NFC-Enabled Menu Ordering System

Design Review

ECE 445
TA: Lydia Majure
Members: Patrick Ding, Patric Takagi, Yau Chan
Team #47
2/12/2013

Table of Contents

I. Introduction:

1.1 Title: NiFty (Near Field) Menu

1.2 Project Description

Customers at restaurants desire prompt service, convenient ordering, and great food. We have no control over the quality of food served; however, our project is designed to increase customer satisfaction by taking advantage of the first two goals and making the restaurant experience as quick and painless as possible.

Another problem that affects restaurants is the hectic queue due to a big lunch or dinner rush. Checks may be misplaced in the kitchen, and customers will be forced to wait for their misplaced order to be prepared. Also, there is an innate bottleneck when only waiters can place orders. A table may be ready to order, but their waiter may be serving food or cleaning up after another table. We wanted to find a solution that would allow patrons to place an order whenever they were ready.

Our idea is to help restaurants solve these problems by implementing a low-cost solution to increasing customer and restaurant staff satisfaction. The idea is creating a menu using NFC technology. NFC tags will be affixed to the restaurant menu with a specifically-designed NFC reader hard-wired to the menu. Customers can use the reader to select menu items by hovering over the appropriate NFC tags on their menus. Since NFC-tags are so low profile (almost as small as a sticker!), restaurants can keep the current design/layout of their menus and integrate our new technology with low switching costs.

The menu will contain an RF module that will send the order directly to the kitchen, reducing the work that waiters need to do. This will allow the restaurant to hire less waiters and will also improve any miscommunication between server and patron. Imagine if you are in a foreign country where you do not speak the native language. Our solution will allow you to intuitively point and select the items you want, keeping difficult communication between server and guest minimal.

Of course, our product will not be attractive to every restaurant. Some restaurants thrive through the friendly service that their staff brings to the table. Our product is designed for fast-casual type restaurants where you are solely interested in getting your food and eating.

1.3 Goals, Functions, Benefits, and Features:

Goals & Functions:

- Improve customer and staff restaurant experience
- Up to 256 menus can be used at once
- Up to 256 items available for purchase on item catalogue (menu)
- Up to 128 unique selections can be transmitted in each order
- Small NFC Reader to detect and send NFC tags to microcontroller memory
- RF communication link between menu and kitchen
- Simple yet powerful computer user interface in kitchen for staff use

Benefits:

- Low-cost compared to other electronic menu options
- Reduction in wait staff required for dine-in customers
- Quick and accurate processing of take-out orders
- Enables precise queueing in the kitchen
- Simple integration with current menu (no significant redesign required)
- Eliminates need for cumbersome communication between server and guest

Features:

- Sleek menu with intuitive point-and-select interface
- Fast wired communication between reader and menu microcontroller
- LCD display to aid order selection/confirmation
- Backlit LCD module for use in dim restaurants
- Quick RF communication between menu and kitchen
- Battery lifetime of up to six months on a single 9V battery

II. Design:

2.1 Block Diagrams

Figure 1: Top Level Block Diagram

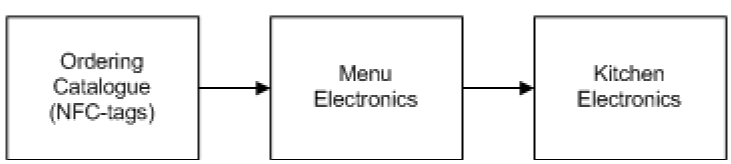
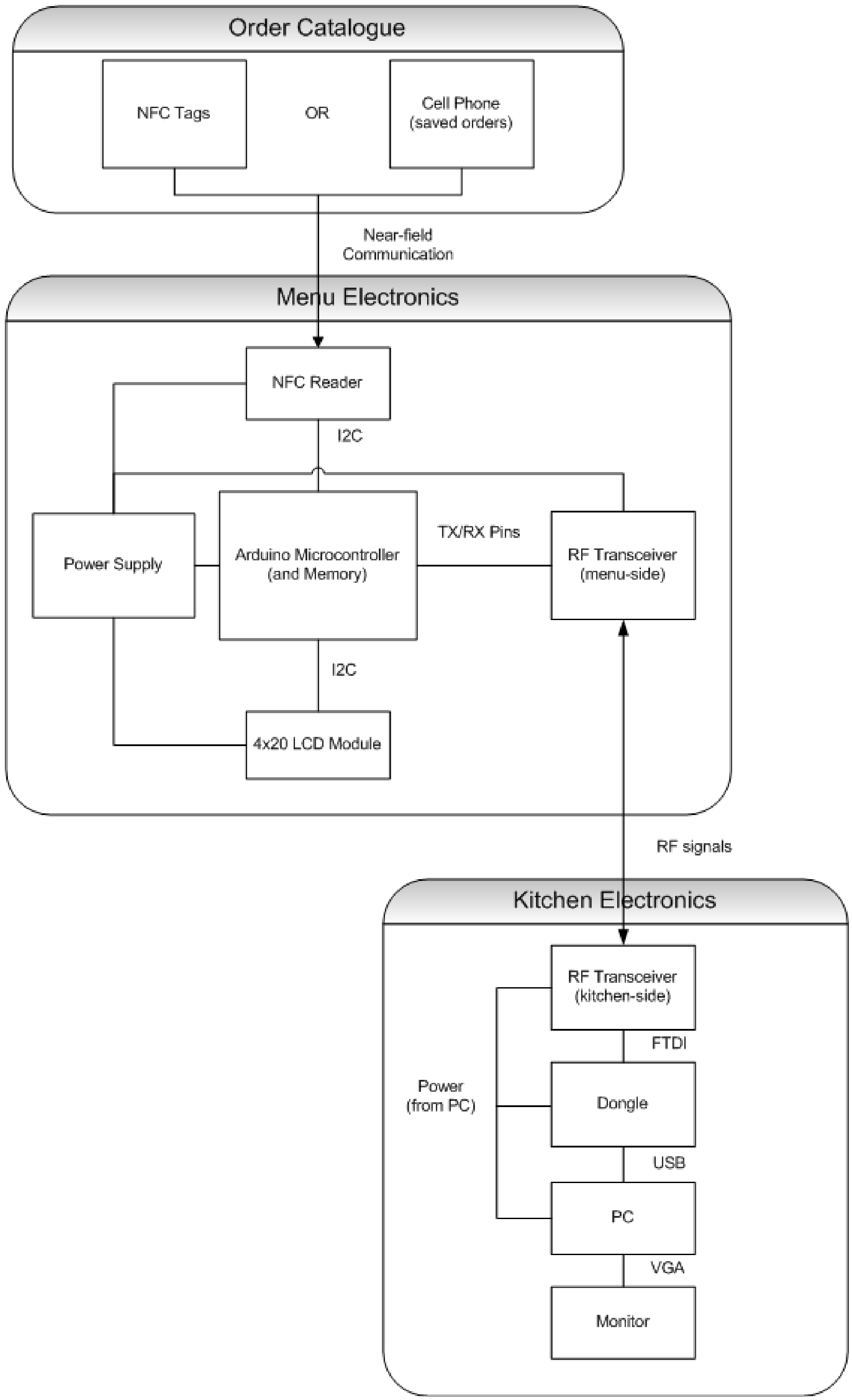


