Logo%20Main%20200

**Boston University**

**Electrical & Computer Engineering**

**EC464 Capstone Senior Design Project**

User's Manual

SharkCam

Submitted to

SharkNinja

GDiPietro@sharkninja.com

by



*Team 28*

Team Members

Sophia Delia sgdelia@bu.edu

Liron Dubin ldubin@bu.edu

Alex HP [alexchp@bu.edu](mailto:alexchp@bu.edu)

Brett Senders bsenders@bu.edu

Kevin Vasquez kvasquez@bu.edu

Submitted: 4/17/2023

#### Table of Contents

[**Executive Summary 2**](#_heading=h.30j0zll)

[**1 Introduction 1**](#_heading=h.1fob9te)

[**2 System Overview and Installation 2**](#_heading=h.3znysh7)

[2.1 Overview 2](#_heading=h.2et92p0)

[2.2 User interface. 2](#_heading=h.tyjcwt)

[2.3 Physical description. 3](#_heading=h.3dy6vkm)

[2.4 Installation 4](#_heading=h.1t3h5sf)

[**3 Operation of the Project 5**](#_heading=h.4d34og8)

[3.1 Operating Mode 1: Normal Operation 5](#_heading=h.2s8eyo1)

[3.2 Operating Mode 2: Abnormal Operations 5](#_heading=h.17dp8vu)

[3.3 Safety Issues 6](#_heading=h.3rdcrjn)

[**4 Technical Background 7**](#_heading=h.26in1rg)

[**5 Relevant Engineering Standards 9**](#_heading=h.lnxbz9)

[**6 Cost Breakdown 10**](#_heading=h.35nkun2)

[**7 Appendices 12**](#_heading=h.1ksv4uv)

[7.1 Appendix A - Specifications 12](#_heading=h.44sinio)

[7.2 Appendix B – Team Information 12](#_heading=h.2jxsxqh)

# Executive Summary

*The SharkCam is a hardware accessory and app for the Shark® AI Ultra 2-in-1 vacuum robot that autonomously photographs events and parties using a smartphone camera. SharkNinja tasked Team 28 with designing an accessory for their robot that augments the device’s functionality and takes advantage of its existing sensors and autonomous capabilities. The rise of apps designed around capturing spontaneous photos, and the resurgence of retro camera technologies such as instant and disposable cameras have demonstrated the growing demand for novel ways to create and share pictures, especially at social gatherings. The SharkCam system will autonomously traverse an indoor event space, capturing candid photos which are uploaded to a database for easy viewing and sharing through a mobile application. The app allows users to set parameters such as capture time and capture interval, takes photos by accessing the phone’s camera, and sends commands to the robot through a network-connected Raspberry Pi. A rotating motorized phone holder facilitates photo-taking in all directions, ensuring the action is captured no matter the robot’s orientation. The phone holder is connected to a detachable telescoping column mounted to a semi-permanent base. An auxiliary obstacle avoidance system utilizes a laser sensor to prevent collisions with the column. The SharkCam accessory is user-installable, removable, and non-destructive to the stock robot.*

# Introduction

The increasing popularity of novelty and retro photography cameras, and even apps like BeReal which encourage people to live in the moment, have demonstrated the demand for new (or old) and more exciting ways to capture experiences, especially at events and parties. The SharkCam

Luckily, SharkNinja is looking to expand their vacuum product line beyond cleaning. The SharkNinja vacuum robot employs an autonomous room navigation system to effectively vacuum spaces without colliding with walls or furniture. To supplement that system, a lidar sensor and front bumper sense collisions with moving objects, and actively redirect the robot before or upon collision. This navigation capability makes the SharkNinja vacuum robot the perfect autonomous vehicle for our product: SharkCam.

SharkCam consists of an application to handle auto-capture and robot control as

well as a telescoping aluminum column to hold the user’s personal device. A time of flight distance sensor prevents collisions between the column and elevated surfaces, while an onboard motor rotates the mounted mobile device for 360 degree photo capture. Photos are saved in Google Firebase and can be downloaded from the photo preview page on the mobile application. Configurable settings, such as autonomous capture toggle and variable capture frequency, allow users to customize their SharkCam experience. Controls featured on the settings page allow the user to start and stop the robot from within the application, eliminating the need to switch between the SharkClean and SharkCam applications while using the SharkCam device.

