Digitizing the Restaurant with Network-Enabled Smart Tables

By
Andrew Chen
Eric Ong
Can Zhou

Design Document for ECE 445, Senior Design, Fall 2020
TA: Sophie Liu

01 October 2020
Group 3

Table of Contents

1. Introduction

1.1. Background

Today's restaurant industry relies on archaic methods of management and customer service that require substantial contact among customers and staff. Every part of the restaurant experience, from seating to ordering to monitoring the restaurant floor currently involves human contact. Moreover, since the onset of COVID-19, there has been a shift in the industry to support more contactless service. For example, COVID-19 regulations require restaurants to have fewer personnel in-house and tighter occupancy caps. Restaurants unable to adapt to this contactless norm have suffered as a consequence: more than 26,000 restaurants have closed since July [1].

Adjustable dining tables in a local peer-to-peer network with integrated pay stations can minimize unnecessary contact during the restaurant experience for three reasons. First, by allowing restaurants to adjust the height of dining tables, they can customize their indoor and outdoor layouts to create flexibility and minimize contact between customers and staff. This allows restaurants to cater to both customers that prefer indoor seating and customers that prefer outdoor seating and also provide ergonomic benefits for customers, increasing the overall comfort throughout the experience [2]. Second, implementing a local wireless network to monitor the dining room will further aid in reducing foot traffic for restaurant staff to inspect tables. Finally, integrating a payment station into the dining table minimizes the need for staff to approach the table and to circulate through the restaurant, therefore decreasing the risk of transmission through contact.

In response to COVID-19 regulations, restaurants have attempted to use outdoor seating to mitigate the risk of transmitting the virus. However, restaurants still lack contact-free interaction in almost every stage of the typical dining experience including: payment, ordering, and food delivery [3] [4]. The constant interactions between staff and customers potentially puts both parties at risk for contamination. Further, governmental institutions are strongly encouraging restaurants to switch to more contactless methods of service throughout the entirety of the restaurant experience [5] [6]. The less interaction between staff and customers, the lower the chance of transmission of the virus.

1.2. Objective

As COVID-19 has changed the restaurant industry to encourage outdoor seating and contactless transactions, archaic technology has hindered the ability for restaurants to adapt to evolving consumer needs. There are two aspects to restaurants we will address in our design. The first is the dependence on customer interactions that may instead be facilitated by a computer system. Our project seeks to keep customer-to-employee interaction to a minimum level in the restaurant experience, requiring less staff at the restaurant while still fulfilling all restaurant functions. The second issue is the lack of seating options with limited available space for customers to dine with indoors. Our project will address this by allowing tables to adjust their heights to adapt to customer seating preference.

To create a more contact-free environment in restaurants, we are developing a table system that connects to an internal restaurant network, allowing for the management of tables and eventually, customer orders as well. The connection to a network will allow restaurants to monitor available tables and keep track of occupancy without the need for personnel to check it manually, enabling fewer employees to risk exposure to the virus from patrons. In addition to this, on the patrons' end, our solution will empower them with the flexibility to pay whenever they please through a payment system integrated directly into the table, which also aids to further minimize customer-to-staff interaction.

To accommodate customer dining preferences while giving restaurant owners more flexibility, we are developing a self-adjusting height system. Having a table that can adjust its height provides an inclusive table model for tall and short people. In addition to this, restaurants will no longer need to buy different types of tables, nor will they need to move the tables around. As restaurants change due to the practices adopted due to COVID regulations,, more long distanced seating has been encouraged along with lower capacities within the restaurant space, but this may change as vaccines for the pandemic arrive and so this flexibility will aid in a smooth transition back to a larger occupancy with more customization options throughout the recovery.

1.3. High-level Requirements:

- The table will adjust its height automatically from 26 inches to 40 inches based on the sitting height of customer knees and whether the customer is sitting or standing at a rate of an inch per duration within 10-20 seconds.
- The status of the table, whether occupied, in need or service, or vacant, will be updated every minute for internal restaurant use.
- The table will house contactless payment methods. The status of the payment of a meal will be available for customers to check without their own equipment.

2. Design

Our design will facilitate less contact at restaurants, ideally to the point where the only time that customers interact with staff is when bringing food out. These tables will synergize nicely with a restaurant system in which customers order food from the restaurant counter, find a location in the restaurant or outdoors, and either bring their food to the table or have it brought to them. Restaurants have made efforts to protect order takers in establishments where restaurant counters are utilized, so limiting employee-to-customer interaction to bringing food out to tables minimizes exposure to guests. Since the table will adjust its height based on customer preferences, the decision of whether to sit at the table or not will be placed in the hands of the customers.

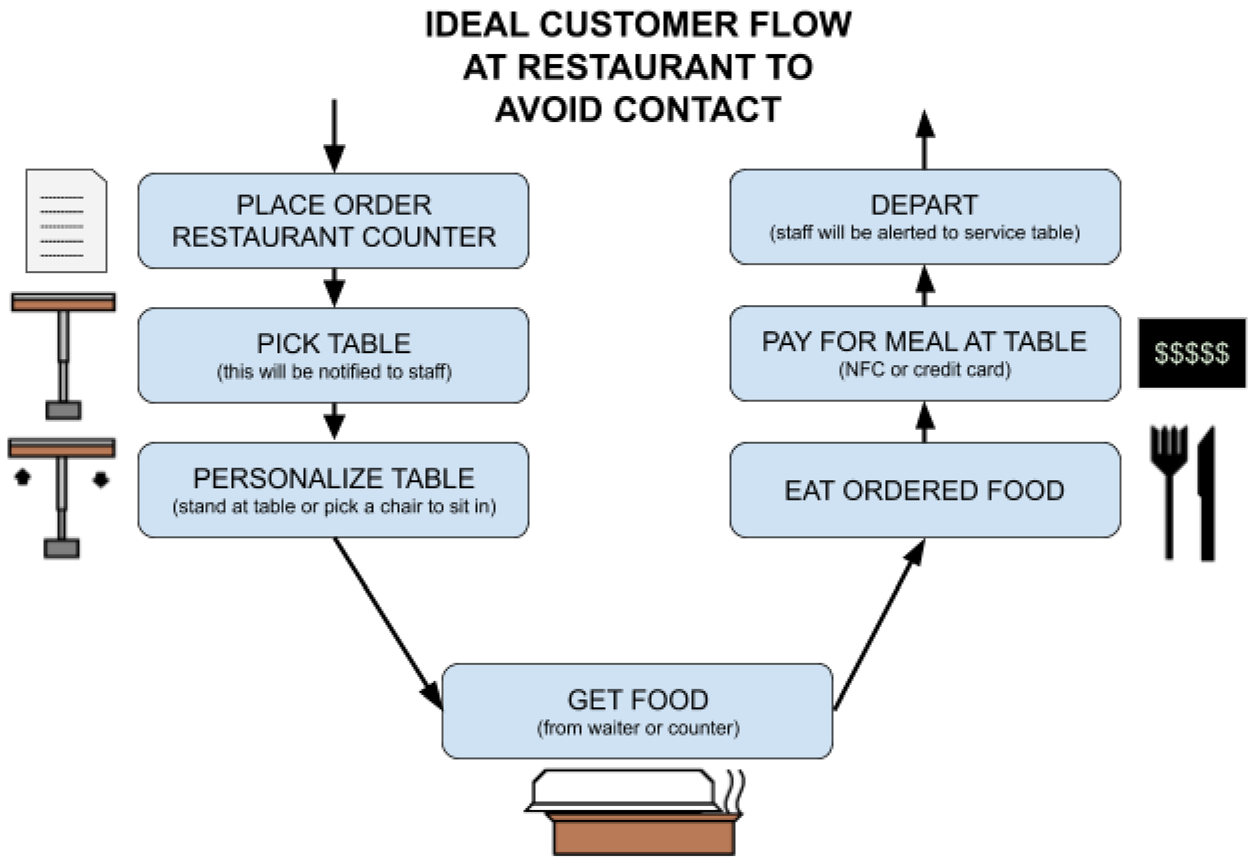

Fig. 1: Ideal diner interaction with table system

2.1. Physical Design:

The design of our table will include a physical table that demonstrates height accommodation. There will be components inside of the table as well as a separate module that will be placed on a chair that will communicate wirelessly with table components via UART sent through a bluetooth module. These other components will support the transmission of the status of the table as well as handle payment from the table for meals. The table will be able to process digital payments via NFC reader and magnetic stripe reader which will be set to send payment information to a central hub used by restaurant staff and the hub will process the payments. Additionally, the central hub will be a computer hosting a web application and communicating with all tables in the restaurant via bluetooth.

Our table will contain the majority of our project's sensors and will measure a foot by two feet for the top surface. The minimum height of the table prototype will be 26 inches, and the table will be vertically adjustable upwards by 14 inches. In implementation, we will hold the table at maximum height and adjust downwards to accommodate customer height. The placement of the height-adjusting module parts is detailed in figure 2. The height adjustment mechanism will consist of a DC motor interfacing with a lead screw which will drive the height of the table itself. The voltage sent to the motor will be determined by a microcontroller which will poll an ultrasonic sensor for the height of the person sitting at the table at any given time, with the table adjusting to max height if it detects no presence with the ultrasonic sensor.

