

GLBowe & Co

Proposal for Doofenshmirtz Restaurant Software

Derek Royse

Brian Hatcher

Gary Danovich

Cody Mathena

Ioan Istrate

Andy Pritt

Bren Pittsenbarger

Jay Bingley

Jonathan Kade

Jacob Antoine

Wajih Hassan

Table of Contents

- [Requirements](#) 3
- [Java/Swing](#) 6
 - [Manager Program](#).....8
 - [Kitchen Program](#).....10
 - [Hostess Program](#).....11
 - [Customer/POS Program](#).....12
- [Android](#) 26
- [Networking](#)
 - [Android Devices](#) 33
 - [Terminals](#) 34
 - [Server](#) 34
 - [Server Interface](#) 36

Requirements

Project Objective

The objective is to develop a piece of Software for the Doofenshmirtz Eatery Inc. Restaurant. This application is capable of being the key efficiency solution for your restaurant. With more organization comes more efficiency and the ability to confidently serve more customers. This includes customized organization for reservations and seating at all tables and the floor layout integration and special event capabilities. Communication between the front and back of the house will be made seamless. Inventory tracking for comparison and projections. Sales reports instantly generated. Tracking of employee schedules, overtime and total sales. This software will allow you to monitor every aspect of your business, detecting and eliminating weak spots in your systems.

Deliverables

Deliverables will be supplied at the end of the project and will consist into a set of files that will contain the **Software** files and documentation.

Training

Training will be provided in house by our engineers. Documentation will be made available in digital and print format.

Development Process

Once the proposal is chosen, we will begin with building a prototype for the **Software**, which will be presented to our first post interview meeting. These initial requirements for the prototype will be based on the notes we took from our interview. Once the prototype is released, tested and approved we will begin work on the actual project, where we will be adding the new requirements (if any) on top of the prototype ones.

When the project requirements are finished we will be presenting them to you in a format of your choosing so that we can make sure everything is accounted for.

As soon as the requirements process is done and over we will begin working on the actual **Software**. We are going to **keep you constantly informed** on the development process and we will be presenting you with the project progress on a weekly basis, this way it will be easy for you to see where and how the project is going, any additional features can be planned accordingly, and anything in the project that is not to your liking, can be changed.

Timeframe

We estimate that the time-frame, in its initial agreed form, for the **Software** to be the 4-6 weeks. The pre-final release will be made available on the 24th of April as discussed.

Testing

Throughout the development process we will be user testing the app within our own company to make sure the **Software** is easy to use and bug free. You and your colleagues will be able to test the app during our weekly progress reports.

App Features

- In Software Reservations
- Menu System
- User Friendly / Colorblind users
- POS (Management/Host/Cook/Waiters)
- Seating Chart (Open Seats)
- Status System (Order taken; customer served; etc.)
- Can place Orders on low inventory items.
- Performance stats for employees
- 'Specials' System
- *Unique Software Solution for a Unique Business*

As you can see, the Software will be extremely helpful for your business and we are delighted that you are considering our services. We are looking forward to getting your input on the **Software** proposal.

Individual Segments

Java/Swing Programs

The Java/Swing team will develop two programs. One will be a program for the restaurant staff, and the other will be a program for customers to make food orders and ask for service from restaurant staff.

Program One

The first program is intended to be used by restaurant staff on a terminal that can be accessed in common areas of the restaurant. It will consist of a main menu with a button for each of the following: Manager, Hosting Staff, Waiting Staff, and Kitchen Staff.

The Manager menu will require a user-name and password that the user will be able to change once they are logged in. They will be able to add new employees within the different sections of the restaurant, as well as remove them.

When the manager adds an employee to the hosting staff, the person they added can log into the terminal by clicking the "Host" button on the main menu and then entering the name that the manager put in for them. When a host/hostess is logged in, they will have a simple, rough diagram of the restaurants floor plan. The diagram will have buttons for each of the tables that will allow the user to assign different statuses for each table, such as "vacant," "occupied," "needs cleaned," etc. They can also assign the different current wait staff on duty to the different tables, which can be seen by the wait staff via the android application.

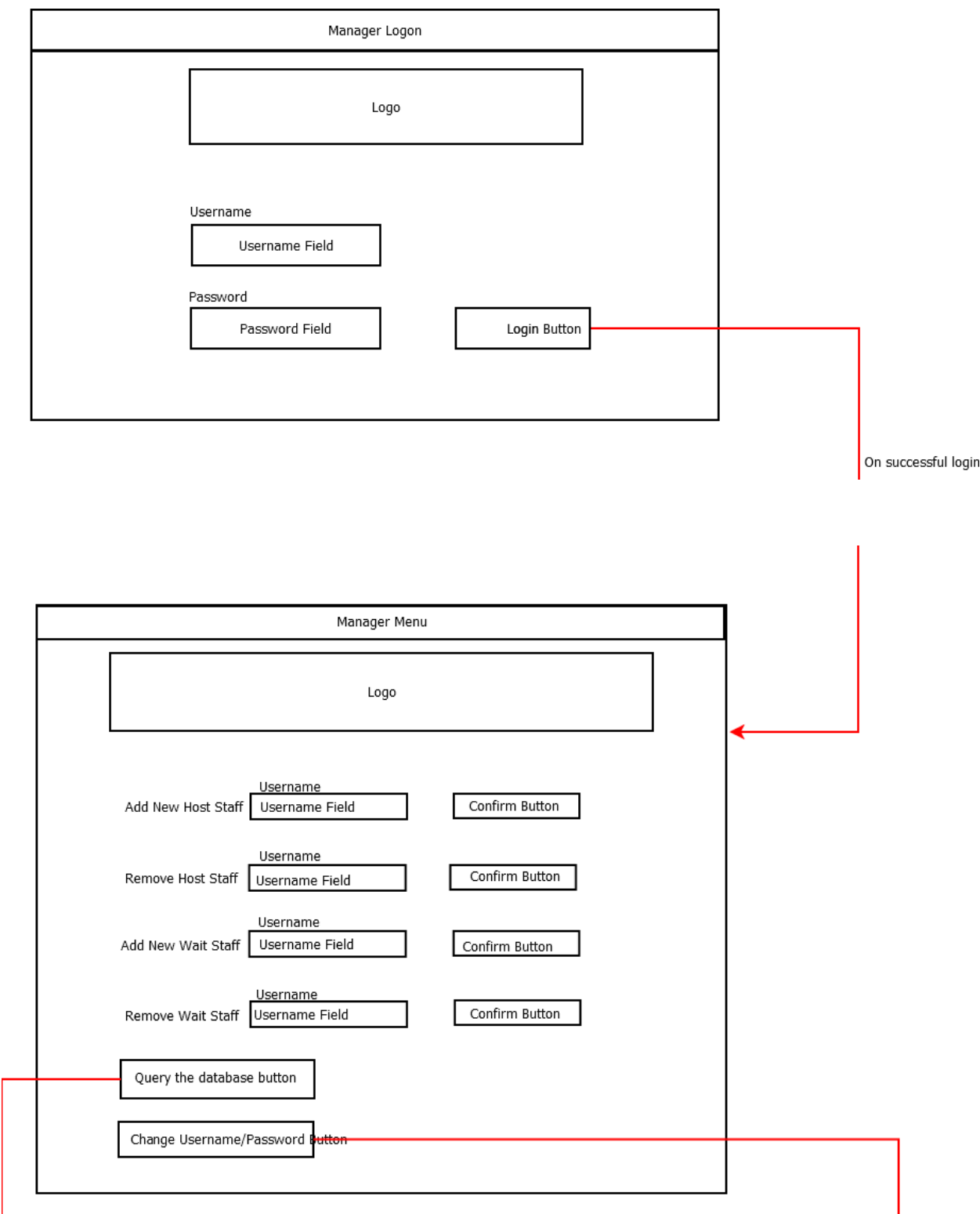
The waiting staff can log into the terminal in the same way that the hosting staff can. Once logged in, members of the wait staff can use the terminal as a “Point of Sale” system. The customer’s total can be printed to a ticket in this menu.

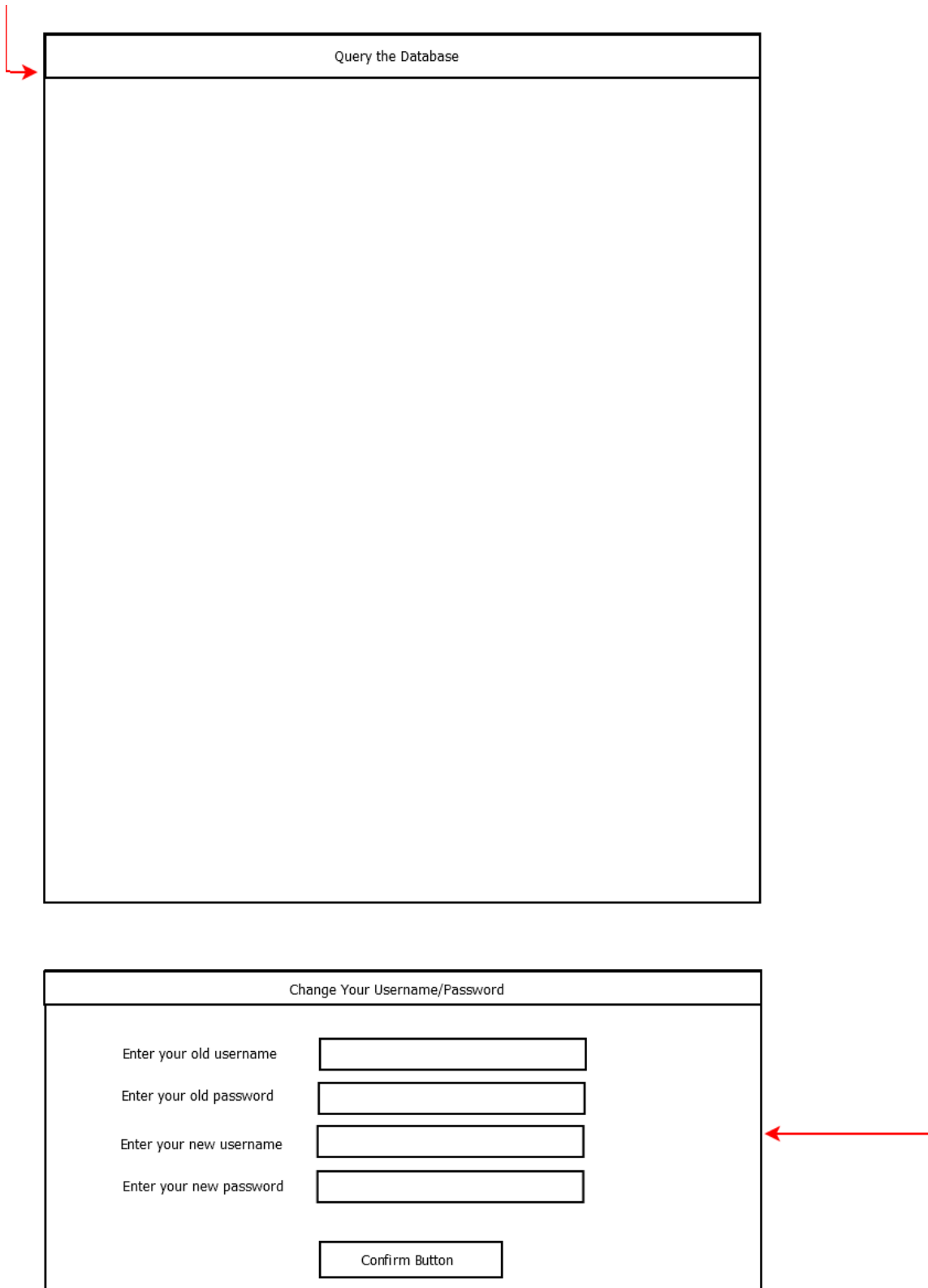
The kitchen staff can log into the terminal in the same way that hosting and waiting staff can. Once logged in, members of the kitchen staff can see what orders customers have placed and what table they are sitting at. They also have the ability to change the status of the order, per table, from waiting to ready. This will then send an alert to the waiter/waitress assigned to that table, so they know to come pick it up.

Program Two

The second program is intended to be used by customers on a terminal that is at each table in the restaurant. The customers can put their food orders in and customize it to their liking, which will be seen directly by the kitchen staff. They will also have the ability to call for the member of the waiting staff assigned to their table, or a manager.

