Non-Contact Interaction with Common Areas Team 2
Final Report
Fall 2020/Spring 2021

Team Lead:
Connor Wilson- connormcintyrew@gmail.com, (303) 437-2926
Team Members:
Jonathan Barstad- jonathan.e.barstad@gmail.com, (303) 913-8554
Andrew Kwavy- andrew.kwavy@gmail.com, (720) 645-9613
Patrice Ouedraogo- patrice.ouedraogo@ucdenver.edu, (347) 371-7268
Alex Quach- alex.quach@ucdenver.edu, (720) 757-8430
Sandeep Singh Rai- singhrai.sandeep@gmail.com, (720) 431-3447

Project Description:

Due to the urgency at hand with the ongoing COVID-19 pandemic, practical solutions to virus transmission reduction are essential to the return of a normally functioning society. The pandemic has presented a very large number of challenges, and it has been this group's task to develop technology capable of reducing viral spread. More specifically, reducing/eliminating virus transmission through physical contact is the focus of this project. This has led to the development of touchless technology capable of eliminating physical contact with commonly touched objects. Last semester, two common areas of interest were selected to focus on, using extensive ideation and selection methodologies. The selected areas of focus are elevators and shared workspaces. Through the use of multiple design processes, two viable solutions have been successfully prototyped: a touchless elevator panel and a "Smart Desk" user detection system. The ideation and down selection processes used are thoroughly outlined later on in this report.

System #1- Touchless Elevator Panel: The touchless elevator panel uses the basic design of a standard elevator panel, but with one major difference. Instead of using a typical elevator button for operation, the touchless elevator panel implements infrared break-beam sensors. These break-beam sensors output an activation signal when broken, and the activation of these sensors is completely contact free. These sensors are present in a recessed cylinder, along with physical buttons at the end of each cylinder. These physical buttons are intended to operate exactly the same as current elevator buttons, although the prototyped solution works slightly different. These physical buttons allow for a physical backup in the event of sensor failure. The prototype of this panel uses an Arduino Nano to interpret the high output signals produced by both the sensor and button, and is also responsible for activating the corresponding LED confirmation.

System #2: Smart Desk System: The smart desk system gives shared workspace users a reliable indication of whether their potential area is clean or dirty. A Jetson Nano is implemented with a camera to provide real time status updates to each desk, through the use of computer vision and AI technology on a live video feed. To provide status updates to users in the area, each workspace has its own status display system consisting of three universally understood icon cut-outs. These icons are backlit with LEDs that are controlled by a Raspberry Pi Zero W, that wirelessly communicates with the Jetson Nano to receive status information. The four potential desks statuses are: 'clean' (shown with a green check mark), 'in-use' (shown with a blue cycle symbol), 'paused' (shown with a blinking blue cycle symbol), and 'dirty' (shown with a red X). Each display unit also contains a depressed cylinder on the side with a break beam sensor. This sensor is used to pause the desk status when in-use if the user wants to step away from the desk without triggering a dirty status. The sensor is also used to reset the desk status from dirty to clean and is intended to be activated after the user or cleaning crew cleans the desk.

Stakeholder Interaction Phase:

Stakeholder Identification and Interaction

The team first initiated contact with stakeholder group representatives via email to schedule convenient meetings via email, in person, over zoom, or over the phone depending on preference. The main stakeholder groups identified include the Air Force Research Laboratory (project sponsor), the Auraria Campus Police Department (project sponsor), the medical community, the technology industry, the service industry, CU Denver professors, staff, and students, as well as educational institutions all throughout the country. Each of these stakeholder groups are currently severely impacted by the pandemic and can all benefit from the decrease in virus transmission.

The team then focused on attaining valuable information by using five main questioning categories: Who Are You, Job/Task, Story, Likes/Dislikes, and Feelings. These categories were selected because they cover a broad area of exploration needed to better understand customer's needs and expectations. By identifying

and understanding stakeholder needs and expectations it was easier to address the objective from the right angle and initiate the process in the right direction.

Results from Stakeholder interaction

After speaking with many of the identified stakeholders, a commonality arose between some answers given. By analyzing the results from speaking with stakeholders, a compiled list of important topics was created, and is shown below.

- People feel relatively safe, but still try to touch as few objects as possible
- Stable and consistent methods of interaction in public areas are needed
- Reduction of physical contact with public objects can reduce the spread of viruses
- Constantly changing social dynamics creates stress on everyone involved
- Sanitizing and cleaning commonly touched items can help to help prevent transmission
- Safe and reliable technological alternatives to human contact, that are easy to use and implement in existing structure

Customer Needs Identification and Organization Phase:

Context Scenarios

In order to better understand and identify potential areas of interest where touchless technology can be most beneficially implemented, several different scenarios were developed. The following six scenarios effectively display the wide application range of touch-less technology in the modern world. Although each of these scenarios are different, the many of the same common ideas apply throughout many of them, suggesting that some solutions have a large amount of implementation opportunity.

Scenario 1: Going to the grocery store	
Scenario Steps	**Possible Solutions**
1) Consumer walks into grocery store	
2) Consumer physically grabs cart and fills it up with products	Touchless cart system, touchless system for selecting products
3) Consumer either pays with self-checkout or with a clerk	
3a) Self-checkout: Consumer touches same buttons as other customers	Touchless method of payment/verifying card
3b) Clerk checkout: Clerk possibly handles cash and/or consumer touches keypad that others have used	Touchless exchange of cash, or touchless way of paying with card and verifying

How:
- Touchless keypad allows contactless verification of card
- Used every time a card transaction occurs, so frequently

Where:
- Gas stations, grocery stores, other in-person stores
- Where in-person monetary transaction occurs

Who:
- Customers
- Store employees/owners

Scenario 2: Going to an office workplace	
Scenario Steps	**Possible Solutions**
1) Employees must badge in and enter a pin code for every door they go through.	- Make badging in contactless by just scanning the badge through a chip reader device
	- As for the pin code, make the panel touchless by using a motion sensor for each button (digit)
	- After badging in successfully, the door will open automatically so avoid touching it
2) Employees share computer workstations with others when they change shifts	- Incorporate standing desks as an option for employees so they don't have to share the same seat
	- Utilize wireless virtual keyboard and issued individual wireless mouse.

How:
- Touchless chip reader for badge
- Touchless motion sensor keypad for pin code.
- Door that opens automatically
- Standing desks

Where:
- Office workplaces

Who:
- Employees

Scenario 3: CU Denver Student going to work and to eat in a public space	
Scenario Steps	**Possible Solutions**
1) Student takes classes only online but goes to work on campus	
2) Student drives to school because she does not want to use public transportation.	
3) Student walks to the school building and opens the door manually	- Door can be automatic
4) Student works on a school computer, touching the mouse and keyboard	- Utilize wireless virtual keyboard and issued individual wireless mouse.
5) Student goes to eat and drink with friend/coworker, pays with cash and a waiter delivers the food	- Robot can take order using a QR reader, accept payment (cash or card), and bring food to reduce total touches overall

How:
- Automate doors, elevator buttons, etc.
- Autonomous robot to take order, payment, and bring food/drink to reduce touch/contact

Where:
- School
- Workplaces
- Eating establishments

Who:
- Students
- Employees
- People who eat out

Scenario 4: Consumer going to a liquor store	
Scenario Steps	**Possible Solutions**
1) Consumer touches the door handles while entering the store	- Make doors automatic which avoids contact for entering the store
2) Consumer opens the fridge to grab a drink	- Automatic door for the fridge which uses motion sensor to open
3) Consumer brings the drinks to register, and employees make contact to scan the items	- Automatic scanner where consumer can scan the drink, which in turn has less contact with employees
4) Employees taking the card to physically enter the price and scan the card	- Cheaper software upgrades for the computer system which can use automatic card reader avoiding unnecessary contact
5) Employees stocking the drinks physically, touching every bottle in the cooler	- Smart cooler which can stock automatically using sensors

How:
- Smart door's which open only one door when you wave at them that uses sensors
- Weight sensors to stock the drinks in the cooler

Where:
- Liquor stores
- Grocery stores

Who:
- Employees
- Consumer

Scenario 5: individual visiting someone in the hospital	
Scenario Steps	**Possible Solutions**
1) Gain entrance to the facility	- Instead of everyone touching the same door handles, an automated opener is used (already commonplace)
2) Use the elevator to reach the proper floor	- Instead of a call button, a camera utilizing computer vision could detect when someone stands still, and call an elevator automatically
	- Instead of pressing physical buttons to select the destination floor, a voice-controlled system could be utilized for selecting the floor
3) Check in with the front desk.	- Rather than filling out any forms on paper or having close contact with a person at a desk, RFID technology could be used to link the visitor's own smart phone to a display on the wall or a stand, using the visitor's phone as a remote controller for interacting with the display.
	- If a visitor does not have a smart phone, a sanitized loaner device could be issued and returned before the visitor leaves
4) Enter room the patient is in	- Doors can be secured until a person presents identification to an RFID scanner (badge for employees, cell phone for visitors), and then open automatically
	- Computer vision technology could be used to detect when someone is waiting to enter rather than just walking by
5) Eating from the cafeteria	- An automated vending system could be used, with the remote cell phone as an interface
How: - Computer vision to detect intent rather than just motion - RFID technology to link personal devices and avoid communal use of different surfaces and minimize person-to-person interaction	
Where: - Hospitals - Doctors' offices - Other medical facilities - Anywhere that requires forms be filled out	
Who: - Medical employees - Visitors to hospitals - Patients attending appointments	

Scenario 6: Going home to an apartment high-rise	
Scenario Steps	**Possible Solutions**
1) Resident arrives at apartment complex, opens front door	- Hands-free door that utilizes computer vision or sensor technology
2) Resident goes into the mail room, opens mailbox with key and retrieves mail	- Hands-free locking/unlocking system utilizing smart phone technology
3) Resident walks up to elevator doors and presses the call button	- Touchless detection device instead of physical buttons
4) Resident steps into elevator and presses floor button	- Touchless detection device instead of physical buttons
5) Elevator reaches floor, resident steps out and walks to apartment door	
6) Resident unlocks apartment door and is now home	- Touchless door locking/unlocking and opening system
How: - Touchless buttons - Touchless doors - Touchless locks	
Where: - Apartment complex	
Who: - Residents - Guests - Utility workers - Food delivery	

Affinity Analysis

Affinity Analysis is a technique which allows the identified customer needs to be easily organized into categories and ranked in order of importance. The categories identified include cleanliness, social conduct, accessibility, efficiency, security, and automation. With help of these categories, one can clearly see that there are a lot of factors involved when reducing or eliminating physical interaction with commonly touched items. Some of the most important customer needs identified include:

a. Efficient cleaning
b. Sanitization of objects
c. Reduction of COVID-19 spread
d. New procedures and methods which are convenient and friendly

e. Safe way to enter buildings
f. Utmost safety for users as well as automation

Below is a summarized version of the affinity analysis in which customer needs were organized and ranked in in their respective categories.

Cleanliness	**Social Conduct**	**Accessibility**	**Efficiency**	**Security**	**Automation**
Clean shared spaces	Reduction of COVID transmission spread	Safe & convenient way of entering building/room	Proposed system needs to be more efficient & effective than current procedures	Ensure the safety & well-being of the customer	As much automation as possible, allowing for minimal human error
Efficient & reliable cleaning and sanitizing	Promote social distancing while maintaining business revenue	Provide extra care and support for the elderly and vulnerable people to COVID	Proposed technology must be cost-effective	Technological security	Minimal customer requirement
Eliminate touching of shared public objects	Make new procedures convenient and friendly as much as possible so people don't get agitated.				

Figure 1: Affinity Analysis

Functional Decomposition

Functional decomposition was used to identify necessary functions that must be included in the proposed technology to meet the demanded requirements. Some of these functions were derived from the affinity analysis. The decomposition was split into three different categories: pre-operation, operation, and post-operation.

The pre-operation phase is focused on initial contact with the user, as well as receiving information. During the operation phase, the focus is directed execution of the desired task. The post-operation phase focuses on user confirmation and returning to the pre-operation state. These three categories may contain additional functions that need to be incorporated for successful operation. Below is the broken-down functional decomposition:

Pre-Operation Phase:

- User detection / Receive information (Audio input, Optical input, other sensor input (weight, sonic, etc.), touch (non-human) input)
- User engagement (Display information, optical/audio output (for user))
- Information storage (computing electronic components)

Operation Phase:

- Process information (what action is requested/required from user and environment input)
- Communicate/interact with desired object
- Perform requested operation (perform operation allowing for touchless interaction between object and user)
- Perform efficiently and effectively (little/no delay, operation successfully executes without direct human contact)
- Assess execution of operation

Post-Operation Phase:

- Display information (optical/audio output (for user))
- Return to pre-operation state
- Be ready for next user engagement

Journey Map:

The journey mapping tool was utilized to help identify where the design concept should be incorporated. This journey map incorporates 3 similarly structured day-to-day activities, going home (an apartment in this case), going to an office building, and going to a grocery store. The Journey Map shown below displays the current manner in which these activities are performed. Analyzing this Journey map has allowed for the easy identification of areas of interest, where the proposed touchless technology can be most beneficially implemented. There are many touch points on this journey map, which can be seen with the hand icon. These are points where human contact is currently necessary for the completion of the task. There are also many insights which point out innovation opportunities, and others that describe the general feelings that the user may be experiencing while completing the certain task.

The zoomed in portion of the journey map shown below illustrates the user in transit in the "going to work" scenario. Touch points have been added to indicate that physical input by the user is required at the corresponding step, which is where introduction of touchless technology should be explored. The channel points show where a flow of information occurs. Emotion indicators have also been added where necessary. Key insights of the interaction process are also included. The journey map made it clear that door handles, elevators, shopping carts, and shared workspaces (desks) would be excellent targets to incorporate touchless technology into. Both of the developed design concepts were inspired by this Journey Map.