The remainder of this document describes proper usage, basic troubleshooting,

and physical descriptions of the device hardware. Welcome to SharkCam!

# System Overview and Installation

## Overview

The brains of the SharkCam prototype are a Raspberry Pi 3A+ which is contained inside the electronics bay and is powered by a lithium polymer battery. The Pi acts as an intermediary between the mobile app and the robot. When the user starts photo capture in the app, it wirelessly sends a signal to the Pi which then relays commands to the robot to move. The Pi also controls the motor used to rotate the phone, and a distance sensor which is used to detect obstacles before they hit the SharkCam’s column.

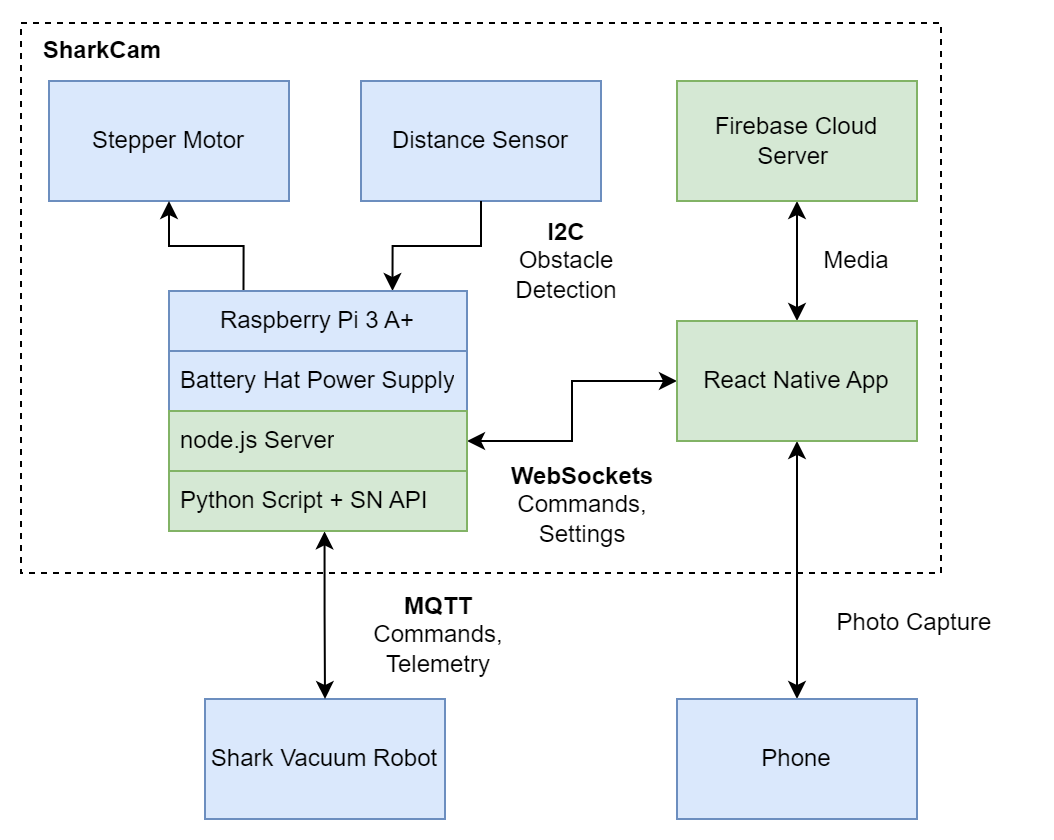


Figure 1: System Block Diagram

## User interface.

The mobile app allows the user to set the desired number of captures during a photo run and the interval between photo captures. It is also where capture can be triggered and photos can be viewed.