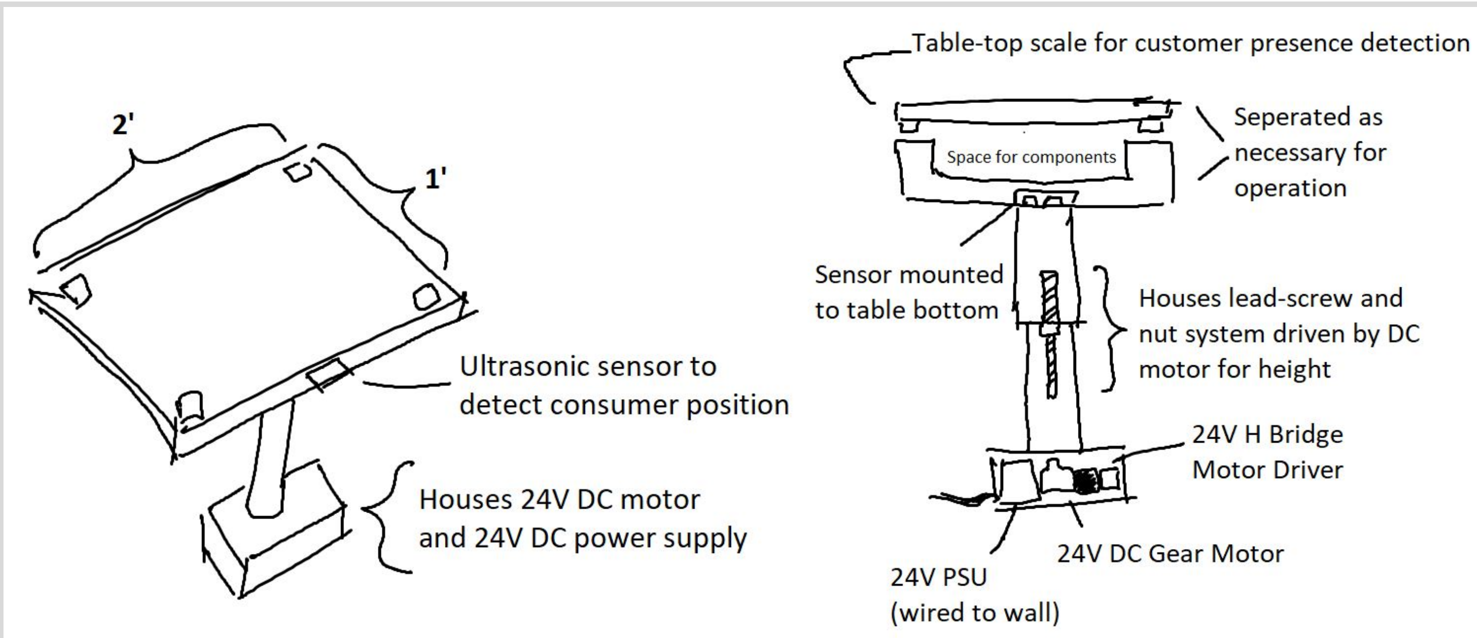

Fig. 2: Table Physical Components Sketch

Also pictured in figure 2 is a table-top scale that will be used to determine customer status for a table network. In tandem with the module that will be attached to a chair, this scale will be used to determine the status of the table, whether it be vacant, requiring service, or occupied. The sensors on the table top scale will consist of four load cells connected to a microcontroller, which will interface with a table module in communication with the table network we are creating. This table module will wirelessly communicate with the central hub of the restaurant to keep track of table status.

Seats will contain sensors embedded inside as well, and will be used to differentiate between a table being ready to service or not. We will create our own force sensing resistor (FSR) for use with the customer presence detection component of our table, since we require detection of whether a person is sitting on it versus something lighter being placed on it. FSRs online have been too small or have had too low of a range for us to consider using them for customer detection. The internals of our custom-made FSR are shown in figure 3.

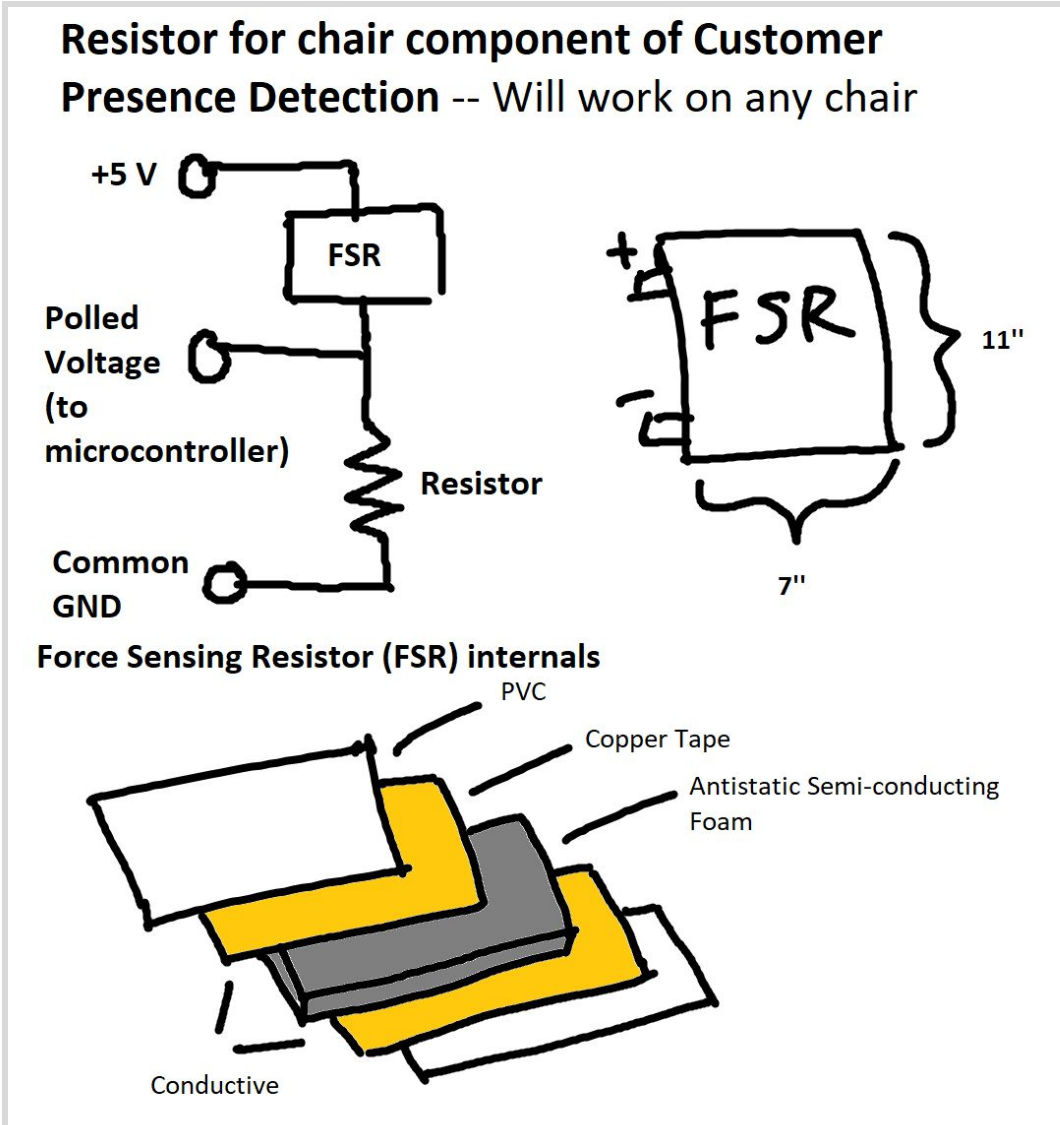

Fig. 3: Chair Component Sketch

2.2. Block Diagram:

Our adjustable table with integrated pay station requires several different groups of components to operate successfully. These components are organized and shown in figure 4, grouped by functionality. Our components require DC power, specifically 5V DC and 24V DC, which will be provided by power supplies in the power group shown. The height adjustment is separated from the networking aspect of our table, but will be powered by the same power source as shown in figure 4. It is important to note that the chair component to our customer presence detection group will be physically separated from the table, and will be powered separately from the other components in our project. Therefore, we have chosen to include its related battery power supply with the customer presence group instead of the power group.

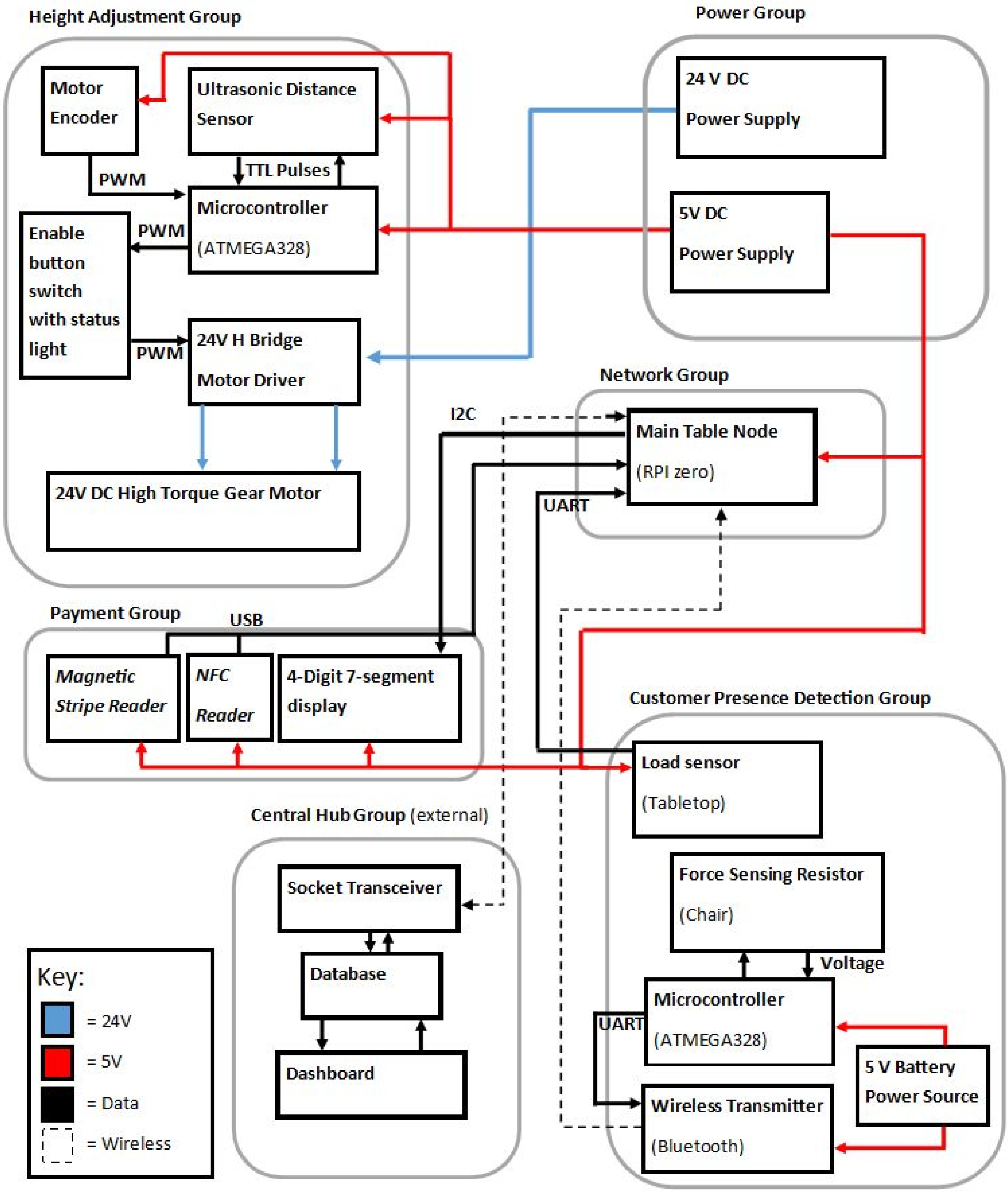

Fig. 4: Block diagram for entire project

2.3. Functional Overview & Block Diagram Requirements:

2.3.1 Height Adjustment Group

Customers come in all different heights, and so in order to best serve them, we have decided to incorporate an auto-adjustable height system to our table. The benefits of having a table at the right height when sitting and doing a task are tremendous [7], and as previously mentioned, having a single set height for customers of varying heights fails to serve them all.The table will be held up with a linear actuator, thus allowing for the overall height to be adjustable. The table will adjust its height accordingly to the customers' heights once they sit down. We plan to make the table adjust the table's height by measuring the distance between the bottom of the table with the customer's knees when they are sitting down using ultrasonic sensors.