Manager Program





Red Line = New Window is Opened

Kitchen Program

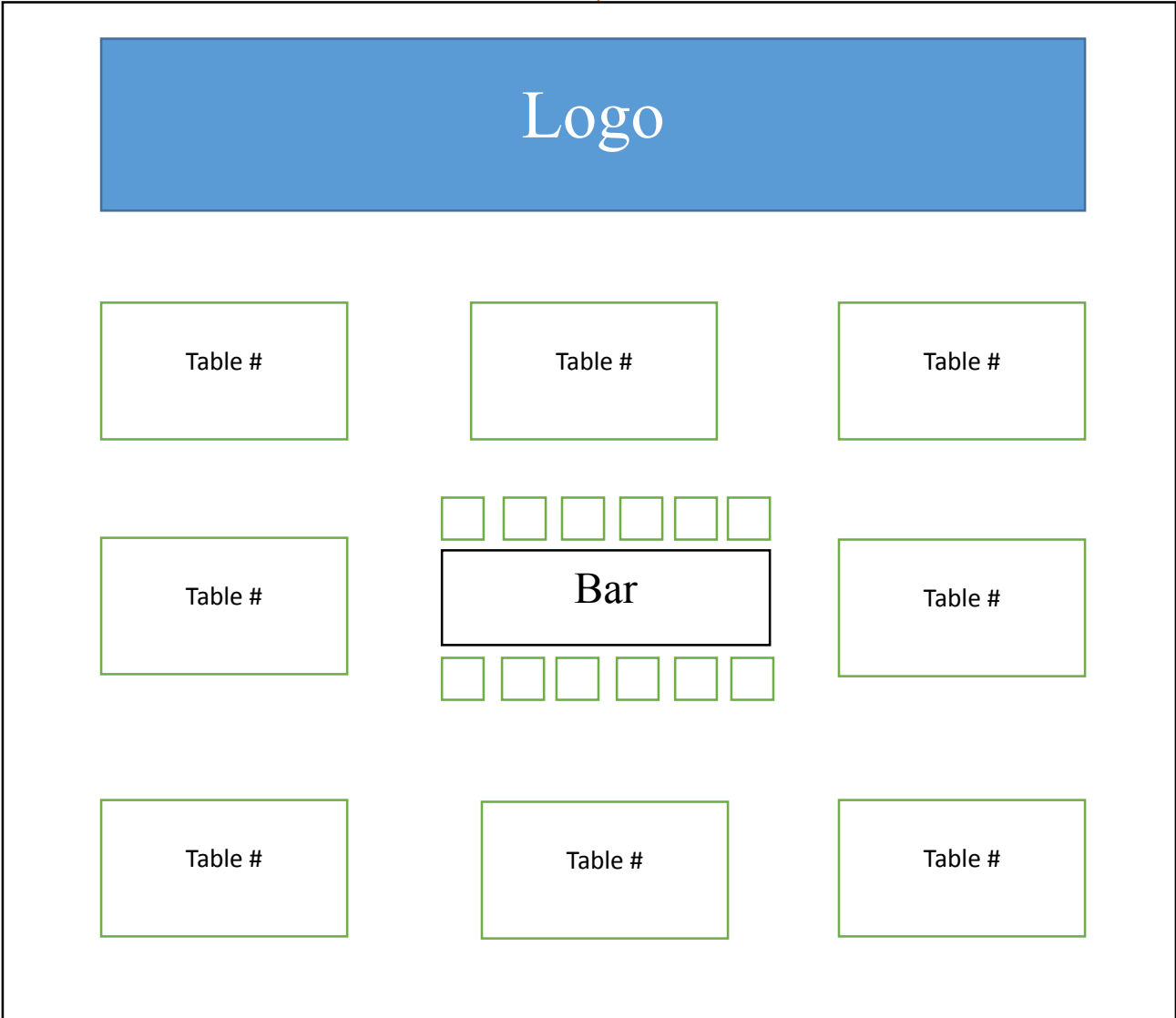
Logo

+	Table #	Time in:	Complete
+	Table #	Time in:	Complete
-	Table #	Time in:	
	Order: Stuff with side of stuff. Hold the stuff		
	Stuff well done		Complete
	Stuff stuff stuff		
+	Table #	Time in:	Complete
+	Table #	Time in:	Complete

Kitchen interface: Our proposed kitchen interface will not require any login information, as it will not have any sensitive material being displayed. The interface consists of a simple scrolling window, with your company logo being displayed on the top. This window will contain a list of all current orders, which can be expanded or collapsed. When collapsed, the row only displays the table number, when the order came in, and allows you to enter the order in as being complete. When expanded, it gives a complete list of all items ordered for that table.

Communication: The kitchen terminal needs to communicate with the wait staff and the database. Once it receives an incoming order, it adds it to the bottom of the list to be completed. Once completed, the system will send an alert out to the wait staff notifying them, and the order is removed from the list. The system will also send pertinent information to the database, such as order times.

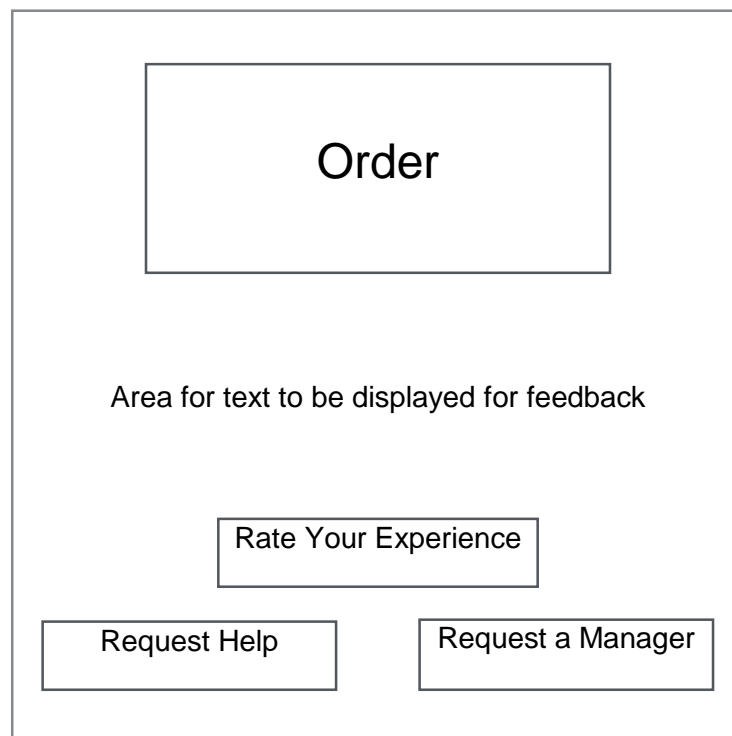
Hostess Program



Host interface: The host terminal opens up to a login screen. After the employee has entered in their login information, they are brought to a window that displays available seating. Once the host has found an available table and clicks on it, the table will then change colors; indicating that it is now in use. Tables will remain colored until they become available again.

Host communication: The host needs to be able to communicate with the database and with the wait staff. To start with, it must be able to compare entered login information with what is stored in the database. From there, after the host has seated the guests, it must send an alert to the wait staff notifying them. The table is then marked as being in use. After the guests have paid their bill, a notification is sent back to the host terminal. This makes the table available for use once again.

Customer/POS Program

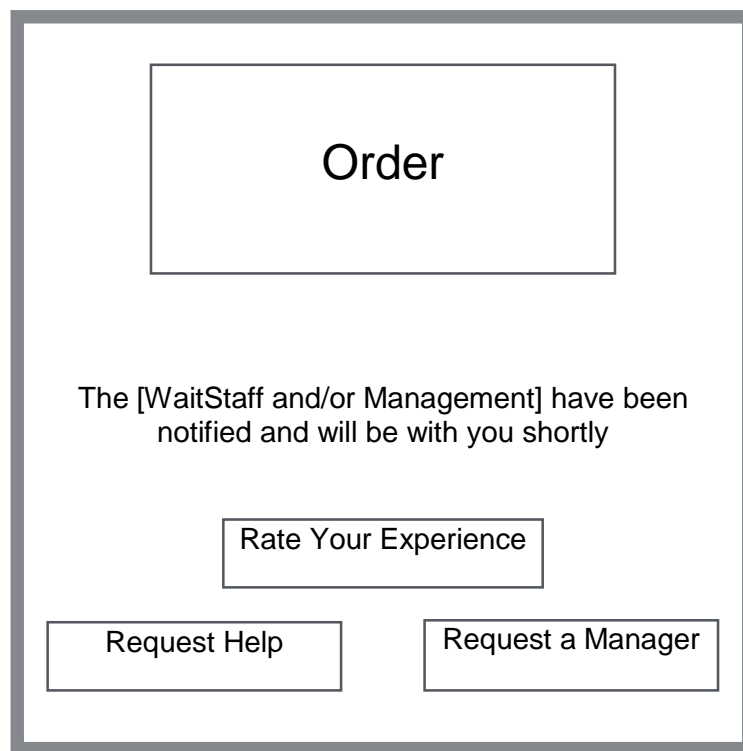


Customer Screens

This is the main customer screen. From here the customer will be able to place an order, summon help, or request a manager. Pressing order will display a hierarchal menu

(to be detailed in another section) that will allow the customer to place their order without the wait staff being present. After order submission the feedback area will display a confirmation message.

Pressing “Request Help will send an alert to the wait staff, notifying them that the table that pressed the button is in need of assistance. This shows what the screen will look like after the user has pressed the button for request help.



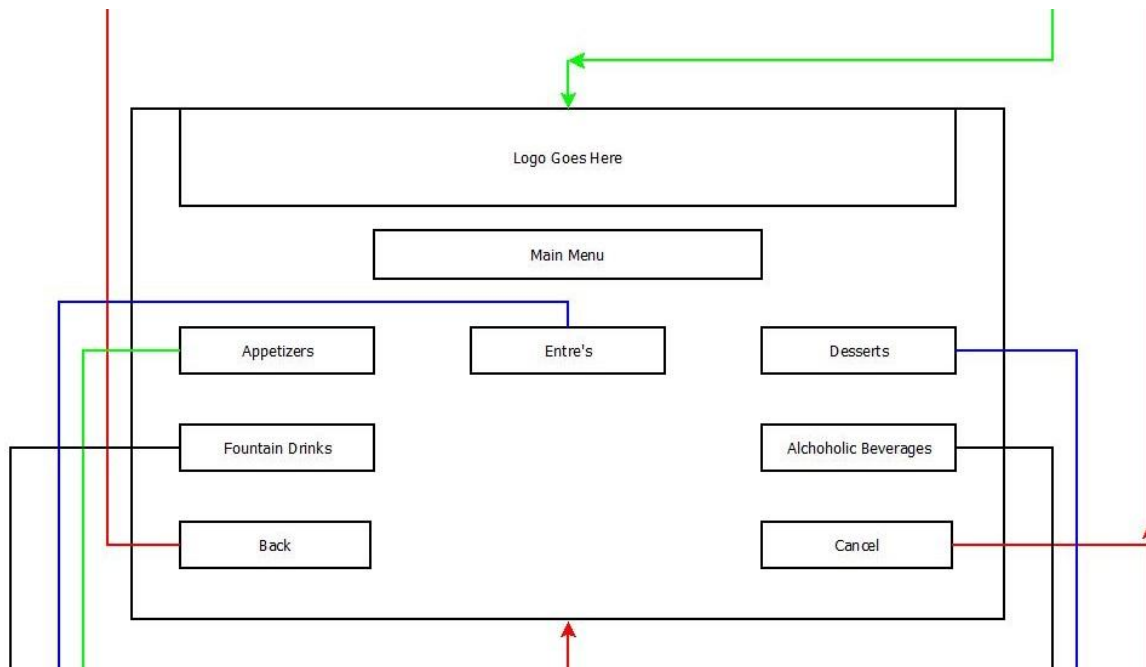
The diagram illustrates the layout of the 'Post Request Help/Management Screen'. It is enclosed in a thick grey border. At the top center is a rectangular box containing the word 'Order'. Below this box is a line of text: 'The [WaitStaff and/or Management] have been notified and will be with you shortly'. Underneath the text is a rectangular button labeled 'Rate Your Experience'. At the bottom of the screen are two rectangular buttons side-by-side: 'Request Help' on the left and 'Request a Manager' on the right.