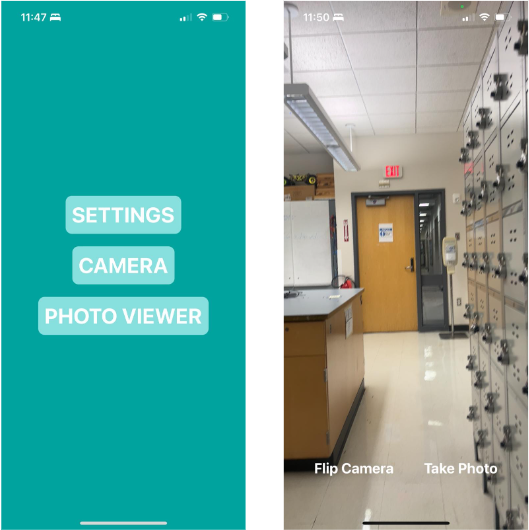


Figure 2: Screenshots of mobile app interface

## Physical description.

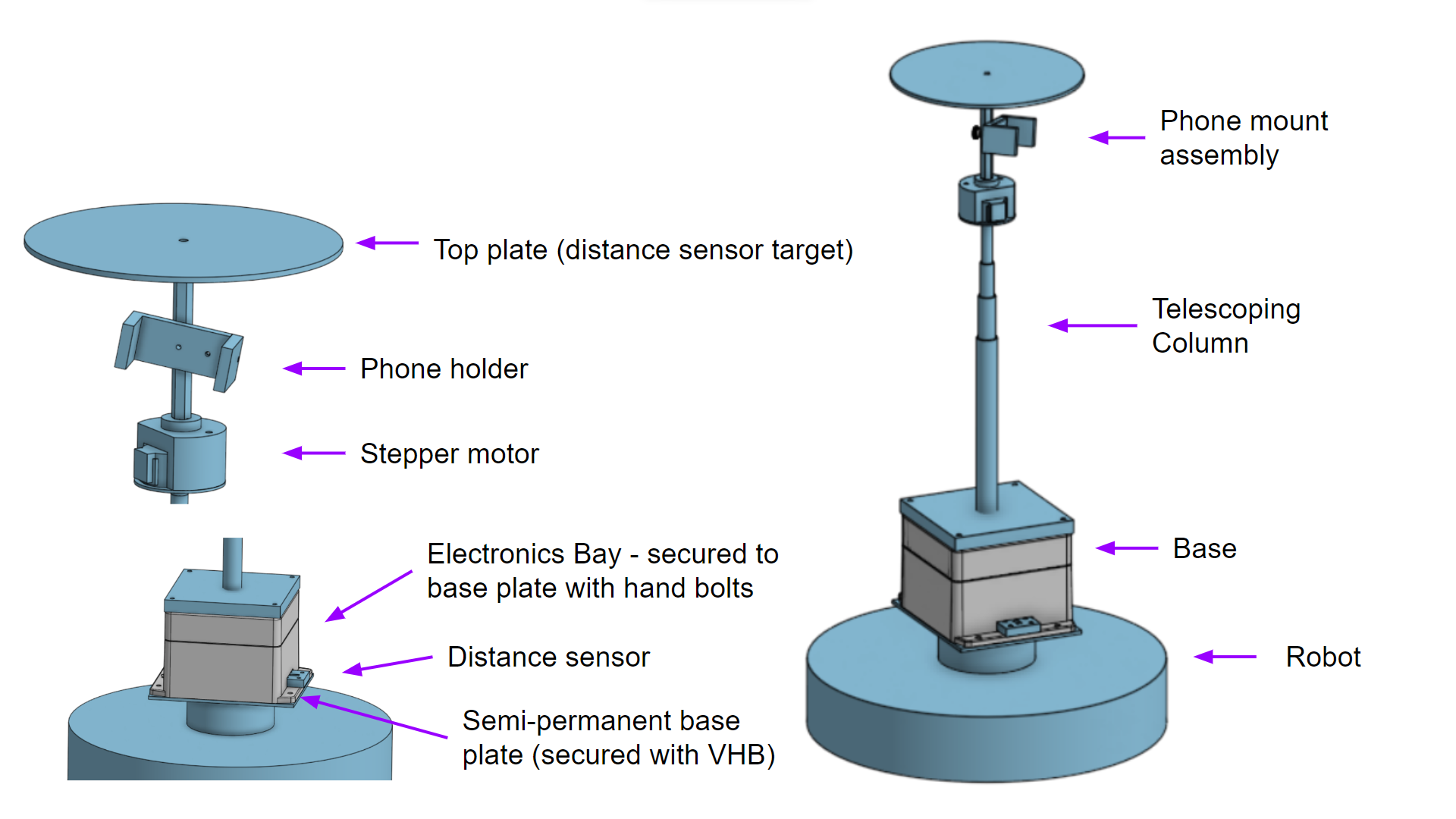
****

Figure 3: Labeled CAD rendering of SharkCam hardware

Sitting above the LIDAR dome of the Shark Robot, the electronics bay houses all the electronics, including the Pi, battery, and distance sensor. The plate on top of the electronics bay is permanently secured to it, and acts as the mount for the telescoping rod. Above that is the phone mount assembly, which allows the user to both secure their phone for photo capture. A plate that sits above the phone works as a reflective surface for the distance sensor which is necessary for obstacle avoidance.

## Installation

To install the SharkCam, the semi-permanent base must first be secured to the top of the robot. Similar to a Command Strip, an adhesive will come pre attached to the bottom of the base. Remove the paper from the bottom of the strip to expose the adhesive. Press the base onto the top of the robot and hold firmly for 30 seconds.

The electronics bay can now be installed. Simply connect the bottom of the bay with the four handbolts. Next, screw in the telescoping column into the top of the bay. The twist locks on the column can be loosened and re-tightened to extend the column and set the desired height.

To store the SharkCam, collapse and unscrew the column. Then, remove the four handbolts to detach the electronics bay. The semi-permanent base can remain and will not interfere with the normal operation of the robot for cleaning. To completely uninstall the base, the next prototype will have a pull tab that will remove the adhesive strip without causing any damage to the robot.

# Operation of the Project

## Operating Mode 1: Normal Operation

The SharkCam works best when the area the robot will be operating in has been mapped. To map a space, the official Shark Clean app can be used. The robot will explore it’s surroundings and use its 360 degree LIDAR and bumper to create a map. It is then highly recommended to set “no go zones” in the app near walls or other obstacles that may interfere with the movement of the robot while the SharkCam column is attached.

For this prototype, the separate SharkCam app must now be installed. In the settings, a capture interval and capture number can be set. The interval determines the amount of time between subsequent photo captures. The capture number sets the amount of photo captures that are taken before the robot returns to the dock.