Enable button: This button will be wired up to the H-bridge for the height locking mechanism.

Requirement	Verification
A button will be used to enable and disable the motor system.	1. Induce a mock signal that is driven from the microcontroller to the motor driver H bridge, and wire it through the switch. 2. Ensure that the signal propagates to the motor driver H bridge when the switch is closed. 3. Ensure that the signal goes to the common ground when the switch is open.

Motor Encoder: Attached to some gear motors is an encoder, which sends data that may be polled by a microcontroller to determine direction the motor is moving.

Requirement	Verification
The motor encoder will interface with the microcontroller to confirm direction of the motor.	1. Connect the motor encoder to a breadboard for polling purposes and to power the encoder with a 5V signal. 2. Induce a signal in the motor and ensure there is a signal from a

	terminal from the encoder using a voltmeter. 3. Reverse the signal across the motor terminal, reversing the direction of the motor, and probe the other encoder terminal.

Ultrasonic distance sensor: An HC-SR04 ultrasonic distance sensor will be used to determine customer presence under the table as well as the sitting height of the customer's legs. When a customer is absent from the table, the table will adjust to maximum height. When a customer sits at the table, the table will adjust downwards until it is above the height of his or her knees by roughly eight inches and stop moving. The sensor will read while the table is adjusting as a failsafe, but the motor will be slow enough (running at 28 RPM) such that if the table came into physical contact with a customer, the customer will not be injured.

Requirement	Verification
Sensor will detect height at least to the nearest inch, and works at the maximum table range of 40 inches.	1. Wire up the HC-SR04 sensor to 5.0V power in the VCC pin, ground in the GND pin, and the echo and trigger pins to two unused digital pins on an Arduino. 2. Connect microcontroller to USB UART bridge and to a computer running the Arduino IDE. 3. Write a program that will continuously read in distance data from the Arduino and print to the IDE Terminal and flash it to the microcontroller. 4. Use an object large enough to block both senders and receivers on the HC-SR04 and move it back and forth to verify that the distance being written on the terminal changes as it continuously polls for data. 5. Place a ruler or tape measure in front of the HC-SR04 sensor (one end just below the sender/receiver) to measure the distance outgoing from

	the sender and receivers. 6. Move the object along the ruler or tape measure and verify that the distance away from the sensor matches the distance data being written to the Terminal within 1 inch up to a maximum range of 40 inches if necessary.

Microcontroller: We will use a microcontroller to dictate the action of the motor based on signals from the enable button, encoder, and ultrasonic distance sensor. We have elected to use the Atmega328p microcontroller, specifically. A microcontroller is necessary because it will interpret the information passed from the ultrasonic sensor and will send the signal to the h-bridge driving the motor, prompting the table to adjust higher or lower. The microcontroller will be used to determine when to adjust and when to hold position. This microcontroller will not be connected to the central hub of our table, as the height system will be independent of the table status.

Requirement	Verification
Microcontroller must be able to take a 10 sample median over 1 second from the samples it reads from the ultrasonic sensor (which emits data at a rate of 40Hz).	1. Connect microcontroller to USB UART bridge and to a computer running the Arduino IDE. 2. Wire up the ultrasonic sensor to the microcontroller AFTER verifying it. 3. Set up the IDE terminal to read at 115.2kbaud. 4. Write a program that selects 10 samples at a rate of one sample every 100ms. 5. In the program, sort the samples and compute the median. Print the 10 samples and the median. 6. Flash the program and observe the output printed in the terminal. Verify that the output calculation matches either by doing the computation by hand or writing unit tests.

5V to 24V H Bridge: Since the microcontroller has a 5V signal and the motor is rated for a maximum of a 24V signal, we need an H bridge to send the necessary voltages to power the motor from either motor terminal.

Requirement	Verification
H bridge will take in a modulated 5V signal and use it to control 24V signals to both sides of the motor	1. Connect the motor encoder to a breadboard for polling purposes and to power the encoder with a 5V signal 2. Induce a signal in the motor and ensure there is a signal from a terminal from the encoder using a voltmeter 3. Reverse the signal across the motor terminal, reversing the direction of the motor, and probe the other encoder terminal

24V DC Gear Motor: The height adjustment of our table will be driven by a motor driving a lead screw with a nut attached to it. Rotating the lead screw will ascend the platform used as our table top or descend it depending on the direction of the rotation.

Requirement	Verification
24V DC Gear Motor must work in both directions to allow it to raise and lower the table	1. Connect the 24V DC power supply to the wall, and connect a 24V DC signal to a breadboard along with the common ground. 2. Use a voltmeter to ensure that the 24V DC signal is as expected before moving on, then keep the power of the power supply off if possible 3. Connect the 24V signal to a 1 kΩ resistor in series with a potentiometer. Place the motor terminals in parallel with the resistor. 4. Ensure that the DC motor works from low to high resistance and thereby works with a 24 V signal. 5. Reverse the terminal connections of the motor and ensure that the motor works the same way across the reverse configuration.

2.3.2 Power Group

The components in the table require different power sources, specifically 24V for the high torque gear motor and 5V for the majority of the components. The specific parts used in this project require DC voltage, so we require a power supply that converts the AC power from a wall outlet to constant DC power for the height adjusting motor. The 5V power supply for the chair will be connected to a separated circuit with the force sensing resistor in the customer presence detection group, so it has been included in that group instead.

24 V DC Power Source: In order to drive the DC Gear Motor, we will employ a 24V power source that is sent through the H bridge in the height-adjustment group.

Requirement	Verification
The power supply should supply a constant 24V DC signal to its 24V DC terminal.	1. Use rubber, insulated gloves as a precaution while handling the power supply. 2. Connect the wall adapter, unplugged, to the live, neutral, and ground terminals on the power supply. 3. Connect a lead from the 24V terminal and a lead from the common ground terminal of the power supply to a breadboard. 4. Carefully connect the wall outlet cable to provide power to the power supply 5. Ensure that the output signal from the power supply reads around 24V DC. Adjust the power supply until the voltage is exactly 24V if it is not.

5 V DC Power Source: For the rest of the components that will be wired in the table. we will use a power supply that supplies 5V DC.

Requirement	Verification
1. If the 5V is provided from a wall outlet, ensure its terminal supplies a constant 5V DC signal 2. Otherwise, if the 5V power supply is battery powered, ensure the signal is at least 4V and less than 5.5V	1. a. Use rubber, insulated gloves as a precaution while handling the power supply b. Connect the wall adapter, unplugged, to the live, neutral, and ground terminals on the power supply c. Connect a lead from the 5V terminal and a lead from the common ground terminal of the power supply to a breadboard d. Carefully connect the wall outlet cable to provide power to the power supply e. Ensure that the output signal from the power supply reads 5V DC. Adjust the power supply until the voltage is exactly 5V if it is not. 2. a. Using a four slot AA battery holder, turn off the connection to the terminals if it is possible.

	b. Slot in four AA rechargeable batteries into the battery holder. Rechargeable batteries have lower voltage, and this will work better for a battery holder that connects all four batteries in series. c. Carefully turn on the battery holder if it was turned off and make sure the leads from the battery holder are separated from each other. d. Ensure that the voltage outputted by the battery holder power supply is between 4 and 5.5 V by using a voltmeter.

2.3.3 Network Group

We plan to create a mesh network of raspberry pi's to track the status of tables in a restaurant. This network will be separated into two types: inter-communication and intra-communication. For the purposes of this project we will demonstrate and deliver one coordinator that will communicate with two table nodes.

Inter-table communication: This communication consists of all of the tables communicating with a single coordinator device. The tables will each have a raspberry pi zero embedded inside and they will all communicate with the coordinator via an ad-hoc network using sockets. This will be a one way communication from the table nodes to the coordinator to update the table's status. We will only build one prototype for the table, but we are planning on having an extra table node with toggleable statuses to demonstrate the reliability of the mesh network.

Requirements	Verification
1. All table nodes will be able to ping the central server node. 2. All table nodes will be able to send information to a single central server node via sockets over an ad-hoc network.	1. Verification process for pinging the central server. a. Allow the central table node to act as a wireless access point to an ad-hoc network. b. Ensure that the central coordinator and the table node are on the same ad-hoc network by selecting the correct network in the network settings for the table node. c. SImply ping the central coordinator from the table node via the 'ping' command and using the correct IP address. 2. We will write a program for the central coordinator node to listen and receive UDP

	packets and a program for each table node to send UDP packets. a. Initialize the listener program on the central table node. b. Send a simple packet of data to the central table node. c. Verify that it is fully received by the coordinator as expected by inspection.

Intra-table communication: This communication will be done between the table and the customer presence detection group via Bluetooth. All the sensors that track customer presence (attached or unattached) to the table will be able to feed data into the main table node via Bluetooth. The rate at which data will be transmitted from the bluetooth module to the table will be one message every five minutes. This is to ensure that should a customer get up to go to the restroom, the table will allow for some time of unoccupancy. This will be in conjunction with the fact that there will be multiple seat sensors so a table will not be completely declared as unoccupied unless all seats for that table report an unoccupied status for more than five minutes.

Requirement	Verification
Data (read in from the force sensing resistor circuit) can be sent via Bluetooth to a table node (Raspberry Pi) at an approximate rate of one message every five minutes.	A. Wire up the Bluetooth module to the microcontroller used in the chair component of the customer presence detection group B. Setup the table node (raspberry pi) to listen on incoming bluetooth connections. C. Send a simple packet from the microcontroller to the table node. D. Verify that the sent and received packet match via inspection.

Status 7-Segment Display : A 4-digit 7 Segment display will be used to show the customer the total amount due for the bill. This will be connected to the Raspberry Pi Zero and output the bill amount received from the central coordinator.

Requirement	Verification
7 Segment display will be able to output	1. Wire up the 7 segment display to the appropriate power, ground, and data pins to the Raspberry Pi.

integer values.	2. Write a program that utilizes the Raspberry Pi's GPIO pins and tm1637 libraries to print numbers on the hex display. 3. Run the program and verify that the printed values on the display match the expected values in the program by inspection.