Figure 2: Detailed Block Diagram



2.2 Block Descriptions:

2.2.1 Overall:

To create this NFC ordering system we will use a menu that acts as an interface for the user. The menu has a RF transmitter that will wirelessly transmit data of the customer's order to a receiver which will transmit the data to a computer via USB port. For our prototype, the computer will translate the data into English and display the order.

NFC Tags:

These tags are small and easily programmable via computer software. The tags do not require power and will be stuck inside the menu to represent various items to order. When a customer scans a picture of the item they want to order, they will actually be scanning the NFC tag underneath the picture. When scanned by the NFC reader, the NFC tags will transmit the data that represents an item to the reader using near field communication technology.

Cell Phone:

Users will be able to use their smartphones to input pre-programmed orders into the system. The goal is to allow users to create an order with the menu, then tap the NFC reader onto their NFC-enabled phones to import their order directly.

2.2.2 Menu Module:

Power Supply:

This will supply power to all the components that require it. The only component that does not require the power supply to function is the NFC tags. The power supply will be connected to microcontroller only, and the microcontroller will distribute power to the components that it is connected to. The microcontroller takes 9V, so the power supply will consist of a 9V battery and a battery box IM361 that will connect the 9V battery to the microcontroller.

NFC Reader:

NFC uses magnetic induction between two loop antennas. One antenna belongs to the initiator (active) and the other, the target (passive). The initiator generates a carrier field in which the target responds by modulating the existing field. For our case, our reader acts as the initiator while the NFC tags act as the target. When we transmit our menu items from a smartphone onto the reader, the smartphone will act as the initiator instead while the reader acts as the target. The NFC reader we will be using is NXP's PN532 transceiver module that is soldered onto a NFC RFID module made by elechouse. This module comes with its own library and has I2C set as the default interface which is what we will use. It will be hardwired to the microcontroller. The range of this reader 4-6 cm requires 5V TTL for I2C.

RF Transceiver:
The transceiver will take the data that was picked up by the reader after the order is verified through the microcontroller. Once the order is confirmed, the transceiver will wirelessly transmit the data to another transceiver component which will be in the kitchen of the restaurant. This transceiver will also receive a confirmation from the kitchen transceiver to notify customers that the order has been received. The transceiver that will be used is the XBEE 1mW series 1 transceiver. This chip will draw 3.3 V at 50 mA from the microcontroller and features 250 kbps max data rate in the 2.4 GHz frequency band. The range for this device will be around 100 meters.

4x20 LCD Module:
This will display the order and acts as an interface that allows the customer to view and then confirm or cancel their order. The display will be a backlit LCD module that receives the data to display from the microcontroller. We will be using the HD44780U (LCD-II) as the display. This display requires 2.7V - 5.5 V of power and can display 80 max characters on the screen at once.

Arduino Microcontroller:
The microcontroller will take the data it receives from the NFC reader and will translate the data into the desired item to display on the display module. It will also be able to receive input from the customer in order to confirm or cancel the order. If confirmed, the microcontroller will transmit the data from the NFC reader to the transmitter. Some memory will be required to store up to 32 unique selections made by the user.

2.2.3 Receiver Module:

RF Transceiver:
This will receive the data from the transmitter and will be transferred into a computer via a USB port. The transceiver will be placed in the kitchen to allow communication from the customer directly to the cooks. When this transceiver has successfully received the order from the customers, it will send confirmation to the menu transceiver. The transceiver used is the XBEE 1mW series 1. This is the same transceiver being used as in the menu.

Dongle:
This is the device which interfaces the kitchen transceiver to the USB which will allow data to be transmitted from the kitchen transceiver into the computer. We will be using the XBEE explorer dongle. This will allow the transceiver to attach to the usb port perfectly. The dongle will regulate the 5V supply from the USB to the 3.3V that the XBEE transceiver needs to run. It also eliminates the need for cables and allows for a small compact unit that can be kept plugged into a computer display.

PC:

Power Supply
This will supply power to all the components that require power. Because the kitchen transceiver will be connected to the computer display through a USB port, the power will come from the computer. The USB supplies power at 5V.

USB Controller:
The first stop for the data coming from the receiver is the USB controller. The computer's controller will parse the information and send it to the memory.

Memory:
The data will be stored into computer memory for our software to utilize.

Software:
Depending on when new data is written into memory, the software will translate the incoming information into an ordered list and process the information for monitor output.

Monitor:
The monitor will take what the software outputs and display a GUI for kitchen staff to use (see Figure 10).

