Control, and Instrumentation*, vol. 1, 5-10 Aug, 1996, pp. 641-646.
- [5] Intellitrack, Inc, "Intellitrack, Inc." March 21, 2005; www.intellitrack.net.
- [6] [author not available]. n.d. [online]. Universal Product Code. Available: http://en.wikipedia.org/wiki/Universal_Product_Code June 2, 2006 [date accessed].
- [7] P. Bhatnagar. (2004, Sept 23). [online]. RFID: Revolutionary Tool or Simply Spyware. Available: http://money.cnn.com/2004/09/07/news/fortune500/retail_rfid/ May 22, 2006 [date accessed].
- [8] US Patent No. 6982640. November 21, 2002. <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=6982640.PN.&OS=PN/6982640&RS=PN/6982640>. May 20, 2006 [date accessed]
- [9] US Patent No. 7050991. October 9, 2001. <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=7050991.PN.&OS=PN/7050991&RS=PN/7050991>. May 20, 2006 [date accessed].
- [10] US Patent No. 6983884. February 19, 2004. <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.htm&r=1&f=G&l=50&s1=6983884.PN.&OS=PN/6983884&RS=PN/6983884>. May 20, 2006 [date accessed].
- [11]. [author not available] . (2003, April 4). [online]. Merloni Unveils RFID Appliances. Available: <http://www.rfidjournal.com/article/articleview/369/1/1/> June 1, 2006 [date accessed].
- [12] Ecrater.com. n.d. [online]. Available: <http://www.ecrater.com/product.php?pid=179407>. June 5, 2006 [date accessed].
- [13] Mobileplanet.com. n.d. [online]. Available: <http://www.ecrater.com/product.php?pid=179407>. June 5, 2006 [date accessed].
- [14] Netgate. n.d. [online]. Available: http://www.netgate.com/product_info.php?products_id=94. June 5, 2006 [date accessed].
- [15] Priority Computer Parts. n.d. [online]. Available: <http://www.prioritycomputerparts.com/listitem.aspx?ItemID=769195>. June 5, 2006 [date accessed].

- [16] Colamco. n.d. [online]. Available:
<http://www.colamco.com/store/product/detail.aspx?product=483392&Source=FromGoogle>. June 5, 2006 [date accessed].
- [17] Case-mod.com. n.d. [online]. Available: <http://www.case-mod.com/store/lilliput-619gl70npct-lcd-touch-screen-vga-monitor-charcoal-p-922.html?src=fr>. Jun 5, 2006 [date accessed].
- [18] I-tech Company. n.d. [online]. Available:
<http://wintech.stores.yahoo.net/widescreen-lcd-monitor-wopw700x.html>. June 5, 2006 [date accessed].
- [19] Microsoft. n.d. [online]. Available: Windows XP Embedded Overview and Benefits,
<http://www.microsoft.com/windows/embedded/eval/xpe/default.mspx>. May 25, 2006 [date accessed].

Appendix A: Visual Basic Source Code

(automatically generated code was removed in the interest of brevity)

colors.bas

```
Attribute VB_Name = "Colors"
Global strNewUPC As String
Global intnewupc As Double
'Global strExpirationDate As String
Const color1 As Long = &HD0D0D0
Const color2 As Long = &HFFFFFF
Const color3 As Long = &HC0C0C0
Const color4 As Long = &HFFFFFF
Declare Sub Sleep Lib "kernel32" (ByVal milliseconds As Long)
Public Function buttonColor() As Long

    buttonColor = color1

End Function
Public Function selectedButtonColor() As Long

    selectedButtonColor = color2

End Function
Public Function formColor() As Long

    formColor = color3

End Function
Public Function textBoxColor() As Long

    textBoxColor = color4

End Function
```

modParseMessage.bas

```
Attribute VB_Name = "modParseMessage"
Option Explicit
Dim cFirstChar As String
Dim strType As String
Dim cSecondChar As String
Dim iFirstSpace As Integer
Dim iCR As Integer
Const ptFORMAT = "Format"
Const ptCLEAR = "Clear Buffer"
Const ptBUFFER_CONTENTS = "Contents of buffer"
Const ptIO_STATUS = "Status of I/O Lines"
Const ptSET_OUTPUT = "Sets Outputs"
Const ptSET_OUTPUT_IO_STATUS = "Set output and get I/O status"
Const ptCHARGE = "Set charge period"
Const ptEXECUTE = "Execute command"
Const ptGATE = "Gate Mode"
Const ptSTORE = "Store Command"
Const ptNUMBER = "Number of Stored IDs"
Const ptLINE = "Line mode"
Const ptNORMAL = "Normal mode"
Const ptPROGRAM = "Programming tag"
Const ptANTENNA = "Antenna command"
Const ptRAM = "RAM Fill"
Const ptRO = "Read Only tag"
Const ptRW = "Read/Write tag"
Const ptMPT = "MPT tag"
Public Function ParseType(strMessage As String) As String

    cFirstChar = Left(strMessage, 1)
    Select Case cFirstChar
        Case Is = "F"
            ParseType = ptFORMAT
        Case Is = "C"
            ParseType = ptCLEAR
        Case Is = "B"
            ParseType = ptBUFFER_CONTENTS
        Case Is = "J"
            ParseType = ptIO_STATUS
        Case Is = "Y"
            ParseType = ptSET_OUTPUT
        Case Is = "H"
            ParseType = ptSET_OUTPUT_IO_STATUS
        Case Is = "Z"
            ParseType = ptCHARGE
        Case Is = "X"
            ParseType = ptEXECUTE
        Case Is = "G"
            ParseType = ptGATE
        Case Is = "S"
            ParseType = ptSTORE
        Case Is = "N"
            ParseType = ptNUMBER
        Case Is = "L"
```

```

    ParseType = ptLINE
Case Is = "E"
    ParseType = ptNORMAL
Case Is = "P"
    ParseType = ptPROGRAM
Case Is = "U"
    ParseType = ptANTENNA
Case Is = "R"
    If Len(strMessage) = 4 Then
        ParseType = ptRAM
    Else
        ParseType = ptRO
    End If
Case Is = "W"
    ParseType = ptRW
Case Is = "M"
    ParseType = ptMPT
End Select

```

End Function

Public Function ParseID(strMessage As String) As String

```

    strType = ParseType(strMessage)
    iFirstSpace = InStr(strMessage, " ") + 1
    iCR = InStr(1, strMessage, vbCr)
    If strType = ptFORMAT Or strType = ptCLEAR Or strType = ptSET_OUTPUT Or strType =
ptCHARGE Or strType = ptNORMAL Then
        ParseID = strType
        ParseID = "Invalid"
    ElseIf strType = ptRO Or strType = ptRW Or strType = ptMPT Then
        ParseID = Mid(strMessage, iFirstSpace, iCR - iFirstSpace)
    Else
        If iCR <= 2 Then
            ParseID = strType & " Invalid"
        End If
        cSecondChar = Mid(strMessage, 2, 1)
        If cSecondChar = "I" Then
            ParseID = strType & " Invalid"
        ElseIf cSecondChar = "M" Or cSecondChar = "R" Or cSecondChar = "W" Then
            ParseID = Mid(strMessage, iFirstSpace, iCR - iFirstSpace)
        End If
    End If
End Function

```

End Function

frmMainMenu.frm

```
Attribute VB_Name = "frmMainMenu"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection
Dim strQuery As String

Dim strNewData As String
Dim strOldData As String
Dim strID As String
Dim strLastID As String

Dim strAddItemsCaption As String
Dim strViewFridgeCaption As String
Dim strRemoveItemsCaption As String
Dim strShoppingListCaption As String
Dim strAlertsCaption As String

Dim i As Integer
Private Sub cmdAddItems_Click()

    'when 'add items' button is clicked, show form, and disable main menu
    MSComm1.Output = "HO0003"
    frmAddItems.Show

End Sub
Private Sub cmdAlerts_Click()

    frmAlerts.Show

End Sub
Private Sub cmdExit_Click()

    End

End Sub

Private Sub cmdList_Click()

    frmShoppingList.Show

End Sub

Private Sub cmdRemove_Click()

    frmRemoveItems.Show

End Sub
```

```

Private Sub cmdView_Click()

    frmViewFridge.Show

End Sub

Private Sub Form_Load()

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Or TypeOf Me.Controls(i) Is ListView Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Or TypeOf Me.Controls(i) Is MSComm Then
            Else
                Me.Controls(i).BackColor = Colors.formColor
            End If
        Next

        Me.BackColor = Colors.formColor

        'on form load initialize date and time
        txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

        MSComm1.CommPort = 1
        MSComm1.Settings = "9600,N,8,1"
        MSComm1.Handshaking = comXOnXoff
        MSComm1.RThreshold = 1
        MSComm1.PortOpen = True
        MSComm1.Output = "ZFF"
        Sleep 500
        MSComm1.Output = "L"
        'MSComm1.Output = Chr(27)

        strAddItemsCaption = frmAddItems.Caption
        strViewFridgeCaption = frmViewFridge.Caption
        strRemoveItemsCaption = frmRemoveItems.Caption
        strShoppingListCaption = frmShoppingList.Caption
        strAlertsCaption = frmAlerts.Caption
        Unload frmAddItems
        Unload frmViewFridge
        Unload frmRemoveItems
        Unload frmShoppingList
        Unload frmAlerts

    End Sub
Private Sub Form_Unload(Cancel As Integer)

    MSComm1.PortOpen = False

End Sub

Private Sub lblEnd_DbIcIck()
    End

```

End Sub

Public Sub tmrTime_Timer()

'timer to refresh date/time on each page
txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

For i = 0 To Forms.Count - 1

 If Forms(i).Caption = strAddItemsCaption Then

 frmAddItems.txtDateTime.Caption = txtDateTime.Caption 'Date & " " & Format(Now, "Medium Time")

 ElseIf Forms(i).Caption = strViewFridgeCaption Then

 frmViewFridge.txtDateTime.Caption = txtDateTime.Caption

 ElseIf Forms(i).Caption = strShoppingListCaption Then

 frmShoppingList.txtDateTime.Caption = txtDateTime.Caption

 ElseIf Forms(i).Caption = strRemoveItemsCaption Then

 frmRemoveItems.txtDateTime.Caption = txtDateTime.Caption

 End If

Next

' Debug.Print "isLoaded('frmAddItems')" & isloaded("frmAddItems")

' If isloaded("frmAddItems") Then frmAddItems.txtDateTime.Caption = Date & " " & Format(Now, "Medium Time") 'Else MSComm1.Output = "HO0000"

' If frmViewFridge.Visible = True Then frmViewFridge.txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

' If frmShoppingList.Visible = True Then frmShoppingList.txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

' If frmRemoveItems.Visible = True Then frmRemoveItems.txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