2.3.4 Payment Group

The table will have NFC reader and magnetic stripe reader for contactless transactions. The payments will be facilitated by a 4-digit 7-segment display used to let customers know how much they owe and when they may pay for their meals. The payment data will be sent to the centralized hub for processing and confirmation.

Payment Confirmation: A 4-digit 7-segment display will be used to display when payments are ready to be made, when payments have been successful, and when payments have been rejected.

Requirement	Verification
4-digit 7-segment display will provide the payment status of the current customer	1. Wire the 7-segment displays to the main table node(raspberry pi zero) 2. Set the central hub to send/receive packets to the table node 3. Set the table node to send/receive packets 4. Set up a program in the table node that will translate the data sent from the central hub to the displays 5. Ensure the node and hub are communicating by using a test program that writes to the 7-segment display 6. Send an arbitrary bill amount to the table node via the central hub 7. Ensure that this amount is displayed displayed on the 7-segment display 8. Pay the test amount through NFC/Magnetic Stripe Reader 9. When the central hub receives the information from the raspberry pi zero, ensure that the table node receives an accepted or rejected status from the central hub.

NFC and Magnetic Payment Modules: For contactless payment, we are using NFC and magnetic payment chip readers for customers to pay without interacting with servers/cashiers. The payment processing needs to be secure and universal.

Requirement	Verification
NFC/Magnetic Stripe Reader will be our payment gateway. It will be on the side of the table and have a visual cue to let customers know where they will pay for their services.	1. Connect the NFC/Magnetic Stripe Reader to the main table node(raspberry pi zero) 2. Ensure the payment are going through by paying a small amount with a NFC/Magnetic Stripe payment and verify the amount went through 3. Listens for bills send from the dashboard 4. Dashboard will initiate a bill and the amount will be on the table's display 5. When a bill is issued, wait for the customer to pay and send the payment info back to the hub. 6. Process the payment via a secure payment service (paypal) and store the bill into the database. 7. An accept or reject payment message will be sent from the service and be used to update the payment status 8. Ensure that the payment message matches the display

2.3.5 Central Hub Group

In order to view all of the tables' statuses, a dashboard will be created to give waiters an easy way to view which tables are occupied or not in the restaurant. This dashboard subsystem will compose of a receiver to poll data from all of the tables, a locally hosted database to store this data, and a simple web-based interface that will poll and track the statuses of the tables from the database in real time in a user-friendly GUI.

Socket Transceiver: A listener process will constantly poll the table statuses. This process will run whenever the entire table system is in use and will write any data it receives to a locally hosted database. The table will send its status to the central coordinator at a rate of one message every thirty seconds (at a minimum). This rate should be more than sufficient to update the status in a timely manner to the coordinator.

Requirements	Verification
1. The central coordinator can receive messages from all table nodes.	1. We will write a program to handle sending and receiving via sockets. a. Connect the central coordinator and one or more table nodes to the same network.

2. Central receiver can broadcast messages to all table nodes.	b. Initialize the receiver program on the central coordinator. c. Initialize the sender program on the table nodes. d. Send a simple packet over UDP from the table node to the central coordinator. e. Verify that the sent and received messages on both machines are the same via inspection. 2. A message can be sent downstream from the coordinator to all nodes. a. Connect the receiver and two or more table nodes to the same network. b. Initialize the receiver program on the table nodes. c. Initialize the sender program on the central coordinator. d. Send a simple packet over UDP from the table node to the central coordinator. e. Verify that the sent and received messages on both machines are the same via inspection.

Database: A database will be used to house the real-time customer data in regards to the status of each table that is in use or free. The database will be hosted on a desktop or laptop and will ensure read/write consistency when writing data from the receiver and also when having data read and published to the front end.

Requirements	Verification
1. Database can write data persistently 2. Data inside of database can be updated 3. Data inside of database can be read	We will run a MongoDB database instance locally on the central coordinator and interface with it via Node.js. MongoDB is a reliable database with a vast amount of documentation for running and interfacing with. 1. Verification procedure for database writes. a. Start up the MongoDB instance on a computer. b. Use MongoDB's Node.js API to write a simple JSON struct into the database c. View the contents of the database either by using the MongoDB shell to print out the collections or using MongoDB Atlas to have a GUI when viewing data inside the

4. Data inside of database can be deleted	database. d. Verify that the struct written in the database matches the initial one via inspection. 2. Verification procedure for database updates. a. Start up the MongoDB instance on a computer. b. Use MongoDB’s Node.js API to write a simple JSON struct into the database, something like key{ value: 0 }. c. View the contents of the database either by using the MongoDB shell to print out the collections or using MongoDB Atlas to have a GUI when viewing data inside the database. d. Use MongoDB’s Node.js APi to update the struct to something like { value: 1} e. Verify that the struct written in the database matches the expected one via inspection. 3. Verification procedure for database reads. a. Start up the MongoDB instance on a computer. b. Use MongoDB’s Node.js API to write a simple JSON struct into the database, something like key{ value: 0 }. c. Use MongoDB’s Node.js APi to read the particular value by its key (with the previous struct that would simply be key). d. Verify that the struct written in the database matches the expected one via inspection. 4. Verification procedure for database deletions. a. Start up the MongoDB instance on a computer. b. Use MongoDB’s Node.js API to write a simple JSON struct into the database, something like { value: 0 }. c. View the contents of the database either by using the MongoDB shell to print out the collections or using MongoDB Atlas to

	have a GUI when viewing data inside the database. d. Use MongoDB's Node.js APi to delete the struct. e. Verify that the struct written in the database matches the expected one via inspection.

Dashboard: The dashboard will be a front end web application that will be locally hosted and not require any internet access. The dashboard is intended for internal use by the restaurant and its staff (namely waiters and waitresses), though the edited information from the dashboard may see use with customers commercially. The internet-dependant nature of the dashboard ensures that should there be a network outage in the area that disables internet access, restaurant employees will be able to continue using the application normally without any drops in performance for monitoring the statuses of their dining room.

Requirement	Verification
1. The dashboard will allow for users to upload and view it. 2. Users will be able to mark where tables are on the uploaded image. 3. The dashboard will query the database and update the status of each table in realtime. 4. Users will be able to assign a	1. Verification process for viewing uploaded dashboard images. a. Initialize and run the dashboard on localhost. b. Navigate to the dashboard, press the upload image button, and select an image (PNG or JPG) to upload. c. Verify that the image rendered on the screen matches the original by inspection. 2. Verification process for marking tables. a. Enter the 'Edit' mode for the dashboard. b. Select the table marker tool. c. Ensure that the markers visibly persist on the screen when finished editing. 3. Verification process for querying database and printing result to dashboard. a. Repeat the process for (2). b. Write a specific status value for a table in the database using either the MongoDB Node.js API or MongoDB Atlas. c. Verify that the status value rendered on the dashboard matches that in the database. 4. Verification process for assigning a bill to a table. a. Users will be able to select a table and a text box will appear.

dollar amount bill to a table.	b. Input a dollar amount bill to assign to the table. c. Press the confirm button. d. View the database and ensure that the bill is registered under that table.

2.3.6 Customer Presence Detection Group

The status of a table will be gauged based on the amount of weight on the physical table itself. An occupied (or even just an unoccupied and dirty table) will be marked as such since the weight of excess food, water, plates, and whatever else the customer may bring will be measured by this pressure sensor. For our prototype, any amount over 1000 grams, or one kilogram, will be considered occupied. In use, restaurants would zero their equipment based on the weight of plates provided as a standard to each table, and this will allow the restaurants to consider the top of the table unoccupied when there are clean plates and cups ready on a vacant table.

Load Sensor: We will use four strain gauges mounted to the bottom of the top of the table to detect weight of objects placed on the table.

Requirement	Verification
1. Each strain gauge should work properly to detect when strain is applied to them to support items on any part of the table surface 2. The four strain gauges should interface with the provided amplifier chip to send a signal to the microcontroller in order to get an accurate weight no matter where on	1. Verification process for strain gauge. a. Secure a strain gauge with tape to a flat surface b. Find a solid medium, made of plastic or wood, to use to press down on the strain gauge c. Connect the white lead of the strain gauge to 5V DC, and connect the black lead to ground d. Connect a voltmeter across the red lead and ground e. Ensure that the red lead outputs different voltages depending on the force applied to the strain gauge using the voltmeter. Repeat this process for the three other strain gauges 2. Verification process for strain gauges initiating amplifier to send a signal. a. Connect the wires of each strain gauge to the amplifier chip as directed by the manufacturer b. Suspend all strain gauges to the same surface in the same orientation c. Connect the load cells and the amplifier chip to

the table an item is placed	5V of power and common ground d. Use a large sheet of solid material, like plastic, to apply force downwards to all strain gauges at once e. Connect the leads from the amplifier chip to a microcontroller, and connect the microcontroller to the Arduino IDE to probe for signal status f. Ensure that different signals are being sent from the amplifier chip depending on how much force is applied downwards via the solid medium to the strain gauges

5 V DC Battery Power Source: The module on the chair will be powered by a separate 5V DC power source.
Requirements are the same as in 2.3.2 under '5V DC Power Source', except this one will be powered by batteries

Microcontroller: This microcontroller will poll the force sensing resistor to determine if a person is sitting on the chair.
Requirements are the same as in 2.3.1 under 'Microcontroller'.

Force Sensing Resistor: A force sensing resistor "pad" will be placed at the seat of a chair positioned for the table, and will be polled by the microcontroller to determine if the table is occupied by detecting if someone is sitting on the chair.

Requirement	Verification
Resistor must decrease in resistance when force is used to compress the resistor in order to perform as a sensor that detects the weight of a person	1. Connect a 5V power supply to a breadboard to supply power to it 2. Connect the 5V signal to a 1 kΩ resistor in series with the force sensing resistor. Place multimeter terminals in parallel with the resistor, and measure resistance. 3. Ensure that the resistance of the resistor decreases when force is applied to the resistor

Wireless Transmitter: A bluetooth module will be used to send a signal from the microcontroller on whether the seat is occupied or not based on the weight on the force sensing resistor
Requirements are the same as in 2.3.3 under 'Intra-table communication'.

2.4. Schematics

2.4.1. Height Adjustment

The height adjustment components of our design will be stored in the top of the table and in the base of the table. Figure 5 details where we plan to place each of the parts used for this function of our project. Our first plan to connect the table top and base is to send a long wire attached to the inside of our table, which will be able to collapse into the table base when the table lowers in operation. If this is not possible, we will keep this cable on the outside of the supporting beam of the table. In implementation for a real restaurant, wiring would be internal to protect from tampering.