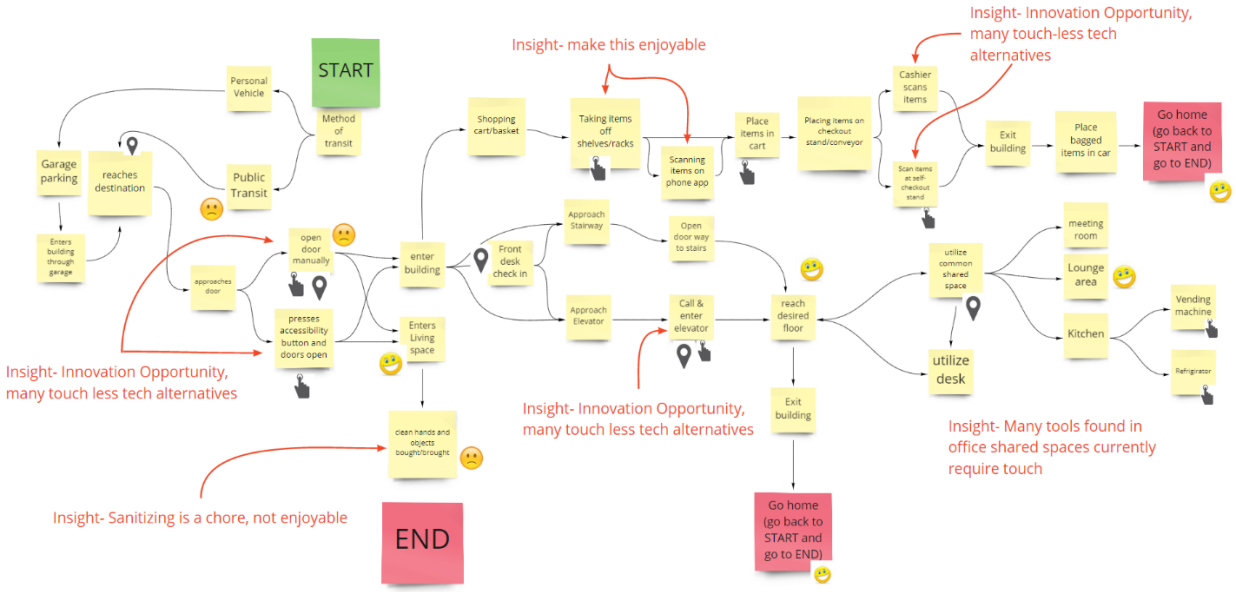


Figure 2: Journey Map

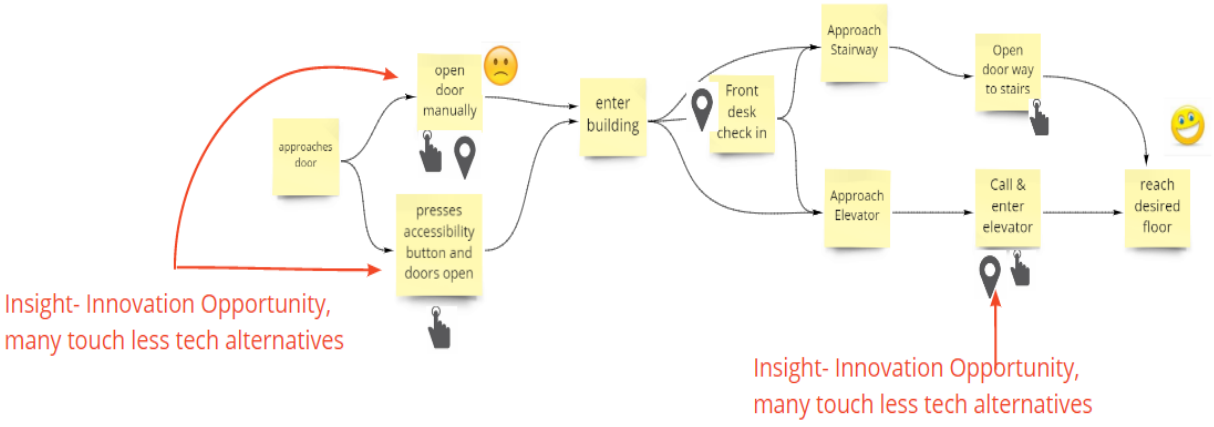


Figure 3: Focused Journey Map

Ideation Phase:

Mind Map

The Mind Map method was also used which is a great brainstorming exercise. Each team member individually spent about 10 minutes to come up with as many ideas as possible. All of the generated ideas were then merged into one mind map with many different categories. The finalized mind map illustrates a broad array of ideas that address the objective from different angles and at different levels, which has helped to reveal solutions that were not initially thought of. Ideas obtained from this Mind Map that were to be potentially incorporated into the design concepts include:

- Signal Detection
- Motion sensors
- RFID (Radio Frequency Identification)
- Computer Vision
- Object recognition
- Gesture recognition

Design by Analogy

Design by analogy was another method used for generating more unique, novel ideas. This was done by coming up with keywords related to the refined opportunity statement and entering these keywords into wordvis.com and asknature.org. Wordvis.com returns similar words, which can provide a different perspective on the opportunity statement overall. Asknature.org returns wildlife-focused articles with information relevant to the keyword, which helped with gaining new perspectives and ideas based off naturally occurring systems. New ideas gained from this method can be seen in the mind map below, where these novel ideas are marked with a red dot. Below is a diagram illustrating the final mind map with design by analogy ideas incorporated, as well as list of the ideas generated in this method and the reasoning behind them.

Figure 4: Mind Map with Design by Analogy Additions

WordVis.com Results:

Handiness – Led us to idea that we can connect everyday devices (smart phones) with technology points (keypads, door operation)
Availability – Create technology that is easily available to a wide number of people. Since the majority of the public already own smart phones, we thought it would be smart to implement the use of smart phone applications into our project.
• Idea – Smart phone app integration.
Skillfulness – Proposed technology needs to cater to all technological skill levels, especially low skill levels. Technology should not require much skill to operate successfully. The key insight of this idea is that the user will be minimally inconvenienced throughout the operation and can successfully be completed with minimal training or skills.
• Idea – Sensing technology capable of operating without the user being aware of the technology.
Modernize – Update antiquated technology that is already in use. This includes the idea of adding new technology on top of existing technology, therefore updating the system overall.
• Idea – Retrofit existing technological touch points (Accessibility button, Elevator buttons, Keypads).
Overhaul – Similar to modernize, but instead of adding on to current technology, we completely redesign system technology, functions and operation. This allows fundamental improvements of the technology in use but can also be much more costly comparted to retrofitting.
• Idea – Complete system redesign of existing technological touch points (doorways, elevator panels, payment points, etc.)

Asknature.org Results:
Article 1: The topic of this article is how innovators used swarm logic to connect their technology to reduce energy use
• Idea – Swarm of sensors for doors that track movement through buildings and open doors ahead of individuals based on movement patterns rather than waiting for individual sensors to pick up on movement. Could also track/predict destinations and spray room/office ahead of time.
Article 2: Lotus leaves self-clean based on geometry
• Idea – Tables, buttons, and door handles can have special geometry that allows it to be more hygienic.
Article 3: Chemicals emitted by red algae reduce the buildup of a biofilm that bacteria grow on. • Idea – Use an automatic chemical cleaner system that reduces biofilm if viruses also accumulate on biofilm.
Article 4: Sea cucumbers emit light after being touched.
• Idea – Things in public can light up after being touched by a person, therefore indicating that it has been touched. After it has been cleaned the indicator will turn off.
Article 5: Stain-resistant fabric with less fluorochemical usage.
• Idea – Antimicrobial fabric overlays that would reduce the spread of the virus upon contact.
Article 6: Targeted filtration technology traps and kills microbes.
• Idea – Air ventilation filters using specialized material to trap and kill bacteria upon air going through it.
Article 7: Antibacterial treatment that disable bacteria's ability to colonize.
• Idea – Synthetic furanones can be used to stop the spread of bacteria which can be installed in ventilation systems, air conditioners, and cleaning products.
Article 8: The skin of pilot whales' resist microorganisms.
• Idea – We can mimic this idea and create a spray which can be used on common items such as buttons, door handles, railings etc.

C-Sketch

The C-Sketch (3-6-5 Sketch) method was used to visualize and further elaborate upon the ideas generated in the previous methods. This was accomplished by having each team member draw 3 concept implementations, then the drawings were rotated to each member for augmenting and idea expansion. This exercise played a big role in the down selection process, as it gave great visual material to analyze. Below are a few of the drawings created, where the multiple layers of editing/revision can be seen by the different colors throughout the drawings (each person used a distinct color.) Two of these sketches were then updated, after being chosen through the down selection process. These two updated C-sketches will be shown in the down selection portion of this report.

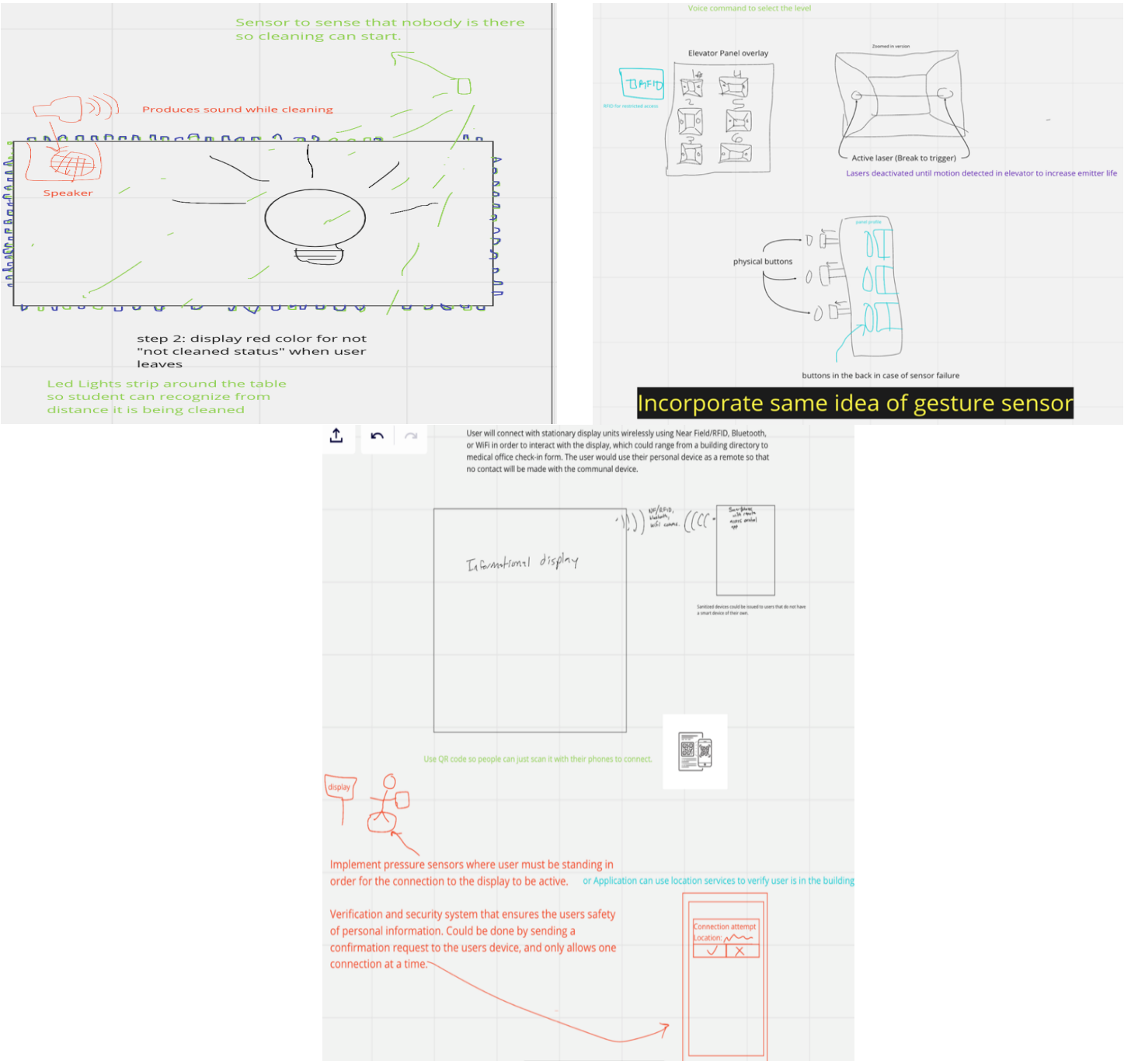


Figure 5: C-Sketches

From the mind map ideation exercise, 41 specific concept solutions were generated. The design by analogy exercise also added 14 more novel concepts to the mind map. There were 10 unique concept designs generated during the 6-3-5 sketch exercise, many of which incorporated ideas obtained during the previous ideation exercises. The ideas generated throughout these ideation exercises were analyzed, and the most novel and feasible ideas were chosen to down-select from.

Down Selection Phase:

The real-win-worth method was used to down select from the most prominent ideas. Each team member was allotted 3 votes for each category: Real (does the technology exist?), Win (does it competitively satisfy our main objective?), and Worth (is it economically feasible?). This process was applied to the mind map and 6-3-5 sketches. The results led to the selection of our two main concepts: a touchless elevator interaction system and a smart desk detection system. Below is the mind map showing the real-win-worth analysis, as well as updated C-sketches of each of the two selected ideas.

The first concept selected was the touchless elevator interaction system, where users can operate the elevator without having to touch the panel. The specifics of this concept will be explored during the prototyping design portion of this report. This concept is very feasible to design but will require special design considerations to acceptably implement in the market due to regulation complexity with elevator environments. This concept was not as novel as it originally appeared, as it was discovered that some elevator companies have already been working on similar concepts. It is important to note that this specific design has not been found through patent searches, but similar patents do exist.

The second concept is a Smart Desk Detection System, where a desk unit constantly updates potential users of its cleanliness. This is done through the use of sensors (both a time-of-flight distance detection & computer vision implementations created) and an LED display used to provide statuses to users. This concept was found to be very feasible due to its relatively low-cost sensors, as well as a real world need for this system. This concept also appears to be novel as preliminary search yielded no similar concepts anywhere.

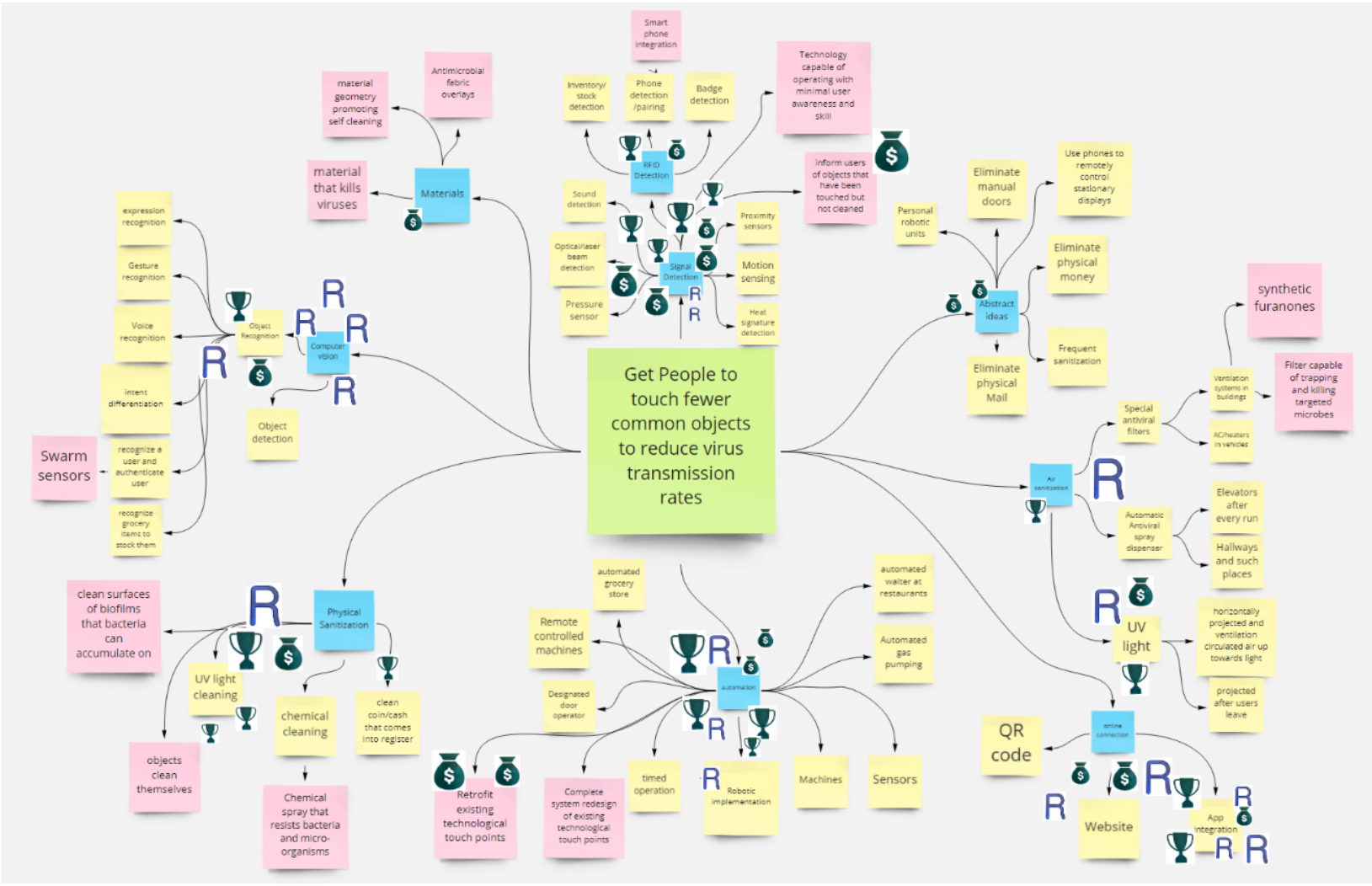


Figure 6: Real-Win-Worth

Updated C-sketch drawings:

Touchless Elevator Concept

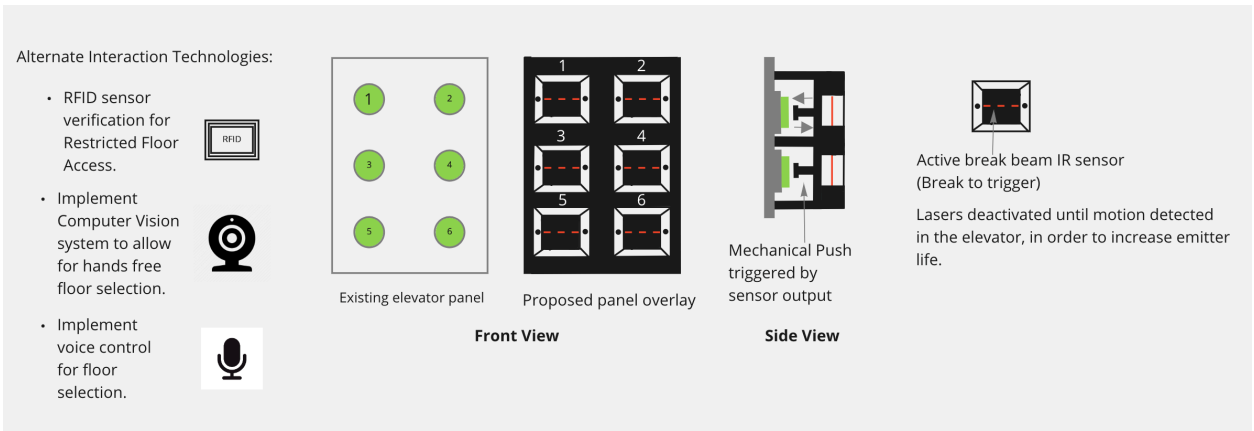


Figure 7: Updated Elevator Panel C-Sketch

Smart Desk Detection Concept

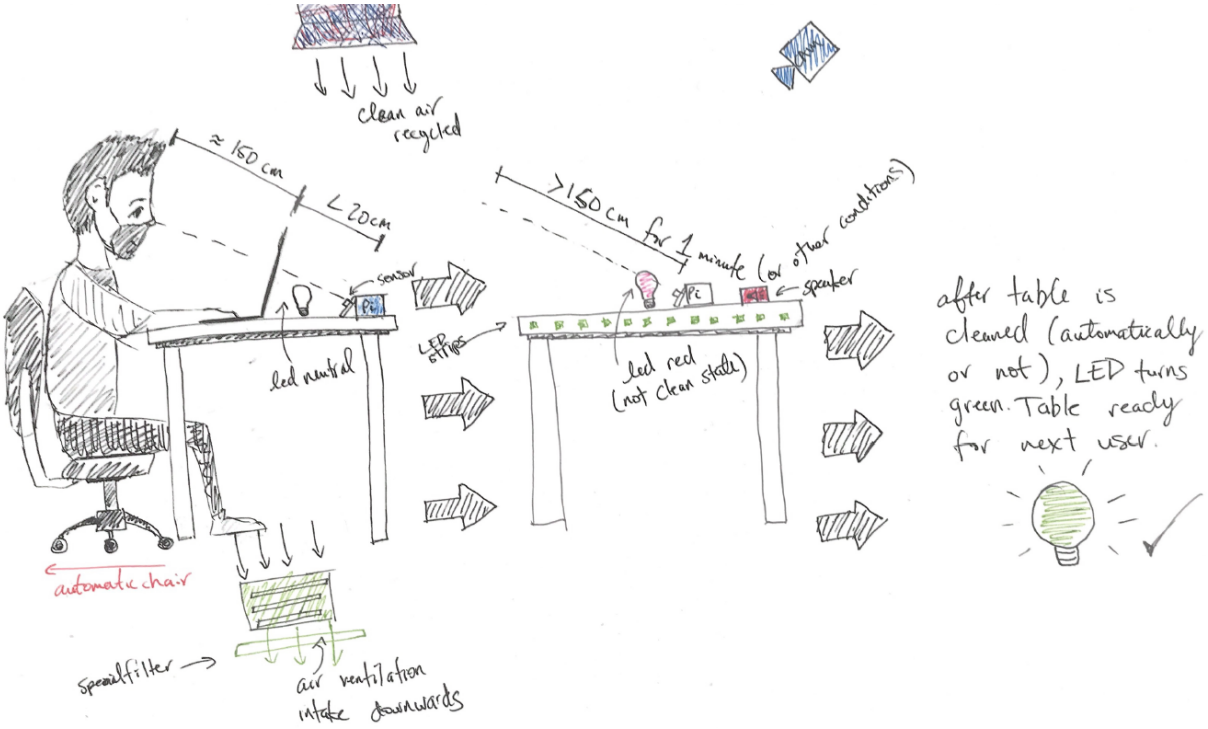


Figure 8: Updated Smart Desk C-Sketch

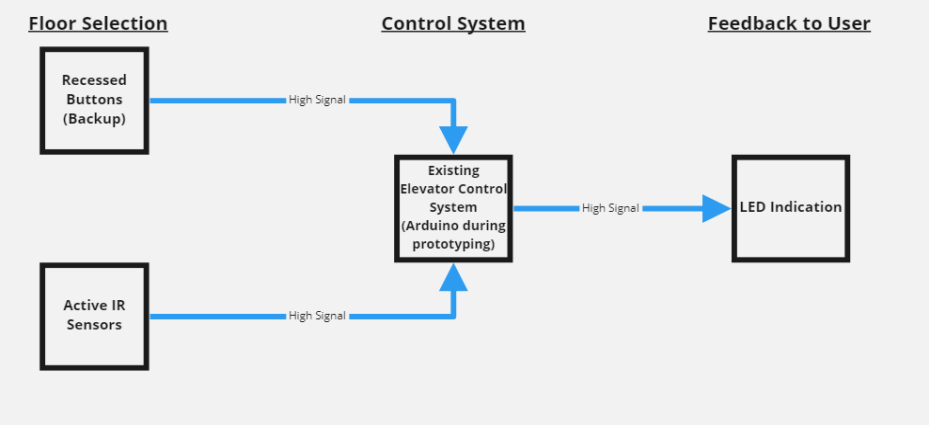
Design methodology, constraints, specifications, and system requirements:

Touchless Elevator Panel:

The basic concept of the elevator panel is that an individual operating the elevator can do so without any physical contact. Instead of buttons, there are depressed cylinders in the panel that constantly have an active infrared beam passing through them. Once any object breaks said infrared beam, the corresponding floor will be selected. In the event of a failure in the break-beam sensor, physical buttons at the back surface of the tube can be pressed to select the floor. The touchless elevator panel was designed to be easily integrated into existing elevator systems, with the operation being simple enough to allow unfamiliar people to successfully operate it. The system was prototyped using a 3D printed panel that includes multiple floor options, as well as the implementation of the break-beam sensors and the backup buttons. On the software level, the code for the panel accounts for emergency situations by having a "hold" function. Overall, the prototype successfully demonstrates the implementation of break-beam sensors to select an elevator floor without any physical contact.

Below is the overall structure for the elevator panel prototype. In an actual implementation, the existing control system inputs would be connected to the break-beam sensors, which would run in parallel with the existing floor selection buttons. The elevator system would also control the LED indication on the panel as it would during normal operation. Without an elevator control system to connect to, an Arduino microcontroller was used to simulate a partial control system for prototyping purposes. This Arduino may also be necessary in an actual implementation to correctly translate the sensor signals, depending on the break-beam sensors used and the control panel configuration. For instance, the break beam sensors used for the prototype operate on 5 volts, whereas an elevator control system may operate between 12 and 24 volts. An Arduino wouldn't be necessary in this case, but the system would require a relay triggered by the sensor that would close the necessary circuit to activate the corresponding button's function. The method in the prototype allows for the prototype user to receive feedback on whether the break-beam sensor activation was successful or not. The prototype is realized using a 3D printed infrastructure.

Overall structure of Touchless Elevator Panel Prototype:



Smart Desk Detection System:

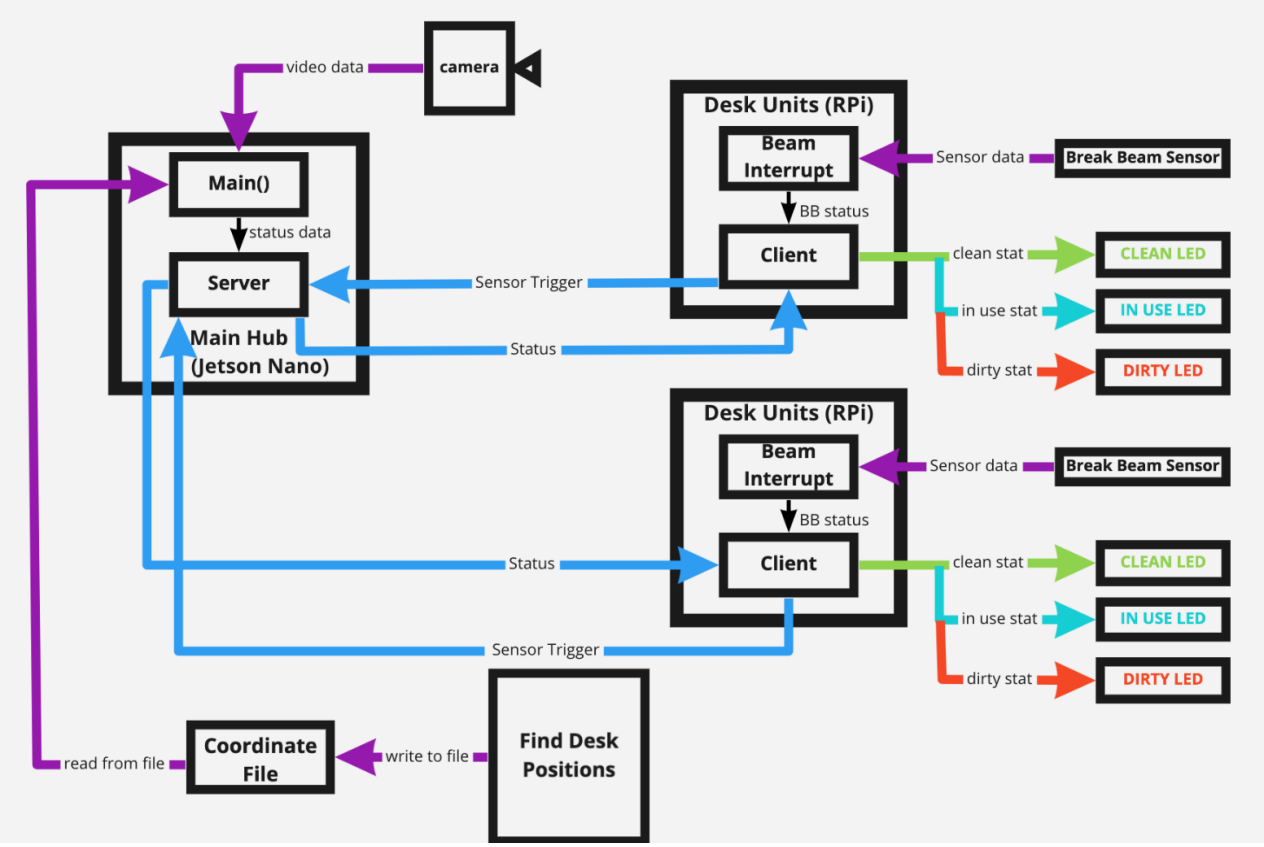
The second developed system addresses the problem of virus transmission through physical contact with an uncleaned desk or shared workspace. The smart desk idea allows for a reliable and accurate method of determining if a desk has been used or not. This system was first realized using a LIDAR distance sensor, that measured the distance from the sensor to the person sitting at the desk. Issues and inefficiencies quickly became apparent, as each desk requires multiple different pieces of hardware, thus creating more failure opportunity. Also, the distance sensor is only capable of detecting objects in a very specific region, leading to issues where the desk is in-use, but the user is undetected due to unexpected positioning. The most comprehensive solution this problem was found through the implementation of computer vision.

The updated smart desk system receives video input from a camera, which then implements Computer Vision and AI technology to detect people in the room. Each desk is then assigned a position on the video feed with the use of a setup program. One of four statuses are assigned to each desk in a room; clean, in-use, paused, or dirty. The distance between each person in the room and their nearest desk is constantly being calculated. When a person is very close to a desk for an extended amount of time, it is assumed that the desk is now in use. When that same person walks away from the desk, the status changes from ‘in-use’ to ‘dirty.’ The computer vision system is in charge of determining the correct status of each desk. On each desk, there is a system responsible for alerting the user which status the desk is, through the use of LED indicators. When a change in the desk status occurs, an update is sent wirelessly from the CV system to the desk system, triggering the appropriate LED (clean- green, in-use- blue, paused- blinking blue, dirty- red). This information is sent to the desk units through the use of a simple socket-level server. In this system, the computer vision side is classified as the “server”, and each desk unit is a “client.”

On each desk system there is also the presence of a break beam sensor. The break beam sensor provides a touchless method of resetting the desk status. When the status of the desk is ‘dirty’, the user would clean the desk, and then activate the break beam sensor to change the status to ‘clean’. Likewise, when the status of the desk is ‘in-use’, the break beam sensor can be used to trigger a ‘pause’ status, which allows the user to leave the workstation without triggering the dirty status. Once they have returned from the break, the camera will trigger the client to return to the ‘in-use’ state. Since the method of communication between client and server is bidirectional, the break beam sensor signal is able to be sent back to the computer vision side of the system, where it is correctly interpreted and used to produce the requested action. The overall smart desk system can be seen in the diagram below.

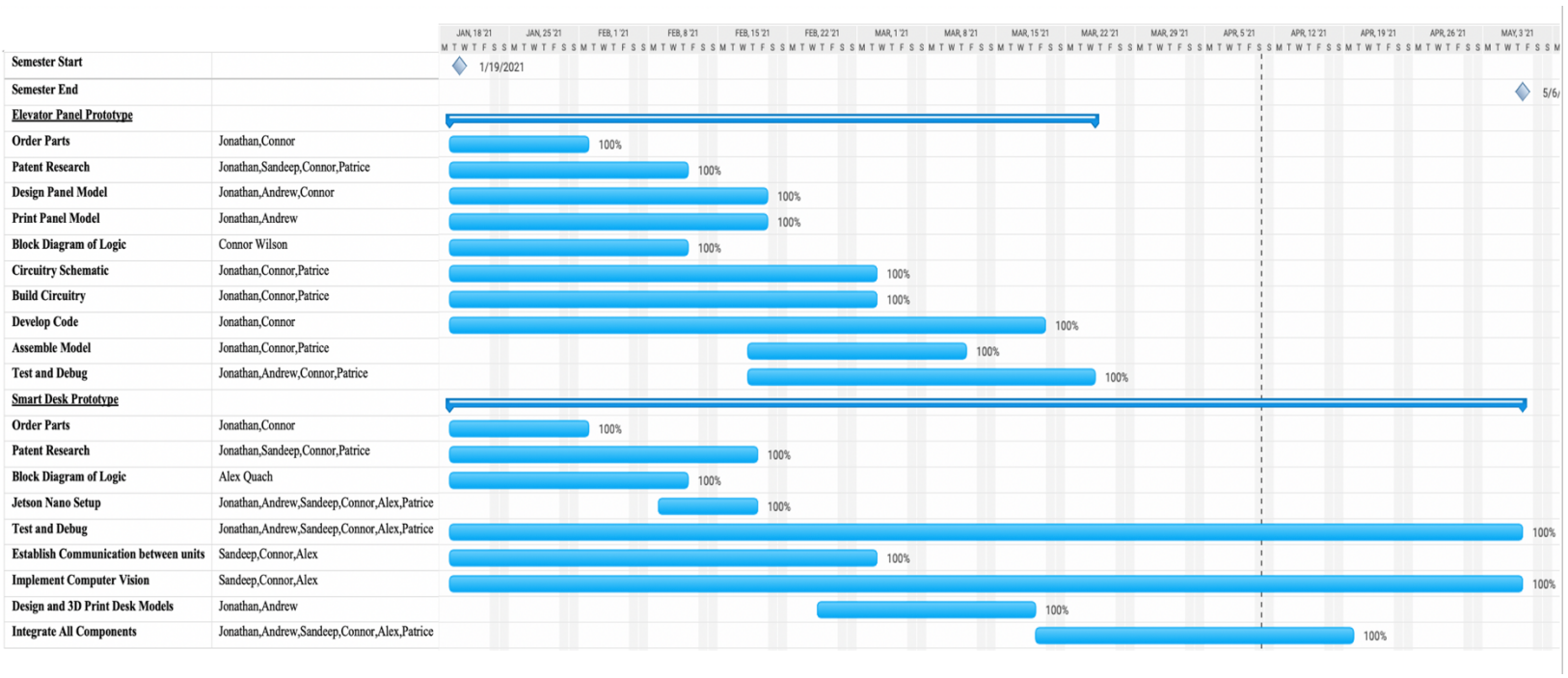
This system was designed to work in classroom environments but can easily be implemented in any environment consisting of a shared workspace. For example, places such as restaurants and bowling allies can also utilize this system. Not only will this be helpful to people looking for a clean desk to use, but also the employees responsible for maintaining clean environments for their customers. The developed system works with both 2D and depth capable cameras.