Post Request Help/Management Screen

The text on the screen will change so that user will know that their request has been received. The process will be the same if the user presses the button to request a manager. Of course “Wait Staff” would be replaced with “Management and vice versa to reflect the different buttons.

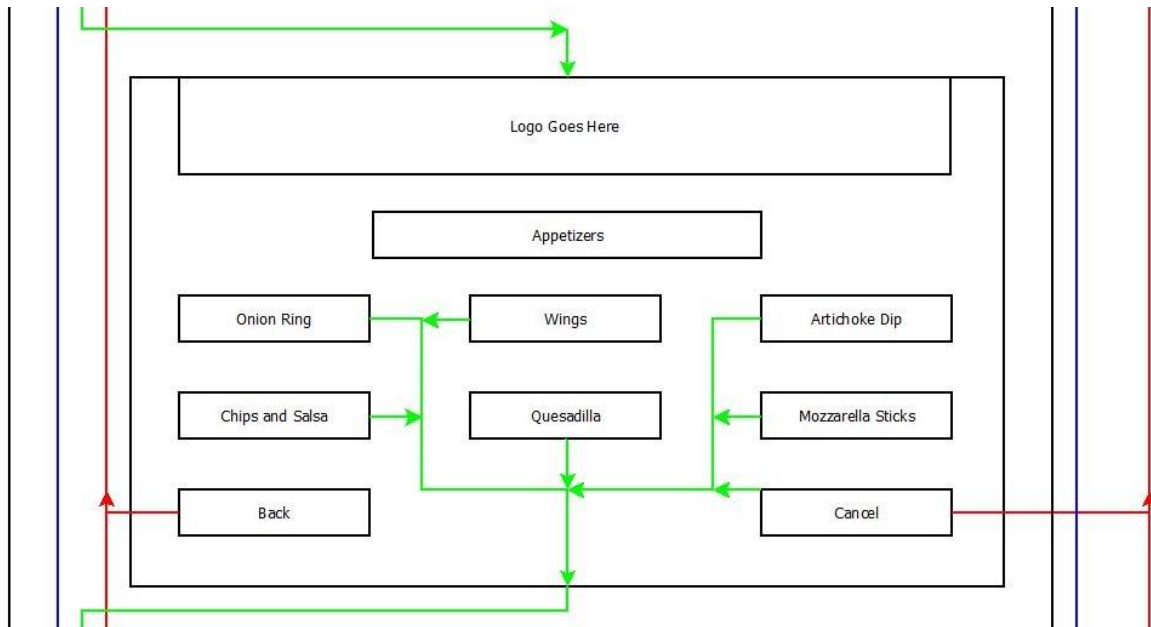
The “Rate Your Experience” Button will allow the customer to leave their feedback on the service right from the table before they leave. When the user has submitted their review, a confirmation message will appear telling them that their input has been received and thank them. The following diagram shows what the Rating screen will look like.

All the Customer will have to do is press their finger to the screen over top of the icon they are wanting to select. Once the button is pressed it will take the customer to a second page.

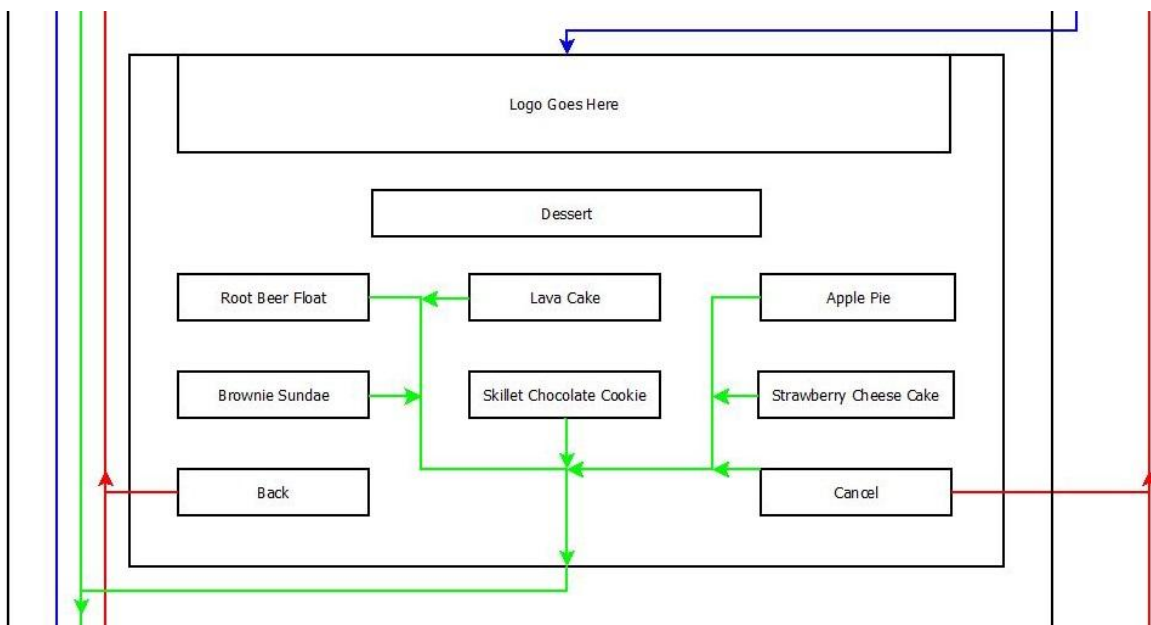


This is the Main Menu page. This is the second page the customer sees upon using the kysok, and is pulled up when the user presses the begin order button. Main Menu is not a button, it is just a label so the customer knows what page they are on. All page Labels are directly below the Logo and on a line by itself. Every page past this point contains a Back and a Cancel Button. The back button will take them back to the previous window and the cancel button will cancel the entire order. Each other button,

when clicked, will take the customer to a new page with more options to choose from. Ex. the Appetizers button will take the customer to a new page with the list of all appetizers sold in the restaurant.

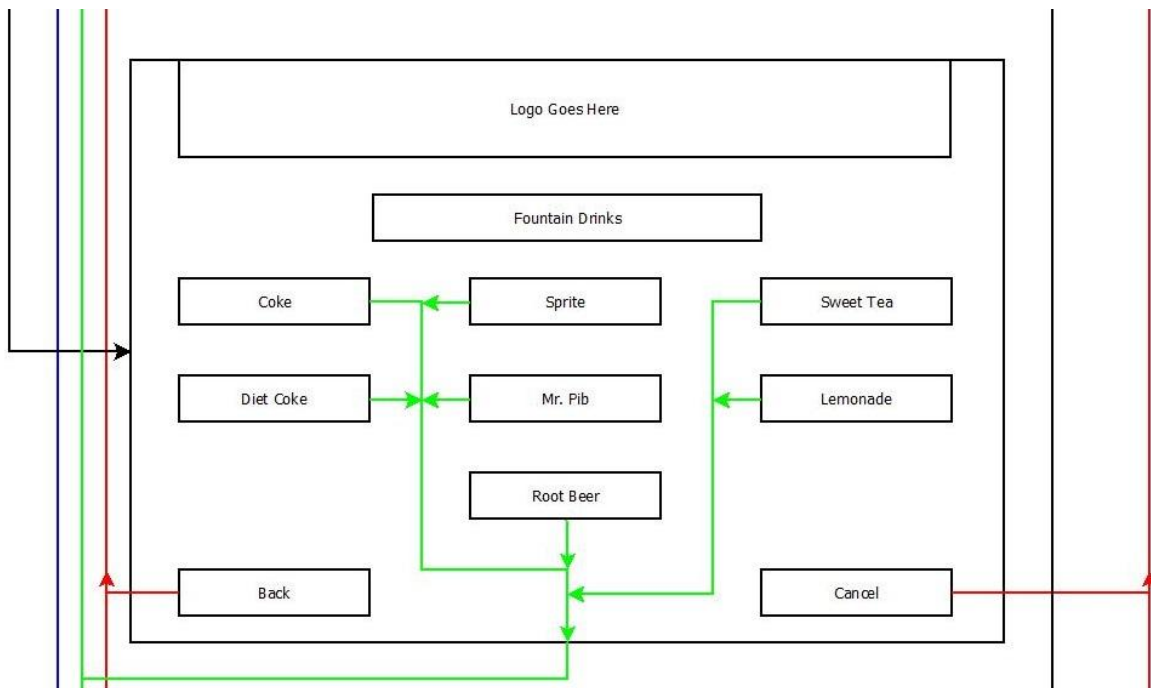


The Appetizers page contains a selection of every appetizer the restaurant sells. Any Appetizer selected will take the customer to the final page where they have the choice to finish, cancel, or add more items to the order.

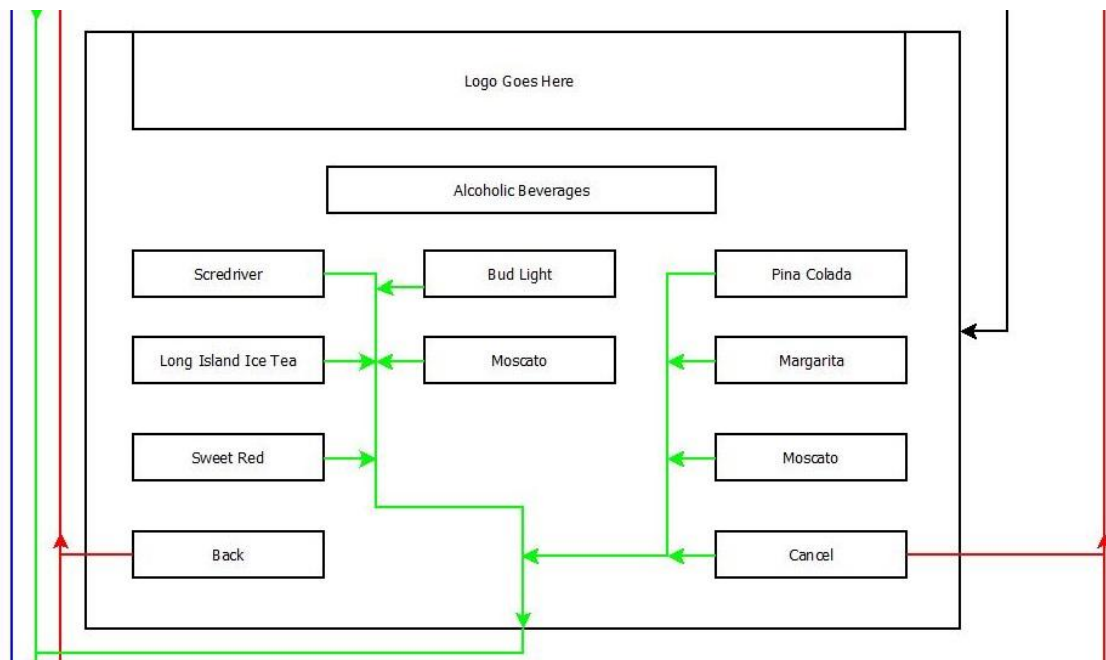


The Desserts page contains a selection of every Dessert the restaurant sells. Any Dessert selected will take the customer to the final page where they have the choice to finish, cancel, or add

more items to the order.

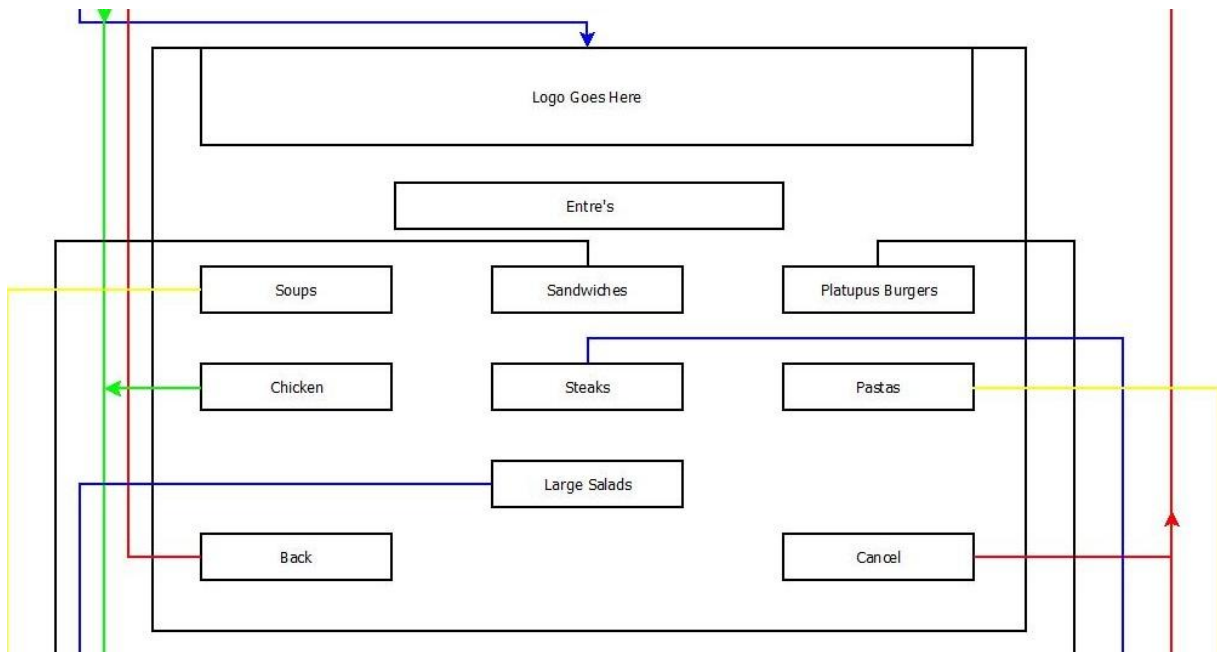


The Fountain Drinks page contains a selection of every fountain drink the restaurant sells. Any Fountain Drink selected will take the customer to the final page where they have the choice to finish, cancel, or add more items to the order.