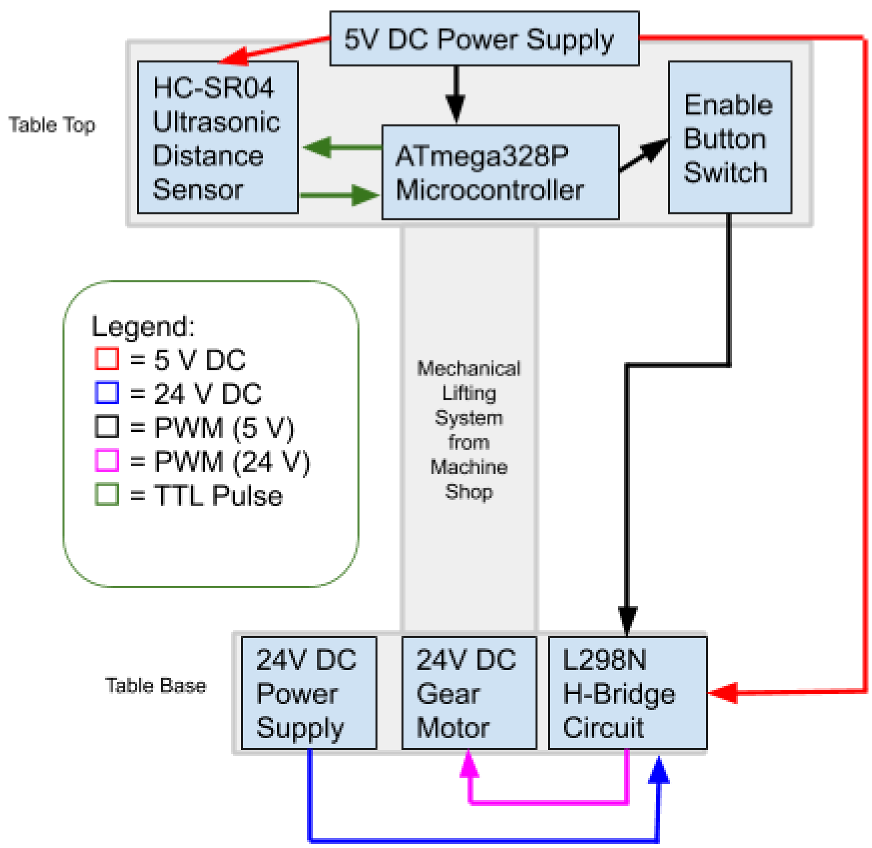

Fig. 5: High-level block diagram detailing the wire connections of each component of the height adjustment group

Due to time constraints and depth of our project, we will use a pre-manufactured H-bridge motor driver that can support a voltage of 24V DC in order to drive our 24V DC gear motor. Manufacturer schematics are provided for this circuit in a datasheet for the L298H motor driver from Handson Technology, shown in figure 6 [10]. In production, this circuit would be substituted for an H bridge chip and necessary power protection instead.

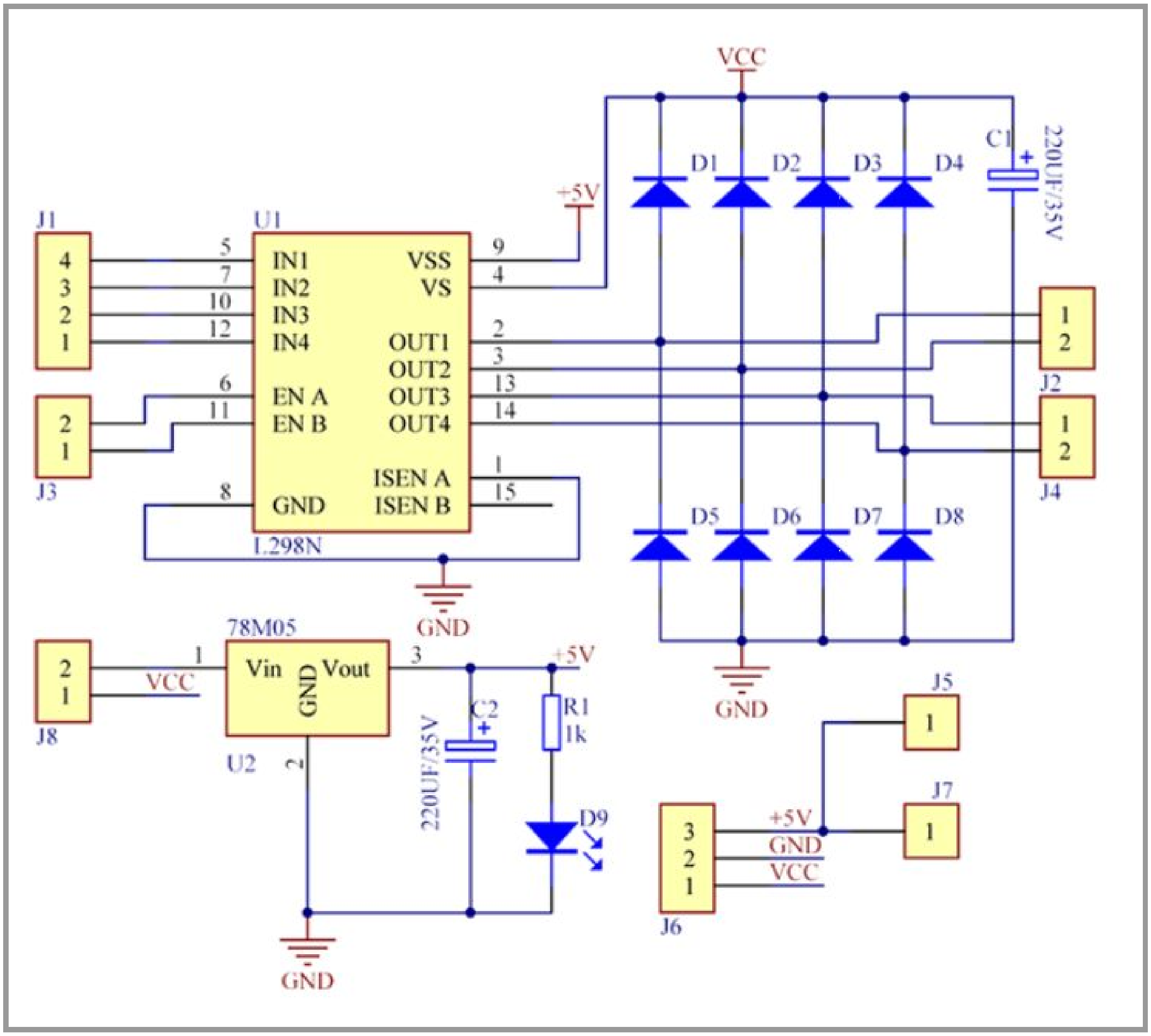

Fig. 6: Handson Technology Schematic for L298N [10]

It is important to note that although the voltage source shown in figure 6 is labeled "12 volts" on the physical board, this circuit actually accepts a power supply voltage from 5 V DC to 35 V DC, so this h bridge will drive our 24 V DC gear motor. Checking the power rating of the components in the h-bridge is paramount for driving a higher voltage motor like the one we will use for height adjustment.

2.4.2. Customer Presence Detection

The customer presence detection components of our table will be stored in the top of the table and in a separate module that will be placed on a chair. Figure 7 details the separate parts necessary to make this group work, where they will be placed on our built model, and how they will be physically connected. The 5V DC power supply shown in figure 7 is the same power supply that will be used for the height adjustment group as shown in figure 5.

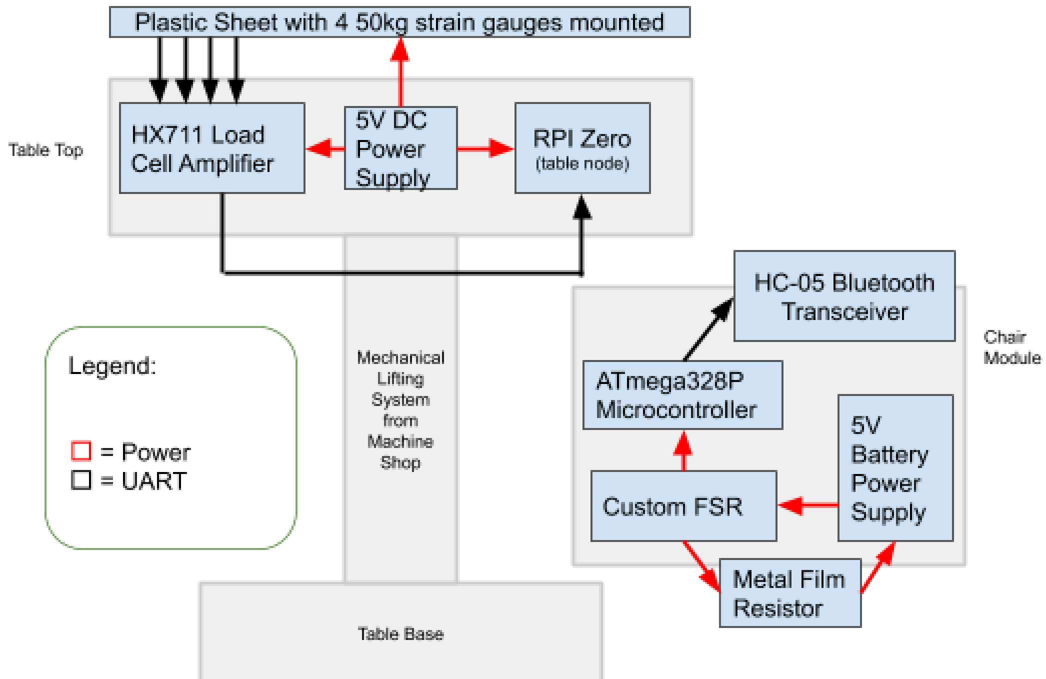

Fig. 7: High-level block diagram detailing the wire connections of each component of the customer presence detection group

Expanding on the chair module shown in figure 7, we will determine the resistance value of the resistor by testing the range that different metal film resistors provide. We plan to use a 1 k ohms resistor in series with our custom force sensing resistor to start, and if different metal film resistors provide more flexible results, we will swap the parts. The circuit we will use to determine which resistor value to use is shown in figure 8.