2.3 Schematic Diagrams:

Figure 3: Menu Schematic

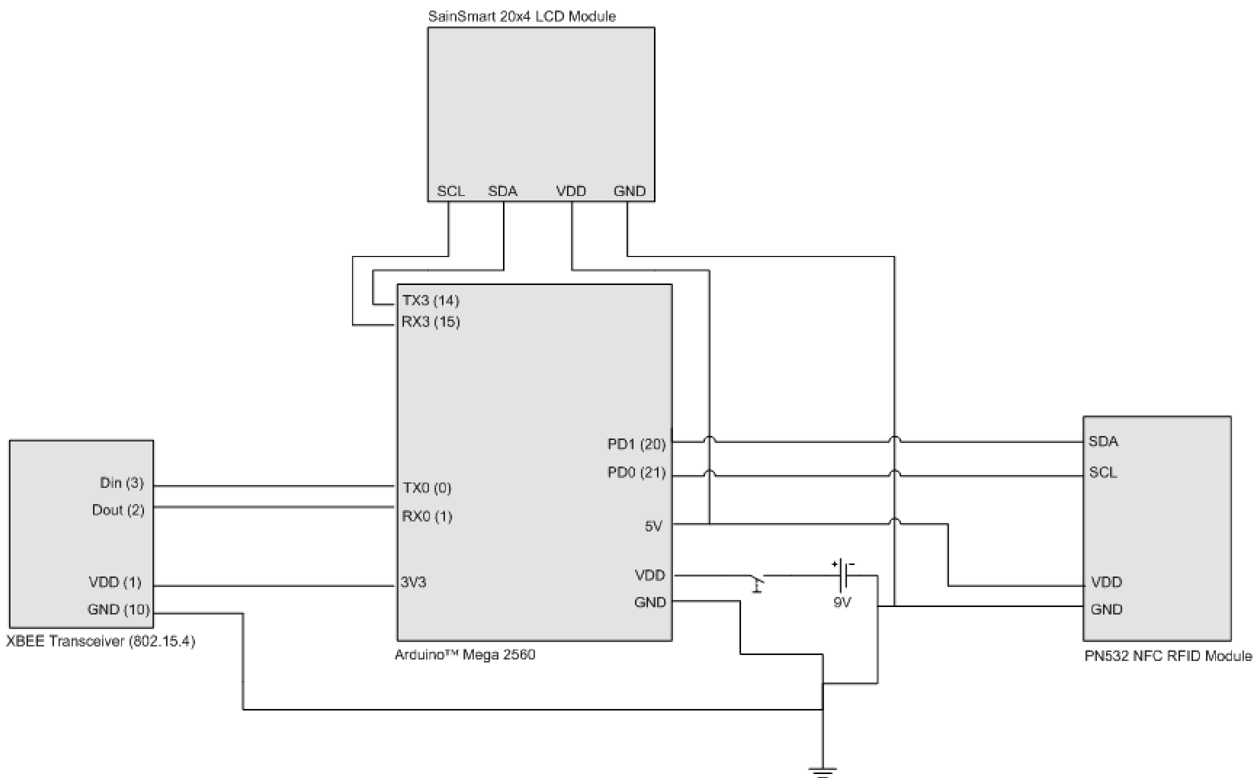
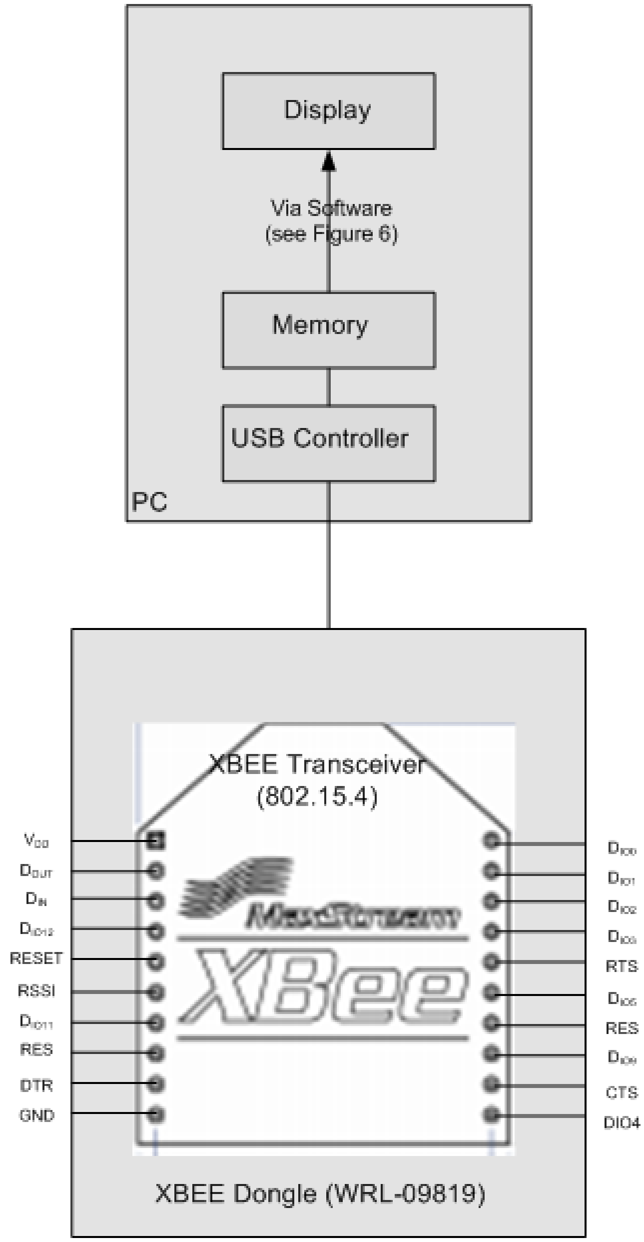