End Sub

Private Sub MSComm1_OnComm()

 strNewData = MSComm1.Input

 strOldData = strOldData & strNewData

 If Right(strNewData, 1) = Chr(10) Then

 strID = ParseID(strOldData)

 strOldData = ""

' If InStrB(strID, "Invalid") = 0 Then

 If strID <> strLastID Then

 Set cn = New ADODB.Connection

 cn.ConnectionString = "DSN=UPC_Database;"

 cn.Open

 strQuery = "UPDATE Fridge AS f SET f.InFridge = FALSE;"

 Set rs = cn.Execute(strQuery)

 If InStrB(strID, "Invalid") = 0 Then

 strQuery = "UPDATE Fridge AS f SET f.InFridge = TRUE, f.LastSeen = " & Now & " WHERE f.TagID IN (SELECT TagID FROM Tags t WHERE t.RFID = " & strID & ") AND f.Active = TRUE;"

 Set rs = cn.Execute(strQuery)

 End If

 strLastID = strID

 End If

End If

End Sub

frmAddItems.frm

```
Attribute VB_Name = "frmAddItems"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection
Private rs2 As ADODB.Recordset
Private cn2 As ADODB.Connection

Dim strQuery As String
Dim intTagID As Integer
Dim bRFIDFlag As Boolean
Dim strRFIDNumber As String
Dim strLast As String
Dim intYesNo As Integer
Dim intNewID As Integer
'Declared constant for Green Light image so it can be easily adpated to different computers
Const ImgGrnPath As String = "C:\Capstone Group 9\VB\Fridge08\traffic-green.jpg"
Const ImgRedPath As String = "C:\Capstone Group 9\VB\Fridge08\traffic-red.jpg"
Public strExpirationDate As String
Dim i As Integer

Private Sub cmdClear_Click()

    'clear all values and set to defaults
    txtUPCValue.Text = ""
    txtRFIDValue.Text = ""
    bRFIDFlag = 0
    frmCalendar.calExpiration.Value = Date
    strExpirationDate = ""
    cmdExpiration.Caption = "Choose Date"
    cmdNone.BackColor = Colors.selectedButtonColor
    cmdExpiration.BackColor = Colors.buttonColor
    imgRFID.Picture = LoadPicture(ImgRedPath)
    imgUPC.Picture = LoadPicture(ImgRedPath)
    cmdType.Caption = "None"
    txtDescription = ""
    'txtInputBox.SetFocus
    ' frmCalendar.Hide

End Sub
Private Sub cmdEnterUPC_Click()

    frmMainMenu.MSComm1.Output = "HO0000"

    ' If Len(txtUPCValue.Text) = 0 Then
    '     frmAddItems.Enabled = False
    '     frmBarcodeEntry.Show
    '     frmBarcodeEntry.ZOrder ()
```

```

'   frmAddItems.Enabled = True
'   Else
'   MsgBox ("UPC already entered. Please clear before entering another.")
'   End If

End Sub
Private Sub cmdExpiration_Click()

    frmMainMenu.MSComm1.Output = "HO0000"
'   frmAddItems.Enabled = False
    frmCalendar.Show

End Sub
Private Sub cmdNone_Click()

    strExpirationDate = ""
    cmdExpiration.Caption = "Choose Date"
    cmdNone.BackColor = Colors.selectedButtonColor
    cmdExpiration.BackColor = Colors.buttonColor

    txtInputBox.SetFocus

End Sub

Private Sub cmdNoUPC_Click()

    frmMainMenu.MSComm1.Output = "HO0000"
    frmNoUPC.Show

End Sub
Private Sub cmdOK_Click()

    If Len(txtUPCValue.Text) = 13 And bRFIDFlag Then
        Set cn = New ADODB.Connection
        cn.ConnectionString = "DSN=UPC_Database;"
        cn.Open
        ' If there is an expiration date, use it, otherwise don't.
        If Len(strExpirationDate) <> 0 Then
            strQuery = "INSERT INTO Fridge (TagID, FridgeUPC, DateAdd, Expires, Active) values (" &
intTagID & ", " & strExpirationDate & ", " & txtUPCValue.Text & ", " & Now & ", " & strExpirationDate & ", TRUE);"
        Else
            strQuery = "INSERT INTO Fridge (TagID, FridgeUPC, DateAdd, Active) values (" & intTagID &
", " & txtUPCValue.Text & ", " & Now & ", TRUE);"
        End If
        Set rs = cn.Execute(strQuery)
'   MsgBox (txtDescription.Text & " has been entered")
        If cmdType.Caption <> "None" Then
            strQuery = "SELECT ItemTypeUPC, TyDesc FROM ItemTypes, Types WHERE
ItemTypes.ItemTypeUPC = " & txtUPCValue.Text & " AND ItemTypes.TypeID = Types.TyID;"
            Set rs = cn.Execute(strQuery)
            If rs.EOF Then
                strQuery = "SELECT TyID FROM Types WHERE Types.TyDesc = " & cmdType.Caption &
"";"
                Set rs = cn.Execute(strQuery)
                strQuery = "INSERT INTO ItemTypes (ItemTypeUPC, TypeID) VALUES (" &
txtUPCValue.Text & ", " & rs!TyID & ");"
            End If
        End If
    End If

```

```

        Set rs = cn.Execute(strQuery)
    Else
        If rs!TyDesc <> cmdType.Caption Then
            strQuery = "SELECT TyID FROM Types WHERE Types.TyDesc = " & cmdType.Caption &
            """,
                Set rs = cn.Execute(strQuery)
                strQuery = "UPDATE ItemTypes SET ItemTypes.TypeID = " & rs!TyID & " WHERE
ItemTypesUPC = " & txtUPCValue.Text & "";"
                Set rs = cn.Execute(strQuery)
            End If
        End If
    End If

    cmdClear_Click
    cn.Close
    Set rs = Nothing
    Set cn = Nothing
Else
    MsgBox ("Please scan both the RFID Tag and the UPC before clicking OK.")
End If

txtInputBox.SetFocus

End Sub
Private Sub cmdType_Click()

    frmMainMenu.MSComm1.Output = "HO0000"
    'if UPC is scanned, show type form
    If Len(txtUPCValue.Text) <> 0 Then
        'show Type form and disable AddItems
        frmChooseType.Show
        'MsgBox ("Under Construction")
    ElseIf Len(txtUPCValue.Text) = 0 Then
        MsgBox ("Please scan an item before selecting type.")
    End If

End Sub
Private Sub Form_Unload(Cancel As Integer)

    ' Debug.Print frmAddItems.Visible

    ' If Me.Visible = True Then
        frmMainMenu.MSComm1.Output = "HO0000"
    ' End If

End Sub

Private Sub imgUPC_DblClick()

    Dim retVal
    retVal = Shell("C:\Ski32.exe", 1)

End Sub

Private Sub txtDescription_Click()

```

```

txtInputBox.SetFocus

End Sub
Private Sub Form_Load()

'on form load initialize default values
strExpirationDate = ""
cmdExpiration.Caption = "Choose Date"
txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

For i = 0 To Me.Controls.Count - 1
    If TypeOf Me.Controls(i) Is CommandButton Then
        Me.Controls(i).BackColor = Colors.buttonColor
    ElseIf TypeOf Me.Controls(i) Is TextBox Then
        Me.Controls(i).BackColor = Colors.textBoxColor
    ElseIf TypeOf Me.Controls(i) Is Timer Then
    Else
        Me.Controls(i).BackColor = Colors.formColor
    End If
Next

Me.BackColor = Colors.formColor

cmdNone.BackColor = Colors.selectedButtonColor

End Sub
Private Sub cmdMainMenu_Click()

'when main menu button is clicked, clear form, hide 'add items' and enable main menu
cmdClear_Click
Unload Me

End Sub
Private Sub txtInputBox_Change()
' This function will handle all the Scanner Inputs, determining whether or not
' an RFID Tag is scanned, or an item barcode. Based on what was scanned, it
' will then search the database to gather all pertinent information and populate
' the display fields on the form. Also, if there is already a Tag UPC and Item
' UPC entered, it will insert data into the database and clear all fields.

'if there are 13 characters in the input box, open database connection
If Len(txtInputBox.Text) = 13 Then
    'if there is already a upc and rfid entered, put them in database, and clear form
    If Len(txtUPCValue.Text) = 13 And bRFIDFlag Then
        Set cn = New ADODB.Connection
        cn.ConnectionString = "DSN=UPC_Database;"
        cn.Open
        If Len(strExpirationDate) <> 0 Then
            strQuery = "INSERT INTO Fridge (TagID, FridgeUPC, DateAdd, Active, Expires) values (" &
intTagID & ", " & txtUPCValue.Text & ", " & Now & ", TRUE, " & strExpirationDate & ");"
        Else
            strQuery = "INSERT INTO Fridge (TagID, FridgeUPC, DateAdd, Active) values (" & intTagID
& ", " & txtUPCValue.Text & ", " & Now & ", TRUE);"
        End If
        Set rs = cn.Execute(strQuery)
        MsgBox (txtDescription.Text & " has been entered")
    End If
End Sub

```

```

cmdClear_Click
cn.Close
Set rs = Nothing
Set cn = Nothing
End If
Set cn = New ADODB.Connection
cn.ConnectionString = "DSN=UPC_Database;"
cn.Open
'if it is a Tag UPC, get the Tag ID from the 'tags' table
If Left(txtInputBox.Text, 7) = "2222222" Then
    strQuery = "SELECT * FROM Tags WHERE TagUPC = " & txtInputBox.Text & ";"
    Set rs = cn.Execute(strQuery)
    intTagID = rs!TagID
    imgRFID.Picture = LoadPicture(ImgGrnPath)
    txtRFIDValue.Text = txtInputBox.Text
    txtInputBox.Text = ""
    bRFIDFlag = True
    Set rs = Nothing
'otherwise, get the item's UPC information from the 'item' and 'type' tables
Else
    'get size a description information
    strQuery = "SELECT * FROM Items WHERE IID = " & txtInputBox.Text & ";"
    Set rs = cn.Execute(strQuery)
    'If no record was found for the UPC scanned, prompt user to enter New Item into database
    If rs.EOF Then
        intYesNo = MsgBox("Item not in database, would you like to add now?", vbYesNo, "Item Not
Found")
        If intYesNo = 6 Then
            strNewUPC = txtInputBox.Text
            txtInputBox.Text = ""
            frmNewitem.txtUPC.Text = strNewUPC
            frmNewitem.Show
        Else
            'run clear command
            cmdClear_Click
        End If
    Else
        If Len(rs!ISize) <> 0 Then
            txtDescription.Text = rs!ISize & " " & rs!IDescription
        Else
            txtDescription.Text = rs!IDescription
        End If
        strQuery = "SELECT TypeID FROM ItemTypes WHERE ItemTypeUPC = " &
txtInputBox.Text & ";"
        Set rs = cn.Execute(strQuery)
        'if there is a typeID get expiration date and type from Types table
        If rs.EOF = False Then
            RunType rs!TypeID
        End If
        imgUPC.Picture = LoadPicture(ImgGrnPath)
        txtUPCValue.Text = txtInputBox.Text
        txtInputBox.Text = ""
    End If
End If
cn.Close
Set rs = Nothing