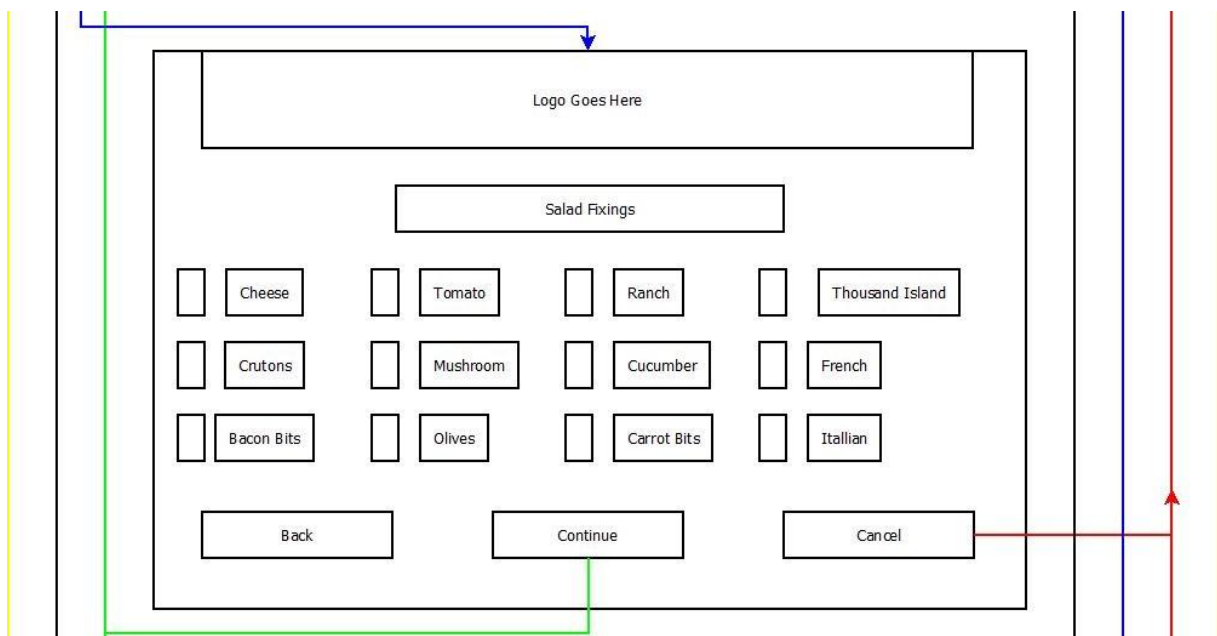


The Alcoholic Beverages page contains a selection of every alcoholic beverage the restaurant sells. Any Alcoholic Beverage selected will take the customer to the final page where they have the

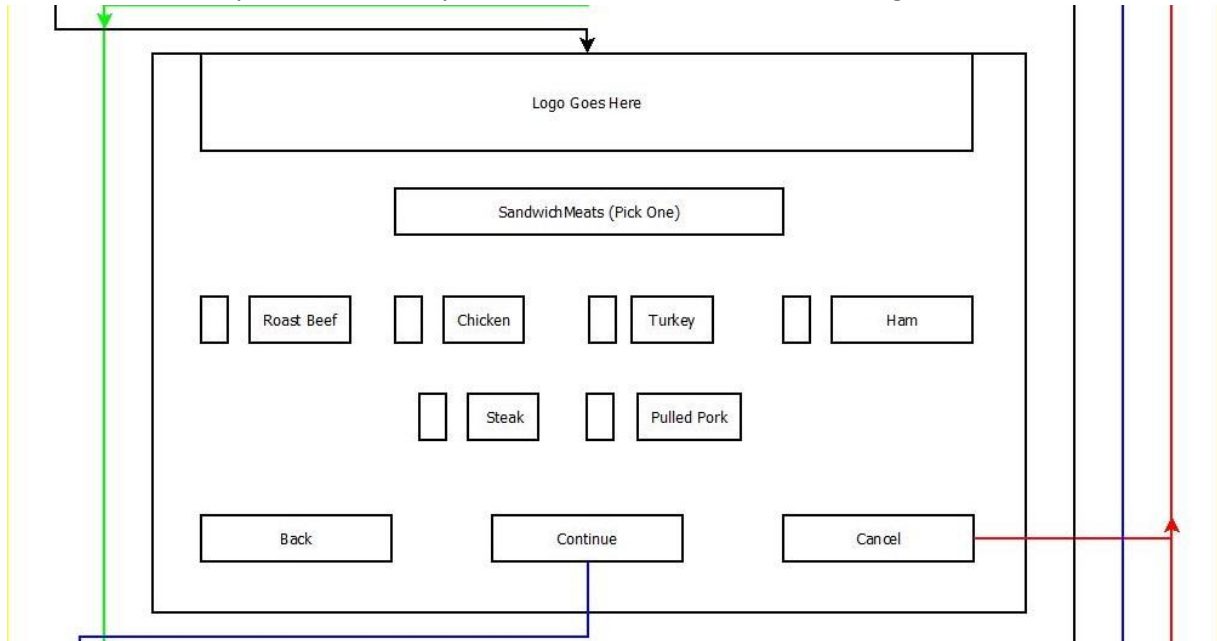
choice to finish, cancel, or add more items to the order.



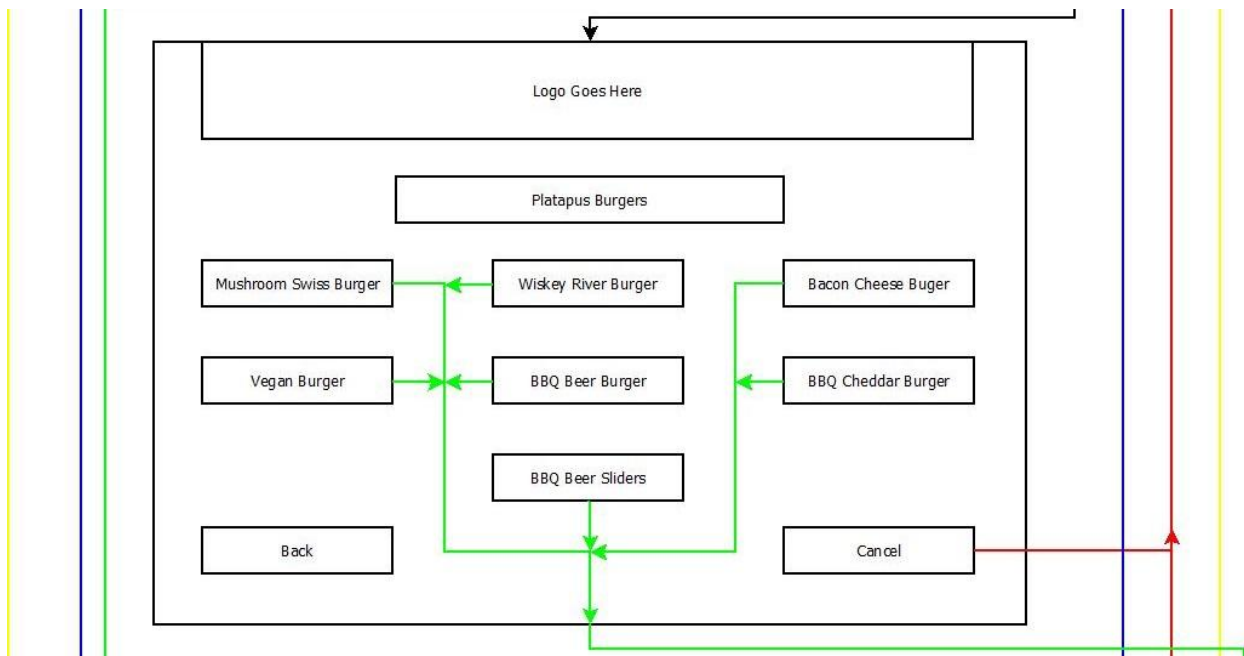
The Entre's pages contains a wide variety of dinners to choose from. Each button takes the customer to a different screen. The Large Salads button will bring them to a page that contains everything they could ever want on a salad. The Pastas button will take them to a page that contains a list of all the pasta dishes our restaurant sells. The Soups button will take the customer to a page containing all the soups the restaurant sells. The Platypus Burgers button will take them to a page that contains all our signature burgers. The Chicken button orders the customer chicken and takes them to the Sides page. The Sandwich button takes the customer to a page where they can pick a meat for their sandwich. Lastly, the Steaks button takes the customer to a page with a variety of different size steaks and how well they are to be cooked.



This page contains a list of different ingredients that the customer might want in their salad. Each ingredient has a checkbox to its left and the customer can select all, none, or any number in between. Once they are finished they can click the continue button to bring them to the final

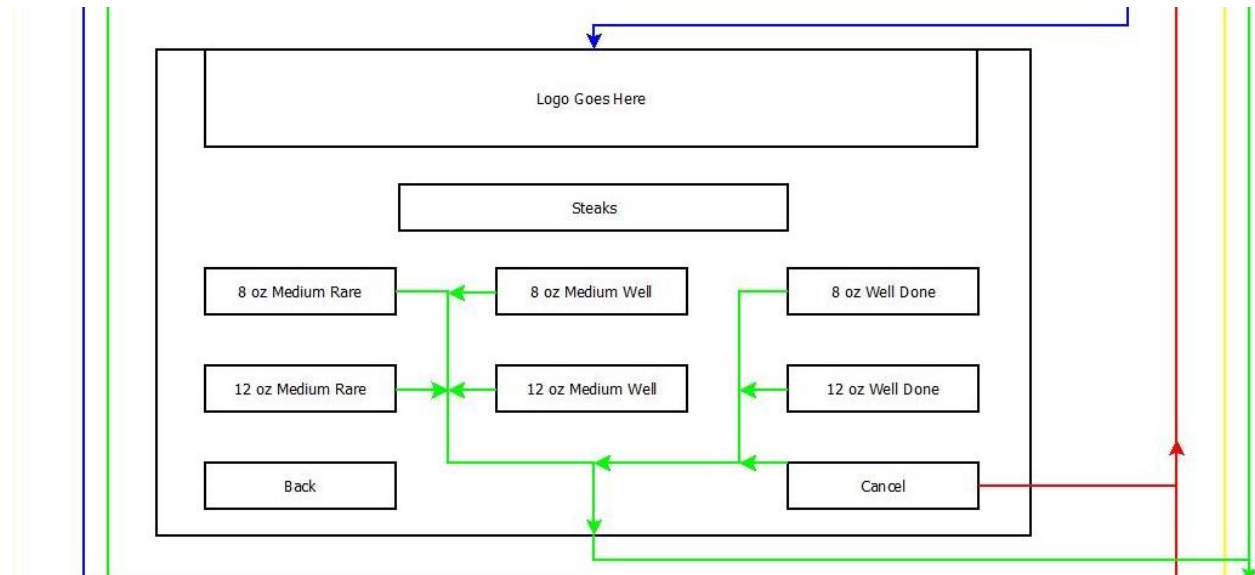


The Customer arrives to this page when they click on a sandwich. Sandwich Meats contains 6 types of meat that are on a radial button. The customer is only allowed 1 meat on their sandwich, if they click on a second one the last one will become unchecked and the new one they selected will be checked. Once they have selected one meat they can click continue to go to the Sandwich Innards page where they can pick from a list of ingredients that can go on a sandwich.