Figure 4: Receiver Schematic



2.4 Process Flow Charts:

Figure 5: Menu Flow Chart

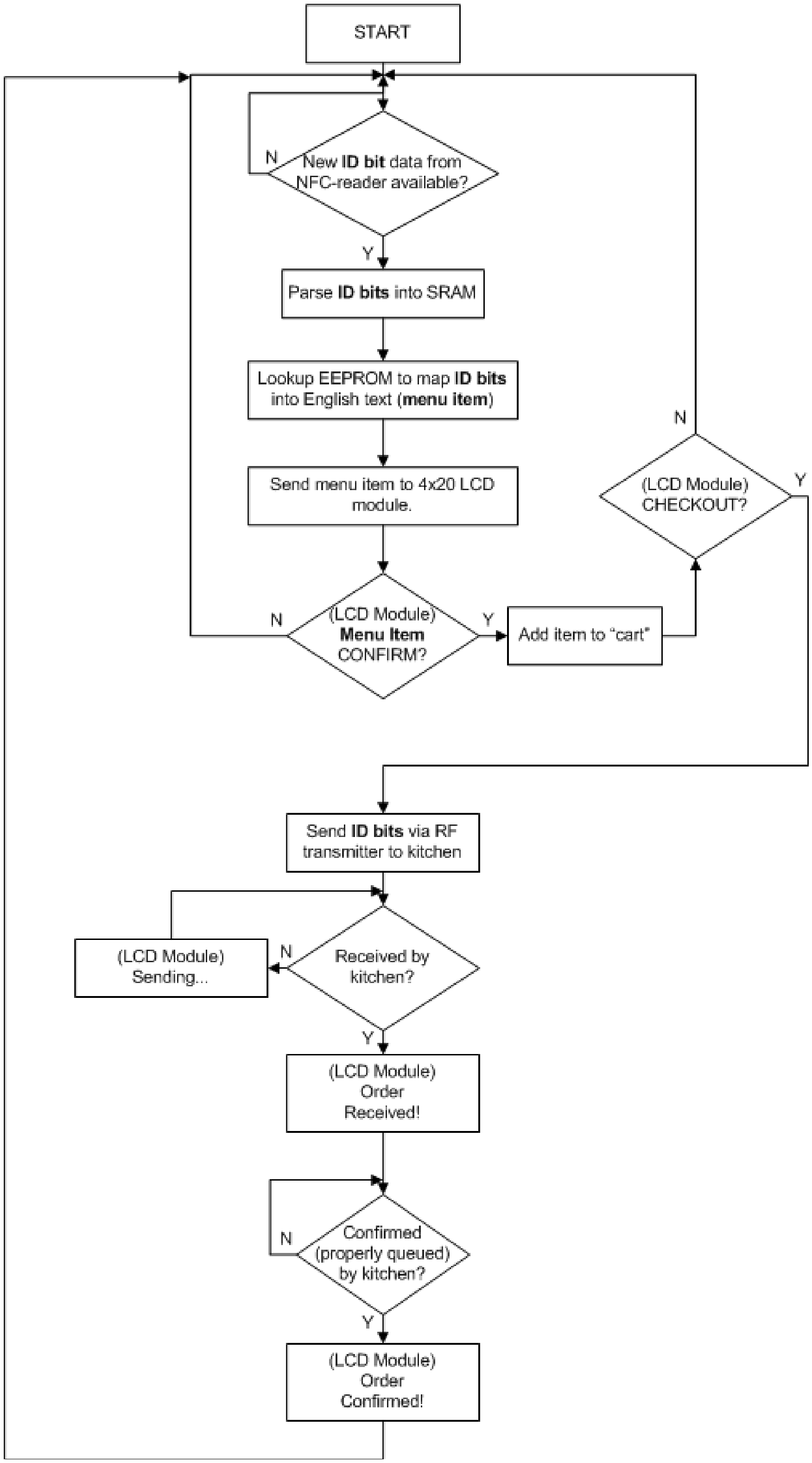
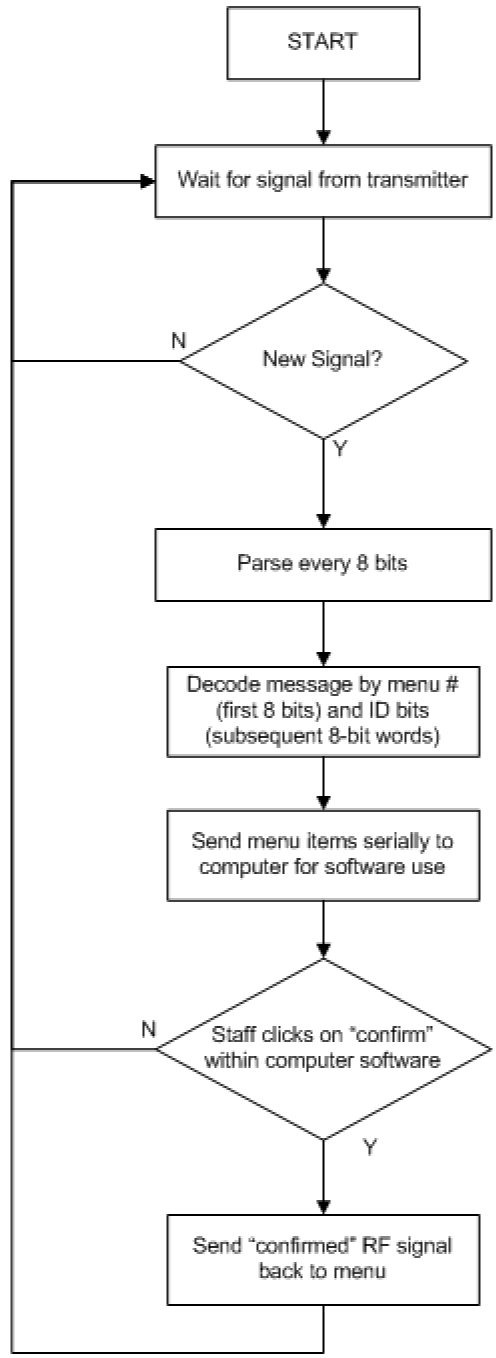


Figure 6: Kitchen Module Flow Chart



2.5 Simulations and Calculations:

2.5.1 Simulation Results / Sample GUI

NFC Reader Preliminary Simulation

The following figures (Figures 7-8) show the expected results from scanning/receiving NFC information. We will be using a smartphone and our NFC chip to test these results and make sure the NFC part of the project works without any hitches. The software is an IDE (Integrated Development Environment) created by Arduino. The pseudocodes for the programs are shown as well:

```
//Example code to write data
Strcpy((char*)block, "Testing - NFC");
Sta = nfc.MifareWriteBlock(blocknum, block);
If(sta)
	Serial.println("Write block successfully:");

//Example code to read data
//read block #4
Sta = nfc.MifareReadBlock(blocknum, block);
If(sta){
	Serial.println("Read block successfully:");
	Nfc.puthex(block,16);
	Serial.println();
}

//read block 5
Sta = nfc.MifareReadBlock(blocknum+1, block);
If(sta){
	Serial.println("Read block successfully:");
	Nfc.puthex(block,16);
	Serial.println();
}

//read block 6
Sta = nfc.MifareReadBlock(blocknum+2, block);
If(sta){
	Serial.println("Read block successfully:");
	Nfc.puthex(block,16);
	Serial.println();
}
```