```

```

        Set cn = Nothing
    End If

End Sub
Private Sub txtUPCValue_Click()

    txtInputBox.SetFocus

End Sub
Private Sub txtRFIDValue_Click()

    txtInputBox.SetFocus

End Sub
Public Sub RunType(typeID As Integer)

    Set cn2 = New ADODB.Connection
    cn2.ConnectionString = "DSN=UPC_Database;"
    cn2.Open
    'get Type ID from ItemType table
    strQuery = "SELECT * FROM Types WHERE TyID = " & typeID & ";"
    Set rs2 = cn2.Execute(strQuery)
    'if there is a value for expiration date, then set it. if not, leave as none
    If rs2!TyExp <> 0 Then
        strExpirationDate = DateAdd("d", rs2!TyExp, Date)
        cmdExpiration.Caption = strExpirationDate
        cmdNone.BackColor = Colors.buttonColor
        cmdExpiration.BackColor = Colors.selectedButtonColor
    End If
    'if there is a value for type, set it, if not, leave as none
    If rs2!TyDesc <> "" Then
        cmdType.Caption = rs2!TyDesc
    End If
    cn2.Close
    Set rs2 = Nothing
    Set cn2 = Nothing

End Sub

```

frmBarcodeEntry.frm

```
Attribute VB_Name = "frmBarcodeEntry"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim strLast As String
Dim strUPC As String
Dim i As Integer
Private Sub cmdNumbers_Click(Index As Integer)

    If Index <= 9 Then
        txtUPC.Text = txtUPC.Text & Index
    ElseIf Index = 11 Then
        If Len(txtUPC.Text) > 0 Then
            txtUPC.Text = Left(txtUPC.Text, Len(txtUPC.Text) - 1)
        End If
    ElseIf Index = 10 Then
        txtUPC.Text = txtUPC.Text & Chr(10)
        txtUPC.Text = ""
    ElseIf Index = 12 Then
        txtUPC.Text = ""
        ' frmAddItems.Enabled = True
        ' frmBarcodeEntry.Hide
        ' Unload Me
        frmAddItems.txtInputBox.SetFocus
        frmMainMenu.MSComm1.Output = "HO0003"
        frmBarcodeEntry.Hide
    ElseIf Index = 13 Then
        txtUPC.Text = ""
    End If

    If Index <> 12 And frmBarcodeEntry.Visible Then
        txtUPC.SetFocus
        txtUPC.SelStart = Len(txtUPC.Text)
    End If

End Sub
Private Sub Form_Load()

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Then
            Else
                Me.Controls(i).BackColor = Colors.formColor
            End If
        Next

    Me.BackColor = Colors.formColor
```

```

End Sub
Private Sub Form_Unload(Cancel As Integer)

    frmAddItems.txtInputBox.SetFocus
    frmMainMenu.MSComm1.Output = "HO0003"

End Sub

Private Sub txtUPC_Change()

    strLast = Right(txtUPC.Text, 1)
    If Len(strLast) > 0 Then
        If Asc(strLast) = 10 Then
            strUPC = Left(txtUPC.Text, Len(txtUPC.Text) - 1)
            ' txtUPC.Text = Left(txtUPC.Text, Len(txtUPC.Text) - 1)
            ' If Len(txtUPC.Text) = 12 Then
            '     txtUPC.Text = "0" & txtUPC.Text
            ' End If
            ' If Len(txtUPC.Text) = 13 Then
            ' If Len(strUPC) = 12 Then
            '     If Left(strUPC, 6) = "222222" Then
            '         strUPC = "2" & strUPC
            '     Else
            '         strUPC = "0" & strUPC
            '     End If
            ' End If
            ' If Len(strUPC) = 13 Then
            '     frmAddItems.Enabled = True
            '     Unload Me
            '     frmAddItems.txtInputBox.Text = strUPC
            '     frmAddItems.txtInputBox.SetFocus
            '     frmMainMenu.MSComm1.Output = "HO0003"
            '     frmBarcodeEntry.Hide
            '     strUPC = ""
            ' End If
            ' txtUPC.Text = strUPC
            ' txtUPC.Text = ""
            End If
        End If

    End Sub

Private Sub txtUPC_KeyPress(KeyAscii As Integer)

    If KeyAscii = 13 Then
        txtUPC.Text = txtUPC.Text & Chr(10)
    End If

End Sub

```

frmNoUPC.frm

```
Attribute VB_Name = "frmNoUPC"  
Attribute VB_GlobalNameSpace = False  
Attribute VB_Creatable = False  
Attribute VB_PredeclaredId = True  
Attribute VB_Exposed = False  
Option Explicit
```

```
Private rs As ADODB.Recordset  
Private cn As ADODB.Connection
```

```
Dim strQuery As String  
Dim strCategory As String  
Dim strUPCPart As String  
Dim intIndex As Integer  
Dim i As Integer  
Private Sub QueryDB()
```

```
    lvwItems.ListItems.Clear  
    Set cn = New ADODB.Connection  
    cn.ConnectionString = "DSN=UPC_Database;"  
    cn.Open  
    strQuery = "SELECT Items.IDescription, Items.IID " & _  
        "FROM Items, ItemCategory, Categories " & _  
        "WHERE Categories.Category='" & strCategory & "' " & _  
        "AND ItemCategory.CategoryID=CATEGORIES.CategoryID " & _  
        "AND Items.IID=ItemCategory.ItemUPC " & _  
        "AND Items.IID LIKE '222229%' " & _  
        "ORDER BY Items.IDescription ASC;"  
    Set rs = cn.Execute(strQuery)  
    If Not rs.EOF Then rs.MoveFirst  
    intIndex = 0  
    Do While Not rs.EOF  
        lvwItems.ListItems.Add , "a" & CStr(rs!IID), rs!IDescription  
        lvwItems.AddItem rs!IDescription  
        lvwItems.ItemData(intIndex) = CInt(Right(rs!IID, 6))  
        rs.MoveNext  
        intIndex = intIndex + 1  
    Loop  
    cn.Close  
    Set rs = Nothing  
    Set cn = Nothing
```

```
End Sub
```

```
Private Sub cmdAll_Click()
```

```
    lvwItems.ListItems.Clear  
    Set cn = New ADODB.Connection  
    cn.ConnectionString = "DSN=UPC_Database;"  
    cn.Open  
    strQuery = "SELECT Items.IDescription, Items.IID " & _  
        "FROM Items " & _  
        "WHERE Items.IID LIKE '222229%' " & _
```

```

        "ORDER BY Items.IDescription ASC;"
    Set rs = cn.Execute(strQuery)
    If Not rs.EOF Then rs.MoveFirst
    intIndex = 0
    Do While Not rs.EOF
        lvwItems.ListItems.Add , "a" & CStr(rs!IID), rs!IDescription
    '     lstItems.AddItem rs!IDescription
    '     lstItems.ItemData(intIndex) = CInt(Right(rs!IID, 6))
        rs.MoveNext
        intIndex = intIndex + 1
    Loop
    cn.Close
    Set rs = Nothing
    Set cn = Nothing

```

End Sub

```
Private Sub cmdCancel_Click()
```

```
    Unload Me
```

End Sub

```
Private Sub cmdBeverage_Click()
```

```
    strCategory = "Beverage"
    QueryDB
```

End Sub

```
Private Sub cmdFruit_Click()
```

```
    strCategory = "Fruit"
    QueryDB
```

End Sub

```
Private Sub cmdLeftovers_Click()
```

```
    strCategory = "Leftovers"
    QueryDB
```

End Sub

```
Private Sub cmdMeat_Click()
```

```
    strCategory = "Meat"
    QueryDB
```

End Sub

```
Private Sub cmdDairy_Click()
```

```
    strCategory = "Dairy"
    QueryDB
```

End Sub

Private Sub cmdNewItem_Click()

```
    strNewUPC = ""
    intnewupc = 0
    Unload Me
    frmNewitem.Show
```

End Sub

Private Sub cmdSeafood_Click()

```
    strCategory = "Seafood"
    QueryDB
```

End Sub

Private Sub cmdProduce_Click()

```
    strCategory = "Fruit"
    QueryDB
```

End Sub

Private Sub cmdVegetable_Click()

```
    strCategory = "Vegetable"
    QueryDB
```

End Sub

Private Sub cmdSelectedItem_Click()

```
    If lvwItems.ListItems.Count = 0 Then
        MsgBox ("There are no available items.")
    Else
        frmAddItems.txtInputBox.Text = Right(lvwItems.SelectedItem.Key, 13)
        Unload Me
    End If
```

End Sub

Private Sub Form_Load()

```
    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Then
            Else
                Me.Controls(i).BackColor = Colors.formColor
            End If
        End If
    Next
```

```
Me.BackColor = Colors.formColor
```

```
lvwItems.ColumnHeaders.Add , "Description", lvwItems.Width - 25
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    frmAddItems.txtInputBox.SetFocus
```

```
    frmMainMenu.MSComm1.Output = "HO0003"
```

```
End Sub
```