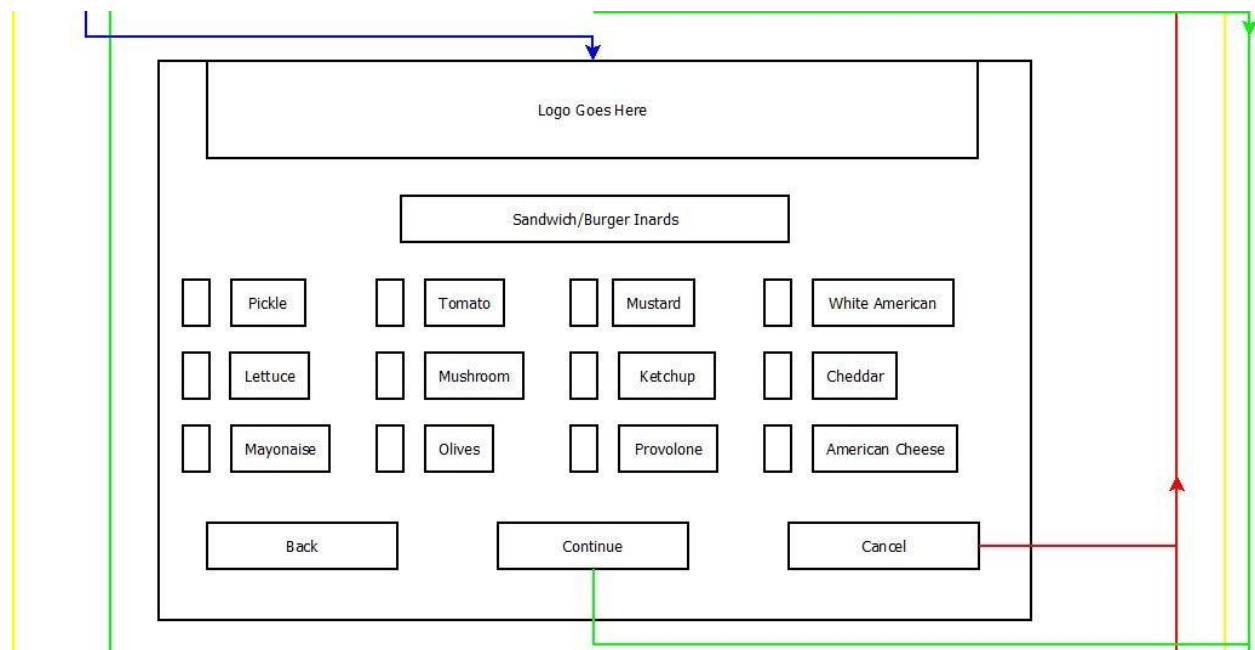


This page contains a list of all the gourmet burgers in the store. After selecting one, the customer

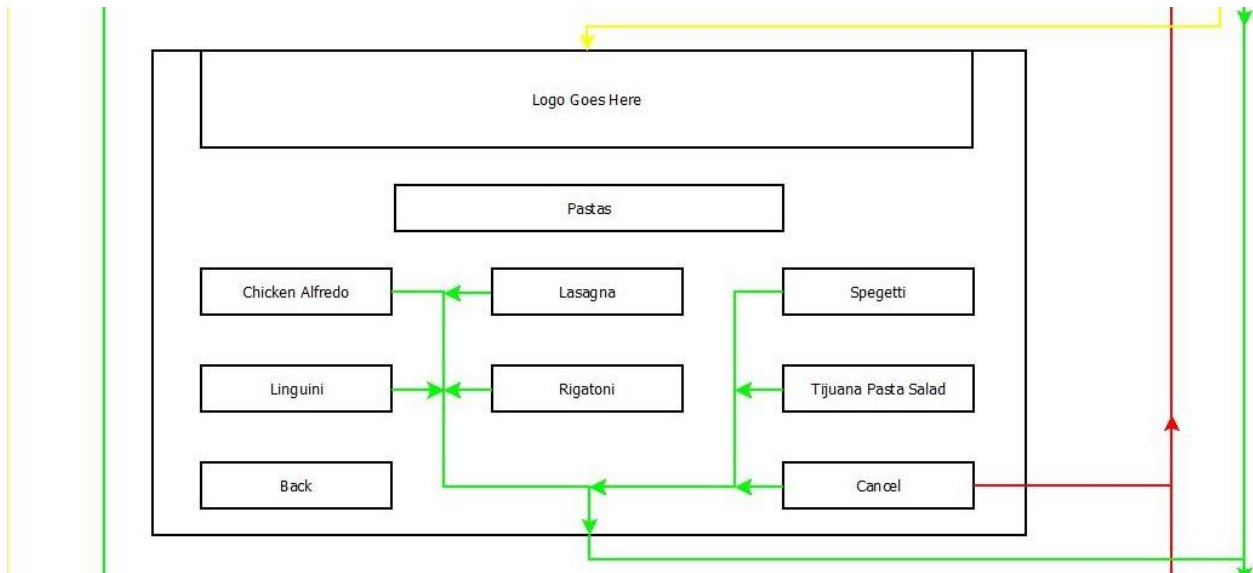
is brought to the Burger Innards page where they can choose from a list of ingredients that could go on the hamburger.



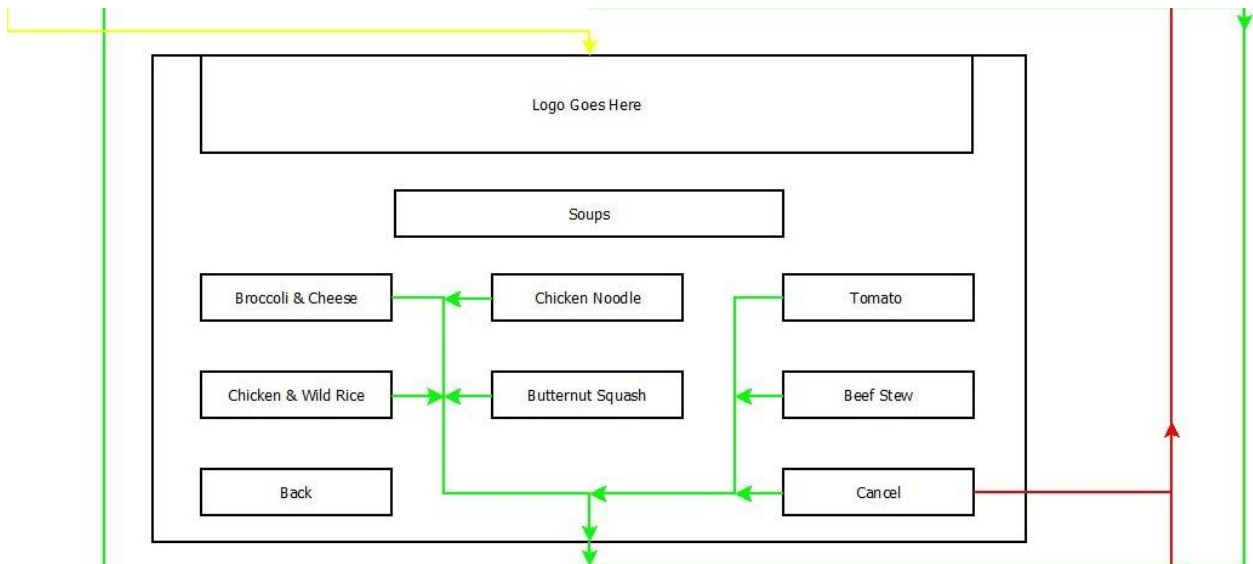
This page contains two different size steaks cooked to various degrees. If any steak button below it is pressed, the customer will be taken to the sides table. The sides table contains a list of the different sides that the restaurant sells.



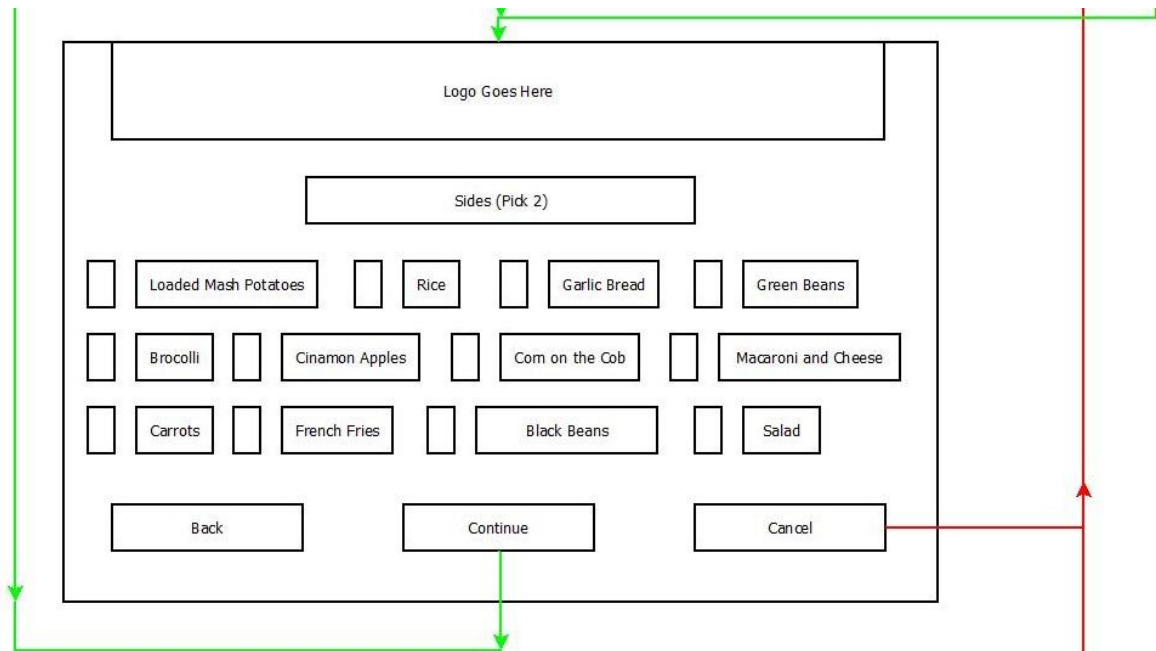
This page contains different Ingredients that the customer may want on his sandwich or burger. They can check them all, none, or any number in between. Once they have picked what they want they can click the continue button to be brought to the Sides page.



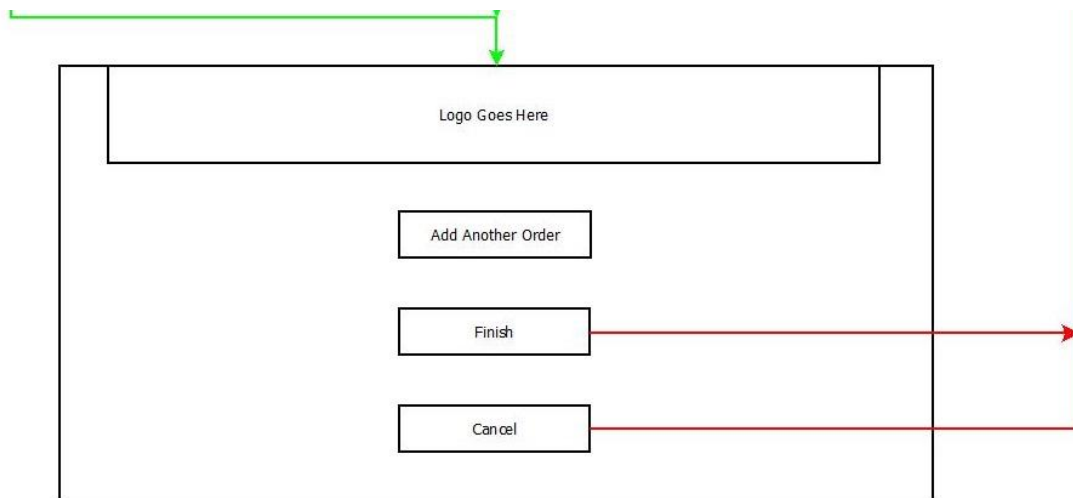
The Pastas page contains all the pastas the restaurant sells. Once the customer picks which dish they want, they are brought to the Sides pages where they can pick two sides.