Figure 7: Read/Write Example from Above Code

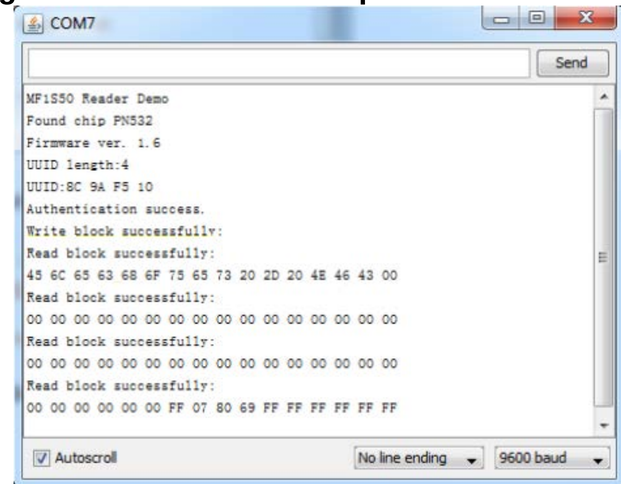


Figure 8.1: Confirmation of Signal Received by Reader

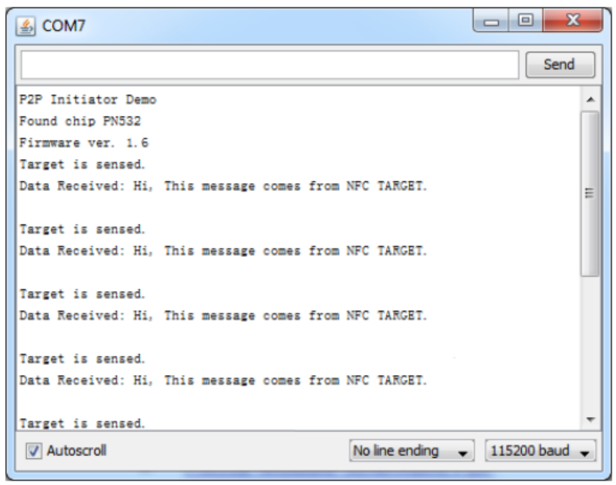
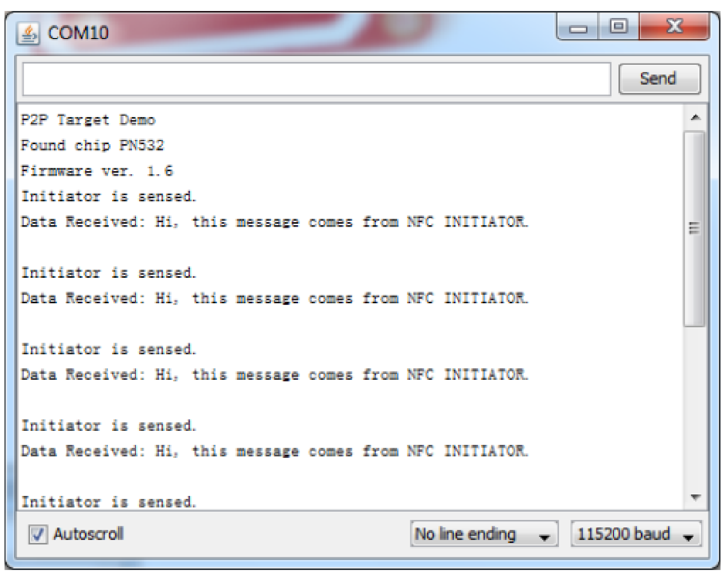
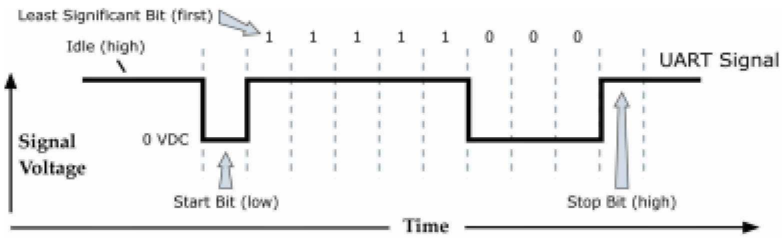