frmNewItem.frm

```
Attribute VB_Name = "frmNewitem"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection

Dim strQuery As String
Dim intWhichBox As Integer
Dim strAdd As String
Const DESCRIPTION_BOX = 1
Const SIZE_BOX = 2
Dim i As Integer
Private Sub cmdBackspace_Click()

    If intWhichBox = DESCRIPTION_BOX Then
        If Len(txtDescription.Text) >= 1 Then
            txtDescription.Text = Left(txtDescription.Text, Len(txtDescription.Text) - 1)
        End If
    '    txtDescription.Text = Left(txtDescription.Text, txtDescription.) & Right(txtDescription.Text,
    Len(txtDescription.Text) - txtDescription.SelStart - txtDescription.SelLength)
    Else
        If Len(txtSize.Text) >= 1 Then
            txtSize.Text = Left(txtSize.Text, Len(txtSize.Text) - 1)
        End If
    End If

End Sub
Private Sub cmdCancel_Click()

    Unload Me

End Sub

Private Sub cmdClear_Click()

    txtDescription.Text = ""
    txtSize.Text = ""

End Sub

Private Sub cmdEnter_Click()

    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    ' strQuery = "SELECT * FROM Items WHERE IID LIKE '2222229%' ORDER BY IID DESC;"
    ' Set rs = cn.Execute(strQuery)
    ' If strNewUPC = "" Then
    '     intnewupc = Val(rs!IID) + 1
```

```

'   strNewUPC = CStr(intnewupc)
'   End If
   strQuery = "INSERT INTO Items (IID, IDescription, ISize) values (" & txtUPC.Text & ", " &
txtDescription.Text & ", " & txtSize.Text & ");"
   Set rs = cn.Execute(strQuery)
   'sleep is necessary to allow database to be populated
   strQuery = "SELECT CategoryID FROM Categories WHERE Categories.Category = " &
cmbCategory.Text & ";"
   Set rs = cn.Execute(strQuery)
   strQuery = "INSERT INTO ItemCategory (CategoryID, ItemUPC) values (" & rs!CategoryID & ", " &
txtUPC.Text & ");"
   Set rs = cn.Execute(strQuery)
   Sleep 500
   cn.Close
   Set rs = Nothing
   Set cn = Nothing

   frmAddItems.txtInputBox.Text = txtUPC.Text

   Unload Me

End Sub

Private Sub cmdKey_Click(Index As Integer)

   strAdd = cmdKey(Index).Caption
   If chkShift.Value = False Then
       strAdd = LCase(strAdd)
   End If

   If intWhichBox = DESCRIPTION_BOX Then
       If txtDescription.Text = "Enter item description here. (i.e. 1% Milk, Coldcuts - Ham)" Then
txtDescription.Text = ""
           txtDescription.Text = txtDescription.Text + strAdd
       Else
           If txtSize.Text = "Enter item size here, if applicable. (i.e. 1 gallon, 1 liter, 1 dozen)" Then txtSize.Text
= ""
               txtSize.Text = txtSize.Text + strAdd
           End If
       End If

       chkShift.Value = False
   End Sub

Private Sub Form_Load()

   Set cn = New ADODB.Connection
   cn.ConnectionString = "DSN=UPC_Database;"
   cn.Open

   intWhichBox = DESCRIPTION_BOX

   For i = 0 To Me.Controls.Count - 1
       If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
           Me.Controls(i).BackColor = Colors.buttonColor
       ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then

```

```

        Me.Controls(i).BackColor = Colors.textBoxColor
    ElseIf TypeOf Me.Controls(i) Is Timer Then
    Else
        Me.Controls(i).BackColor = Colors.formColor
    End If
Next

Me.BackColor = Colors.formColor

If txtUPC.Text = "0000000000000" Then
    strQuery = "SELECT count(IID) as count FROM Items WHERE IID LIKE '2222229%';"
    Set rs = cn.Execute(strQuery)
    txtUPC.Text = "2222229" & Format(rs!Count, "000000")
End If

strQuery = "SELECT Category FROM Categories ORDER BY Category ASC"
Set rs = cn.Execute(strQuery)

rs.MoveFirst

Do While Not rs.EOF
    cmbCategory.AddItem rs!Category
    rs.MoveNext
Loop

cn.Close
Set rs = Nothing
Set cn = Nothing

' cmdNone.BackColor = Colors.selectedButtonColor

End Sub

Private Sub txtDescription_GotFocus()

    intWhichBox = DESCRIPTION_BOX
    If txtDescription.Text = "Enter item description here. (i.e. 1% Milk, Coldcuts - Ham)" Then
    txtDescription.Text = ""

End Sub
Private Sub txtSize_GotFocus()

    intWhichBox = SIZE_BOX
    If txtSize.Text = "Enter item size here, if applicable. (i.e. 1 gallon, 1 liter, 1 dozen)" Then txtSize.Text =
    ""

End Sub

```

frmChooseType.frm

```
Attribute VB_Name = "frmChooseType"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Dim strQuery As String
Dim intIndex As Integer
Private Sub cmdCancel_Click()

    Unload Me

End Sub
Private Sub cmdNewType_Click()

    frmNewType.Show

End Sub

Private Sub cmdSelectType_Click()

    If lvwTypes.ListItems.Count = 0 Then
        MsgBox ("There are no available items.")
    Else
        frmAddItems.RunType CInt(Right(lvwTypes.SelectedItem.Key, Len(lvwTypes.SelectedItem.Key) -
1))
        Unload Me
    End If

End Sub
Private Sub Form_Load()

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Then
        Else
            Me.Controls(i).BackColor = Colors.formColor
        End If
    Next

    Me.BackColor = Colors.formColor

    lvwTypes.ColumnHeaderHeaders.Add , , "Description", 375
    lvwTypes.ColumnHeaderHeaders.Add , , "Days to expire", 200

    Populate_List

End Sub
Public Sub Populate_List()
```

```

lvwTypes.ListItems.Clear
Set cn = New ADODB.Connection
cn.ConnectionString = "DSN=UPC_Database;"
cn.Open
strQuery = "SELECT * " & _
           "FROM Types " & _
           "ORDER BY Types.TyDesc ASC," & _
           "Types.TyExp ASC;"
Set rs = cn.Execute(strQuery)
If Not rs.EOF Then rs.MoveFirst
intIndex = 1
Do While Not rs.EOF
    lvwTypes.ListItems.Add , "a" & CStr(rs!TyID), rs!TyDesc
    lvwTypes.ListItems(intIndex).SubItems(1) = rs!TyExp
    lvwTypes.ItemData(intIndex) = CInt(Right(rs!TyID, 6))
    rs.MoveNext
    intIndex = intIndex + 1
Loop
cn.Close
Set rs = Nothing
Set cn = Nothing

End Sub

Private Sub Form_Unload(Cancel As Integer)

    frmAddItems.txtInputBox.SetFocus
    frmMainMenu.MSComm1.Output = "HO0003"

End Sub

```

frmNewType.frm

```
Attribute VB_Name = "frmNewType"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection

Dim strQuery As String
Dim intWhichBox As Integer
Dim strAdd As String
Const DESCRIPTION_BOX = 1
Const SIZE_BOX = 2
Dim i As Integer
Private Sub cmdBackspace_Click()

    If intWhichBox = DESCRIPTION_BOX Then
        If Len(txtDescription.Text) >= 1 Then
            txtDescription.Text = Left(txtDescription.Text, Len(txtDescription.Text) - 1)
        End If
    Else
        If Len(txtSize.Text) >= 1 Then
            txtSize.Text = Left(txtSize.Text, Len(txtSize.Text) - 1)
        End If
    End If

End Sub
Private Sub cmdCancel_Click()

    Unload Me

End Sub

Private Sub cmdClear_Click()

    txtDescription.Text = ""
    txtSize.Text = ""

End Sub

Private Sub cmdEnter_Click()

    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    ' strQuery = "SELECT * FROM Items WHERE IID LIKE '2222229%' ORDER BY IID DESC;"
    ' Set rs = cn.Execute(strQuery)
    ' If strNewUPC = "" Then
    '     intnewupc = Val(rs!IID) + 1
    '     strNewUPC = CStr(intnewupc)
    ' End If
```

```

    strQuery = "INSERT INTO Types (TyDesc, TyExp) values ('" & txtDescription.Text & "', " &
CInt(txtSize.Text) & ");"
    Set rs = cn.Execute(strQuery)
    Sleep 500
    cn.Close
    Set rs = Nothing
    Set cn = Nothing

' frmAddItems.txtType.Text = txtDescription.Text
' frmAddItems.

Unload Me

End Sub

Private Sub cmdKey_Click(Index As Integer)

    strAdd = cmdKey(Index).Caption
    If chkShift.Value = False Then
        strAdd = LCase(strAdd)
    End If

    If intWhichBox = DESCRIPTION_BOX Then
        If txtDescription.Text = "Enter type name here. (i.e. Milk, Coldcuts)" Then txtDescription.Text = ""
        txtDescription.Text = txtDescription.Text + strAdd
    Else
        If txtSize.Text = "Enter number of days until expiration here (i.e. 4, 12)" Then txtSize.Text = ""
        txtSize.Text = txtSize.Text + strAdd
    End If

    chkShift.Value = False

End Sub

Private Sub Form_Load()

    intWhichBox = DESCRIPTION_BOX

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Then
        Else
            Me.Controls(i).BackColor = Colors.formColor
        End If
    Next

    Me.BackColor = Colors.formColor

' cmdNone.BackColor = Colors.selectedButtonColor

End Sub

Private Sub Form_Unload(Cancel As Integer)

```

```

frmChooseType.Populate_List

End Sub
Private Sub txtDescription_GotFocus()

    intWhichBox = DESCRIPTION_BOX
    If txtDescription.Text = "Enter type name here. (i.e. Milk, Coldcuts)" Then txtDescription.Text = ""

    For i = 0 To 25
        cmdKey(i).Enabled = True
    Next

    For i = 36 To 48
        cmdKey(i).Enabled = True
    Next

    chkShift.Enabled = True

End Sub
Private Sub txtSize_GotFocus()

    intWhichBox = SIZE_BOX
    If txtSize.Text = "Enter number of days until expiration here (i.e. 4, 12)" Then txtSize.Text = ""

    For i = 0 To 25
        cmdKey(i).Enabled = False
    Next

    For i = 36 To 48
        cmdKey(i).Enabled = False
    Next

    chkShift.Enabled = False

End Sub

```

frmCalender.frm

```
Attribute VB_Name = "frmCalendar"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim i As Integer
Private Sub calExpiration_Click()

    ' frmAddItems.Enabled = True
    frmAddItems.cmdExpiration.Caption = calExpiration.Value
    frmAddItems.strExpirationDate = calExpiration.Value
    ' frmAddItems.txtInputBox.SetFocus
    ' frmCalendar.Hide
    Unload Me
End Sub

Private Sub cmdMonthBack_Click()

    If calExpiration.Month = 1 Then
        calExpiration.Month = 12
        calExpiration.Year = calExpiration.Year - 1
        lblYear.Caption = calExpiration.Year
    Else
        calExpiration.Month = calExpiration.Month - 1
    End If
    lblMonth.Caption = MonthName(calExpiration.Month)
End Sub

Private Sub cmdMonthForward_Click()

    If calExpiration.Month = 12 Then
        calExpiration.Month = 1
        calExpiration.Year = calExpiration.Year + 1
        lblYear.Caption = calExpiration.Year
    Else
        calExpiration.Month = calExpiration.Month + 1
    End If
    lblMonth.Caption = MonthName(calExpiration.Month)

End Sub
Private Sub cmdYearBack_Click()

    calExpiration.Year = calExpiration.Year - 1
    lblYear.Caption = calExpiration.Year

End Sub
Private Sub cmdYearForward_Click()

    calExpiration.Year = calExpiration.Year + 1
    lblYear.Caption = calExpiration.Year

End Sub
```

```

Private Sub Form_Load()

    calExpiration.Visible = True
    calExpiration.Month = Month(Now)
    calExpiration.Day = Day(Now)
    calExpiration.Year = Year(Now)
    lblMonth.Caption = MonthName(Month(Now))
    lblYear.Caption = Year(Now)

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Then
            Else
                Me.Controls(i).BackColor = Colors.formColor
            End If
        Next

    Me.BackColor = Colors.formColor

End Sub

Private Sub Form_Unload(Cancel As Integer)

    frmAddItems.cmdNone.BackColor = Colors.buttonColor
    frmAddItems.cmdExpiration.BackColor = Colors.selectedButtonColor

    frmAddItems.txtInputBox.SetFocus
    frmMainMenu.MSComm1.Output = "HO0003"
End Sub