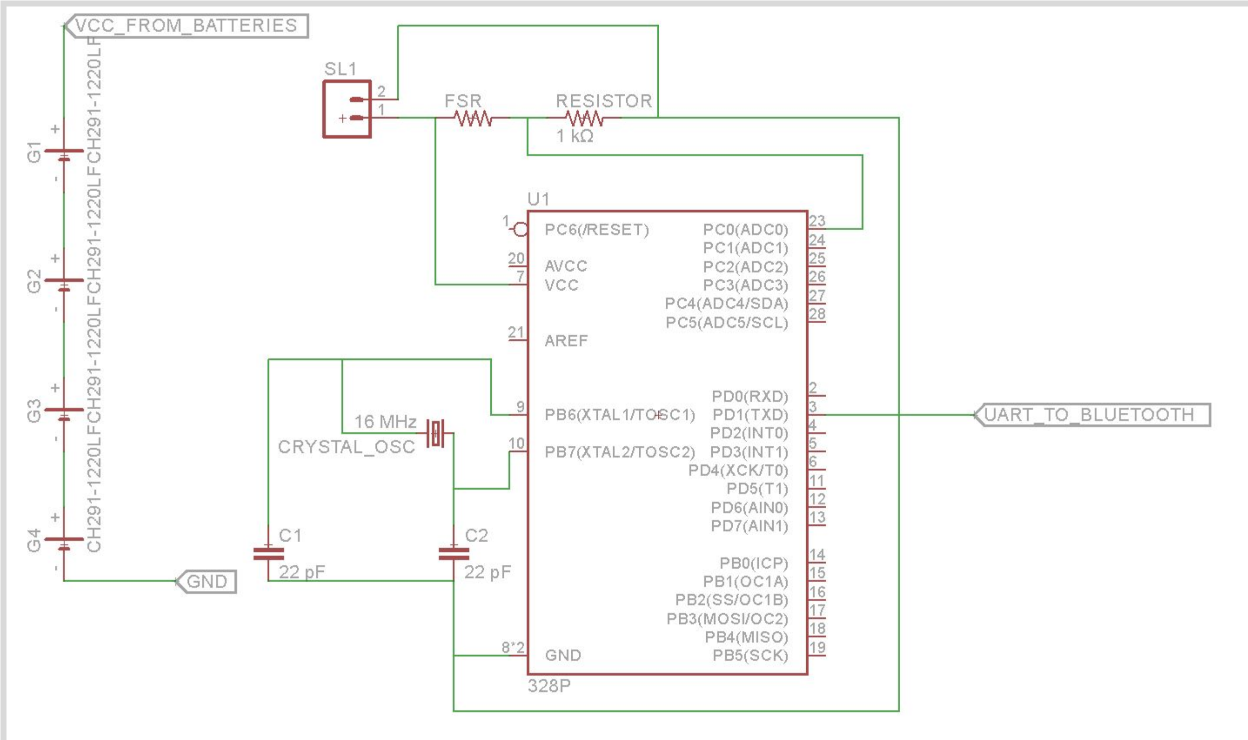

Fig. 8: Schematic for chair component of customer presence detection, using ATMEGA328 from microcontroller library from SPARKFUN Library [12]

2.5. Plot

Based on a publication that collected information on the speed of DC motors versus PWM duty cycles [13], we know that the relationship between the average voltage of a PWM signal and the speed of motors is not linear in practice. The graph from this publication is shown in figure 9. Since we have elected to use a high torque motor rated at a speed of 28 RPM, we expect our results to be scaled down accordingly. We expect results to be similar to this graph regardless of our choice of DC motor, and based on our own testing, we will seek a duty cycle that allows for our table to stay within our speed constraint requirement while avoiding disruption on the top of the table as much as possible.

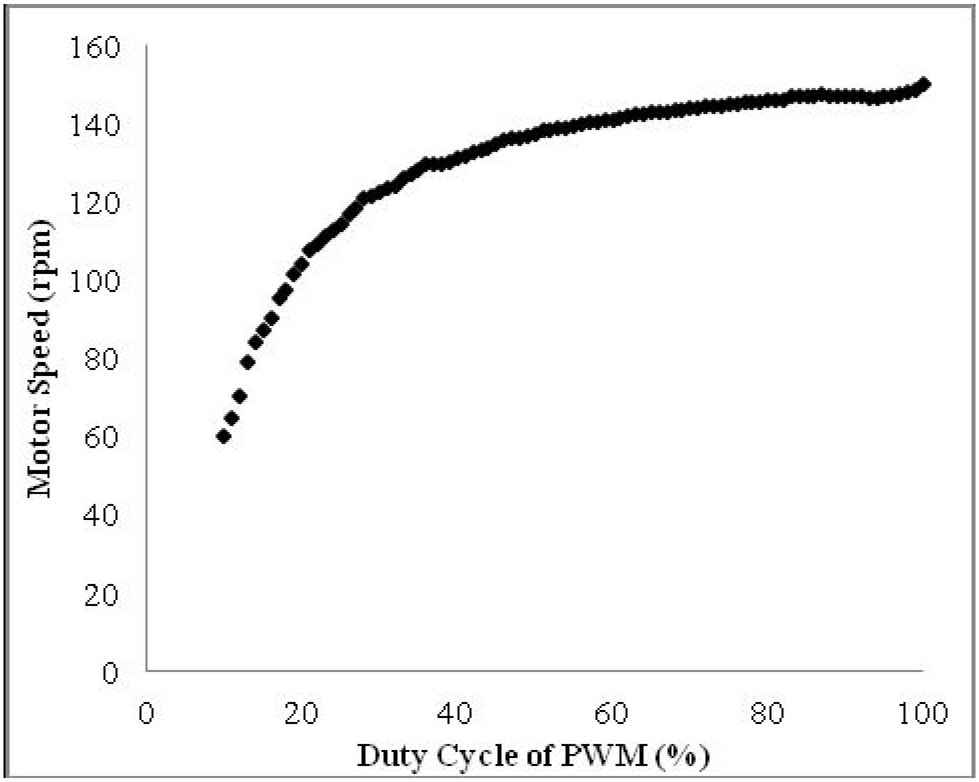

Fig. 9: Duty Cycle of PWM signal versus Motor Speed as found Calcium alginate microbead production research paper [13]

The plot shown in figure 9 is important because each DC gear motor will react differently, and unique measurements will need to be taken from any table that uses this design to tailor a PWM signal that works best for it. The other components, like the ultrasonic sensor, will have errors that may be mitigated through software solutions instead of relying on component performance.

2.6. Software Architecture

2.6.1. Algorithms for Hardware

2.6.1.1 Height adjustment

The height adjustment of our table will be driven by a microcontroller that polls an ultrasonic sensor and the table's height adjustment motor encoder. We will initially poll the sensor every second, but if this frequency proves to be not accurate enough, we will adjust accordingly. The microcontroller will keep track of the latest 10 distances polled, find the median of these numbers, then consider this median the distance to the object under the table. The height will adjust in response to the measured distance according to the following cases:

- If the number is more than 8 inches but less than 21 inches, to the nearest inch, the table will adjust downwards.
- If the number is less than 8 inches or more than 21 inches, the table will adjust upwards.
- If the encoder reports to the microcontroller that it is turning and the distance does not change for an entire cycle, the table will stop raising, as this suggests that the table is at either maximum or minimum height.

Our initial plan is to autonomously adjust the height of the table by using the ultrasonic sensor, but if the readings by the microcontroller are not accurate enough while moving, being off by more than an inch, we will use the motor encoder to assign a separate threshold to the microcontroller to use with the ultrasonic sensor while the table is adjusting.

Our table will start from its maximum height of 40 inches and adjust downwards until it reaches a distance of about 8 inches from the customer's legs, to the nearest inch. Allowing for this inch of error, the table will ideally be able to adjust to a leg height of at worst about 19 inches from the ground provided it does not stop moving downwards until it is about 7 inches from the top of the legs. Therefore, our planned threshold of 21 inches of distance comes from considerations that the minimum height of the table is greater than 21 inches tall, and at most the table will be able to accommodate legs 21 inches from the table's maximum height.

2.6.1.2 Customer Presence

The overall state of the table will be determined by two factors:

- Table-top Status -- Labels the table surface as clean or unclean.
- Chair Occupancy -- Labels the chairs at the table as occupied or not occupied.

A table will be declared as clean or unclean based on the weight of all the items on the surface of the table at any moment in time. When customers dine in, the food served to them along with the additional beverage glasses increase the overall weight, even after the customers depart. The table top will be measured from load cells (detailed in figures 7 and 8) on the top of the table with a threshold of 1 kilogram above where it is zeroed.

Chairs surrounding the table will each determine whether it is vacant or occupied via the use of a force sensing resistor (detailed in figure 3). These chairs will all forward their status to the table node. The chair module to declare itself as vacant if and only if *all* chairs at that table report that there is no customer sitting atop it for at least five minutes. This will minimize the false positive rate of declaring the table as having vacant seating.

A diagram of the states of these two factors and the overall state result is shown below. The overall status will be forwarded to the dashboard once per minute.

Table top	Chair module	Overall Table Status
Clean	Vacant	Ready for Customers
Clean	Occupied	Occupied
Dirty	Vacant	Needs Service
Dirty	Occupied	Occupied

Fig. 10: All possible states for a table to be in with the customer-desired state highlighted in green.

2.6.2. Table Network Architecture

2.6.2.1 Software Architecture

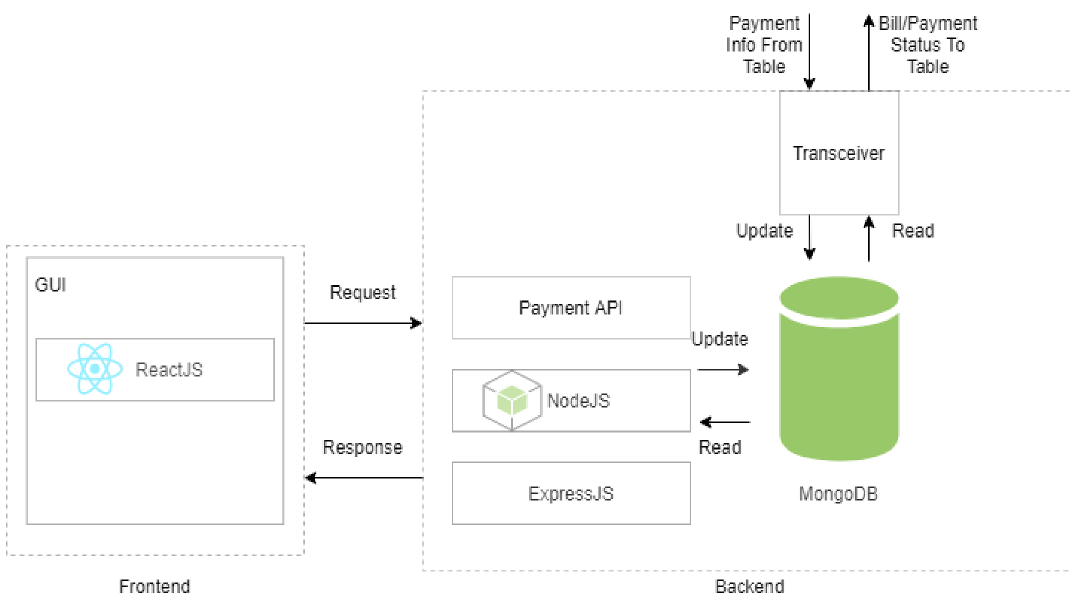

Fig. 11: Software Architecture Diagram

2.6.2.2 Database

For data storage, our backend will utilize a locally hosted MongoDB database which would be updated/read by a Node.js runtime environment and our table transceiver. The database will be running locally, as this will prevent downing when there is an internet outage. We need persistent storage to bookkeep real-time information of each table and it will be used to update the dashboard's frontend. Figure 12 details the data and data type we want in our data collection.