Overall diagram of desk concept:



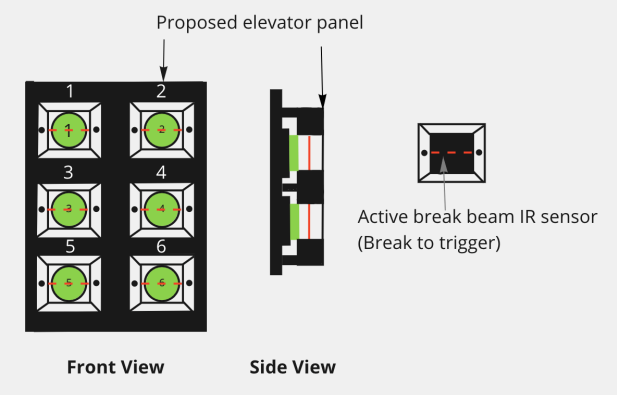
Engineering Documents:

Gannt chart detailing this semesters work:



Semester Start		1/19/2021
Semester End		5/6/
Elevator Panel Prototype		
Order Parts	Jonathan,Connor	100%
Patent Research	Jonathan,Sandeep,Connor,Patrice	100%
Design Panel Model	Jonathan,Andrew,Connor	100%
Print Panel Model	Jonathan,Andrew	100%
Block Diagram of Logic	Connor Wilson	100%
Circuitry Schematic	Jonathan,Connor,Patrice	100%
Build Circuitry	Jonathan,Connor,Patrice	100%
Develop Code	Jonathan,Connor	100%
Assemble Model	Jonathan,Connor,Patrice	100%
Test and Debug	Jonathan,Andrew,Connor,Patrice	100%
Smart Desk Prototype		
Order Parts	Jonathan,Connor	100%
Patent Research	Jonathan,Sandeep,Connor,Patrice	100%
Block Diagram of Logic	Alex Quach	100%
Jetson Nano Setup	Jonathan,Andrew,Sandeep,Connor,Alex,Patrice	100%
Test and Debug	Jonathan,Andrew,Sandeep,Connor,Alex,Patrice	100%
Establish Communication between units	Sandeep,Connor,Alex	100%
Implement Computer Vision	Sandeep,Connor,Alex	100%
Design and 3D Print Desk Models	Jonathan,Andrew	100%
Integrate All Components	Jonathan,Andrew,Sandeep,Connor,Alex,Patrice	100%

Touchless panel concept design:



Elevator Panel Prototype Code:

arduino_elevator_simple_finished §

```
#define PROTOTYPING

#include "pins.h"
#include "prototyping.h"

void setup() {
/* Set all button pins to input_pullup
 * Pins will remain low until button is pressed
 * connecting input pins to ground and triggering
 * the input
 */
  pinMode(BUTTON_1, INPUT_PULLUP);
  pinMode(BUTTON_2, INPUT_PULLUP);
  pinMode(BUTTON_3, INPUT_PULLUP);
  pinMode(BUTTON_4, INPUT_PULLUP);
  pinMode(BUTTON_OPEN, INPUT_PULLUP);
  pinMode(BUTTON_CLOSE, INPUT_PULLUP);
  pinMode(BUTTON_CALL, INPUT_PULLUP);

/*
 * Set all output signal pins to output.
 */
  pinMode(SIGOUT_1, OUTPUT);
  pinMode(SIGOUT_2, OUTPUT);
  pinMode(SIGOUT_3, OUTPUT);
  pinMode(SIGOUT_4, OUTPUT);
  pinMode(SIGOUT_OPEN, OUTPUT);
  pinMode(SIGOUT_CLOSE, OUTPUT);
  pinMode(SIGOUT_CALL, OUTPUT);

}

void loop() {

/*
 * Each block is repeated for each input/output pair
 * If a LOW (sensor trigger or button press) is
 * detected, trigger the output. If not, turn the
 * output off.
 */

  if(!digitalRead(BUTTON_1)){ digitalWrite(SIGOUT_1, OUTPUT_ON); }
  else{ digitalWrite(SIGOUT_1, OUTPUT_OFF); }

  if(!digitalRead(BUTTON_2)){ digitalWrite(SIGOUT_2, OUTPUT_ON); }
  else{ digitalWrite(SIGOUT_2, OUTPUT_OFF); }

  if(!digitalRead(BUTTON_3)){ digitalWrite(SIGOUT_3, OUTPUT_ON); }
  else{ digitalWrite(SIGOUT_3, OUTPUT_OFF); }

  if(!digitalRead(BUTTON_4)){ digitalWrite(SIGOUT_4, OUTPUT_ON); }
  else{ digitalWrite(SIGOUT_4, OUTPUT_OFF); }

  if(!digitalRead(BUTTON_OPEN)){ digitalWrite(SIGOUT_OPEN, OUTPUT_ON); }
  else{ digitalWrite(SIGOUT_OPEN, OUTPUT_OFF); }

  if(!digitalRead(BUTTON_CLOSE)){ digitalWrite(SIGOUT_CLOSE, OUTPUT_ON); }
  else{ digitalWrite(SIGOUT_CLOSE, OUTPUT_OFF); }

  if(!digitalRead(BUTTON_CALL)){ digitalWrite(SIGOUT_CALL, OUTPUT_ON); }
  else{ digitalWrite(SIGOUT_CALL, OUTPUT_OFF); }

}
```

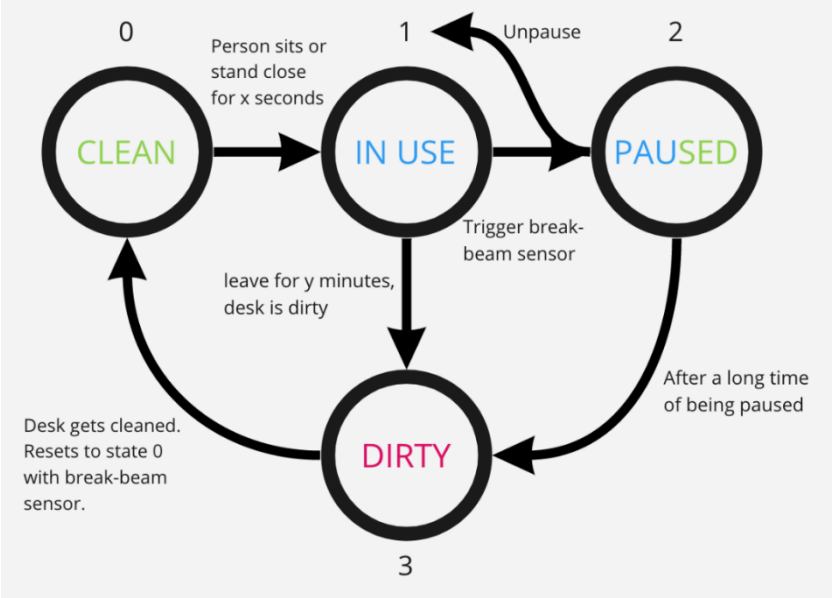
arduino_elevator_simple_finished § pins.h § prototyping.h §

```
/*
 * For the prototype, we have the outputs normally OUTPUT_OFF, to
 * trigger the white LED color, and the outputs switch to OUTPUT_ON
 * during a button press to turn off the green and blue so that
 * the LEDs turn red. For actual implementation, the OUTPUT_ON and OUTPUT_OFF
 * will need to be reversed. To accomplish this, comment out the prototyping
 * macro at the beginning of the main file.
 */

#ifdef PROTOTYPING

#define OUTPUT_ON   LOW
#define OUTPUT_OFF  HIGH

#endif

#ifndef PROTOTYPING

#define OUTPUT_ON   HIGH
#define OUTPUT_OFF  LOW

#endif
```

arduino_elevator_simple_finished § pins.h § prototyping.h

```
#ifndef PINS_H
#define PINS_H

/*
 * Define button pins -
 * Pin names in comments are directly compatible
 * with Arduino Uno Rev3 and Arduino Nano
 * as of March 2021
 */

#define BUTTON_1      2   //D2
#define BUTTON_2      3   //D3
#define BUTTON_3      4   //D4
#define BUTTON_4      5   //D5
#define BUTTON_OPEN   6   //D6
#define BUTTON_CLOSE  7   //D7
#define BUTTON_CALL   8   //D8

#define SIGOUT_1      19  //D19 or A5
#define SIGOUT_2      18  //D18 or A4
#define SIGOUT_3      17  //D17 or A3
#define SIGOUT_4      16  //D16 or A2
#define SIGOUT_OPEN   15  //D15 or A1
#define SIGOUT_CLOSE  14  //D14 or A0
#define SIGOUT_CALL   13  //D13

#endif
```

Smart Desk concept logic:



Smart Desk Code (Distance Sensor Program):

```
import RPi.GPIO as GPIO
import time
import numpy as np
import adafruit_vl53l0x
import board
import busio
class ultrasonic_sensor:
    def __init__(self, trig=18,echo=24,led_in_use=23, led_clean=17, led_dirty=25, state=0, data_length=20, sensor=True):
        GPIO.setmode(GPIO.BCM)

        if sensor == False:
            self.GPIO_TRIGGER = trig
            self.GPIO_ECHO = echo

            GPIO.setup(self.GPIO_TRIGGER, GPIO.OUT)
            GPIO.setup(self.GPIO_ECHO, GPIO.IN)
        else:
            i2c = busio.I2C(board.SCL, board.SDA)
            self.tof = adafruit_vl53l0x.VL53L0X(i2c)

        self.led_in_use = led_in_use
        self.led_clean = led_clean
        self.led_dirty = led_dirty


        GPIO.setup(self.led_in_use, GPIO.OUT)
        GPIO.setup(self.led_dirty, GPIO.OUT)
        GPIO.setup(self.led_clean, GPIO.OUT)

        self.cleaned = True
        self.paused = False
        self.data_length = data_length
        self.state = state

        self.dist_q = np.ones(data_length, dtype=float)*400 # set array to have high values initially (should have no one there)
        self.set_state(0)

    def distance_ultrasonic(self):
        GPIO.output(self.GPIO_TRIGGER, False)
        time.sleep(.00001)
        GPIO.output(self.GPIO_TRIGGER, True)
        time.sleep(.00001)
        GPIO.output(self.GPIO_TRIGGER, False)

        StartTime = time.time()
```

```
        StopTime = time.time()

        while GPIO.input(self.GPIO_ECHO) == 0:
            StartTime = time.time()

        while GPIO.input(self.GPIO_ECHO) == 1:
            StopTime = time.time()

        return (StopTime - StartTime)*34300/2 # cm

    def distance_tof(self):
        return self.tof.range/10

    def evaluate_q(self):
        if self.state == 0:
            if sum(self.dist_q[0:4])/5 < 75:
                self.set_state(1)
        elif self.state == 1:
            if self.dist_q[0] < 10 and self.dist_q[1] < 10 and self.dist_q[2] < 10:
                self.set_state(2)
            elif sum(self.dist_q)/self.data_length > 350:
                self.set_state(3)
        elif self.state == 2:
            while self.paused:
                cmd = input('desk paused, enter "p" to resume: ')
                if cmd == 'p':
                    self.paused = False
            if not self.paused:
                self.set_state(1)
        elif self.state == 3:
            while not self.cleaned:
                cmd = input('desk dirty, enter "c" to clean: ')
                if cmd == 'c':
                    self.cleaned = True
            if self.cleaned:
                self.set_state(0)

    def set_state(self, state=0):
        if state == 0:
            GPIO.output(self.led_clean, True)
            GPIO.output(self.led_dirty, False)
            GPIO.output(self.led_in_use, False)
            self.pause = False
        elif state == 1:
            GPIO.output(self.led_clean, False)
            GPIO.output(self.led_dirty, False)
            GPIO.output(self.led_in_use, True)
            self.pause = False
        elif state == 2:
            GPIO.output(self.led_clean, True)
            GPIO.output(self.led_dirty, False)
            GPIO.output(self.led_in_use, True)
            self.paused = True
        elif state == 3:
            GPIO.output(self.led_clean, False)
            GPIO.output(self.led_dirty, True)
            GPIO.output(self.led_in_use, False)
            self.cleaned = False
            self.pause = False
        self.state = state


if __name__ == '__main__':
    try:
        u = ultrasonic_sensor()
        while True:
            dist = u.distance_tof()
            if dist > 400:
                dist= 400
            u.dist_q[1:] = u.dist_q[0:u.data_length-1]
            u.dist_q[0] = dist
            u.evaluate_q()
            time.sleep(1)
            print(dist, u.state)
    except KeyboardInterrupt:
        print("measurement stopped")
    finally:
        GPIO.cleanup()
```

Smart Desk Desk Location Setup Code (Computer Vision Program-2D Camera):

```
import cv2

vid = '/dev/video0' #location of video file or camera stream
pic_name = 'frame1.jpg' #name of picture being saved
pic = '/home/frame1.jpg' #location of saved photo from video
fileloc = "/home/sdptt/Documents/xylocation.txt" #location of coordinate text file

# Opens the Video file and saves first frame as image
cap= cv2.VideoCapture(vid)
cap.set(3,1280)
cap.set(4,720)
ret, frame = cap.read()
cv2.imwrite(pic_name,frame)

# function to display the coordinates of
# of the points clicked on the image
def click_event(event, x, y, flags, params):

    # checking for left mouse clicks
    if event == cv2.EVENT_LBUTTONDOWN:

        # displaying the coordinates
        # on the Shell
        print(x, ' ', y)
        #write coordinates to file
        location = x, y
        file1.write(str(location))
        file1.write("\n")
        # displaying the coordinates
        # on the image window
        font = cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(img, str(x) + ',' +
                    str(y), (x,y), font,
                    1, (255, 0, 0), 2)
        cv2.imshow('image', img)

# driver function
if __name__=="__main__":

    # reading the image
    img = cv2.imread('frame1.jpg')
    # displaying the image
    cv2.imshow('image', img)
    #open file
    file1 = open(fileloc, "w")
    # setting mouse hadler for the image
    # and calling the click_event() function
    cv2.setMouseCallback('image', click_event)
    # wait for a key to be pressed to exit
    cv2.waitKey(0)
    #close file
    file1.close()
    # close the window
    cv2.destroyAllWindows()
```