Figure 8.2: Confirmation of Signal Received by Smartphone



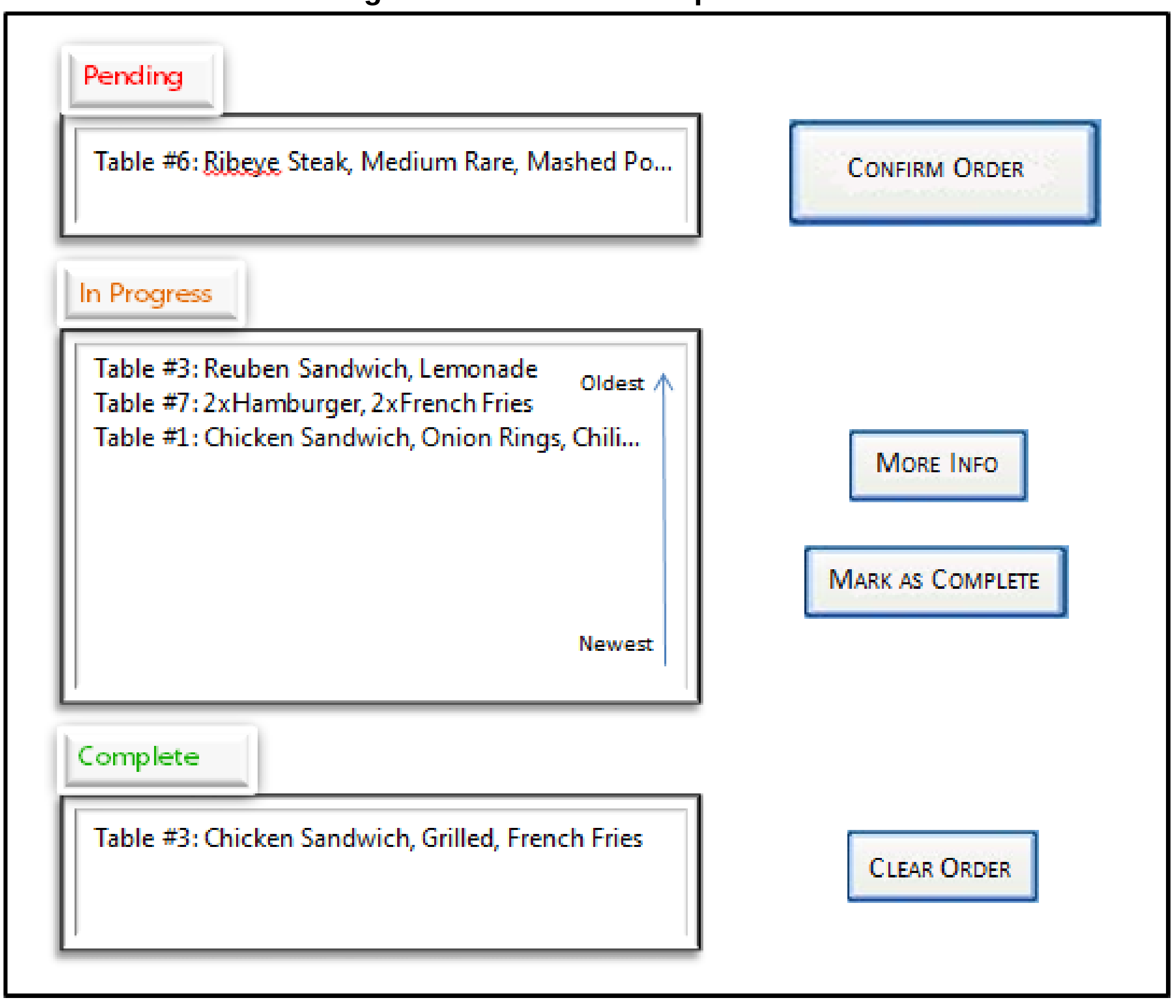
RF Data Transmission
We have researched how data is sent through the transceiver. Messages are sent in the following format, with a low start bit and a high stop bit. Eight bits are sent with each signal.

Figure 9: Example of RF Word Submission



Computer GUI
We have also been considering the user interface for our software, and a rough design is presented below. By clicking the buttons next to the list, orders can be moved down the line from “Pending” to “In Progress” to “Complete”. Each table has the option of placing orders from either multiple menus or a single menu, whichever is most convenient for the user.

Figure 10: Model of Computer GUI



2.5.2 Calculations

Baud Rate Calculations

Arduino quartz oscillator operates at 16 MHz.

NFC Reader operates between 106 - 400 kbps bitrate:

@ **400 kHz**: 16MHz/400kHz = 40
Frequency of NFC Reader stepped down by 40x
Each item = 1 byte. 1 byte / (400E3/8) **= 0.00002 s per scan**

@ **106 kHz**: 16MHz/106kHz = 151
Frequency of NFC Reader stepped down by 151x
1 byte / (106E3/8) **= 0.000075 s per scan** (still very fast!)

RF Transceiver operates between 1.2 - 250 kbps bitrate:

@ **250kHz**: 16MHz/250kHz = 64
Frequency for transceiver will step down by 64x
Full order of 128 items = 128 bytes. 128 bytes / (250E3/8)
= 0.004 seconds per order (fast!)

@ **125kHz**: 16MHz/125kHz = 128
Frequency for transceiver will step down by 128x
128 bytes / (125E3/8) = **0.008 seconds per order**

@ **12.5kHz**: 16MHz/12.5kHz = 1280
Frequency for transceiver will step down by 1280x
128 bytes / (12.5E3/8) = **0.082 seconds per order**

@ **1.2kHz**: 16MHz/1.2kHz = 13,333.33
Frequency for transceiver will step down by 13,333.33x
128 bytes / (1.2E3/8) **= 0.853 seconds per order**

Battery Life Calculations

Since these menus are vital to a restaurant's operation, we decided against rechargeable batteries. We assume that a restaurant would simply have a stock of 9V's on stock that they would replace when necessary. This would be much easier than popping out 9V's to recharge whenever they drained. The following are typical capacities of 9V batteries by material:

Alkaline	~565 mAh
Zinc-Carbon	~400 mAh
Lithium	~1200 mAh

We expect our circuit to draw between 50mA and 200 mA of power. Please see Table 1 on the next page for the sensitivity analysis.

Table 1: Hours of Expected Use

Capacity/Draw	25mA	50mA	75mA	100mA
400 mAh	16.0	8.0	5.3	4.0
565 mAh	22.6	11.3	7.5	5.7
1200 mAh	48.0	24.0	16.0	12.0

We estimate that users will spend about 10 minutes learning the interface and selecting each order. If the battery lasts for ~ 20 hours (using Lithium battery), then this would equate to 200 orders, or about a week's worth of orders.

We will need to do further testing when the system is fully integrated to find total power draw. We will use all methods possible to limit the power draw of the system. At the moment, battery costs seem a little steep for sustained use.

III. Requirements and Verifications:

Requirement	Verification
1. Menu Power Supply (9V Battery): • Must be able to supply a voltage within the range of 8V-10V (safe range) • Must be able to supply a minimum current rating of 25mA to a maximum current rating of 100mA at load ranges between 360 and 90 ohms	1. a. Use a DMM to monitor the voltage b. While testing with loads between 360 and 90 ohms, we will use a DMM to monitor the current
2. Menu NFC Tags (Passive Element): • Must be able to properly store messages • Must be able to transmit messages to the NFC Reader (functionality test)	2. a. We will use the smartphone application called TagWriter (by NXP) and compare the contents read by the smartphone with the content programmed into the NFC tag through Arduino's IDE b. We will write a plain text message to the tag through Arduino's IDE
3. Menu NFC Reader:	3.