The table occupancy value will be one of the following:

- Ready for Customers
- Needs Service
- Occupied

Meanwhile the bill status will be one of the following:

- Assigned but not paid
- Assigned and paid
- No bill issued

```
{
    "id": <Int>,
    "occupancy_status": <String>,
    "bill_status": <String>,
    "bill_amt": <Float>
}

{
    "date": <Date>,
    "sales": <Array>
}
```

Fig 12. Collection Format

2.6.2.3 Dashboard

Our dashboard will be implemented in React.js and it will be updated in real-time by querying the backend every ten seconds. Restaurant employees will be able to upload an image of the dining area floor plan and mark where tables will be through an edit mode. The tables on the dashboard will change color depending on the status of the table's occupancy.

The color representation will be the following:

- Green - Ready for Customers
- Yellow - Needs Service
- Red - Occupied

They will also be able to tap on a table to view a textbox of information such as bill status and bill amount. Bill for selected tables can be inputted here to be sent for payment and then the bill amount will be submitted to our database.

2.6.2.4 Transceiver

Our project will employ an ad-hoc network that will allow all table nodes to communicate with a single coordinator as shown in figure 13 below. The table nodes and the coordinator will communicate by passing messages to each other via UDP sockets. The transceiver that exists on the coordinator will update the local database with table status updates it receives and will send out customer bills to assigned tables.

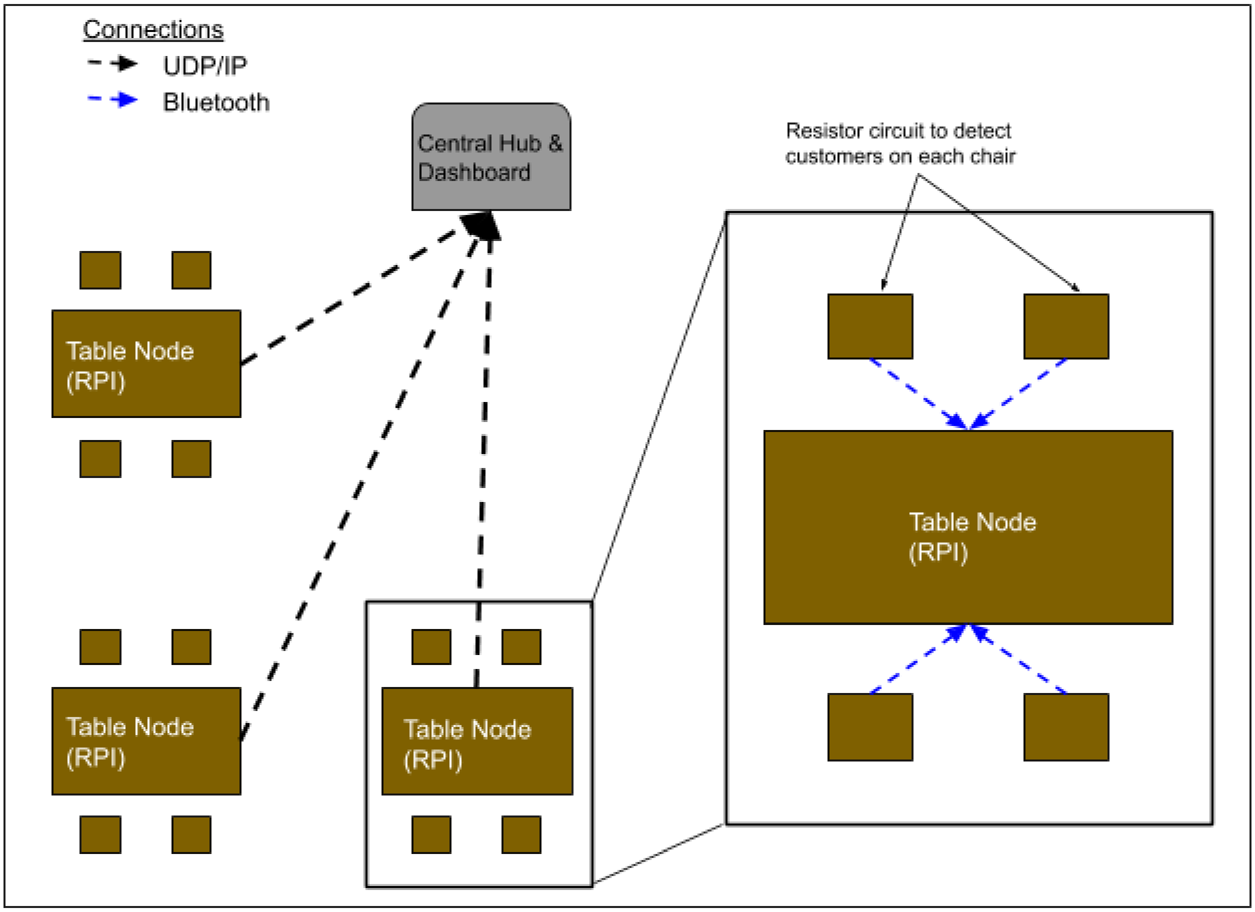

Fig 13. Network topology diagram.

2.6.2.5 Payment

The restaurant employees will be able to use the dashboard to initiate and assign bills to each table. This bill will show up as a dollar amount on a 7 segment display attached to each table. To pay for the bill, customers will use the NFC and Magnetic stripe reader connected to each table's raspberry pi zero. When a charge is initiated, a program using the libnfc library will read in payment information from the readers and forward it to the transceiver on the central coordinator over UDP. The payment info will be processed and validated by secure payment API's provided by Paypal. Once the payment goes through, an 'success' or 'failed' payment message will be updated in the database for the corresponding table and this message will also be forwarded to the customer as well.

The status representation on the display will be the following:

- Blank will imply that there is no bill
- Digits will display the billed amount when prompting customers to pay
- The display will display four zeros when payment is successful
- Flashing digits will imply a declined payment

2.7. Tolerance Analysis:

Height Adjustment

The main physical component of our project is the concept of a self-adjusting table. In implementation, the self-adjusting table will rely on an ultrasonic distance sensor to determine when the table needs to raise itself or lower. The mechanism that will allow the table itself to become taller or shorter will rely on a lead screw, mechanicalically, that is driven by a gear motor. The table needs to support a significant amount of weight. We aim to possibly lift up to 10 or 20 kilograms of mass, so we have elected to employ a high torque motor running at a comparatively high voltage to achieve this task. The motor needs to be slow enough to not noticeably hinder the performance of the ultrasonic sensor, so the motor we have selected has a low RPM rating.

The most critical aspect of our project's performance is for the table to be able to adjust on its own based on customer leg height under the table. For our tolerance analysis, we will investigate the ultrasonic sensor that is providing the height group's microcontroller chip with information, which will ultimately lead to the signal fed to the motor.

The goal of our height adjustment system is to be within an inch of the desired height above potential customers' legs. Considering this accuracy, it would be useful to be accurate with the ultrasonic sensor to the nearest 2 centimeters. The sensor should be able to measure to at least 40 inches, the maximum height of our table, or approximately 102 centimeters.
We are using the HC-SR04 ultrasonic sensor for our project. According to its datasheet, the sensor can read distances from 2cm to 400cm with a ranging accuracy of 3mm [8]. This distance measurement is under ideal conditions, so objects like fabric and the angles caused by clothing may affect the accuracy of this measurement. The temperature of the room and noise from other signals in the room may also affect the sensor.

The distance from our ultrasonic sensor is determined as our microcontroller prompts the sensor to output an 40kHz burst of 8 signals. This signal is prompted by the TRIG pin of the sensor. The ECHO pin of the ultrasonic sensor reports the time it took to receive any signal sent as a result of input to the TRIG pin, and this will be polled by our microcontroller in order to calculate the distance from the table top to the legs of anyone sitting in a chair seated at the table. We may calculate the distance from the table top with the equation in equation 1.

$$Distance = \frac{Duration\ from\ ECHO\ pin\ in\ \mu s \bullet Velocity}{2}$$

{1}

We need to divide by two because this duration involves sending a signal, anticipating the signal bouncing, and waiting for the signal to traverse back the distance. Since the duration will be given to the nearest µs by the sensor, we simplify this equation in equation 2.

$$Distance = \frac{Duration \bullet 340\ \mu m/\mu s}{2} = \frac{Duration \bullet 0.034\ cm/\mu s}{2}$$

{2}

By default, the sensor uses a pulse that lasts 10 µs, of which time the ECHO pin is not active. Adding to that one µs due to possible error from rounding to the next µs, we calculate the tolerance of our sensor's calculated distance in equation 3.

$$Distance = \frac{11\mu s \bullet 0.034\ cm/\mu s}{2} = \frac{0.374\ cm}{2} = 0.187\ cm$$

{3}

This error is within our accepted tolerance of error to the nearest inch. The actual error from the ultrasonic sensor is expected to be worse due to the moving tabletop and environmental factors such as temperature, noise, and materials the signals bounce off.

Table Sensing

There are a number of delays and possible areas for error when maintaining the status of the table. Starting with the tables themselves, we declared that a table would mark itself as unoccupied if and only if the surface ia not weighed down with too many excess objects (uncleaned food) and if *all* chairs were empty for at least five minutes.

The load cell for the table has a tolerance of 1 kg more than the “empty” weight (the weight we start at when there are no extra items on the table and we zero the load cell), so there may be a false positive if extra items on a clean table (such as extra menus) are left on that table. Should an error arise, it should be quite clear that any possibility of error will likely be falsely marking the table as occupied as opposed to vacant. We believe that it would be better to make this mistake as opposed to the other way around, otherwise two customers may try and be seated at the same table which would be a harder conflict for restaurant staff to resolve than simply realizing the table was not properly cleared, thus blocking them from seating customers there since they would assume it really is occupied.