Smart Desk Server-side Code (Computer Vision Program-2D Camera):

```
import jetson.inference
import jetson.utils
import cv2
import socketio
import eventlet
import sys
import copy
from threading import Thread
from collections import deque

class Jetson_System:
    # function below sets the neural network, the camera being accessed, and opens a display
    def __init__(self):
        self.net = jetson.inference.detectNet("ssd-mobilenet-v2", threshold=0.5)
        self.camera = jetson.utils.gstCamera(1280, 720, "/dev/video0")
        self.display = jetson.utils.glDisplay()
        # add data structure for desks {desk ID: (position, status)}
        self.test_desk = [(560,431), 'C', [], deque(), False] #[location, status, 20 frame data, 20 second data, pause button status]
        self.desks = {1:self.test_desk, 2:self.test_desk, 3:self.test_desk} # dictionary "1" is the key, ((0,0), "clean", ....) is the data

        self.set_desk_coord() # open coordinate file and set coordinates
        print(self.desks)

        self.setup_server()
        server_thread = Thread(target=self.start_server) # create the thread to start the server
        server_thread.daemon = True # will die when program dies
        server_thread.start() # start the thread that starts the server

    # function sets up the server
    def setup_server(self):
        self.sio = socketio.Server()
        self.app = socketio.WSGIApp(self.sio)
        self.call_backs()

    # function starts the server
    def start_server(self):
        eventlet.wsgi.server(eventlet.listen(('', 5000)),self.app)

            self.sio.emit('server_message1',self.desks[1][1])
            #print("sending 1")
            self.sio.sleep(1)

    def send_desk2(self):
        while True:
            self.sio.emit('server_message2',self.desks[2][1])
            #print("sending 2")
            self.sio.sleep(1)

    def send_desk3(self):
        while True:
            self.sio.emit('server_message3',self.desks[3][1])
            #print("sending 3")
            self.sio.sleep(1)



    #function below defines all the events related to self.sio
    def call_backs(self):
        @self.sio.event
        def connect(sid, environ):
            print('connect ', sid)
            self.sio.start_background_task(self.send_desk1)
            self.sio.start_background_task(self.send_desk2)
            self.sio.start_background_task(self.send_desk3)

        @self.sio.event
        def my_message1(sid, data):
            print('Recieved data from node {}: {} '.format(sid, data))
            self.desks[1][4] = True

```

```
        @self.sio.event
        def my_message2(sid, data):
            print('Recieved data from node {}: {} '.format(sid, data))
            self.desks[2][4] = True

        @self.sio.event
        def my_message3(sid, data):
            print('Recieved data from node {}: {} '.format(sid, data))
            self.desks[3][4] = True

        @self.sio.event
        def disconnect(sid):
            print('disconnect ', sid)

    # function below does everything
    def main_function(self):
        try:
            while self.display.IsOpen():
                img, width, height = self.camera.CaptureRGBA()
                detections = self.net.Detect(img,width,height)
                self.display.RenderOnce(img,width,height)
                for ID,desk in self.desks.items():
                    distances = self.find_distance(desk, detections, 150)
                   # print('id:' , ID,distances)
                    status = self.find_status(ID, desk, distances)

                self.display.SetTitle("FPS: " + str(self.net.GetNetworkFPS()) + " | " + "Stat: " + str(self.desks[1][1]))
        except:
            print("Display closed")
            self.server_thread.stop()
            sys.exit()

    # center datatype: (x,y)
    def calc_distance(self, center1, center2):
        return ((center1[0]-center2[0])**2 + (center1[1]-center2[1])**2)**.5

    # find distance between a point (singular desk) and persons
    def find_distance(self, desk, detections, max_dist):
        persons = []
        for detection in detections:
            if detection.ClassID == 1: # class id 1 = "person"
                dist = self.calc_distance(detection.Center, desk[0])
                if dist < max_dist: # only add person that is close enough
                    persons.append(dist) # desk[0] is center pos
        return persons

    # function finds if the person is close within 20 frames
    def find_status(self, ID, desk, persons):
        stat = 0
        in_use = False
        if len(persons) >= 1:
            desk[2].append(1) # desk[2] is data_1s for one detection person was detected
        else:
            desk[2].append(0)
        if len(desk[2]) >= 20: # after 1 second of recording, determine if person was there that second
            if sum(desk[2]) >= 10:
                stat = 1
            if len(desk[3]) < 15:
                desk[3].append(stat)
            else:
                desk[3].popleft()
                desk[3].append(stat)
            self.set_status(ID, desk)
            desk[2] = []
```

```
    # based on the average of 20 seconds and 'pause' button on client end, the function sets the status
    def set_status(self,ID, desk):
        if desk[1] == 'C':
            if sum(desk[3]) >= 10: # desk[3] is 20 second data
                desk[1] = 'I'
                desk[4] = False
        elif desk[1] == 'I':
            if sum(desk[3]) <= 5:
                desk[1] = 'D'
            if desk[4]:
                desk[1] = 'P'
                desk[3].clear()
                desk[4] = False
        elif desk[1] == 'P':
            if sum(desk[3]) >= 10:
                desk[1] = 'I'
                desk[4] = False
        else: # desk[1] is 'dirty'
            if desk[4]:
                desk[1] = 'C'
                desk[3].clear()
                desk[4] = False
        print('Desk', ID , 'status:',sum(desk[3]), desk[1])

    # function opens a file, read it line by line, then modifies or adds (i created 3 up top. if there is more, it'll add) desks
    def set_desk_coord(self):
        file1 = open('xylocation.txt', 'r')
        for i,line in enumerate(file1):
            desk = copy.deepcopy(self.test_desk) # copy desk data
            coord = eval(line) # eval will evaluate the string '(x,y)' and turn it into a tuple type (x,y)
            desk[0] = coord # change coordinate in desk data
            self.desks[i+1] = desk #set desk to updated data
            print(self.desks[i+1])

sys = Jetson_System()
sys.main_function()
```

Smart Desk Desk Location Setup Code (Computer Vision Program-Depth Camera):

```
#!/usr/bin/env python3
import cv2
import depthai as dai

fileloc = "xyzlocation.txt" #location of coordinate text file
#open file
file1 = open(fileloc, "w")
stepSize = 0.05

# Start defining a pipeline
pipeline = dai.Pipeline()

# Define a source - two mono (grayscale) cameras
monoLeft = pipeline.createMonoCamera()
monoRight = pipeline.createMonoCamera()
stereo = pipeline.createStereoDepth()
spatialLocationCalculator = pipeline.createSpatialLocationCalculator()
xoutDepth = pipeline.createXLinkOut()
xoutSpatialData = pipeline.createXLinkOut()
xinSpatialCalcConfig = pipeline.createXLinkIn()
xoutDepth.setStreamName("depth")
xoutSpatialData.setStreamName("spatialData")
xinSpatialCalcConfig.setStreamName("spatialCalcConfig")

# MonoCamera
monoLeft.setResolution(dai.MonoCameraProperties.SensorResolution.THE_720_P)
monoLeft.setBoardSocket(dai.CameraBoardSocket.LEFT)
monoRight.setResolution(dai.MonoCameraProperties.SensorResolution.THE_720_P)
monoRight.setBoardSocket(dai.CameraBoardSocket.RIGHT)
outputDepth = True
outputRectified = False
lrcheck = False
subpixel = False

# StereoDepth
stereo.setOutputDepth(outputDepth)
stereo.setOutputRectified(outputRectified)
stereo.setConfidenceThreshold(255)
stereo.setLeftRightCheck(lrcheck)
stereo.setSubpixel(subpixel)
monoLeft.out.link(stereo.left)
monoRight.out.link(stereo.right)
spatialLocationCalculator.passthroughDepth.link(xoutDepth.input)
stereo.depth.link(spatialLocationCalculator.inputDepth)
```

```python
topLeft = dai.Point2f(0.55, 0.55)
bottomRight = dai.Point2f(0.6, 0.6)
spatialLocationCalculator.setWaitForConfigInput(False)
config = dai.SpatialLocationCalculatorConfigData()
config.depthThresholds.lowerThreshold = 100
config.depthThresholds.upperThreshold = 10000
config.roi = dai.Rect(topLeft, bottomRight)
spatialLocationCalculator.initialConfig.addROI(config)
spatialLocationCalculator.out.link(xoutSpatialData.input)
xinSpatialCalcConfig.out.link(spatialLocationCalculator.inputConfig)

# Pipeline is defined, now we can connect to the device
with dai.Device(pipeline) as device:
    device.startPipeline()
    # Output queue will be used to get the depth frames from the outputs defined above
    depthQueue = device.getOutputQueue(name="depth", maxSize=4, blocking=False)
    spatialCalcQueue = device.getOutputQueue(name="spatialData", maxSize=4, blocking=False)
    spatialCalcConfigInQueue = device.getInputQueue("spatialCalcConfig")
    color = (255, 255, 255)
    print("Use WASD to move location selector. Press G to capture location coordinates.")

    while True:
        inDepth = depthQueue.get() # Blocking call, will wait until a new data has arrived
        inDepthAvg = spatialCalcQueue.get() # Blocking call, will wait until a new data has arrived
        depthFrame = inDepth.getFrame()
        depthFrameColor = cv2.normalize(depthFrame, None, 255, 0, cv2.NORM_INF, cv2.CV_8UC1)
        depthFrameColor = cv2.equalizeHist(depthFrameColor)
        depthFrameColor = cv2.applyColorMap(depthFrameColor, cv2.COLORMAP_HOT)
        spatialData = inDepthAvg.getSpatialLocations()
        for depthData in spatialData:
            roi = depthData.config.roi
            roi = roi.denormalize(width=depthFrameColor.shape[1], height=depthFrameColor.shape[0])
            xmin = int(roi.topLeft().x)
            ymin = int(roi.topLeft().y)
            xmax = int(roi.bottomRight().x)
            ymax = int(roi.bottomRight().y)
            fontType = cv2.FONT_HERSHEY_TRIPLEX
            cv2.rectangle(depthFrameColor, (xmin, ymin), (xmax, ymax), color, cv2.FONT_HERSHEY_SCRIPT_SIMPLEX)
            cv2.putText(depthFrameColor, f"X: {int(depthData.spatialCoordinates.x)} mm", (xmin + 10, ymin + 20), fontType, 0.5, color)
            cv2.putText(depthFrameColor, f"Y: {int(depthData.spatialCoordinates.y)} mm", (xmin + 10, ymin + 35), fontType, 0.5, color)
            cv2.putText(depthFrameColor, f"Z: {int(depthData.spatialCoordinates.z)} mm", (xmin + 10, ymin + 50), fontType, 0.5, color)

        cv2.imshow("depth", depthFrameColor)
        newConfig = False
        key = cv2.waitKey(1)
        if key == ord('q'):
            break
        elif key == ord('w'):
            if topLeft.y - stepSize >= 0:
                topLeft.y -= stepSize
                bottomRight.y -= stepSize
                newConfig = True
        elif key == ord('a'):
            if topLeft.x - stepSize >= 0:
                topLeft.x -= stepSize
                bottomRight.x -= stepSize
                newConfig = True
        elif key == ord('s'):
            if bottomRight.y + stepSize <= 1:
                topLeft.y += stepSize
                bottomRight.y += stepSize
                newConfig = True
        elif key == ord('d'):
            if bottomRight.x + stepSize <= 1:
                topLeft.x += stepSize
                bottomRight.x += stepSize
                newConfig = True

        elif key == ord('g'):
            print(int(depthData.spatialCoordinates.x),' ',int(depthData.spatialCoordinates.y),'',int(depthData.spatialCoordinates.z))
            #write coordinates to file
            location = int(depthData.spatialCoordinates.x), int(depthData.spatialCoordinates.y), int(depthData.spatialCoordinates.z)
            file1.write(str(location))
            file1.write("\n")

        if newConfig:
            config.roi = dai.Rect(topLeft, bottomRight)
            cfg = dai.SpatialLocationCalculatorConfig()
            cfg.addROI(config)
            spatialCalcConfigInQueue.send(cfg)
```

"Smart Desk" Server-side Code (Computer Vision Program-Depth Camera):

```python
import cv2
import socketio
import eventlet
import sys
import copy
from threading import Thread
from collections import deque
import depthcam

class Jetson_System:
    # function below sets the neural network, the camera being accessed, and opens a display
    def __init__(self):
        self.cam = depthcam.depthcam()
        # add data structure for desks {desk ID: (position, status)}
        self.test_desk = [(54,67,689), 'C', [], deque(), False] #[location, status, 20 frame data, 20 second data, pause button status]
        self.desks = {1:self.test_desk, 2:self.test_desk, 3:self.test_desk} # dictionary "1" is the key, ((0,0), "clean", ....) is the data

        self.set_desk_coord() # open coordinate file and set coordinates
        print(self.desks)

        self.setup_server()
        server_thread = Thread(target=self.start_server) # create the thread to start the server
        server_thread.daemon = True # will die when program dies
        server_thread.start() # start the thread that starts the server

    # function sets up the server
    def setup_server(self):
        self.sio = socketio.Server()
        self.app = socketio.WSGIApp(self.sio)
        self.call_backs()

    # function starts the server
    def start_server(self):
        eventlet.wsgi.server(eventlet.listen(('', 5000)),self.app)

    #  the three send functions below sends messages to each client (1,2,3)
    def send_desk1(self):
        while True:
            self.sio.emit('server_message1',self.desks[1][1])
            #print("sending 1")
            self.sio.sleep(1)

    def send_desk2(self):
        while True:
            self.sio.emit('server_message2',self.desks[2][1])
            #print("sending 2")
            self.sio.sleep(1)

    def send_desk3(self):
        while True:
            self.sio.emit('server_message3',self.desks[3][1])
            #print("sending 3")
            self.sio.sleep(1)



    #function below defines all the events related to self.sio
    def call_backs(self):
        @self.sio.event
        def connect(sid, environ):
            print('connect ', sid)
            self.sio.start_background_task(self.send_desk1)
            self.sio.start_background_task(self.send_desk2)
            self.sio.start_background_task(self.send_desk3)

        @self.sio.event
        def my_message1(sid, data):
            print('Recieved data from node {}: {} '.format(sid, data))
            self.desks[1][4] = True

        @self.sio.event
        def my_message2(sid, data):
            print('Recieved data from node {}: {} '.format(sid, data))
            self.desks[2][4] = True

        @self.sio.event
        def my_message3(sid, data):
            print('Recieved data from node {}: {} '.format(sid, data))
            self.desks[3][4] = True

        @self.sio.event
        def disconnect(sid):
            print('disconnect ', sid)

    # function below does everything
    def main_function(self):
        try:
            print(self.desks)
            self.cam.do_it(1000, self.desks, self.find_distance, self.find_status) # inside here, loop through distances comparing to EACH desk, then set desk status
```

```
        except:
            print("Error somewhere")
            sys.exit()

    # center datatype: (x,y)
    def calc_distance(self, center1, center2):
        return ((center1[0]-center2[0])**2 + (center1[1]-center2[1])**2 + (center1[2]-center2[2])**2)**.5

    # find distance between a point (singular desk) and persons
    def find_distance(self, desk, detections, max_dist):
        persons = []
        for detection in detections:
            dist = self.calc_distance(detection, desk[0])
            if dist < max_dist: # only add person that is close enough
                persons.append(dist) # desk[0] is center pos
        return persons

    # function finds if the person is close within 20 frames
    def find_status(self, ID, desk, persons):
        stat = 0
        in_use = False
        if len(persons) >= 1:
            desk[2].append(1) # desk[2] is data_1s for one detection person was detected
        else:
            desk[2].append(0)
        if len(desk[2]) >= 30: # after 1 second of recording, determine if person was there that second
            if sum(desk[2]) >= 15:
                stat = 1
            if len(desk[3]) < 15:
                desk[3].append(stat)
            else:
                desk[3].popleft()
                desk[3].append(stat)
            self.set_status(ID, desk)
            desk[2] = []

    # based on the average of 20 seconds and 'pause' button on client end, the function sets the status
    def set_status(self,ID, desk):
        if desk[1] == 'C':
            if sum(desk[3]) >= 10: # desk[3] is 20 second data
                desk[1] = 'I'
                desk[4] = False
        elif desk[1] == 'I':
            if sum(desk[3]) <= 5:
                desk[1] = 'D'
            if desk[4]:
                desk[1] = 'P'
                desk[3].clear()
                desk[4] = False
        elif desk[1] == 'P':
            if sum(desk[3]) >= 10:
                desk[1] = 'I'
                desk[4] = False
        else: # desk[1] is 'dirty'
            if desk[4]:
                desk[1] = 'C'
                desk[3].clear()
                desk[4] = False
        print('Desk', ID , 'status:',sum(desk[3]), desk[1])

    # function opens a file, read it line by line, then modifies or adds (i created 3 up top. if there is more, it'll add) desks
    def set_desk_coord(self):
        file1 = open('xyzlocation.txt', 'r')
        for i,line in enumerate(file1):
            desk = copy.deepcopy(self.test_desk) # copy desk data
            coord = eval(line) # eval will evaluate the string '(x,y)' and turn it into a tuple type (x,y)
            desk[0] = coord # change coordinate in desk data
            self.desks[i+1] = desk #set desk to updated data
            #print(self.desks[i+1])




sys = Jetson_System()
sys.main_function()
```