• Must be able to properly receive stored messages from the target • Must be supplied 5V (±0.5 V) • Must be able to forward the received messages to the microcontroller at 250 kbit/s with 0.1% bit-error	a. Use TagWriter to receive plain text messages b. Probe the 5V feed from the microcontroller with a DMM c. Use Arduino's IDE in relay messages and spot for 0.1% bit-error
4. Menu RF Transceiver: • Must be supplied with 2.8-3.4V • Must operate at ISM (industrial, scientific, and medical radio band) 2.4 GHz frequency band. • Must work at a range of approximately 100 meters. • Must transmit data to and from kitchen transceiver and microcontroller at 250kbps	4. a. Using a DMM, we will probe the voltage supplied in order to verify that it falls within the desired range at all times b. For a frequency range test, we will use a spectrum analyzer and run a frequency sweep with 1 MHz step size to find the frequency range (This must be between 2.4-2.5 GHz) c. For distance testing, we will manually guess and check starting from 100 meters and observe for accuracy on our computer display (i.e. submit a selection and observe the response from the GUI) d. Transceiver will be wired to an oscilloscope to observe data transfer rate on a time scale. A square wave at 250kHz will be transmitted to match the baud rate
5. Menu Display: • Must be able to properly display messages that were read from the NFC Reader	5. a. Will be given a test message from the microcontroller (typed from Arduino's IDE) and spot for 0.1% bit-error
6. Menu Microcontroller:	6.

• Must operate from a 9V battery power supply (7V-12V safety range). • Must output data to LCD display correctly. • Must receive correct data from NFC reader. • Must transmit correct data to menu transceiver. • Must be able to hold 2560 bits of memory for menu items. • Must be able to transmit and receive data from transceiver at 250kbps. • Must be able to regulate voltage from the power supply and provide other components with power.	a. While the microcontroller is powered, we can use a DMM to monitor the voltage across and the current drawn from the battery b. Test code will be developed (using provided library's from Arduino) to interface between the LCD and microcontroller which will properly display the messages read from the NFC reader (which can be tested with a TagWriter as mentioned in #2 Verif.) c. Using TagWriter to transmit a message to the NFC reader and to the microcontroller, we will use Arduino's IDE to view the data and spot for 0.1% bit-error d. Test code will be written to display (or TagWriter will be used) the data that the microcontroller will send to the transceiver while also spotting for 0.1% bit-error e. Dummy memory will be written to the first 3 MB of EEPROM, and then read to an output file. Input/output test files will be Vimdiff'd to prove that they are identical f. Transceiver will be wired to an oscilloscope to observe data transfer rate on a time scale. A square wave at 250 KHz will be transmitted to match baud rate g. We will use a DMM to probe voltage outputs from the power supply and the feeds
7. Receiver RF Transceiver: • Must be supplied with 2.8-3.4V • Must operate at ISM (industrial, scientific, and medical radio band) 2.4 GHz frequency band. • Must work at a range of approximately 100 meters. • Must transmit data to and from kitchen transceiver	7. a. We will use a DMM to probe the regulated voltage from the dongle b. For a frequency range test, we will use a spectrum analyzer and run a frequency sweep with 1 MHz step size to find the frequency range. This must be between 2.4-2.5 GHz c. For distance testing, we will manually guess and check starting from 100 meters and

and microcontroller at 250kbps	observe for accuracy on our computer display (i.e. submit a selection and observe the response from the GUI) d. Transceiver will be wired to an oscilloscope to observe data transfer rate on a time scale. A square wave at 250kHz will be transmitted to match the baud rate
8. Dongle • Must regulate voltage of 5V from USB to 2.8-3.4V for the transceiver • Must allow for transfer of data from kitchen transceiver to USB	8. a. Probe the VCC pin (pin1) of the dongle with a DMM to verify that the voltage supplied is within the correct range b. Also create test code on computer display to verify the contents from the transceiver are being transmitted to computer with 0.1% bit-error
9. Receiver Computer Display (software) • Must be able to parse the signal and display the appropriate messages from the kitchen transceiver	9. a. Software will test corner cases/simultaneous entry and show the data on the computer screen

3.2 Tolerance analysis:
The most critical part of the project is ensuring that the proper order reaches the kitchen. There are two components that make this possible -- the signals fed from NFC reader to menu microcontroller, and also the RF signal that is sent from the menu to the kitchen. For our tolerance analysis, we will focus on ensuring that the signals sent from NFC reader to menu microcontroller are quick and precise.

NFC technology is relatively quick--we would like the time delay between selecting a menu item and having it appear on the LCD display in ~1.5 seconds. However, the tradeoff is power consumption. We will be balancing speed vs power consumption of the device to make sure that selections feel intuitive, natural, and lag-free to the user, while minimizing the amount of power that the NFC reader draws from the battery module.

To test this feature, we will have the NFC reader draw various amounts of power from the battery pack by varying the bitrate. We will test for an upper-bound and lowerbound of the power drawn and how it correlates to a delay in response time. We hope to find a fair compromise between a high-speed circuit and long battery life.

We will also be doing a similar test with the RF transmitter, limiting the bitrate to maximize power while still providing fast response.

3.3 Contingency Plan:

Battery

If battery voltage does not fall within 8-10V, we will regulate the voltage using resistors in series to deliver proper voltage to our circuit. Likewise, if current rating is different from expected, we will use resistors in parallel to regulate current.

NFC Reader

If NFC Reader is found to consume too much power, we will throttle the baudrate to limit the amount of current draw.

Transceivers

If transceiver is found to consume too much power, we will throttle the baudrate to limit the amount of current draw. If 100m distance is too far for accurate communication, we will measure a shorter distance that results in 0.1% bit error and quote this operating range on our product specs.

Receiver Computer Display (software)

If software GUI depicted in Figure 10 cannot be made as sleekly as imagined, we will modify the GUI so it remains powerful while sacrificing ease of use. Software may use keyboard shortcuts/arrow keys instead of a more intuitive mouse.