Chairs use a force sensing resistor as discussed above that will assume that there is no person sitting atop of it if there is less than 20kg of weight over the sensor. It was also discussed that the table would not declare all the tables as unoccupied unless *all* chairs report that they are unoccupied for at least five minutes to allow for customers to get up and use the restroom if needed. This introduces a bit of uncertainty in measuring, however, should one or more chair sensors fail.

Since hardware can never always be 100% reliable in sensing the table and chair weights, we can estimate the probability of the table marking a false positive based on the probability of hardware failure on the two parts as:

$$P_{table} = P_{load\ cell} + P_{chairs}$$

Where:

- P_{table} is the probability of the table falsely declaring the table as any state but 'Vacant'.
- $P_{load\ cell}$ is the probability that the load cell falsely declares it is occupied when it is not.
- P_{chair} is the probability that at least one chair falsely marks itself as occupied for more than five minutes.

The last point of failure that can occur when monitoring the table status is when the table sends its status over to the dashboard. The dashboard will poll the table appropriately once per minute since table statuses likely will not change too frequently. This rate can be adjusted in between development and deployment to production as the use cases change. By polling only once per minute, we reduce the possibility of noisy inputs that the table may send over.

3. Cost and Schedule

3.1. Labor Cost Analysis

Our labor cost estimate for the development of this project is approximately $37.50 an hour, the median hourly wage for a new Electrical Engineer graduate, 10hr/wk for 3 people for 16 weeks.

$$3 \cdot \$37.50/hr \cdot 10hr/wk \cdot 16wk \cdot 2.5 = \$45,000$$

3.2. Cost of Parts

Name of Part	Cost
24V High Torque DC Gear Motor	$32.99
Ultrasonic Distance Sensor	$8
Breadboards	$9
L298N H Bridge	$6.69
Large Safety Button	$8
24 V DC Power Supply Unit	$11.97
2x 4-slot AA Battery Packs	$6.99
2 sheets Antistatic Semiconducting Foam	$14.99
Roll of Copper Tape	$7.99
Roll of Gaffer's Tape	$7.97
Arduino Uno IDE + ATMEGA Microprocessor combo	$22.79
Arduino Bluetooth module	$12.99
USB NFC reader	$42.75
USB Magnetic Stripe Reader	$15.99
MicroUSB to USB Port Hub	$6.99
4-Digit Tube LED Segment Display	$5.49
Raspberry Pi Zero (2x)	$14.00 x 2
Total Cost of Parts	**$250.59**

Combining the price of labor and parts, this project is worth about $45,250. The majority of the material part cost can be decreased by purchasing the materials in bulk and designing fully proprietary circuits by making one's own version of the bluetooth module, H bridge, and 24 V DC power supply.

3.3. Schedule

	Andrew Chen	Eric Ong	Can Zhou
09/27	Continue working on design document, start proto-typing ad-hoc network.	Touch base with machine shop on final design, build chair FSR	Continue researching on what api to use for payments, also learn react.js
10/04	Test bluetooth wireless module to communicate with a raspberry pi and send data through a one-way connection.	Wire power supply, collect information on the ultrasonic sensor	Start building the payment api for our web server.
10/11	Write senders and a receiver for table nodes to send data to the central coordinator with just mock data.	Consider making PCB for H-bridge system, test operation of motor system	Start building the frontend and have multiple design for feedback
10/18	Integrate and test with Eric's table data and prototype the 7-segment display for the table.	Work with Andrew on getting table status from chair module	Finishing the frontend and backend for payment
10/25	Coordinate with Can on database schema and API's and deploy database to integrate with frontend.	Calibrate top of table to weigh items placed on top properly and report status to table hub	Integrating the payment api with our frontend and database
11/1	Assist with any hardware/software integrations/debugging.	Tighten algorithm for the self-adjusting module	Testing the software And assist with any hardware integration.

4. Ethics and Safety

There will be circuits and moving parts in our table, so it is necessary to address safety concerns to protect the internals of our device and shield users from danger. In use, this project's height adjustment aspect will involve interaction with motorized table legs. This moving part necessitates care for both safety and durability of the constructed prototype and any models based on it. One specific safety concern lies in the possibility that the motor will possibly injure customers due to unintended performance due to glitches in the algorithm or inaccurate readings from the sensor used to adjust the table height. To mitigate this, we have included a safety button on the table that may be used to halt the table's adjustment. There will also be manual buttons in the table to adjust the table position, which will allow for adjustment out of any positions that may be considered safety concerns. This addresses IEEE ethic code II.9, as these are measures in place "to avoid injuring others". [9]

The high voltage supplied to the DC motor and related h-bridge pose a separate safety concern. The table will be supplied 24V and 5V power supplies with currents that could potentially be fatal if discharged through the heart. All terminals with different voltages will be contained within the table to prevent interaction with any components that could potentially discharge electricity.

In addition, we will be using NFC/Magnetic Stripe payment for users to pay for their services. Therefore, it is essential that we can securely process a user's card payment to protect it from being accessed externally. This addresses IEEE ethic code II.1, "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices" [9].

There are other safety concerns to consider for our design if it is used in practice. Since this product is intended for restaurant use, water and food will come in contact with the table often. The table requires measures to prevent water from coming into contact with its internal electronic components. We are using sensors throughout our table, so it is necessary to maintain these sensors or know when one fails, as errors in the table could lead to physical risks with the moving parts. Information from the sensors on our table must be maintained because its accurate performance is required for both the operation of the table and the network the table is connected to.

We will test our design intensively. This will prevent potential safety hazards and repeated incidents. We will also strive to consider feedback from others on how to improve the safety of our design. This addresses IEEE ethic code I.5, "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest" [9]. Safe operation of our design is of prime importance for integration into the restaurant industry.

5. Contingency Plan

In accordance to the new Fall 2020 course policies, this contingency plan will detail how the project will change in scope and deliverables in the event that the University of Illinois at Urbana-Champaign shuts down due to the COVID-19 pandemic.

Design Changes

On the occasion that the University closes, students will lose access to the machine shop in ECEB. This would impair our ability to create a physical table for the final demo, and so a table will not be used to demo our project. The circuits and subsystems inside will still be available to demo individually.

1. ***Self-Adjusting Table Height Subsystem:*** The self-adjusting table height subsystem will no longer modify the height of the table based on distance polled beneath the table. Instead, a proof-of-concept circuit will be developed that will power a gear motor to adjust the height of a nut attached to a lead screw. This mechanical system is the basis of the physical height adjustment of our table. For the physical implementation of the height system, the nut will bear the weight of the top of the table, so showing the position of the nut will be sufficient to demonstrate the full capability of the height adjustment system. The motor will raise or lower the nut based on the input of an ultrasonic sensor, which will be as our original design intended, but they will no longer be embedded inside of a table.

2. ***Customer Presence-Sensing Subsystem:*** Force sensing resistor for sensing customer weight on chairs will be less robust, built using available foil rather than copper tape, and have a less tuned threshold. The remaining microcontroller reading this input and sending to the table unit will remain the same.

3. ***Table LAN Subsystem:*** LAN subsystem will stay largely the same. Table units will communicate and interact with each other using the bare hardware with no tables to store them.

4. ***Payment Subsystem:*** Payment subsystem will stay largely the same. Payments will be demoed using a payment module and sent to the table unit using the bare hardware with no tables to store the physical system.

5. ***Dashboard Subsystem:*** Dashboard subsystem will remain unchanged.

6. Citations

[1] Yelp, ‘Increased Consumer Interest in May Correlates with COVID-19 Hot Spots in June, According to the Yelp Economic Average‘, *Yelp*, 2020 [Online]. Available: https://www.yelpeconomicaverage.com/yea-q2-2020.html [Accessed 17-Sep-2020].

[2] J. LongStreet, ‘Pennsylvania restaurants need 50% capacity to reopen’, *The Philadelphia Inquirer*, 2020 [Online]. Available: https://www.inquirer.com/food/indoor-dining-philadelphia-pandemic-coronavirus-capacity-restaurant-week-20200915.html [Accessed 17-Sep-2020].

[3] PYMNTS, ‘Restaurants That Aren’t Ready For A Mobile Future Risk Losing Out‘, *PYMNTS*, 2020 [Online]. Available: https://www.pymnts.com/news/mobile-payments/2019/restaurants-mobile-ordering-food-delivery-ritual/ [Accessed 17-Sep-2020].

[4] MRM Staff, ‘The Future of Dining: Industry Expert Insights‘, *MRM*, 2020 [Online]. Available: https://modernrestaurantmanagement.com/the-future-of-dining-industry-expert-insights/ [Accessed 17-Sep-2020].

[5] CDC, 2020 [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/community/organizations/business-employers/bars-restaurants.html [Accessed 17-Sep-2020].

[6] R. Sutton, ‘NYC’s Indoor Dining Comeback Fails Restaurant Workers. Here’s Why’, *Eater*, 2020 [Online]. Available: https://ny.eater.com/2020/9/16/21437886/indoor-dining-workers-regulations-cuomo-nyc-restaurants [Accessed 17-Sep-2020].

[7] R. C. Schafer, ‘The Art of Chiropractic Adjustment: Part II‘, *Dynamic Chiropractic*, 1991 [Online]. Available: https://www.dynamicchiropractic.com/mpacms/dc/article.php?id=44159 [Accessed 17-Sep-2020].

[8] Elecfreaks, Ultrasonic Ranging Module HC-SR04 Datasheet, Sparkfun [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf [Accessed 17-Sep-2020].

[9] IEEE, ‘IEEE Code of Ethics’, *IEEE*, 2020 [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed 17-Sep-2020].

[10] Handson Technology, *L298N Dual H-Bridge Motor Driver*, 2012 [Online]. Available: http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf [Accessed 30-Sep-2020]

[11] ECE Illinois, ‘Salary Averages’, *ECE Illinois*, 2018 [online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages [Accessed 28-Sep-2020].

[12] Sparkfun Electronics, *Github* 2020 [Online]. Available: https://github.com/sparkfun/SparkFun-Eagle-Libraries [Accessed 28-Sep-2020].

[13] J. Teknologi, ‘Development of a flicking system for producing calcium alginate microbeads‘, 2016 [Online]. Available: https://www.researchgate.net/publication/303887514_Development_of_a_flicking_system_for_producing_calcium_alginate_microbeads [Accessed 29-Sep-2020].