Smart Desk Client-side Code (Computer Vision Program for Both Camera Types):

```
# for server operations
import socketio
import signal
import sys

# for GPIO operations
import RPi.GPIO as GPIO

from enum import Enum
from time import sleep

class DeviceState(Enum):
    DIRTY = 0
    CLEAN = 1
    INUSE = 2
    PAUSE = 3

class ServerState(Enum):
    DISCONNECTED = 0
    CONNECTED = 1

# Defining GPIO pins
SENSOR_PIN = 19
CLEAN_LED = 16
INUSE_LED = 20
DIRTY_LED = 21

# define the server client
sio = socketio.Client()
#add = 'http://localhost:5000' # for testing on same machine
add = 'http://172.20.10.12:5000' # set server IP here

# The following variables contain the states for the
#   server and device states.
# They need to be modifable by multiple threads (main,
#   GPIO callback, and socketio event).
# Those other thread functions need to use the keyword
#   global so that they pick up the main thread's instance.
#
device_state = DeviceState.CLEAN
server_state = ServerState.DISCONNECTED


def set_clean_state():
    global device_state
    P.ChangeDutyCycle(0)
    GPIO.output(DIRTY_LED, GPIO.LOW)
    GPIO.output(CLEAN_LED, GPIO.HIGH)
    device_state = DeviceState.CLEAN
    print ('set_clean_state')

def set_inuse_state():
    global device_state
    GPIO.output([CLEAN_LED, DIRTY_LED], GPIO.LOW)
    P.ChangeDutyCycle(100)
    device_state = DeviceState.INUSE
    print ('set_inuse_state')

def set_dirty_state():
    global device_state
    P.ChangeDutyCycle(0)
    GPIO.output(CLEAN_LED, GPIO.LOW)
    GPIO.output(DIRTY_LED, GPIO.HIGH)
    device_state = DeviceState.DIRTY
    print('set_dirty_state')

def set_pause_state():
    global device_state
    GPIO.output([CLEAN_LED, DIRTY_LED], GPIO.LOW)
    P.ChangeDutyCycle(50)
    device_state = DeviceState.PAUSE
    print('set_pause_state')
```

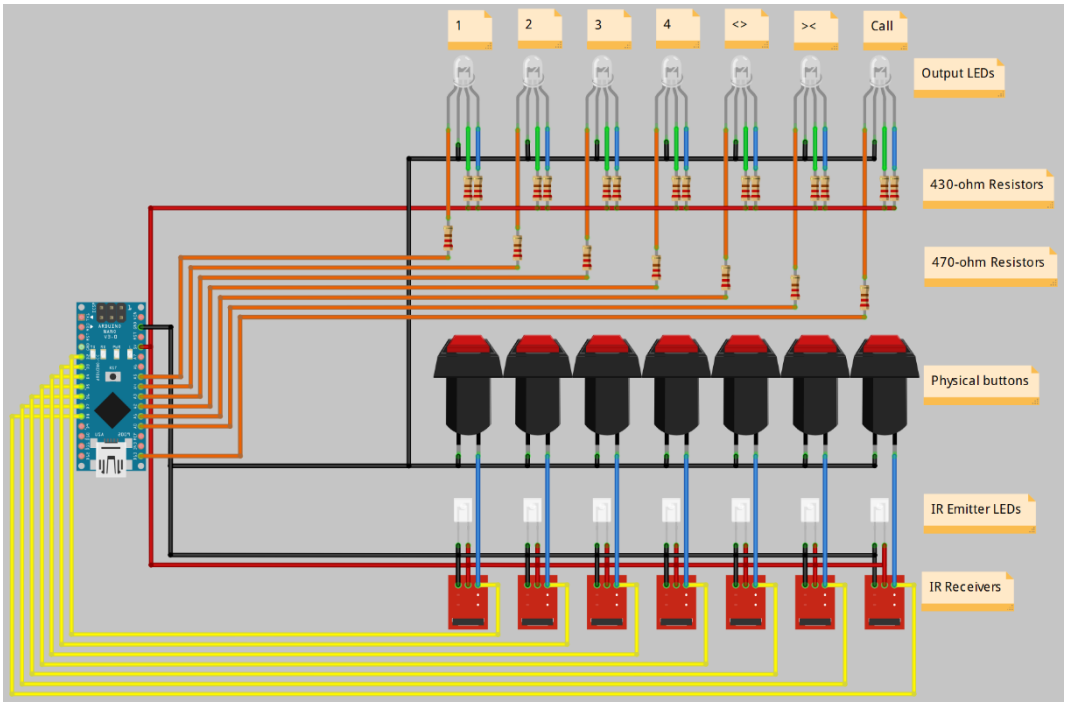
```
#sends pause state to server
def send_server_pause():
    sio.emit('my_message3','P')
    print('send_server_pause')

#sends clean state to server
def send_server_clean():
    sio.emit('my_message3','C')
    print('send_server_clean')

# no operation state
def nop_state():
    pass

#combining device and server state to use in transion of states
def create_string_state():
    global server_state
    global device_state
    return '{0}:{1}'.format(server_state.name, device_state.name)

# The following dictionaries provide the mapping of
# states to functions.
sensor_transistion = dict()
sensor_transistion['DISCONNECTED:DIRTY'] = set_clean_state
sensor_transistion['DISCONNECTED:CLEAN'] = set_inuse_state
sensor_transistion['DISCONNECTED:INUSE'] = set_dirty_state
sensor_transistion['DISCONNECTED:PAUSE'] = nop_state
sensor_transistion['CONNECTED:DIRTY'] = send_server_clean
sensor_transistion['CONNECTED:CLEAN'] = nop_state
sensor_transistion['CONNECTED:INUSE'] = send_server_pause
sensor_transistion['CONNECTED:PAUSE'] = nop_state

server_transistion = dict()
server_transistion['C'] = set_clean_state
server_transistion['I'] = set_inuse_state
server_transistion['D'] = set_dirty_state
server_transistion['P'] = set_pause_state

# runs when status is received from server
# sets pins as necessary
# status_1 for when only using the in-use pin
# status_2 for when using both in-use and pause pins
def set_status(status):
    next_state = server_transistion[status]
    next_state()

# runs when client connects to server
@sio.event
def connect():
    global server_state
    print('connection established')
    server_state = ServerState.CONNECTED

# runs when 'server_message1' is received from server
@sio.event
def server_message3(data):
    print('from server: ',data)
    set_status(data)

# runs when client disconnects
@sio.event
def disconnect():
    global server_state
    print('disconnected from server')
    server_state = ServerState.DISCONNECTED
    #sys.exit(0)
```

```
# runs on keyboard interrupt - ctrl+c
def signal_handler(sig,frame):
    P.stop() # stops PWM
    GPIO.cleanup() # resets GPIO pins
    sys.exit(0)

# runs when sensor is tripped
def sensor_callback(channel):
    print('sensor tripped')
    current = create_string_state()
    #print('current = {0}'.format(current))
    next_state = sensor_transistion[current]
    next_state()


if __name__ == '__main__':
    # configures gpio pins
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(SENSOR_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup([CLEAN_LED, INUSE_LED, DIRTY_LED], GPIO.OUT)
    P = GPIO.PWM(INUSE_LED,2)
    P.start(0)
    GPIO.add_event_detect(SENSOR_PIN, GPIO.FALLING, callback=sensor_callback, bouncetime=250)
    set_clean_state()
    print('GPIO setup complete')

    # configures keyboard interrupt and server connection
    signal.signal(signal.SIGINT, signal_handler)

    while server_state == ServerState.DISCONNECTED:
        try:
            sio.connect(add)
        except socketio.exceptions.ConnectionError as err:
            print("Connection Error: {0}".format(err))
            sleep(10)

    print("Connected")

    sio.wait()
```

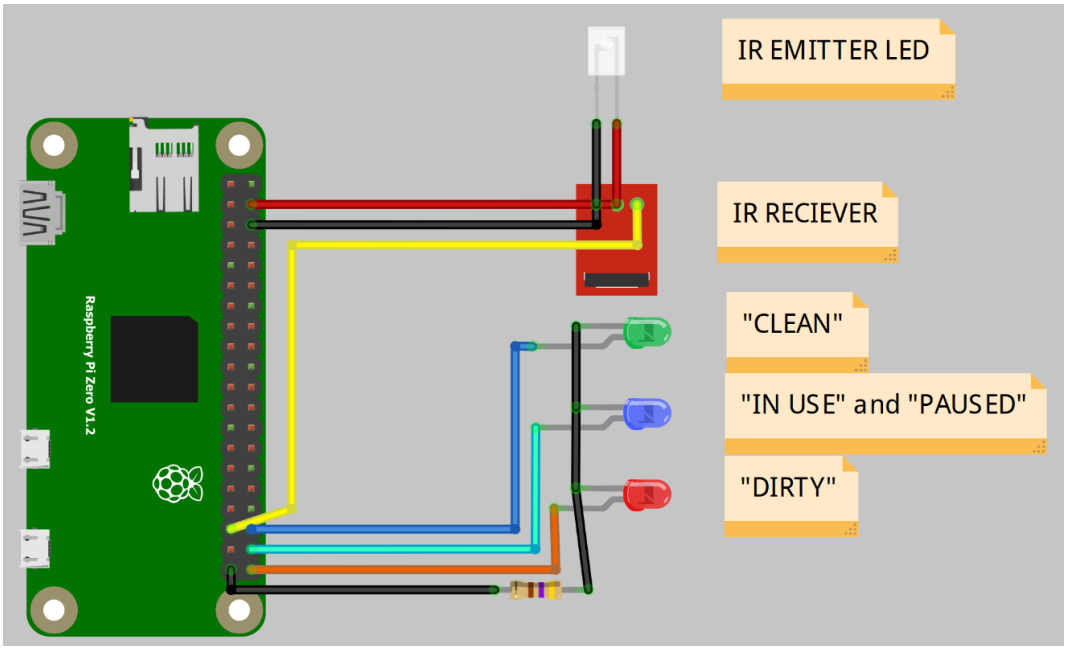
Computer Design Tools:

Our team decided to use the design program *Fritzing* to develop wiring schematics for both of our proposed systems. Below is the schematic for the touchless elevator panel, as well as the schematic for the desk units in the smart desk system. Also, our team used Fusion 360 to develop 3D models for both the elevator panel, and the desk system housing. Below are screenshots of these 3D models.

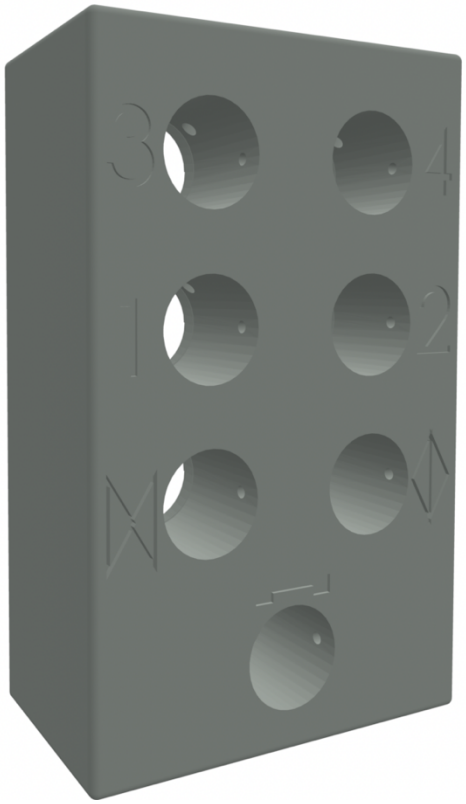
Elevator Panel Wiring Schematic:



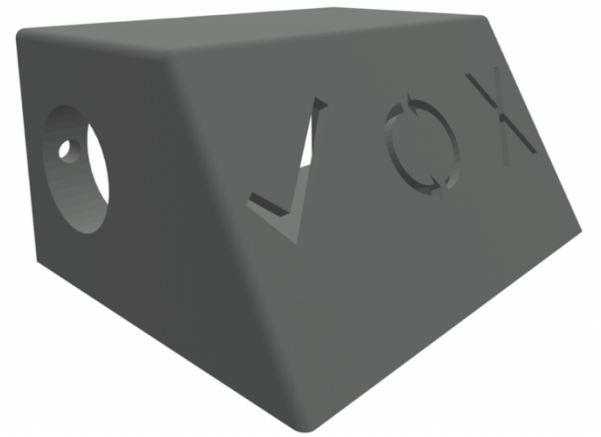
Desk Unit Wiring Schematic:



Elevator Panel 3D Model:



Desk Prototype Housing 3D Model:



Patent and Standards Research Related to Design:

Patent research for both developed prototypes was conducted using both Google Patents and the United States Patent and Trademark database. The main purpose of this patent research was to determine if any similar systems have already been invented and patented. Patents of interest were found by searching for keywords (like smart desk, desk, elevator, elevator panel…). Below are the results of the conducted patent research for both systems.

Touchless Elevator Panel:

Using Google Patent research as a preliminary database, it appears that the idea of a touchless elevator panel invention is not new. Many similar projects of touchless elevator panels were developed due to the SARS (severe acute respiratory syndrome) pandemic in the early 2000s, with the same overall goal of reducing viral transmission from physical contact with elevator systems. The following invention is remarkably similar to the touchless panel explored throughout this report: Neng-Fu Chuang, "non-contact optical Sensor Switch assembly of elevators "U.S. Patent 1419A1, issued January 01, 2006. This invention implements non-contact interaction using optical sensors, also including a physical button backup. These optical sensors work similarly to the break-beam sensors, allowing for the elevator to be operated without any physical contact.

Additionally, there is the CleanLift project developed by Porcupine Solutions. This project includes a similar touchless elevator panel design, utilizing both optical and the break-beam sensor technology to trigger touchless commands. Light emitting diodes were also used to display the status of the selected floor. Other related patents include JP3195888U and CN101425416A. These relevant patents serve as confirmation that our developed system is both needed and well designed.

It is also important to note that all elevators must meet the ASME A17.1 Safety Code for elevators and escalators. The non-contact control panel system needs to satisfy this code if it was to be implemented in an

actual elevator, as well as other required measures such as ADA compliance. The main focus of this project was to develop a functional touchless panel, with less emphasis on code compliance.

Smart Desk System:

The basic research using Google Patents did not produce any results relevant to a workspace capable of automatically determining a cleanliness status. However, there is a patent for a system called intelligent desk (Patent CN204861807U issued December 16, 2015) that uses various sensors to enable a desk to be aware of its environment, as well as the users' overall comfort and posture. Research using the United States Patent and Trademark Office (USPTO) database brought up a slightly similar patent. The invention is called Smart Desk or Smartdesk (Govil, et al. 'Smart desk', U.S. Patent D883277 issued May 5, 2020), but this invention addresses "Ergonomic seating system, tilt-lock control and remote powering method and apparatus" (patent abstract). Further research was conducted using the USPTO database, but no other relevant patents were found.