The Soups page contains all the soups the restaurant sells. Once the customer picks which dish they want, they are brought to the Sides page where he can pick two sides.



The Sides page contains all the sides the restaurant sells. All the dishes all have checkboxes beside them. Once the customer picks two sides they can click continue to go to the final page. If a customer tries to select more than 2 sides and error message will pop up.

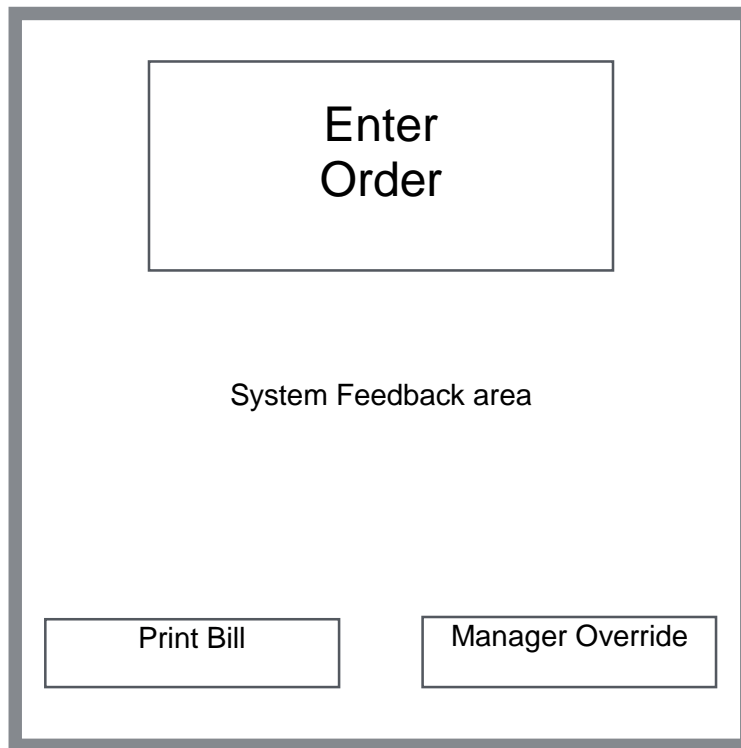


This is the final page. The customer has the choice to add something else to the order, finish the order, or cancel it. If Add Another Order is selected then the current item will be saved into a list and the customer will be taken back to the main menu's page. If the finish button is selected, the order is then transmitted to the kitchen and a copy is sent to the Data Base. The orders will be sent to the kitchen as plain text and be displayed in a way that the cooks can understand what the customer wants. If cancel is selected then it cancels the entire order and takes the customer back to the first interface page.

Rating Screen

	Terrible		Fair		Excellent
Food	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Service	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Speed	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Satisfaction	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Overall Experience	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="button" value="Cancel"/>		<input type="button" value="Submit"/>			

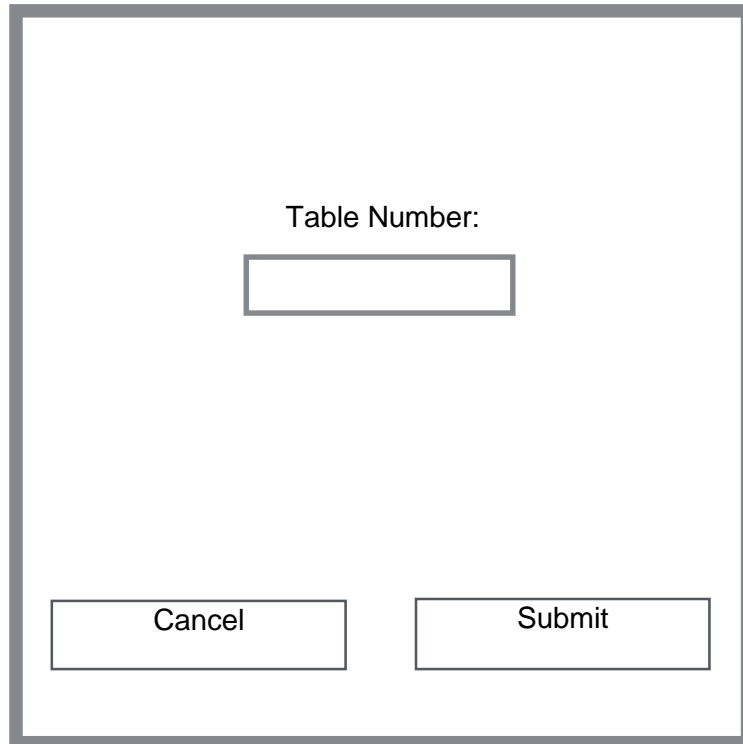
The circles on the screen indicate radio buttons where the user can only pick one of the options to give their opinion on the topic. Hitting the cancel button will take the user back to the main screen without submitting the results. Conversely, the Submit button will send the results to the server to be used for usage in statistics by management. After submission the user will see a confirmation message that their feedback has been submitted.



Main POS Screen

This screen will be the main interface that the wait staff use. Hitting the “Enter Order” button will allow the waiter to input the the table number, then enter the order that they took from the customers. Converting their shorthand to a uniform system for the database and allowing the bill to be totaled. If the customers put their order in by themselves, then the waiter can just hit the print bill button, input the table number and get the bill. If they put their own order in themselves but then wanted to add something to the order, they can hit the “Enter Order” button and go about it as they would if they were entering the order from scratch, only they will just have to enter what is being added! Finally if mistakes are made or management needs to remove items from the bill, we have a button for a manager override.

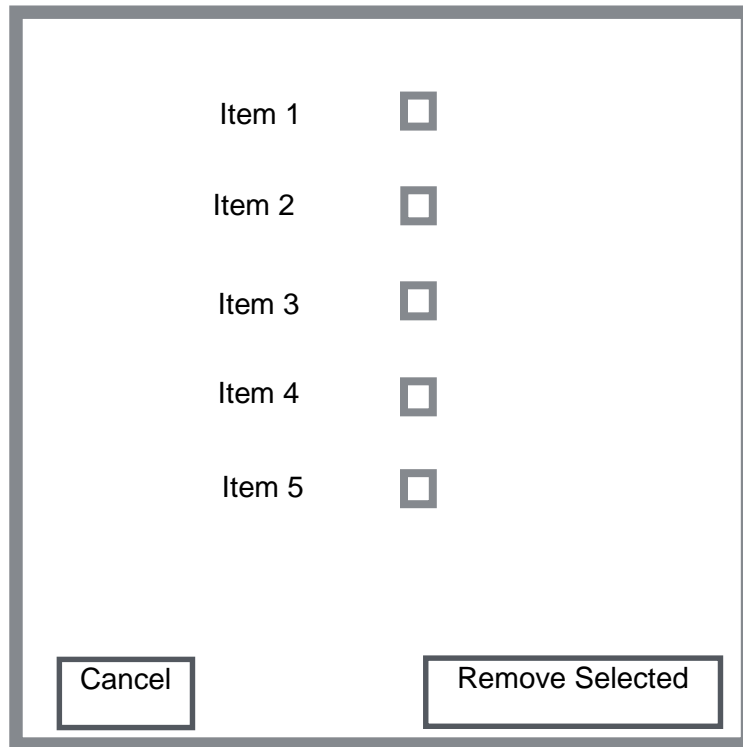
Table Input



A screenshot of a web form titled "Table Input". The form is enclosed in a thick gray border. Inside, the text "Table Number:" is centered above a single-line text input field. At the bottom of the form, there are two buttons: "Cancel" on the left and "Submit" on the right.

All those buttons lead to this screen to input the number of the table so that we can then take actions by adding to orders, removing items from an order, and printing the bill. When you hit the "Manager Override" button it will ask for a username and password first. If the user pressed the buttons to add or remove things from the order then entering a table number and hitting the submit button will display the order to remove things or the menu to add things, whichever the case may be. If the user had hit the "Print Bill" button to get to this screen, entering the table number and then submit will return the user to the main screen with a confirmation message that the bill is printing.

Remove Item Screens



Item 1	<input type="checkbox"/>
Item 2	<input type="checkbox"/>
Item 3	<input type="checkbox"/>
Item 4	<input type="checkbox"/>
Item 5	<input type="checkbox"/>

CancelRemove Selected

This screen shows where the list of what has been ordered so with check boxes so that the manager can select one or more of the items to remove from the order. If things are selected and the manager hits the “Remove Selected” button, those items will be stricken from the order. Conversely, if the user hits cancel the system will return to the main menu.

Android App

The android application will handle the work that the wait staff has to accomplish. The home screen of the mobile app will have two buttons and the app will have notifications at the top left corner for alerts as shown in Figure 1-1.

The table ready button will show a screen to type which table is now open in a text box and a send button to send to the hostess from the wait staff that a table is now open for the next customer in Figure 1-2.

The take order button, on the home screen, will show a screen to the waiter/waitress with a list to add an order and keep track of the orders as shown in figure 1-3. When an order in the menu is clicked then a screen will open to write the order and a send button will send to the kitchen. This is shown on Figure 1-4. These orders are in list form to be able to go back to them and then make the bill at the POS just like in figure 1-3.

Finally, the alert notification when dropped down and clicked will go to a screen displaying all the alerts in list form. Alerts will consist of: if a customer at a table needs help, if the people are at their table, if the food in the kitchen is ready, or kitchen problems arise delaying food. This is shown in figure 1-5. Also, if you click an alert you will be brought to a screen with a delete button and will be able to delete an alert if they are done with it as shown in figure 1-6.

These pictures are the buttons to go to the "Table Ready" and the "Take Order" screens. (Figures 1-2 and 1-3, respectively).