```

frmShoppingList.frm

```
Attribute VB_Name = "frmShoppingList"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection

Dim strQuery As String
Dim intIndex As Integer
Dim i As Integer
Private Sub cmdAddList_Click()

    frmAddList.intSID = 0
    frmAddList.Show

End Sub

Private Sub cmdEditRemove_Click()

    If lvwShopList.ListItems.Count = 0 Then
        MsgBox ("There are no available items.")
    Else
        If CInt(Right(lvwShopList.SelectedItem.Key, Len(lvwShopList.SelectedItem.Key) - 1)) <> 0 Then
            frmAddList.intSID = CInt(Right(lvwShopList.SelectedItem.Key,
Len(lvwShopList.SelectedItem.Key) - 1))
            frmAddList.Show
        End If
    End If

End Sub

Private Sub cmdMainMenu_Click()

    Unload Me

End Sub

Private Sub cmdViewFridge_Click()

    Unload Me
    frmViewFridge.Show

End Sub

Private Sub Form_Load()

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
            Me.Controls(i).BackColor = Colors.buttonColor
        End If
    Next i

End Sub
```

```

    ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then
        Me.Controls(i).BackColor = Colors.textBoxColor
    ElseIf TypeOf Me.Controls(i) Is Timer Then
    Else
        Me.Controls(i).BackColor = Colors.formColor
    End If
Next

```

```

Me.BackColor = Colors.formColor

```

```

txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

```

```

lvwShopList.ColumnHeaders.Add , , "Qty", 50
lvwShopList.ColumnHeaders.Add , , "Size", 136
lvwShopList.ColumnHeaders.Add , , "Description", 400

```

```

RefreshList

```

```

End Sub

```

```

Sub RefreshList()

```

```

    lvwShopList.ListItems.Clear
    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    strQuery = "SELECT ShQty, ShSize, ShDesc, SID FROM ShoppingList ORDER BY ShDesc Asc"
    Set rs = cn.Execute(strQuery)
    If Not rs.EOF Then rs.MoveFirst
    intIndex = 1
    Do While Not rs.EOF
        lvwShopList.ListItems.Add , "a" & CStr(rs!SID), rs!ShQty
        If IsNull(rs!ShSize) Or rs!ShSize = "" Then
            lvwShopList.ListItems(intIndex).SubItems(1) = "N/A"
        Else
            lvwShopList.ListItems(intIndex).SubItems(1) = rs!ShSize
        End If
        lvwShopList.ListItems(intIndex).SubItems(2) = rs!ShDesc
        intIndex = intIndex + 1
        rs.MoveNext
    Loop
    cn.Close
    Set rs = Nothing
    Set cn = Nothing

```

```

End Sub

```

frmAddList.frm

```
Attribute VB_Name = "frmAddList"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection

Dim strQuery As String
Dim intWhichBox As Integer
Dim strAdd As String
Public intSID As Integer
Const DESCRIPTION_BOX = 1
Const QTY_BOX = 2
Const SIZE_BOX = 3
Dim i As Integer

Private Sub cmdAdd_Click()

    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    If intSID = 0 Then
        strQuery = "INSERT INTO ShoppingList (ShDesc, ShQty, ShSize) VALUES (" &
        txtDescription.Text & ", " & Val(txtQty.Text) & ", " & txtSize.Text & ");"
    Else
        strQuery = "UPDATE ShoppingList SET ShDesc = " & txtDescription.Text & ", ShQty = " &
        Val(txtQty.Text) & ", ShSize = " & txtSize.Text & " WHERE SID = " & intSID & ";"
    End If
    Set rs = cn.Execute(strQuery)
    Sleep 500
    Unload Me
    cn.Close
    Set rs = Nothing
    Set cn = Nothing
    frmShoppingList.RefreshList

End Sub

Private Sub cmdBackspace_Click()

    If intWhichBox = DESCRIPTION_BOX Then
        If txtDescription.Text = "Enter item description here." Then txtDescription.Text = ""
        If Len(txtDescription.Text) >= 1 Then
            txtDescription.Text = Left(txtDescription.Text, Len(txtDescription.Text) - 1)
        End If
    ElseIf intWhichBox = QTY_BOX Then
        If Len(txtQty.Text) >= 1 Then
            txtQty.Text = Left(txtQty.Text, Len(txtQty.Text) - 1)
        End If
    Else
        If Len(txtSize.Text) >= 1 Then
```

```

        txtSize.Text = Left(txtSize.Text, Len(txtSize.Text) - 1)
    End If
End If

End Sub
Private Sub cmdCancel_Click()

    Unload Me

End Sub

Private Sub cmdClear_Click()

    If intSID = 0 Then
        txtQty.Text = ""
        txtDescription.Text = ""
        txtSize.Text = ""
    Else
        Set cn = New ADODB.Connection
        cn.ConnectionString = "DSN=UPC_Database;"
        cn.Open
        strQuery = "DELETE FROM ShoppingList WHERE SID = " & intSID & ";"
        Set rs = cn.Execute(strQuery)
        Sleep 500
        Unload Me
        cn.Close
        Set rs = Nothing
        Set cn = Nothing
        frmShoppingList.RefreshList
    End If

End Sub

Private Sub cmdKey_Click(Index As Integer)

    strAdd = cmdKey(Index).Caption
    If chkShift.Value = False Then
        strAdd = LCase(strAdd)
    End If

    If intWhichBox = DESCRIPTION_BOX Then
        If txtDescription.Text = "Enter item description here." Then txtDescription.Text = ""
        txtDescription.Text = txtDescription.Text + strAdd
    ElseIf intWhichBox = QTY_BOX Then
        txtQty.Text = txtQty.Text + strAdd
    Else
        txtSize.Text = txtSize.Text + strAdd
    End If

    chkShift.Value = False

End Sub
Private Sub Form_Load()

    intWhichBox = DESCRIPTION_BOX

```

```

For i = 0 To Me.Controls.Count - 1
    If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
        Me.Controls(i).BackColor = Colors.buttonColor
    ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then
        Me.Controls(i).BackColor = Colors.textBoxColor
    ElseIf TypeOf Me.Controls(i) Is Timer Then
        Else
            Me.Controls(i).BackColor = Colors.formColor
        End If
Next

Me.BackColor = Colors.formColor

If intSID = 0 Then
    cmdClear.Caption = "Clear"
    cmdAdd.Caption = "Add item to shopping list"
Else
    cmdClear.Caption = "Remove item from list"
    cmdAdd.Caption = "Save changes"
    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    strQuery = "SELECT ShSize, ShDesc, ShQty FROM ShoppingList WHERE SID = " & intSID & ";"
    Set rs = cn.Execute(strQuery)
    If IsNull(rs!ShDesc) Then
        txtDescription.Text = ""
    Else
        txtDescription.Text = rs!ShDesc
    End If
    If IsNull(rs!ShQty) Then
        txtQty.Text = ""
    Else
        txtQty.Text = rs!ShQty
    End If
    If IsNull(rs!ShSize) Then
        txtSize.Text = ""
    Else
        txtSize.Text = rs!ShSize
    End If
    cn.Close
    Set rs = Nothing
    Set cn = Nothing
End If

End Sub
Private Sub txtDescription_GotFocus()

    intWhichBox = DESCRIPTION_BOX
    If txtDescription.Text = "Enter item description here." Then txtDescription.Text = ""

    For i = 0 To 48
        cmdKey(i).Enabled = True
    Next

    chkShift.Enabled = 48

```

End Sub

Private Sub txtQty_GotFocus()

intWhichBox = QTY_BOX

For i = 0 To 25

cmdKey(i).Enabled = False

Next

For i = 36 To 48

cmdKey(i).Enabled = False

Next

chkShift.Enabled = False

End Sub

Private Sub txtSize_GotFocus()

intWhichBox = SIZE_BOX

If txtSize.Text = "Enter item size here, if applicable. (i.e. 1 gallon)" Then txtSize.Text = ""

For i = 0 To 48

cmdKey(i).Enabled = True

Next

chkShift.Enabled = 48

End Sub

frmViewFridge.frm

```
Attribute VB_Name = "frmViewFridge"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection

Dim strQuery As String
Dim intIndex As Integer
Dim intLastSelected As Integer
Dim i As Integer
Private Sub RefreshList()

    If lvwFridge.ListItems.Count <> 0 Then
        intLastSelected = lvwFridge.SelectedItem.Index
    Else
        intLastSelected = -1
    End If

    lvwFridge.ListItems.Clear

    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    strQuery = "SELECT Items.IID, Items.ISize, Items.IDescription, Fridge.Expires, Fridge.ID FROM
Items, Fridge WHERE Items.IID=Fridge.FridgeUPC AND Fridge.InFridge = TRUE ORDER BY
Items.IDescription ASC;"
    Set rs = cn.Execute(strQuery)
    If Not rs.EOF Then rs.MoveFirst
    intIndex = 1
    Do While Not rs.EOF
        If IsNull(rs!ISize) Then
            lvwFridge.ListItems.Add , "a" & CStr(rs!ID), "N/A"
        Else
            lvwFridge.ListItems.Add , "a" & CStr(rs!ID), rs!ISize
        End If
        lvwFridge.ListItems(intIndex).SubItems(1) = rs!IDescription
        lvwFridge.ListItems(intIndex).SubItems(2) = Format(rs!Expires, "mm/dd/yy")
        intIndex = intIndex + 1
        rs.MoveNext
    Loop
    cn.Close
    Set rs = Nothing
    Set cn = Nothing

    If intLastSelected > -1 And intLastSelected <= lvwFridge.ListItems.Count Then
        lvwFridge.ListItems.Item(intLastSelected).Selected = True
    End If
```

```

End Sub
Private Sub Form_Load()

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Then
            Else
                Me.Controls(i).BackColor = Colors.formColor
            End If
        Next

    Me.BackColor = Colors.formColor

    txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

    lvwFridge.ColumnHeaders.Add , , "Size", 108
    lvwFridge.ColumnHeaders.Add , , "Description", 380
    lvwFridge.ColumnHeaders.Add , , "Exp. Date", 100

    tmrRefresh.Enabled = True

End Sub

Private Sub cmdMainMenu_Click()

    Unload Me

End Sub

Private Sub cmdAddList_Click()

    If lvwFridge.ListItems.Count = 0 Then
        MsgBox ("There are no available items.")
    Else
        Set cn = New ADODB.Connection
        cn.ConnectionString = "DSN=UPC_Database;"
        cn.Open
        strQuery = "SELECT FridgeUPC FROM Fridge WHERE Fridge.ID = " &
Right(lvwFridge.SelectedItem.Key, Len(lvwFridge.SelectedItem.Key) - 1) & ";"
        Set rs = cn.Execute(strQuery)
        strQuery = "INSERT INTO ShoppingList (ShUPC, ShSize, ShDesc, ShQty) values (" &
rs!FridgeUPC & ", " & lvwFridge.SelectedItem & ", " & lvwFridge.SelectedItem.SubItems(1) & ", 1);"
        Set rs = cn.Execute(strQuery)
        cn.Close
        Set rs = Nothing
        Set cn = Nothing
        Sleep 500
        MsgBox ("Item has been added to Shopping List.")
    End If

End Sub