It has been concluded that our proposed smart desk system may be patent eligible. Our primary contacts at the Air Force Research Lab (Dr. McIntire and Ms. Harlow) have already initiated the patent application process with an invention disclosure writeup. The smart desk system can be modified to work in many different environments, such as offices, schools, restaurants. If a desk or table must be adapted to incorporate the smart desk system, it must follow the standard of such technology (OSHA, ANSI/BIFMA, and other pertinent standards).

Proof of Concept Results:

Touchless Elevator Panel:

The elevator panel prototype our team developed successfully demonstrates the capabilities of using an IR break-beam sensor as a contactless alternative to the traditional elevator button. Although interfacing this prototype with an actual elevator was not possible due to time constraints, it is strongly believed that the prototype does what is necessary for successful operation. In real elevator systems each elevator button sends a high voltage signal to the control system when activated, and this high signal output is demonstrated in the prototype though LED activation. Also, testing with multiple different activation devices was conducted to thoroughly determine the effectiveness of the BB sensor.

Below is a table containing the results of this testing, as well as a video demonstration of the prototype. The results of this testing show that the break-beam sensor reliably activates when the activation device is large enough to block the infrared light beam. In the case of the apartment key, it was not thick enough to completely block the IR light, resulting in 0 successful activations. This should not be too big of an issue, as even a slight tilt to the key resulted in successful activation.

Elevator Panel Testing Results:

ACTIVATION DEVICE	***# OF SUCCESSFUL ACTIVATIONS***
EXPO MARKER	50/50
MECHANICAL PENCIL	49/50
CAR KEY	50/50 at (90-degree)
	46/50 at (0-degree)
FINGER	50/50
APARTMENT KEY	50/50 at (90-degree)
	0/50 at (0-degree)
IPHONE CORNER	50/50
KNIFE	50/50

Elevator Panel Demonstration Video:
OneDrive:
Links to elevator panel demo video: https://olucdenver-my.sharepoint.com/:v:/g/personal/connor_wilson_ucdenver_edu/EdREeNgLUp1JtW0Pw6OrlFABczJPY8Phcunv4e7ow3KC3g?e=FFRy3c
YouTube https://youtu.be/dGr-fb71py8

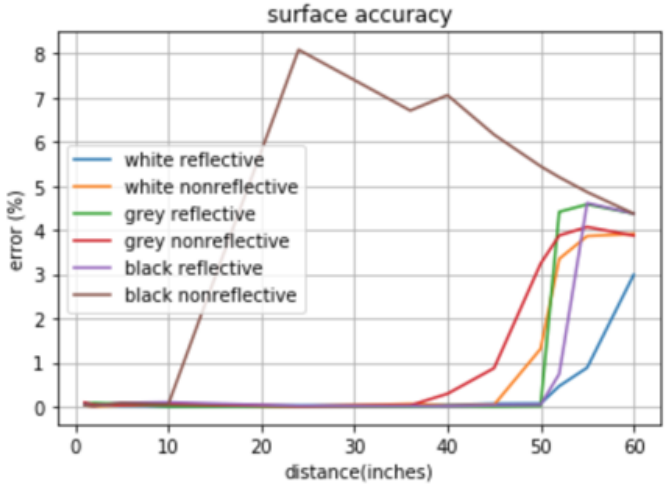
Smart Desk Detection System:

The smart desk system was successfully prototyped in three different variations. The first variation demonstrates the concept of displaying a desk status with the utilization of a distance sensor aimed toward the user's body. Below is a chart outlining distance sensor testing, as well as a demonstration video.

The second variation is the computer vision system. Due to this system being primarily software based, the test evaluation will be done by evaluating the results of key objectives and functionality of the system. At a higher level, the system will need to be able to identify and measure the distance between the user and the desk. Then it will send a status to the desk modules, alerting users of the desk's status. Below is a chart outlining essential functions of the system, as well as an evaluation of each function. Overall, the developed prototype provides a working method of determining the status of the observed desks, as well as a working method of notifying users. Also below are demonstration videos showing the computer vision system, as well as the manual "offline" operation mode.

The third variation is very similar to the last, but this time the system is designed to work with an OAK-D depth capable camera. The system works even better on this camera than it does with the single point of view camera used in the second system variation. The implementation of depth data allows for more accurate desk statuses, as each desk and person on the feed is now defined with three coordinates instead of two. Also, the person tracking was found to be much more reliable when utilizing this device.

Distance Sensor Testing Results:



Desk System Objectives (Computer Vision Program):

Objective	***Evaluation***
Locating of workstations	Desk coordinates are mapped effectively
Desk assignment from file	Correct reading from the mapping file
Person detection	Executed by person detection code successfully, system has periodic difficulties with detecting people in certain positions

Distance calculation	Calculates cartesian distance between person and desk units
Status calculation	Implemented by knowing the previous state, as well as computer vision code
Server/client communication	TCP connection successfully established between the server and the client, allowing for bidirectional communication
Multiple client capability	Successfully connected multiple clients (Raspberry Pis) to one server (Jetson Nano)
Pause/reset capability	Pause/reset functionality are successfully achieved by tripping break-beam sensor
LED status output	Corresponding LED status output are triggered correctly based off of desk status determined in CV system
Manual override	Manual override kicks in successfully when server is down, allowing for user to set desk status with break-beam sensor
Reconnection	When client establishes connection to the server it automatically overrides manual operations
Overall system performance	System functions as expected, assuming ideal circumstances and camera positioning

Desk System Demo Videos:

Video 1: Demonstration of the desk system utilizing the time-of-flight distance sensor:
OneDrive: https://olucdenver-my.sharepoint.com/:v:/g/personal/connor_wilson_ucdenver_edu/EfS3ZK1uwiJFlnh0e4DGdOoBwjf0vcNO7_PTN-k4TgjNFA?e=WiGBXP
YouTube: https://youtu.be/flx4MRha6pM

Video 2: Demonstration of "stand-alone" operation, allowing the users to manually set the desk status with break-beam sensor.
OneDrive: https://olucdenver-my.sharepoint.com/:v:/g/personal/connor_wilson_ucdenver_edu/EerVCQinIW5JkPCdyWl1EVgBrVaydxU1p0DZFy7DNE3kiw?e=HHEabj
YouTube: https://youtu.be/VtdsKwqtLVg

Video 3: Demonstration of complete system showing computer vision operation, updating of desk status LEDs, as well as demonstration of the break-beam controlled pause and reset functions. Desk modules would be on each desk in actual implementation but were organized under the monitor for video simplicity.
OneDrive: https://olucdenver-my.sharepoint.com/:v:/g/personal/connor_wilson_ucdenver_edu/EaiAVE987DZBs3aeXLINU68Bb69TAlYWXIMVrBAEfqBJEg?e=vgrpc5
YouTube: https://youtu.be/Hrq3-UzjKWo

Video 4: Demonstration of complete system showing computer vision operation with a depth camera implementation, updating of desk status LEDs, as well as demonstration of the break-beam controlled pause and reset functions. Desk modules would be on each desk in actual implementation but were organized under the monitor for video simplicity. Also, only two desk units were utilized in this demonstration video. This demonstration provides a showcase of the depth capability, allowing for more accurate desk statuses.

OneDrive: https://olucdenver-my.sharepoint.com/:v:/g/personal/jonathan_barstad_ucdenver_edu/EUbPAI3aPStInyCseG7iWbABPhtDMDKaw65HWWdEhdzKeA?e=HbuPfx
YouTube: https://youtu.be/t8UScklG-90

Project Budget:

All prototypes developed by the team were successfully completed while still maintaining a cost-effective budget. The end total of this team's budget was $137.41 under budget within school funding. An under-budget was achieved by proper resource management and extensive material research.

Semester	Purchased Materials	Individual Cost	Quantity	Total Cost
Fall 2020	TOF Sensor	14.95	2	29.9
	PIR Sensor	12.9	1	12.9
	PIR Sensor	6.89	1	6.89
	BB Sensor	9.89	3	29.67
	Solenoid	19.95	2	39.9
	Component Pack	7.86	1	7.86
	Wire pack	6.98	1	6.98
	Arduino Nano	15.98	1	15.98
	USB cables	3.93	3	11.79
Spring 2021	BB Sensor	7.12	2	14.24
	BB Sensor	7.81	8	62.48
	LEDs	8.99	1	8.99
	Filament	19.99	2	39.98
	Buttons	5.89	1	5.89
	Sandpaper	8.99	1	8.99
	Spray paint	6.94	2	13.88
	wire	14.57	1	14.57
	Jetson Nano(2GB)	59.99	2	119.98
	Jetson Nano(2GB)	49.99	3	149.97
	Jetson Nano(4GB)	89.99	1	89.99
	Power Supplies	10.99	2	21.98
	Power Supplies	8.99	3	26.97
	Power Supplies	9.99	1	9.99
	SD Cards	7.49	6	44.94
	SD Cards	19.11	1	19.11
	Rpi ZeroW	18.48	3	55.44
	M&K	24.99	4	99.96
	Jetson Nano fan (4GB)	14.95	1	14.95
	Jetson nano fan (2GB)	9.98	1	9.98
	Micro-SD cards	8.48	1	8.48
	USB Webcam	**29.98**	**2**	**59.96**
		Total Spent		**$ 1062.59**
		Total Budget		**$ 1200**
		Available Budget		**$ 137.41**

List of Project Design and Implementations Task and Responsibilities:

Jonathan Barstad

Elevator Panel Prototype:

- Developed code responsible for correct prototype operation
- Integration of electrical components
- Assembly of the overall system
- Testing and troubleshooting

Desk Prototype:

- Development and testing of client-side code:
    - Interoperates status updates from the server and correctly updates the LED display.
    - Data sent to server for correct operation of pause & reset functions based off of break-beam sensor activation
- Created fritzing wiring diagram detailing the desk units
- Setup and troubleshooting of Jetson Nano and Raspberry Pi devices
- Setup of desk unit hardware

Andrew Kwavy

- Setup and troubleshooting of Jetson Nano
- Developed multiple iterations of 3D models
    - 3D model of the elevator panel and associated components
    - 3D model of desk unit housing and status identifiers
- Testing of desk system objectives
- Development of stand-alone desk operation

Patrice Ouedraogo

- Extensive patent research
- Setup and troubleshooting of Jetson Nano and Raspberry Pi devices
- Integration of electrical components on both prototypes
- Setup of desk unit hardware
- Testing of desk system objectives

Alex Quach

Desk Prototype:

- Defined and organized sever-side (Jetson Nano/main hub) program's structure, data, and functions
- Developed server-side code:
    - Computer vision that detects users and calculates distances from workstations.
    - Logic to set desk (client) statuses based on information from CV program and client messages
- Integrated the individual coding components to complete working server-side program:
    - Linked data from computer vision to desk logic code to change desk statuses internally
    - Implemented multithreading for server communication to send desk data/receive client info
    - Developed code to pull in desk location from data file and coordinate them in the server system
- Debugged the program to ensure functionality and quality

Sandeep Singh Rai

Desk Prototype

- Setup and troubleshooting of Jetson Nano and Raspberry Pi devices
- Development and research of communication between systems
- Developed Initial Proof of concept for client/server Communication

    - Communication was designed using restful API
    - Format: HTTP://<state_led>/<on/off>
    - Successfully demonstrated setting of state LEDs using own server and client test application
- Implemented communication failure design in client
    - Modified client code base to handle both sunny day and failure scenario
    - Refactored client code to use state flow logic using function pointers instead of cascading if-else statements
    - Troubleshot and rectified bug handling GPIO call back
- Setup of desk unit hardware
- Testing of desk system objectives

Connor Wilson (Team Lead)

-Fulfilled team lead responsibilities:

- Ensured the on-time completion of various tasks & milestones
- Lead effective sponsor communication
- Assigned project roles to each team member

-Elevator Panel Prototype:

- Integration of electrical components
- Assembly of the overall system
- Testing and troubleshooting
- Created multiple detailed diagrams explaining the panel system

-Desk Prototype:

- Developed code responsible for setting up location of desks on camera feed
- Developed code responsible for wireless bidirectional communication between the Jetson Nano and the Raspberry Pi
- Created fritzing wiring diagram detailing the desk units
- Setup and troubleshooting of Jetson Nano and Raspberry Pi devices
- Testing and troubleshooting of multiple code iterations

Jonathan E. Barstad

Jonathan.e.barstad@gmail.com • (303) 913-8554
www.linkedin.com/in/jonathanbarstad • Littleton, CO

Electrical Engineer

Dynamic, innovative, and versatile upcoming Electrical Engineering graduate with robust understanding of electrical and electronic systems maintenance, inspection, and installation. Extensive experience in electrical feedback and control systems for diesel/electric locomotives and military aircraft avionics and electrical systems, creating innate ability to anticipate and overcome design conflicts to facilitate seamless projects. In-depth knowledge of prototype designs, programming code, circuitry design, power systems, electrical drive systems, power systems analysis, power electronic systems, device electronics, and hardware/software interface.

Areas of Expertise

- Advanced Troubleshooting
- Systems Analysis
- Data Analytics
- Inspections and Testing
- Quality Assurance
- Safety Compliance & Training
- Installations and Repairs
- Technical Problem-Solving
- Critical Thinking

Professional Experience

BNSF Railway, Denver, CO — 2015 – Present

Electrician, Diesel Engines

Inspect, test, and diagnose electrical malfunctions in systems and equipment using testing devices, software, and comprehensive visual inspections. Read and interpret electrical schematics and blueprints to locate and troubleshoot problems. Complete requisite repairs, reassemble equipment, calibrate, and perform robust tests before equipment is returned to service. Repair and replace defective wiring and relays and install electrical equipment using diverse array of electrician and hand tools. Service and maintain all electrical equipment on board trains and on track systems encompassing safety, HVAC, and lighting control systems.

- Rank in top 5% of 1400+ electricians' company-wide based on skill level.
- Level 2 journeyman, qualified for advanced troubleshooting.
- Collaborate with team members to ensure compliance with company and federal safety rules and procedures.

United States Marine Corps, San Diego, CA — 2010 – 2015

RTCASS Technician IMA (Enlisted), Collateral Duty Inspector, IMRL Program Manager

Managed $17M+ inventory of airborne weapon assemblies, shop replaceable assemblies, automatic test equipment, and ancillary apparatus. Supervised 15 subordinate personnel and facilitated testing and repair of avionic and armament tools and equipment across 10+ aircraft variances. Identified unsafe conditions and conducted safety training to mitigate hazards. Utilized in-depth knowledge of functional operation of electronic systems and units to test and diagnose equipment malfunctions. Conducted disassembly, troubleshooting, reassembly, and installation in compliance with schematics. Completed specialized training in avionics systems, electrical systems, and components.

- Recorded zero preventable losses, maintained detailed inventory logs.
- Performed conformity audits and inspections to ensure alignment with repair standards.
- Trained incoming IMRL asset manager.
- Secret security clearance expired 2016

Education

Bachelor of Science in Electrical Engineering, Minor in Computer Engineering

University of Colorado Denver, Denver, CO, Anticipated Graduation May 2021

- GPA: 3.724
- Dean's List Six of Eight Eligible Semesters

- Coursework Includes Electrical Drive Systems, Power Electronic Systems & Lab, Autonomous Vehicle Design, Device Electronics, Power Systems Analysis & Lab, Advanced Electromagnetic Fields, Hardware/Software Interface.
- Senior Project sponsored by the Air Force Research Laboratory: Collaboration with interdisciplinary team of electrical and mechanical engineers to develop innovative methods, technologies, and/or techniques for sanitary hands-free/touch-free interaction with everyday items to help prevent the spread of COVID-19. Tasked with assisting with prototype designs, programming code, and designing circuitry for two designs prototypes:
    - elevator panel incorporating infrared sensors to detect when person intends to press button, triggering appropriate signal without physical interaction with button.
    - "smart desk" system that will detect when person has used desk, table, or other workstation, notify central system and provide visible indication that area needs to be cleaned, as well as an indication that the area has been sanitized.
- Phase 1 Finalist in the 2021 OpenCV AI Competition sponsored by Intel Corporation and Microsoft Azure: Competing on a university-sponsored team of electrical engineers tasked with producing innovative solutions to COVID related issues through the use of computer vision and AI technology.