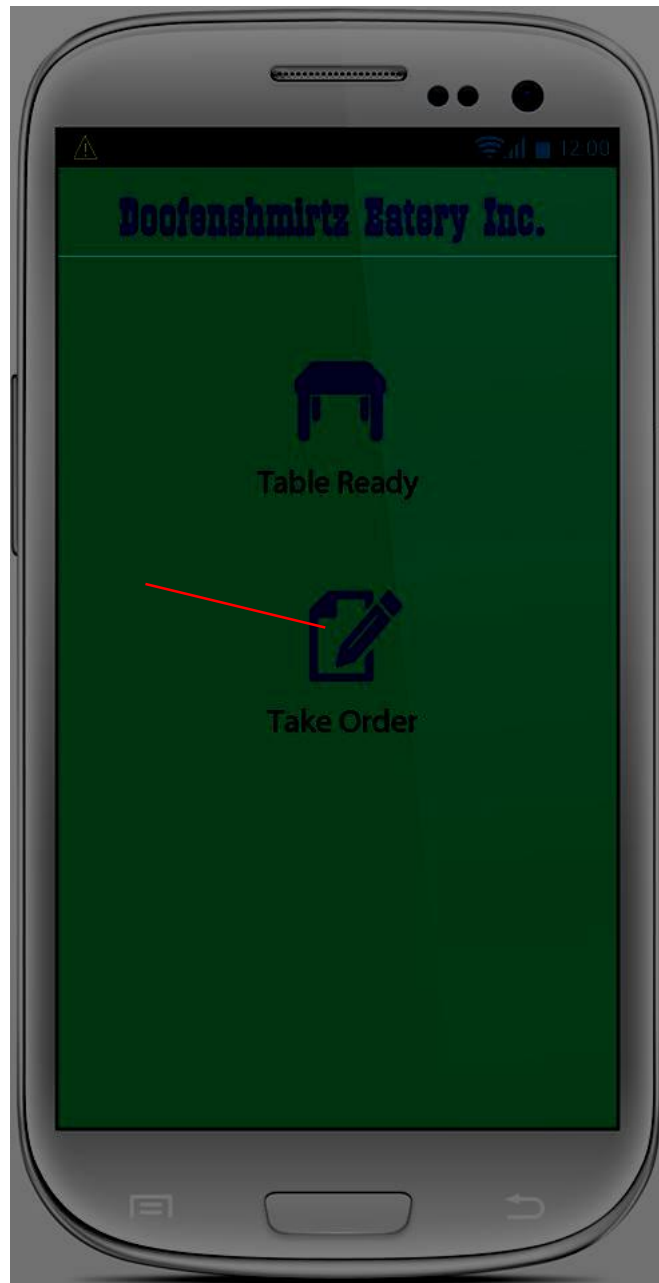
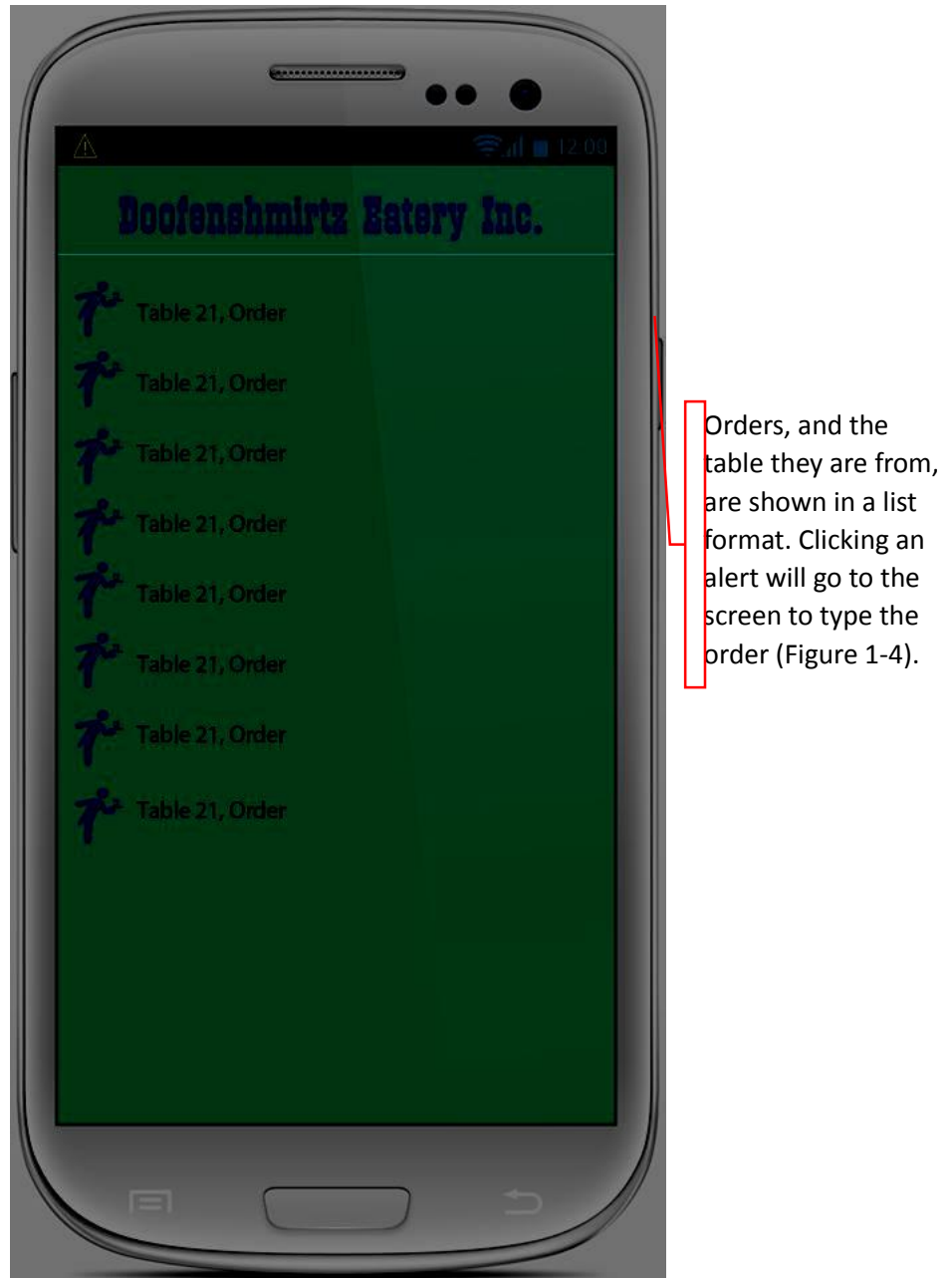


Figure 1-1: Home Screen



Figure 1-2: Table Ready Screen



Orders, and the table they are from, are shown in a list format. Clicking an alert will go to the screen to type the order (Figure 1-4).

Figure 1-3: List of Orders



Figure 1-4: Making and Sending the Order

When an alert comes in, this bar shows the alert symbol (yellow triangle).

Alerts for the wait staff are stored in a list fashion. Clicking an alert will expand it (Figure 1-6).

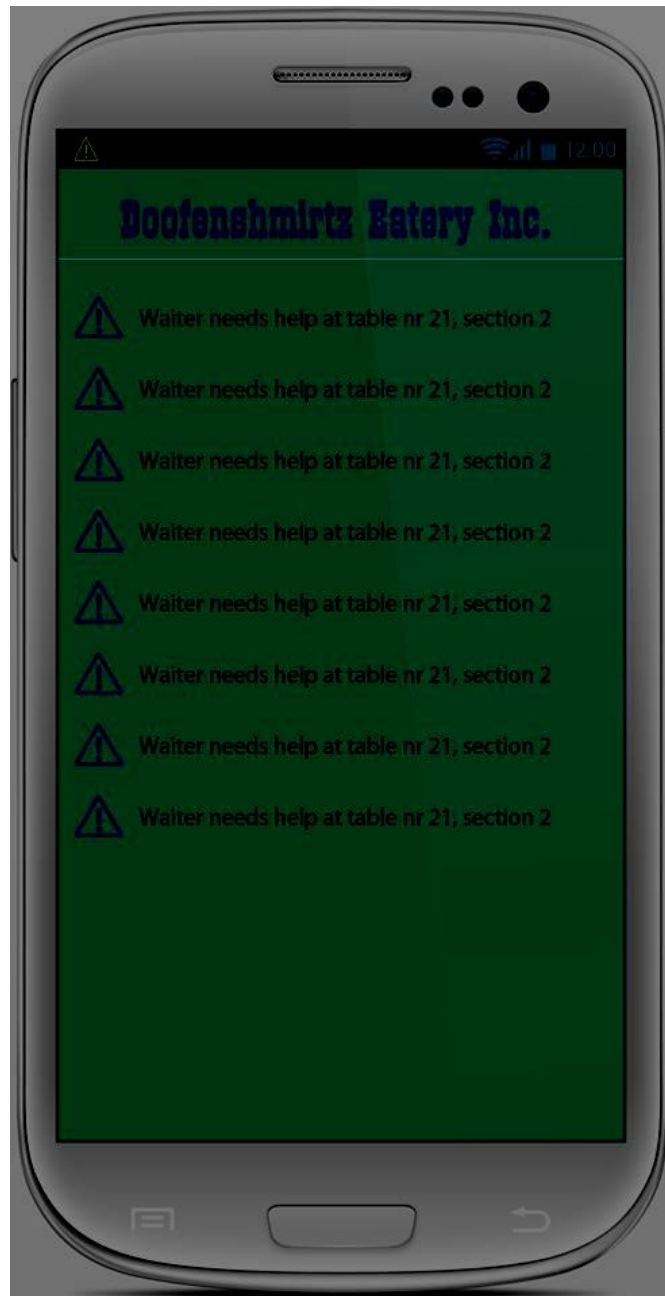


Figure 1-5: Alert Screen

Clicking on an alert shows this drop-down option. If the staff clicks this, the alert is then deleted from the alert list.



Figure 1-6: Delete Alert Screen

Networking

All hand-held devices and terminals will be connected through a central server located in the restaurant. This will be a local network that only periodically requires a connection to the internet in order to back up the database. The server will contain a SQLite database that stores the restaurant's statistical information and other data that will allow the numerous devices involved in this system to communicate quickly and efficiently. Periodically (at least daily, but possibly more often if the client's wishes), the server will automatically make both a local backup of the database and also upload an additional backup to cloud storage for safe keeping.

Android Devices:

Hand-held devices used in the restaurant will run on the Android operating system. They will utilize AsyncTask class to allow background threads to listen for data from the server. This class allows to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers. The threads send requests using Java's built in socket class to send and receive messages from the server software. These messages will include things

such as alerts, statistical information on seating and popular dishes, and the status of all the tables in the restaurant.

Terminals:

Terminals will function according to the same principles as the Android devices in regards to their networking functions. While Android devices are restricted to using a wireless connection to the server, the terminals, depending on their hardware, could potentially be connected to a Local Area Network through Ethernet cables for a more reliable connection.

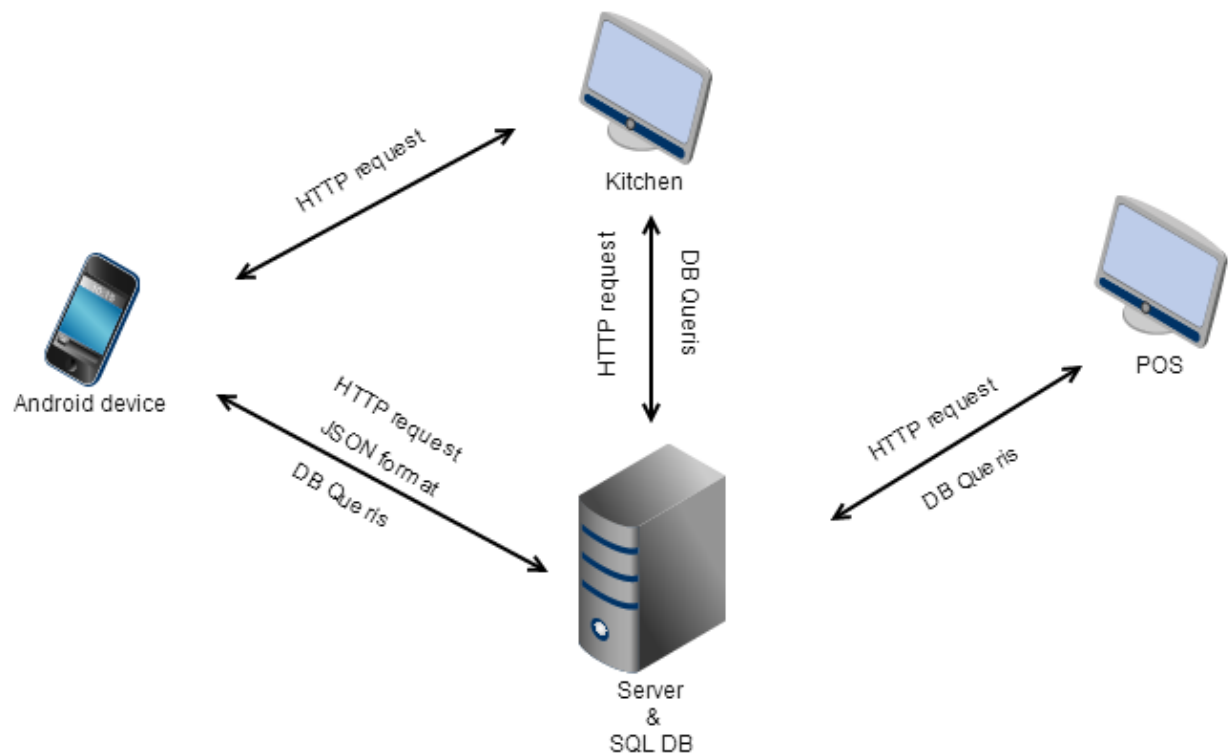
Server:

The server at the heart of Doofenshmirtz Eatery's system will run a simple server program created by our developers. It is a lightweight Java application which utilizes a local SQLite database. The server will utilize multiple threads to listen on specific ports depending who is trying to establish connection with the server. For example if a Kitchen terminal is trying it will connect to port 1499 and if POS connect it would connect to port 1455.

While some data can be communicated through primitive variables such as integers or Boolean values, some more complicated pieces of data must be transferred as objects. In these cases we will use the JSON Format to serialize the objects and transmit them from devices to the server, and vice-versa.