```

```
Private Sub cmdViewList_Click()
```

```
    Unload Me  
    frmShoppingList.Show
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    tmrRefresh.Enabled = False
```

```
End Sub
```

```
Private Sub tmrRefresh_Timer()
```

```
    RefreshList
```

```
End Sub
```

frmRemoveItems.frm

```
Attribute VB_Name = "frmRemoveItems"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private rs As ADODB.Recordset
Private cn As ADODB.Connection

Dim strQuery As String
Dim intIndex As Integer
Dim intLastSelected As Integer
Dim i As Integer
Private Sub cmdAddList_Click()

    If lvwRemove.ListItems.Count = 0 Then
        MsgBox ("There are no available items.")
    Else
        Set cn = New ADODB.Connection
        cn.ConnectionString = "DSN=UPC_Database;"
        cn.Open
        strQuery = "SELECT f.FridgeUPC, i.IDescription, i.ISize FROM Fridge AS f, Items AS i WHERE
f.ID = " & Right(lvwRemove.SelectedItem.Key, Len(lvwRemove.SelectedItem.Key) - 1) & " AND i.IID =
f.FridgeUPC;"
        Set rs = cn.Execute(strQuery)
        strQuery = "INSERT INTO ShoppingList (ShUPC, ShSize, ShDesc, ShQty) values (" &
rs!FridgeUPC & ", " & rs!ISize & ", " & rs!IDescription & ", 1);"
        Set rs = cn.Execute(strQuery)
        cn.Close
        Set rs = Nothing
        Set cn = Nothing
        Sleep 500
        MsgBox ("Item has been added to Shopping List.")
    End If

End Sub

Private Sub cmdMainMenu_Click()

    Unload Me

End Sub

Private Sub cmdRemove_Click()

    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    intIndex = CInt(Right(lvwRemove.SelectedItem.Key, Len(lvwRemove.SelectedItem.Key) - 1))
    strQuery = "UPDATE Fridge SET Active = FALSE WHERE ID = " & intIndex & ";"
    Set rs = cn.Execute(strQuery)
    cn.Close

End Sub
```

```

Set rs = Nothing
Set cn = Nothing
MsgBox ("Selected item removed from the refrigerator.")
PopulateList

End Sub

Private Sub cmdViewList_Click()

    Unload Me
    frmShoppingList.Show

End Sub

Private Sub Form_Load()

    For i = 0 To Me.Controls.Count - 1
        If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
            Me.Controls(i).BackColor = Colors.buttonColor
        ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Then
            Me.Controls(i).BackColor = Colors.textBoxColor
        ElseIf TypeOf Me.Controls(i) Is Timer Then
        Else
            Me.Controls(i).BackColor = Colors.formColor
        End If
    Next

    Me.BackColor = Colors.formColor

    txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")

    lvwRemove.ColumnHeaders.Add , , "Description", 390
    lvwRemove.ColumnHeaders.Add , , "Last seen in fridge", 200

    tmrRefresh.Enabled = True

End Sub

Private Sub PopulateList()

    If lvwRemove.ListItems.Count <> 0 Then
        intLastSelected = lvwRemove.SelectedItem.Index
    Else
        intLastSelected = -1
    End If

    lvwRemove.ListItems.Clear
    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    strQuery = "SELECT Items.IID, Items.IDescription, Fridge.LastSeen, Fridge.ID FROM Items, Fridge
WHERE Fridge.Active = TRUE AND Fridge.InFridge = FALSE AND Items.IID=Fridge.FridgeUPC
ORDER BY Items.IDescription ASC;"
    Set rs = cn.Execute(strQuery)
    If Not rs.EOF Then rs.MoveFirst

```

```

intIndex = 1
Do While Not rs.EOF
    lvwRemove.ListItems.Add , "a" & CStr(rs!ID), rs!IDescription
    If IsNull(rs!LastSeen) Then
        lvwRemove.ListItems(intIndex).SubItems(1) = "never in fridge"
    Else
        lvwRemove.ListItems(intIndex).SubItems(1) = rs!LastSeen
    End If
    rs.MoveNext
    intIndex = intIndex + 1
Loop
'lvwRemove.SortKey = 0
cn.Close
Set rs = Nothing
Set cn = Nothing

If intLastSelected > -1 And intLastSelected <= lvwRemove.ListItems.Count Then
    lvwRemove.ListItems.Item(intLastSelected).Selected = True
End If

End Sub

Private Sub Form_Unload(Cancel As Integer)

    tmrRefresh.Enabled = False

End Sub

Private Sub tmrRefresh_Timer()

    PopulateList

End Sub

```

frmAlerts.frm

```
Attribute VB_Name = "frmAlerts"  
Attribute VB_GlobalNameSpace = False  
Attribute VB_Creatable = False  
Attribute VB_PredeclaredId = True  
Attribute VB_Exposed = False  
Option Explicit  
  
Private rs As ADODB.Recordset  
Private cn As ADODB.Connection  
  
Dim strQuery As String  
Dim intIndex As Integer  
Dim i As Integer  
Dim intLastSelected As Integer  
Private Sub chkAll_Click()  
  
    If chkAll.Value = 1 Then  
        chkExpired.Value = 1  
        chkExpiresSoon.Value = 1  
        chkOOF.Value = 1  
        PopulateList  
    End If  
  
End Sub  
Private Sub chkExpired_Click()  
  
    If chkExpired.Value = 0 Then  
        chkAll.Value = 0  
    End If  
  
End Sub  
  
Private Sub chkExpiresSoon_Click()  
  
    If chkExpiresSoon.Value = 0 Then  
        chkAll.Value = 0  
    End If  
  
End Sub  
  
Private Sub chkOOF_Click()  
  
    If chkOOF.Value = 0 Then  
        chkAll.Value = 0  
    End If  
  
End Sub  
  
Private Sub cmdAddList_Click()  
  
    If lvwAlerts.ListItems.Count = 0 Then  
        MsgBox ("There are no available items.")  
    Else
```

```

    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    strQuery = "SELECT f.FridgeUPC, i.ISize, i.IDescription FROM Fridge AS f, Items AS i WHERE
f.ID = " & Right(lvwAlerts.SelectedItem.Key, Len(lvwAlerts.SelectedItem.Key) - 1) & " AND
f.FridgeUPC = i.IID;"
    Set rs = cn.Execute(strQuery)
    strQuery = "INSERT INTO ShoppingList (ShUPC, ShSize, ShDesc, ShQty) values (" &
rs!FridgeUPC & ", " & rs!ISize & ", " & rs!IDescription & ", 1);"
    Set rs = cn.Execute(strQuery)
    cn.Close
    Set rs = Nothing
    Set cn = Nothing
    Sleep 500
    MsgBox ("Item has been added to Shopping List.")
End If

```

End Sub

```
Private Sub cmdMainMenu_Click()
```

```
    Unload Me
```

End Sub

```
Private Sub cmdRemove_Click()
```

```

    If lvwAlerts.ListItems.Count = 0 Then
        MsgBox ("There are no available items.")
    Else
        Set cn = New ADODB.Connection
        cn.ConnectionString = "DSN=UPC_Database;"
        cn.Open
        intIndex = Cint(Right(lvwAlerts.SelectedItem.Key, Len(lvwAlerts.SelectedItem.Key) - 1))
        strQuery = "UPDATE Fridge SET Active = FALSE WHERE ID = " & intIndex & ";"
        Set rs = cn.Execute(strQuery)
        cn.Close
        Set rs = Nothing
        Set cn = Nothing
        MsgBox ("Selected item removed from the refrigerator.")
        PopulateList
    End If

```

End Sub

```
Private Sub cmdViewList_Click()
```

```

    Unload Me
    frmShoppingList.Show

```

End Sub

```
Private Sub Form_Load()
```

```
    txtDateTime.Caption = Date & " " & Format(Now, "Medium Time")
```

```

lvwAlerts.ColumnHeaders.Add , , "Alert Type", 100
lvwAlerts.ColumnHeaders.Add , , "Size", 100
lvwAlerts.ColumnHeaders.Add , , "Description", 290
lvwAlerts.ColumnHeaders.Add , , "Exp. Date", 100

tmrRefresh.Enabled = True

For i = 0 To Me.Controls.Count - 1
    If TypeOf Me.Controls(i) Is CommandButton Or TypeOf Me.Controls(i) Is CheckBox Then
        Me.Controls(i).BackColor = Colors.buttonColor
    ElseIf TypeOf Me.Controls(i) Is TextBox Or TypeOf Me.Controls(i) Is ComboBox Or TypeOf
Me.Controls(i) Is ListView Or TypeOf Me.Controls(i) Is ListView Then
        Me.Controls(i).BackColor = Colors.textBoxColor
    ElseIf TypeOf Me.Controls(i) Is Timer Or TypeOf Me.Controls(i) Is MSComm Then
    Else
        Me.Controls(i).BackColor = Colors.formColor
    End If
Next

Me.BackColor = Colors.formColor

chkAll.Value = 1

End Sub

Private Sub PopulateList()

    If lvwAlerts.ListItems.Count <> 0 Then
        intLastSelected = lvwAlerts.SelectedItem.Index
    Else
        intLastSelected = -1
    End If

    lvwAlerts.ListItems.Clear
    Set cn = New ADODB.Connection
    cn.ConnectionString = "DSN=UPC_Database;"
    cn.Open
    intIndex = 1
    If chkExpired.Value = 1 Then
        strQuery = "SELECT f.ID, f.FridgeUPC, f.Expires, i.IDescription, i.ISize FROM Fridge AS f, Items
AS i WHERE f.FridgeUPC = i.IID AND f.Active = TRUE AND f.Expires < NOW ORDER BY f.Expires
ASC;"
        Set rs = cn.Execute(strQuery)
        If Not rs.EOF Then rs.MoveFirst
        Do While Not rs.EOF
            lvwAlerts.ListItems.Add , "a" & CStr(rs!ID), "Expired"
            If IsNull(rs!ISize) Then
                lvwAlerts.ListItems(intIndex).SubItems(1) = "N/A"
            Else
                lvwAlerts.ListItems(intIndex).SubItems(1) = rs!ISize
            End If
            lvwAlerts.ListItems(intIndex).SubItems(2) = rs!IDescription
            lvwAlerts.ListItems(intIndex).SubItems(3) = rs!Expires
            rs.MoveNext
            intIndex = intIndex + 1
        Loop
    End If
End Sub