Bachelor of Science in Computer Information Systems

Columbia College, Columbia, MO, Attended 2012-2013

- 12 credit hours completed

Professional Training

Electrical Feedback, Control, and Power Systems for Diesel/Electric Locomotives, BNSF Railway Technical Training Center, 2015 - 2020
Electrical and Avionics Systems and Components, United States Marine Corps, 2011 -2012

Technical Proficiencies

Cadence ORCAD Pspice, MATLAB/Simulink, Mathcad, C Programming, Python Programming, Verilog Programming, Arduino Programming, FPGA Design Software (Quartus and Xilinx), Robot Operating System, Microsoft Word, Excel, PowerPoint, OneNote

Awards

Honorable Discharge, Marine Corps Good Conduct Medal, OpenCV AI Competition 2021 Phase 1 Finalist

Affiliations

Tau Beta Pi (Engineering Honor Society)

Andrew Kwavy

Mechanical Engineer

Westminster, CO 80234
(720) 645-9613 | Andrew.Kwavy@gmail.com

SUMMARY STATEMENT

College educated with 3 years of military experience. Graduating with a Mechanical Engineering degree in May 2021. Excellent in detecting errors, seeing the big picture, thinking out of the box, solving problems, analyzing systems, leading, and communicating with others. Strives to exceed professional and personal goals and has strong written and verbal communication skills.

PROFESSIONAL WORK EXPERIENCE

Imagery Systems Engineer, August 2020 – Present
Maxar Technologies, Longmont, CO

I am responsible for the monitoring, troubleshooting, and verification of the Maxar factory flow from order through delivery for various customers. My duties include executing collection strategies in support of projects, real time, and tasking requirements, perform necessary ground troubleshooting when an issue occurs, and work with MOC personnel during anomaly resolution.

Mechanic, May 2018 – Present
Army National Guard, Longmont, CO
Job Type: Reservist

My job is repairing and maintaining ACs and heaters, but my main role is being a soldier and follow orders from my chain of command to accomplish whatever tasks we have and conduct training in a variety of things.

OTHER WORK EXPERIENCE

Armed Security Officer, May 2019 – August 2020
SCIS Securitas, Longmont, CO

Merchandise Associate, June 2018 – September 2018
The Home Depot, Louisville, CO

Customer Service, October 2016 – January 2017
Costco Wholesale, Thornton, CO

Stocker, March 2016 – November 2016
The Home Depot, Louisville, CO

Parking Lot Attendant, August 2015 – December 2015
Ace Parking, Beverly Hills, CA

Cashier, November 2012 – June 2014
Good Food Market, Pasadena, CA

EDUCATION

Bachelor of Science in Mechanical Engineering, January 2016 – May 2021
University of Colorado at Denver, Denver, CO
Graduation date: May 2021

Associate of Science, January 2013 – December 2015
Pasadena Community College, Pasadena, CA

PROJECTS

Animated Processing Television, University of Colorado at Denver, 2019

Wrote a program that utilizes both, Arduino and Processing to display different animation screens on Processing window that changes screens when clicking buttons on a remote control that is connected to Arduino.

Quadcopter Drone, Pasadena City College, 2015
Designed and built a plywood-framed Quadcopter using SolidWorks, AutoCAD, and a laser cutter.

Go-Kart, Pasadena City College, 2014
Built a plywood-framed Go-Kart with junk parts.

Nano Technology Research, Pasadena City College, 2014
Conducted experiments on chemical compounds and recorded data for a Professor from UCLA.

PROFESSIONAL SKILLS

Linux
SolidWorks
AutoCAD
MATLAB
Arduino
C++
Java
Microsoft Office

CERTIFICATIONS

Active Secret Clearance

Professional TRAINING

Utilities Equipment Repairer, November 2018 – March 2019
Army Ordnance School, Fort Lee, VA

Patrice D. Ouedraogo

Electrical Engineering Senior Student

ouepatrice@gmail.com • (347) 371-7268
Aurora, Colorado

- Electrical Engineering Undergraduate with strong communication, problem solving skills and a background in production and testing.
- knowledge of electrical Engineering designs, basic programming code, circuit design and analysis, embedded systems and hardware/software interface.
- Team player with the ability to take on various or new tasks

EDUCATION

University of Colorado Denver**,** *Bachelor of Science in* ***Electrical Engineering*** *Anticipated Dec. 2021*
GPA: 3.74/4.0

Other institutions and training:
Ashworth Online College *Electrician Certification Dec. 2018*
International Institute for Water and Environment(2IE) *2006*

ACADEMIC &PROFESSIONAL EXPERIENCE

Senior Design Project: Non-Contact Interaction with Common Areas *sponsored by*: **The Air Force Research Lab**
Spring 2021

- Extensive patent research
- Work with a multidisciplinary group to prototype a non-contact interaction system for elevator and the smart desk
- Integration of electrical components on both prototypes

Circuit Design and Fabrication: *simulated, built, and soldered PCB* **Fall 2020 - Spring 2021**

1. Apply theory in the design and analysis of electrical circuits and use ORCAD/Pspice simulation to verify functional requirements of circuit
2. use benchtop laboratory equipment including function generators, D.C power supplies, oscilloscopes, and multimeters
3. populate a printed circuit board, solder components and verify circuit operation
4. RF filter and 90-Degree Hybrid Design:

- Theory, Design, fabrication and Test of Rf filter (Project 1) and 90-Degree Hybrid Coupler (project 2) using ADS design tool
- . Presented project 2 to classmates and submitted final report that included recommendations and amendments form lecturer.

Hardware/ Software Design *Spring 2019. – Summer 2021*

- Model digital systems using hardware description language (structural, register-transfer-language, and behavioral) techniques in Verilog (Quartus and Xilinx)
- Customized AXI IP interface and explore hardware and software design alternatives using Xilinx tool with the PYNQ board
- Apply digital circuit design techniques to the microarchitecture construction of a computer processor (MIPS processor Design)

Mount Sopris Instruments Inc. *Denver, CO US 10/2016-09/2018*
Electromechanical Assembler

- Assembled logging probes and surface related equipment using work instructions and other assembly documentation.
- Repair: disassembled and replace all defective parts as instructed in the diagnostic report, reassembled tools and filled out required documents.
- Tested repaired and new tools and made the necessary adjustments for optimal operation at the shop.

- Worked according to ISO 9001:2008 standards related to the company's Quality Management System (QMS)

SMA America, Denver, *Denver, CO US 09/15 – 10/2016*

Electromechanical Assembler

- Read specifications to determine component parts and assembly sequences of the solar inverters
- manufactured various parts or units in accordance with written Standard Operating Procedures
- Used basic tools and power tools such as crimpers, strippers, power screwdrivers, and torque wrenches and jigs
- Performed riveting and drilling
- Attached labels and mark other identifying information on parts and units

OTHER WORK EXPERIENCE

Cooper Lighting *Aurora, CO US*

Receiving and Electrical Assembler 05/14 – 09/15

Cargill Meat Solution *Fort Morgan, CO, US*

Wizard Trimmer Operator 09/11 – 04/12

Price Wise Discount *New York, NY, US*

Store Associate 02/10 – 08/11

Regional Directorate of Agriculture and hydraulic *Ddg,Burkina Faso*

Water Technician 12/06 – 08/09

OTHER SKILLS:

OSHA 10-hour General Industry training
Proficient in Microsoft Word, access and Excel, PowerPoint and internet, cycle counting,
Fluent in French
Work well with others in a Productivity Driven Environment (keep hour by hour board updated)
Ability to be cross trained.

ALEX QUACH

Denver, CO 80249
Phone: (720)757-8430 · Email: alex.quach@ucdenver.edu alexquach1221@gmail.com

PROFILE

- Quality driven upcoming Electrical Engineering (EE) graduate with experience in professional engineering projects
- Excel in use of EE lab equipment and programming (Python, C/C++, Java, microprocessors, FGPA)
- Determined to learn and improve my knowledge and skills through experience and graduate level courses
- Self-starter who performs and communicates well in high demanding multicultural team environments

EDUCATION

University of Colorado Denver, *Bachelor of Science in* ***Electrical Engineering*** **Anticipated May 2021**

- Minor: Computer Engineering and Computer Science
- GPA: 3.73/4.0

PROFESSIONAL & ACADEMIC EXPERIENCE

Software Development Research Assistant, *University of Colorado Anschutz* **Mar 2020 – Nov 2020**

- Developed a novel software system in Python language for auditory neuroscience research
- Generated unit test procedures for subsystems and acquired test data to analyze requirements
- Configured hardware design files to compile with Python software and Windows platform
- Collaborated on research projects with EE faculty and peers, and the neuroscience professionals

Senior Design Project: Smart Desk, *sponsored by*: **The Air Force Research Lab** **Aug 2020 - Present**

- Developing a computer vision smart desk solution using Arduino microprocessors, Raspberry Pi's, and sensors to reduce COVID transmission through physical touch in public
- Collaborate with electrical/mechanical peers and professionals to prototype ideas and generate solutions
- Engage with sponsor, stakeholders, and potential customers to evaluate customer needs and requirements

Administrative Assistant, *Quach Electric (family business)* **Jan 2017 - Present**

- Interpret electrical building schematics and installed electrical components following safety standards
- Manage paperwork, permits, and administrative tasks
- Coordinate business communication between clients, local jurisdictions, and other entities
- Handle business finances such as balancing budgets and procuring material using Microsoft Office tools

IT Support Assistant, *University of Colorado Denver, School of Public Affairs* **Apr 2018 – Present**

- Support entire School of Public Affairs IT services independently from May '18 – November '18
- Troubleshoot and diagnose hardware and software computer issues for faculty, staff, and students
- Maintain computer lab, classroom, and meeting room computers and technology

FPGA: *Verilog/Python implementation on Intel and Xilinx development boards* **Aug 2018 – May 2020**

- Implemented behavioral, block & RTL Verilog code to run digital circuit modules and designs
- Designed custom AXI IP in Xilinx to process imaging data and accelerate computer vision process
- Programmed in Verilog to control Intel and PYNQ FPGA development boards and onboard hardware
- Verified & validated acquired data of integrated MIPs computer architecture model using testbench design

Circuit & Electronics: *simulated, built, and soldered PCB* **Aug 2017 - Present**

- Design analog circuits with OrCAD/Pspice to verify functional requirements of circuit
- Test and build circuits with lab measurement equipment: oscilloscope, DMM, function generator, etc.

ADDITIONAL INFORMATION

- Outside of school and work, I enjoy playing volleyball, drawing and painting, and learning new things

Sandeep Singh Rai

Westminster, CO 80234
Phone: (720) 431-3447 · Email: sandeepsingh.rai@ucdenver.edu singhrai.sandeep@gmail.com

SUMMARY OF QUALIFICATIONS

- College Educated, knowledgeable, detail-oriented, problem solving skills and multilingual individual with experience in sales and customer service industry
- Oversaw the hiring, training, and motivating staff to ensure high quality and standards are met
- Troubleshooted basic and routine customer issues that are technical in nature
- Managed Inventory, ordering/purchasing and applied appropriate cost control measures
- Technical skills include Microsoft Office and basic knowledge of C++, Python, MATLAB

EDUCATION

B.S in Electrical Engineering **Expected Graduation 12/2021**
University of Colorado, Denver, CO
B.S in Nuclear Engineering
Idaho State University, Pocatello, ID *2009-2012*
Narayana Jr. College
Andhra Pradesh, India *2008*

WORK EXPERIENCE

UNITED STATES ARMY
Interior Electrician *May 2020 – Present*

- Understood the principles of electricity. Install, maintain, and safely distribute electricity which include transformers, circuit breakers, electrical boxes, and lightning rods. Testing and fixing any short circuits in wires and faulty equipment and read blueprints and wire plans to determine layouts and repair.

Testing Assistant
University of Colorado, CO Jan 2020 – May 2020

- Administers and oversees tests according to strictly prescribed guidelines and standards. Prepares and organizes testing materials in accordance with individual testing requirements and with established testing protocol and guidelines. Assist in the enforcement of test security policies and procedures before, during, and after testing

Assistant Manager
Corner Store, Denver, CO 2017-2018

- Managed a team of 10 employees that generated a total revenue of $500k Dollars
- Ensured high standards of all health, safety and sanitation requirements are met in accordance with federal, provincial and local standards

Customer Service Representative
Convergys (DirectTV Reseller), Pocatello, ID 2014-2015

- Attended attentively to customer needs and concerns. Demonstrated empathy, while maximizing opportunity, for building rapport with the customer. Conducted needs-based selling by using non-scripted probing techniques

Assistant Manager 2013-2014
Shift Leader 2010-2013
Chevron, McCammon, ID

- Managed a team of 14 employees that generated a total revenue of $36 million Dollars. Managed budgets, costs, revenues, schedules, and operations
- Coordinated sharing of resources among different locations to ensure coverage was adequate to drive sales and Managed inventory, ordering/purchasing and applied appropriate cost control measures

CONNOR WILSON

Connor.Wilson@ucdenver.edu | (303)-437-2926 | Highlands Ranch, CO 80130

Summary

Extremely motivated and eager student in the pursuit of a BS in Electrical Engineering. Undergone multiple senior level engineering courses related to the power field, such as Electric Drive Systems, Power Electronic Systems, Power Systems Analysis, and Control Systems Analysis. Strong performance under stressful situations and a heavy emphasis on being reliable and adaptable. Consistently a successful leader on team projects, managing tasking, and delivering against strict deadlines.

Education

Senior Standing Electrical Engineering student at the University of Colorado Denver 12/2021
GPA: 3.745/4.0
Recognized on the Dean's list for the past six semesters.
Member of the Tau Beta Pi Engineering Honor Society.

Educational Projects

- Working as appointed team lead on Capstone design project sponsored by Air Force Research Laboratory. Project entails creating a solution to reducing virus spread through human contact with everyday items. Responsible for project planning, assignment completion, and ensuring effective team collaboration and communication.
- Finalist in the OpenCV AI Competition 2021, as a member of CU Denver's "LynxVision" team.
- Designed and created a fully functioning 24-hour digital clock with the use of Boolean logic schematics. The clock ran on an Altera FPGA board using Quartus software, and included start, stop, reset, and preset functions.
- Designed Multiple successful projects involving OrCAD simulation software, such as building a 0-100KHz two stage low pass filter using OPAMPS, with only 5KΩ and 10KΩ resistors (Project parameters). Also Built a functional full bridge diode rectifier that successfully converts an AC input to a DC output.

Soft Skills:

- Strong work ethic
- Peer relationships
- Detail Oriented
- Time management
- Reliable
- Organization and planning
- Team player/leader

Technical Skills:

- Microsoft Word, PowerPoint, Excel
- Power-pole board analysis
- Good foundation in C language programming
- Experience with MATLAB and Simulink
- OrCAD simulation software
- Physical circuit analysis with DMM and Oscilloscope utilization

Experience

Assistant Technician *04/2017 - 07/2017,* 04/2018 - 07/2018

Johnny Stubbs Mechanical | Lone tree, CO

Aided HVAC technicians in the field by assisting with technical work. Included extensive job training with many of the hands-on aspects of the job, as well as frequently leading customer interactions.

- Maintained consistent on-time/early arrival for all appointments to check/fix systems and equipment.
- Involved in completing advanced repairs on HVAC equipment, components, and systems.
- Performed semi-skilled work with minimal supervision.
- Collected and organized parts and equipment to fulfill customer and job order requirements.
-

Office Assistant *05/2019 - 08/2019*

Colorado Help at Home | Centennial, CO

Filed and audited all company expenses from 2016-2019. Conducted meetings with advertisement agencies.

- Gained valuable professional experience by working in an office environment.
- Maintained consistent on-time/early arrival to office.

Activities and Interests

Some of my interests outside of engineering include traveling, hiking, camping, skiing, and personal fitness.