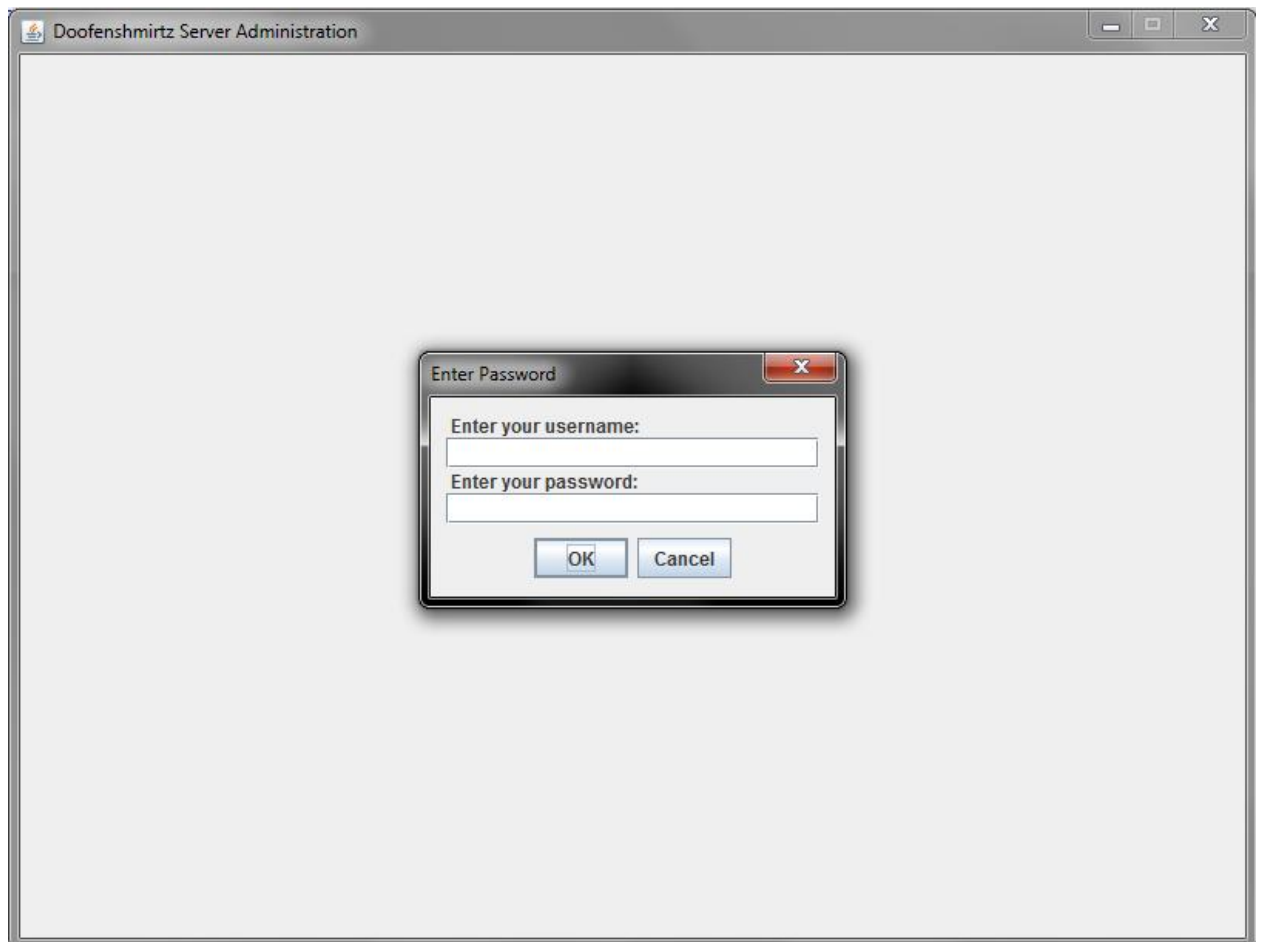
Due to the portable nature of the Java language, the server software can be run on Linux, Mac, or Windows systems, and as such the client will be able to upgrade or expand to as many terminals as they wish during the lifetime of this system.

A simplified diagram of the restaurant's network can be seen here:



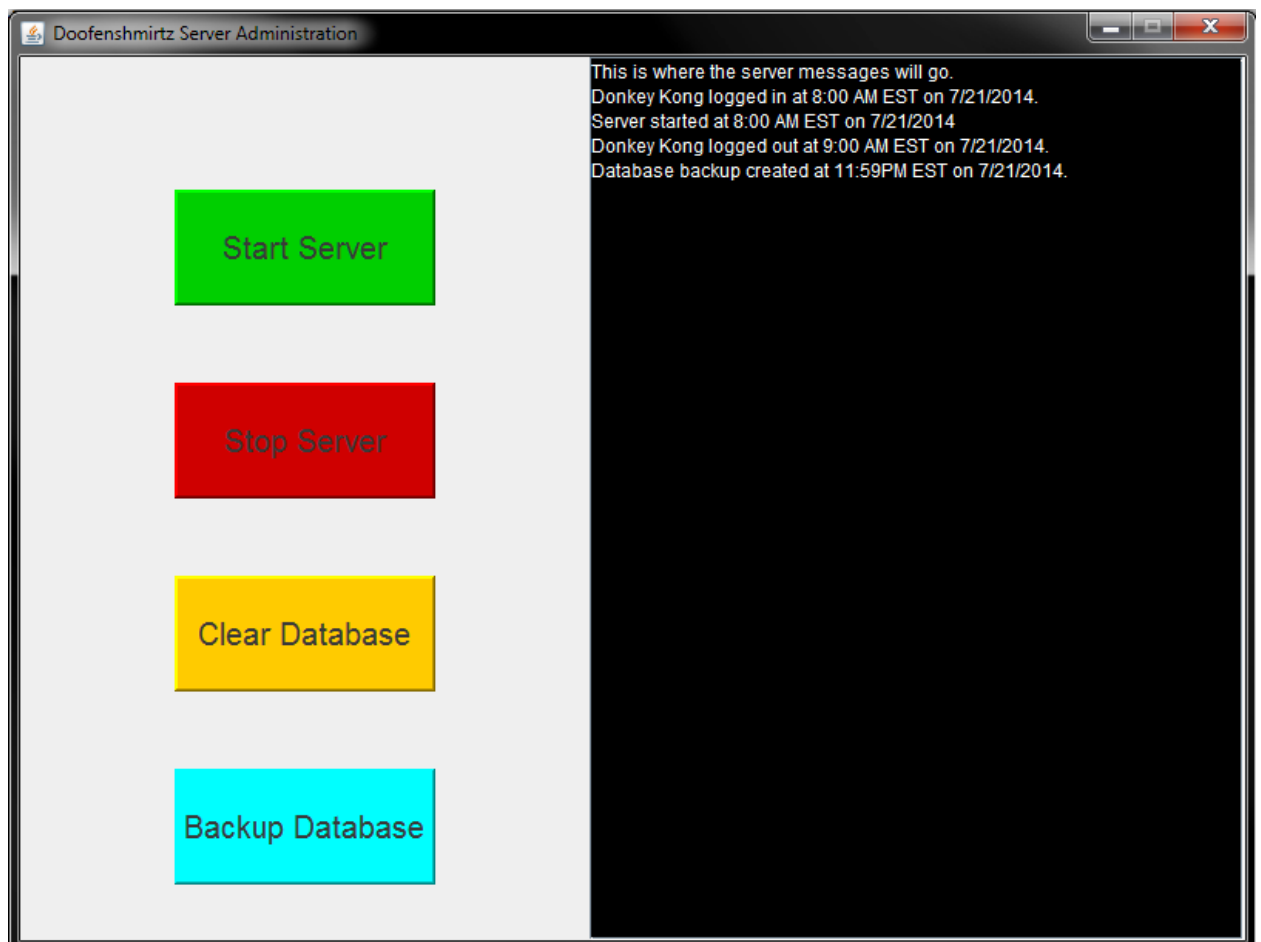
Server Interface:

The goal of the user interface for the server software is to provide an authorized employee, regardless of their technical skills to perform a very small selection of routine server tasks quickly and without risk to the valuable data held within. To this end, the server software is protected by a user name and password screen, as seen below.

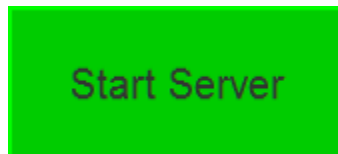


Access to the server administration program is restricted to anyone without a valid user-name and password combination. These combinations can only be created by the store owner, and can also be revoked by the store owner as well. User-name and password pairs will be stored in a separate database and encrypted in order to keep them secure. In the event of catastrophic data loss in which the store manager's password is lost, we will be able to create a new password for him/her.

Once a user has logged in, they will be presented with the following screen:



The user interface for our server was designed to be simple and easy to use with little to no training, technical or otherwise. Starting on the left, we have the Start Server button:



This button will, as one might expect, start the server. After a successful server start up, this button will change to a button that says "Server Started" and become unclickable. In the event the server fails to start, the message area on the right will display an error message that a user may use to diagnose the problem and use to communicate with our customer support team in order to speed the completion of any technical support tasks. Moving down, the next button is "Stop Server".



Again, we strive for simplicity. This button stops the server. It has no effect if the server is not already running. This button should only be used in an emergency or in the event that the server needs to be rebooted (in combination with the "Start Server" button).

The next button could potentially represent a large risk to our client's data integrity. It is the “Clear Database” button.



Clear Database

This button will allow an authorized user to clear all data from the database. This will not remove user accounts, as they are stored in a separate database. Pressing this button will trigger an alert message, asking the user if they are sure they wish to perform this action. If the user accepts this action, a special backup of the database will be made, and a new, empty database will be generated, ready for the client's use. This button is only intended to be used on exceptionally rare occasions, such as database corruption or in the event the client wishes to 'start fresh' with their computational data.

The final button is the “Backup Database” button.

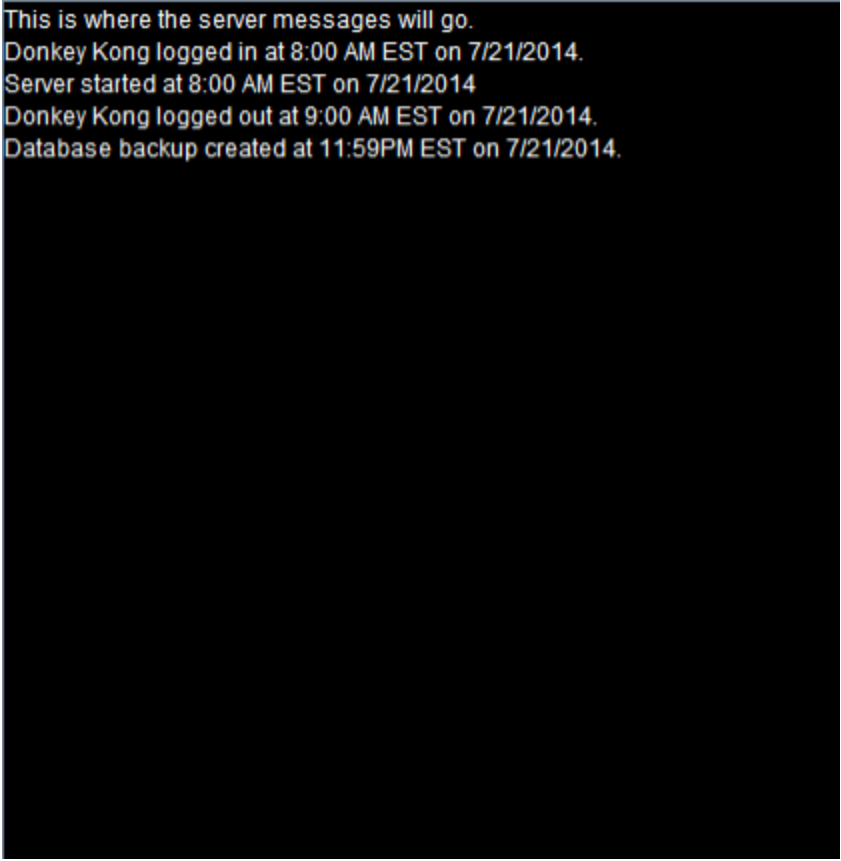


Backup Database

This button will make a 'hard' backup of the database. While the nightly, automatic backups are replaced every other day, the user generated backups are

kept indefinitely. This will allow users to keep a snapshot of data at important milestones (monthly, annually, etc.) or to make a quick backup in the case that the server must be shutdown manually.

The final element of the user interface is the message panel:



```
This is where the server messages will go.  
Donkey Kong logged in at 8:00 AM EST on 7/21/2014.  
Server started at 8:00 AM EST on 7/21/2014  
Donkey Kong logged out at 9:00 AM EST on 7/21/2014.  
Database backup created at 11:59PM EST on 7/21/2014.
```

This message log will record user log-in and log-outs, server starts and stops, and the clearing and backing up of the database. Additionally, this screen will provide error messages that will help users to communicate their problems more accurately and quickly with support staff. The information in this panel could

also be used to identify what actions users are performing during server administration.