```

```

End If
If chkExpiresSoon.Value = 1 Then
    strQuery = "SELECT f.ID, f.FridgeUPC, f.Expires, i.IDescription, i.ISize FROM Fridge AS f, Items
AS i WHERE f.FridgeUPC = i.IID AND f.Active = TRUE AND f.Expires < NOW + 2 AND f.Expires >=
NOW ORDER BY f.Expires ASC;"
    Set rs = cn.Execute(strQuery)
    If Not rs.EOF Then rs.MoveFirst
    Do While Not rs.EOF
        lvwAlerts.ListItems.Add , "b" & CStr(rs!ID), "Expires Soon"
        If IsNull(rs!ISize) Then
            lvwAlerts.ListItems(intIndex).SubItems(1) = "N/A"
        Else
            lvwAlerts.ListItems(intIndex).SubItems(1) = rs!ISize
        End If
        lvwAlerts.ListItems(intIndex).SubItems(2) = rs!IDescription
        lvwAlerts.ListItems(intIndex).SubItems(3) = rs!Expires
        rs.MoveNext
        intIndex = intIndex + 1
    Loop
End If
If chkOOF.Value = 1 Then
    strQuery = "SELECT f.ID, f.FridgeUPC, f.Expires, i.IDescription, i.ISize FROM Fridge AS f, Items
AS i WHERE f.FridgeUPC = i.IID AND f.Active = TRUE AND f.InFridge = FALSE ORDER BY
f.Expires ASC;"
    Set rs = cn.Execute(strQuery)
    If Not rs.EOF Then rs.MoveFirst
    Do While Not rs.EOF
        lvwAlerts.ListItems.Add , "c" & CStr(rs!ID), "Out of Fridge"
        If IsNull(rs!ISize) Then
            lvwAlerts.ListItems(intIndex).SubItems(1) = "N/A"
        Else
            lvwAlerts.ListItems(intIndex).SubItems(1) = rs!ISize
        End If
        lvwAlerts.ListItems(intIndex).SubItems(2) = rs!IDescription
        If IsNull(rs!Expires) Then
            lvwAlerts.ListItems(intIndex).SubItems(3) = ""
        Else
            lvwAlerts.ListItems(intIndex).SubItems(3) = rs!Expires
        End If
        rs.MoveNext
        intIndex = intIndex + 1
    Loop
End If
cn.Close
Set rs = Nothing
Set cn = Nothing

If intLastSelected > -1 And intLastSelected <= lvwAlerts.ListItems.Count Then
    lvwAlerts.ListItems.Item(intLastSelected).Selected = True
End If

End Sub

Private Sub Form_Unload(Cancel As Integer)

    tmrRefresh.Enabled = False

```

End Sub

Private Sub tmrRefresh_Timer()

 PopulateList

End Sub

Appendix B: Web Source Code

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
<meta name="description" content="RFID Refrigerator Web Interface" />
<meta name="keywords" content="RFID, Refrigerator, Northeastern University, Capstone" />
<meta name="author" content="Capstone Group 9 / Original design: Andreas Viklund -
http://andreasviklund.com/" />
<link rel="stylesheet" type="text/css" href="andreas03.css" />
<title>RFID Refrigerator</title>
</head>

<body>
<div id="thetop">
<a id="top"></a>
<p class="hide">Skip to: <a href="#sitemenu" accesskey="2">Site menu</a> | <a href="#maincontent"
accesskey="3">Main content</a></p>
</div>

<div id="container">
<div id="main">

<div id="logo">
<h1>[<a href="index.html" accesskey="4">IntelliFridge</a>]</h1>
<span id="tagline">Who has time to open the door?</span>
</div>

<div id="intro">

<br /><br /><h2><a id="maincontent"></a>Remote Fridge Manager</h2>
<p>Use the menu to the right to navigate to various refrigerator options.</p>
</div>

<div class="clear"></div>
<h3 class="headerstyle">Welcome to the IntelliFridge</h3>
<p>
</p>
<p><center></center></p>

</div>

<div id="sidebar">
<center></center>
<h2 class="sidelink menuheader"><a id="sitemenu"></a>Main menu:</h2>

<a class="sidelink" href="browse.html">Browse Refrigerator</a>
<span class="hide"> | </span>
<a class="sidelink" href="shoplist.html">Shopping List</a>
<span class="hide"> | </span>
```

```
<a class="sidelink" href="alerts.html">Alerts</a>  
<span class="hide"> | </span>
```

```
</div>  
<div class="clear">&nbsp;</div>  
</div>
```

```
<div id="footer"><p>&copy; 2007 Capstone Group 9. Design by <a href="http://andreasviklund.com/"  
class="credit">Andreas Viklund</a> of <a href="http://jokkmokk.biz" title="ITUS  
Jokkmokk">Jokkmokk</a>.</p></div>
```

```
</body>  
</html>
```

browse.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
<meta name="description" content="RFID Refrigerator Web Interface" />
<meta name="keywords" content="RFID, Refrigerator, Northeastern University, Capstone" />
<meta name="author" content="Capstone Group 9 / Original design: Andreas Viklund -
http://andreasviklund.com/" />
<link rel="stylesheet" type="text/css" href="andreas03.css" />
<title>RFID Refrigerator</title>
</head>

<?php

// Login to mySQL and select database "pictures"
$db = odbc_connect("UPC_Database", "", "");

echo"
<body>
<div id="thetop">
<a id="top"></a>
<p class="hide">Skip to: <a href="#sitemenu" accesskey="2">Site menu</a> | <a
href="#maincontent" accesskey="3">Main content</a></p>
</div>

<div id="container">
<div id="main">

<div id="logo">
<h1>[<a href="index.html" accesskey="4">IntelliFridge</a>]</h1>
<span id="tagline">Who has time to open the door?</span>
</div>

<div id="intro">

<br /><br /><h2><a id="maincontent"></a>Remote Fridge Manager</h2>
<p>Use the menu to the right to navigate to various refrigerator options.</p>
</div>

<div class="clear"></div>;

$result = odbc_exec($db,"SELECT Items.ISize, Items.IDescription, Format(Fridge.Expires, 'mm-dd-yy')
AS expdate, Fridge.InFridge, Items.IID FROM Items, Fridge WHERE Items.IID = Fridge.FridgeUPC
AND Fridge.Active = TRUE;");
echo"
<h3 class="headerstyle">Refrigerator Contents</h3>
<p>
<table align=center>
<tr>
<td width="75"><b>Size</b></td>
<td width="200"><b>Description</b></td>
```

```

                <td width="75"><b>Expires</b></td>
                <td width="75"><b>In Fridge?</b></td>
                <td width="75"></td>
            </tr>";
    $i=0;
    while($row = odbc_fetch_array($result)){
        if($i % 2 == 0){
            $c="";
        }
        else{
            $c="#EEEEEE";
        }
        echo"
        <tr bgcolor=\"$c\">
            <td width="75">".$row['ISize']."</td>
            <td width="200">".$row['IDescription']."</td>
            <td width="75">".$row['expdate']."</td>";
            if($row['InFridge']){
                echo"<td width="75" align="center">yes</td>";
            }
            else{
                echo"<td width="75" align="center">no</td>";
            }
            echo"<td width="75"><a href="/shoplist.html?do=add&upc=".$row['IID'].">add
to<br>shopping list</a></td>
        </tr>";
        $i++;
    }
    echo"</table></p>";

```

```

echo"<p></p>

```

```

</div>

```

```

<div id="sidebar">
<center></center>
<h2 class="sidelink menuheader"><a id="sitemenu"><a>Main menu:</h2>

```

```

<h2 class="selected">Browse Refrigerator</h2>
<span class="hide"> | </span>
<a class="sidelink" href="/shoplist.html">Shopping List</a>
<span class="hide"> | </span>
<a class="sidelink" href="/alerts.html">Alerts</a>
<span class="hide"> | </span>

```

```

</div>

```

```

<div class="clear">&nbsp;</div>

```

```

</div>

```

```

<div id="footer"><p>&copy; 2007 Capstone Group 9. Design by <a href="http://andreasviklund.com/"
class="credit">Andreas Viklund</a> of <a href="http://jokkmokk.biz/" title="ITUS
Jokkmokk">Jokkmokk</a>.</p></div>

```

```
</body>  
</html>";
```

```
odbc_free_result($result);  
odbc_close($db);  
?>
```

shoplist.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
<meta name="description" content="RFID Refrigerator Web Interface" />
<meta name="keywords" content="RFID, Refrigerator, Northeastern University, Capstone" />
<meta name="author" content="Capstone Group 9 / Original design: Andreas Viklund -
http://andreasviklund.com/" />
<link rel="stylesheet" type="text/css" href="andreas03.css" />
<title>RFID Refrigerator</title>
</head>

<?php

// Login to mySQL and select database "pictures"
$db = mysql_connect("UPC_Database", "", "");
$dothis=$_GET['do'];
$upc=$_GET['upc'];
$qty=$_POST['qty'];
$size=$_POST['size'];
$desc=$_POST['desc'];
$del=$_POST['del'];

echo"
<body>
<div id="thetop">
<a id="top"></a>
<p class="hide">Skip to: <a href="#" sitemenu" accesskey="2">Site menu</a> | <a
href="#" maincontent" accesskey="3">Main content</a></p>
</div>

<div id="container">
<div id="main">

<div id="logo">
<h1>[<a href="index.html" accesskey="4">IntelliFridge</a>]</h1>
<span id="tagline">Who has time to open the door?</span>
</div>

<div id="intro">

<br /><br /><h2><a id="maincontent"></a>Remote Fridge Manager</h2>
<p>Use the menu to the right to navigate to various refrigerator options.</p>
</div>

<div class="clear"></div>;

if($dothis=='add'){
    echo"<h3 class="headerstyle">Add Items to Shopping List</h3>
    <p>
    <form method="post" action="/shoplist.html">
    <table align="center">
```