3.4 Ethics:

The purpose of this project is to provide a user with a more convenient way to order and enjoy food at a restaurant. In addition, the project should make it easier for restaurant staff to manage their orders during peak demand times. Since this product will be used by both customers and staff, proper safety is a must. If any unsafe bugs are discovered, we must work to develop a solution and recall the faulty products. This follows the first code of the IEEE Code of Ethics:

1. To accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;

The menu will also be marketed to restaurants around the country, so it must be able to deliver on all of the features that we advertise. Data should not and will not be fabricated to deceive potential customers. This correlates to the third code of the IEEE Code of Ethics:

3. To be honest and realistic in stating claims or estimates based on available data;

Our team needed to learn and study how both near-field and radio frequency communication works before attempting to design the system. We also needed to learn how an Arduino behaves in order to succeed in our design. We believe this relates to codes 5 and 6 of the IEE Code of Ethics:

5. To improve the understanding of technology, its appropriate application, and potential consequences;

6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

In this senior design class, we will be expected to interact with our peers and instructors technically and professionally as well (in peer reviews, weekly TA meetings, etc). If needed, we will not hesitate to seek advice from (or give advice to) the people that we find necessary. This correlates to the seventh, eighth, and tenth codes of the IEEE Code of Ethics:

7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

8. To treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;

10. To assist colleagues and co-workers in their professional development and to support them in following this code of ethics;

Finally, we are representing ourselves, our peers, our professors, and our school by working on this project. We will need to be sure to act in a way that honors all of these stakeholders. This follows the ninth code of the IEEE Code of Ethics.

9. To avoid injuring others, their property, reputation, or employment by false or malicious action;

IV. Cost & Schedule:

4.1 Cost Analysis

Labor

Name	Hourly Rate	Total Hours Invested	Total = Hourly Rate x 2.5 x Total Hours Invested
Patrick Ding	$35.00	180	$15,750
Yau Chan	$35.00	180	$15,750
Patric Takagi	$35.00	180	$15,750
Total Labor Costs		**540**	**$47,250**

Parts

Item (P/N)	Unit Cost	Quantity	Total Cost ($)
LED’s	$0.42	15	$6.30
PCB’s	$0.00	1	$0.00
Battery Box (IM361)	$6.50	1	$6.50
RLC Components	variable	variable	$10.00
Backlit LCD Display (LCD-00790)	$15.00	1	$15.00
RF Transceiver (WRL-10534)	$22.95	2	$45.90
NFC Tags (NTAG203)	$1.10	20	$22.00
NFC-Reader	$34.60	1	$34.60
Menu Design Costs (housing, etc.)	$30.00	1	$30.00
Dongle	$24.95	1	$24.95
Total Parts Costs			**$195.25**

Grand Total

Section	Total
Labor	$47,250.00
Parts	$195.25
Total	**$47,445.25**

4.2 Schedule

Week	Responsibility	Member
2/04	Finalize and hand in proposal	Patrick
	Research and design NFC tags; order parts	Yau
	Research and design NFC reader; order parts	Patric
2/11	Design in-menu microcontroller; design PCB and order parts	Patrick
	Research LCD display requirements and support Patrick with microcontroller task; order parts	Yau
	Design RF interface between menu and receiver	Patric
2/18	Prepare for Design Review	Patrick
	Design RF receiver -> computer interface (kitchen)	Yau
	Support Yau on RF receiver task and order parts	Patric
2/25	Design LCD module	Patrick
	Design NFC reader module	Yau
	Design RF interface	Patric
3/4	Assemble Microcontroller to LCD module and power supply	Patrick
	Assemble NFC reader module	Yau
	Assemble both transmitter and receiver modules	Patric
3/11	Program the microcontroller and LCD module	Patrick
	Program the NFC reader interface. Design first draft of PCB	Yau

	Program the USB to computer display module	Patric
3/18 (Spring Break)	Individual Progress Report	Patrick
	Individual Progress Report	Yau
	Individual Progress Report	Patric
3/25	Test and verify Microcontroller and LCD module	Patrick
	Test and verify the NFC reader module and power supply	Yau
	Test and verify the transmitter and receiver module	Patric
4/01	Assemble housing	Patrick
	Create phone connectivity interface	Yau
	Test/Edit PCB design	Patric
4/08	Last chance for final PCB design	Patrick
	Refine prototype	Yau
	Implement extra features	Patric
4/15	Test fully integrated system	Patrick
	Prepare for final demo	Yau
	Last-minute revisions/improvement	Patric
4/22	Prepare for final demo	Patrick

	Prepare final presentation	Yau
	Prepare final report	Patric
4/29	Turn in lab notebooks	Patrick
	Complete final written report	Yau
	Checkout and part return	Patric

V. References:

Ethics:

United States. IEEE. 7.8. IEEE Code of Ethics. Washington DC: , 2013. Web.

LCD Module:

Elger. "I2C LCD Display." Projects:lcd_module. N.p., n.d. Web. 23 Feb. 2013. <http://elger.org/wiki/projects/lcd_module>.

"SainSmart IIC/I2C/TWI Serial 2004 20x4 LCD Module Shield For Arduino UNO MEGA R3." SainSmart. N.p., n.d. Web. 21 Feb. 2013. <http://www.sainsmart.com/sainsmart-iic-i2c-twi-serial-2004-20x4-lcd-module-shield-for-arduino-uno-mega-r3.html>.

Microcontroller:

"Arduino - ArduinoBoardMega2560." Arduino - ArduinoBoardMega2560. N.p., n.d. Web. 01 Oct. 2012. <http://arduino.cc/en/Main/ArduinoBoardMega2560>.

Fried, Limor. "Arduino Tutorial - Learn Electronics and Microcontrollers Using Arduino!"Ladyadanet Blog. N.p., 27 Apr. 2012. Web. 19 Feb. 2013. <http://www.ladyada.net/learn/arduino/index.html>.

NFC:

"PN532 NFC RFID Module Kits -- Arduino Compatible." Elechouse.com. N.p., n.d. Web. 19

Feb. 2013. <http://www.elechouse.com/elechouse/index.php?main_page=product_info&cPath 90_93&products_id=2205>.

Jahn, Rene. "NFC/RFID for Beagleboard Xm with Java." Blog SIB Visions RSS. N.p., 26 Nov. 2012. Web. 19 Feb. 2013. <http://blog.sibvisions.com/2012/11/26/nfcrfid-for beagleboard-xm-with-java/>.

Power:

"Nine-volt battery." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 27 Feb. 2013. Web. 28 Feb. 2013. <http://en.wikipedia.org/w/index.php?title=Nine-volt_battery&oldid=540922147>

Transceiver:

XBee. "WRL-08664." XBee 1mW Chip Antenna. N.p., n.d. Web. 18 Feb. 2013. <https://www.sparkfun.com/products/8664>.

XBee. "WRL-09819." XBee Explorer Dongle. N.p., n.d. Web. 18 Feb. 2013. <https://www.sparkfun.com/products/9819>.