```

        <tr>
            <td width="50"><b>Quantity</b></td>
            <td width="50"><b>Size</b></td>
            <td width="400"><b>Description</b></td>
        </tr>";
for($i=0;$i<10;$i++){
    echo"<tr>";
    if(isset($_GET['upc']) && $i==0){
        $result = odbc_exec($db,"SELECT ISize, IDescription FROM Items WHERE
IID='".$_Supc.'"");
        $row = odbc_fetch_array($result);
        echo"<td width="50"><input type="text" size="5" maxlength="5"
name="qty[]" value="1"></td>
        <td width="100"><input type="text" size="10" maxlength="50"
name="size[]" value="".$row['ISize']."></td>
        <td width="350"><input type="text" size="50" maxlength="50"
name="desc[]" value="".$row['IDescription']."></td>";
    }
    else{
        echo"<td width="50"><input type="text" size="5" maxlength="5"
name="qty[]"></td>
        <td width="100"><input type="text" size="10" maxlength="50"
name="size[]"></td>
        <td width="350"><input type="text" size="50" maxlength="50"
name="desc[]"></td>";
    }
    echo"</tr>";
}
echo"</table>
</p>
<p><center>
<input type="submit" name="add" value="add to shopping list">
<input type="reset" name="clear" value="clear">
</center></p>
</form>";
}

else if($dothis=='edit'){
    echo"<h3 class='headerstyle'>Edit Items in Shopping List</h3>
<p>
<form method="post" action="/shoplist.html">
<table align="center">
    <tr>
        <td width="50"><b>Quantity</b></td>
        <td width="50"><b>Size</b></td>
        <td width="350"><b>Description</b></td>
        <td width="50"><b>Remove?</b></td>
    </tr>";
    $result=odbc_exec($db,"SELECT * FROM ShoppingList;");
    while($row=odbc_fetch_array($result)){
        echo"<tr>
            <td width="50"><input type="text" size="5" maxlength="5"
name="qty[]" value="".$row['ShQty']."></td>
            <td width="50"><input type="text" size="10" maxlength="50"
name="size[]" value="".$row['ShSize']."></td>

```

```

                <td width="350"><input type="text" size="50" maxlength="50"
name="desc[]" value="" .$row['ShDesc']. "></td>
                <td width="50"><input type="checkbox" name="del[]"
value="" .$row['SID']. "></td>
            </tr>";
        }
        echo"</table>
    </p>
    <p><center>
    <input type="submit" name="update" value="update shopping list">
    </center></p>
    </form>";
}

else{
    if(isset($_POST['add'])){
        $num=0;
        foreach($qty as $q){
            if($q!=0){
                $num++;
            }
        }
        for($i=0;$i<$num;$i++){
            $result = odbc_exec($db,"INSERT INTO ShoppingList (ShSize, ShDesc,
ShQty) VALUES (" . $size[$i]. ", " . $desc[$i]. ", " . intval($qty[$i]). ");");
        }
    }
    else if(isset($_POST['update'])){
        $result=odbc_exec($db,"SELECT * FROM ShoppingList;");
        while($row=odbc_fetch_array($result)){
            $num[]=$row['SID'];
        }
        $i=0;
        foreach($num as $n){
            foreach($del as $d){
                if(intval($d) == $n){
                    $result=odbc_exec($db,"DELETE FROM ShoppingList
WHERE SID=" . $n. ";");
                }
                else{
                    $result=odbc_exec($db,"UPDATE ShoppingList SET
ShQty=" . intval($qty[$i]). ", ShDesc=" . $desc[$i]. ", ShSize=" . $size[$i]. " WHERE SID=" . $n. ";");
                }
            }
            $i++;
        }
    }
    $result = odbc_exec($db,"SELECT * FROM ShoppingList;");
    echo"
    <h3 class="headerstyle">Shopping List</h3>
    <p>
    <table align="center">
        <tr>
            <td width="50"><b>Qty</b></td>
            <td width="50"><b>Size</b></td>
            <td width="300"><b>Description</b></td>

```

```

        </tr>";
    $j=0;
    while($row = odbc_fetch_array($result)){
        if($j % 2 == 0){
            $c="";
        }
        else{
            $c="#EEEEEE";
        }
        echo"
        <tr bgcolor=\\".$c.\">
            <td width=\\"50\">".$row['ShQty']. "</td>
            <td width=\\"50\">".$row['ShSize']. "</td>
            <td width=\\"300\">".$row['ShDesc']. "</td>
        </td>";
        $j++;
    }
    echo"</table></p>";
}

echo"<p></p>

</div>

<div id=\\"sidebar\">
<center><img src=\\"/img/microfridge_open_s.jpg\"></center>
<h2 class=\\"sidelink menuheader\"><a id=\\"sitemenu\"></a>Main menu:</h2>

<a class=\\"sidelink\" href=\\"browse.html\">Browse Refrigerator</a>
<span class=\\"hide\"> | </span>
<h2 class=\\"selected\">Shopping List</h2>
<span class=\\"hide\"> | </span>
<p>
- <a href=\\"/shoplist.html\">View Shopping List</a><br />
- <a href=\\"/printlist.html\" target=\\"_blank\">Print Shopping List</a><br />
- <a href=\\"/shoplist.html?do=add\">Add Items to List</a><br />
- <a href=\\"/shoplist.html?do=edit\">Edit Items in List</a>
</p>
<a class=\\"sidelink\" href=\\"alerts.html\">Alerts</a>
<span class=\\"hide\"> | </span>

</div>
<div class=\\"clear\">&nbsp;</div>
</div>

<div id=\\"footer\"><p>&copy; 2007 Capstone Group 9. Design by <a href=\\"http://andreasviklund.com\"
class=\\"credit\">Andreas Viklund</a> of <a href=\\"http://jokkmokk.biz\" title=\\"ITUS
Jokkmokk\">Jokkmokk</a>.</p></div>

</body>
</html>";

odbc_free_result($result);
odbc_close($db);
?>

```

printlist.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
<meta name="description" content="RFID Refrigerator Web Interface" />
<meta name="keywords" content="RFID, Refrigerator, Northeastern University, Capstone" />
<meta name="author" content="Capstone Group 9 / Original design: Andreas Viklund -
http://andreasviklund.com/" />
<title>Shopping list</title>
</head>

<body>

<?php
$db = odbc_connect("UPC_Database", "", "");
$result = odbc_exec($db,"SELECT * FROM ShoppingList;");
echo"
<p><b>Shopping List</b>
<table>
    <tr>
        <td width=50><b>Qty</b></td>
        <td width=300><b>Description</b></td>
    </tr>";
while($row = odbc_fetch_array($result)){
    echo"
    <tr>
        <td width=30>".$row['ShQty']."</td>
        <td width=200>".$row['ShDesc']."</td>
    </tr>";
}
echo"</table></p>";

?>
</body>
</html>
```

alerts.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1" />
<meta name="description" content="RFID Refrigerator Web Interface" />
<meta name="keywords" content="RFID, Refrigerator, Northeastern University, Capstone" />
<meta name="author" content="Capstone Group 9 / Original design: Andreas Viklund -
http://andreasviklund.com/" />
<link rel="stylesheet" type="text/css" href="andreas03.css" />
<title>RFID Refrigerator</title>
</head>

<?php

// Login to mySQL and select database "pictures"
$db = odbc_connect("UPC_Database", "", "");

echo"
<body>
<div id="thetop">
<a id="top"></a>
<p class="hide">Skip to: <a href="#">Site menu</a> | <a
href="#">Main content</a></p>
</div>

<div id="container">
<div id="main">

<div id="logo">
<h1><a href="index.html" accesskey="4">IntelliFridge</a></h1>
<span id="tagline">Who has time to open the door?</span>
</div>

<div id="intro">

<br /><br /><h2><a id="maincontent"></a>Remote Fridge Manager</h2>
<p>Use the menu to the right to navigate to various refrigerator options.</p>
</div>

<div class="clear"></div>;

//Generate main content based on menu selection
$result = odbc_exec($db,"SELECT Items.ISize, Items.IDescription, (Fridge.Expires<Date()) AS expired,
(DateAdd('d',2,Fridge.Expires)=Date()) AS soon1, (DateAdd('d',1,Fridge.Expires)=Date()) AS soon2,
(Fridge.Expires=Date()) AS today, Format(Fridge.Expires,'mm-dd-yy') AS Expires, Fridge.InFridge
FROM Items, Fridge WHERE (((Items.IID)=[Fridge].[FridgeUPC]) AND ((Fridge.Active)=True)),");
echo"
<h3 class="headerstyle">Alerts</h3>
<p>
<table align=center>
<tr>
<td width="100"><b>Alert</b></td>
```

```

        <td width="75"><b>Size</b></td>
        <td width="250"><b>Description</b></td>
        <td width="75"><b>Expires</b></td>
    </tr>";
while($row = odbc_fetch_array($result)){
    if($row['expired']){
        $alert="Expired";
        $c="#FF9999";
    }
    else if($row['soon1'] || $row['soon2']){
        $alert="Expires Soon";
        $c="#FFFFFF";
    }
    else if($row['today']){
        $alert="Expires Today";
        $c="#FFF999";
    }
    else if($row['InFridge']==FALSE){
        $alert="Out of Fridge";
        $c="#CCFFFF";
    }
    else{
        $alert="";
        $c="";
    }
    if($alert!=""){
        echo"
        <tr bgcolor=\"$c\">
            <td width="100">\".$alert.\"</td>
            <td width="75">\".$row['ISize'].\"</td>
            <td width="250">\".$row['IDescription'].\"</td>
            <td width="75">\".$row['Expires'].\"</td>
        </tr>";
    }
}
echo"</table>
</p>
<p>
<table>
    <tr>
        <td width="20"></td>
        <td width="15" style="border:solid 1px\" bgcolor="#FFFFFF"></td>
        <td width="100">Expires Soon</td>
        <td width="15" style="border:solid 1px\" bgcolor="#FFF999"></td>
        <td width="100">Expires Today</td>
        <td width="15" style="border:solid 1px\" bgcolor="#FF9999"></td>
        <td width="100">Expired</td>
        <td width="15" style="border:solid 1px\" bgcolor="#CCFFFF"></td>
        <td width="100">Out of Fridge</td>
        <td width="20"></td>
    </tr>
</table></p>";

echo"<p></p>

```

```
</div>
```

```
<div id="sidebar">
```

```
<center></center>
```

```
<h2 class="sidelink menuheader"><a id="sitemenu"></a>Main menu:</h2>
```

```
<a class="sidelink" href="browse.html">Browse Refrigerator</a>
```

```
<span class="hide"> | </span>
```

```
<a class="sidelink" href="shoplist.html">Shopping List</a>
```

```
<span class="hide"> | </span>
```

```
<h2 class="selected">Alerts</h2>
```

```
<span class="hide"> | </span>
```

```
</div>
```

```
<div class="clear">&nbsp;</div>
```

```
</div>
```

```
<div id="footer"><p>&copy; 2007 Capstone Group 9. Design by <a href="http://andreasviklund.com/" class="credit">Andreas Viklund</a> of <a href="http://jokkmokk.biz/" title="ITUS Jokkmokk">Jokkmokk</a>.</p></div>
```

```
</body>
```

```
</html>";
```

```
odbc_free_result($result);
```

```
odbc_close($db);
```

```
?>
```