The phone can then be placed into the mount. Photo capture is triggered by pressing start in the app. The robot will then move about the space, At the specified interval, it will pause in place, and the motor will rotate the phone 90 degrees four times in order to capture in all directions.

If the robot encounters an obstacle entering the path of the column, it will pause and then rotate until the obstacle is no longer in the path of the column. Once the robot has docked, the phone can be removed and photos can be viewed in the app

To charge the Pi, a USB C cable can be plugged into the Battery Hat, which will quick-charge the battery.

## Operating Mode 2: Abnormal Operations

In the event of unexpected behavior, press the “Dock” button on the top face of the robot. The robot should automatically navigate back to the dock. If the robot is unable to dock itself, carefully pick up the robot and manually place it back on the dock. The LED’s should light up to indicate that the device is charging.

One potential issue that may be encountered in this prototype is an infinite obstacle avoidance loop. This SharkCam prototype was developed without access to the firmware of the SharkRobot. This imposes limitations on the ability of the system to control the robot. Because the cleaning autonomous mode of the robot is used for photo capture, the robot has a tendency to move close to walls where obstacles may interfere with the column. It is possible for the robot to end up in an infinite loop where obstacle avoidance is triggered, the robot rotates away from the obstacle, and then once operation is resumed, the robot turns back into the obstacle. A workaround for this issue is to use the official Shark Clean app to set “no go zones” along walls or other obstacles so that the robot avoids them.

In a SharkNinja implementation of the SharkCam, the robot would be able to “remember” where obstacles are the way it does when the bumper is triggered. This would prevent infinite loops.

## Safety Issues

The SharkCam does not pose any additional safety risks to the SharkNinja Vacuum Robot. Be careful when the device is in operation. It might bump into you. It has an obstacle avoidance system to prevent collisions, but the large footprint of the device means that users should still be aware of their surroundings when the device is moving. Users should also be aware of the safety instructions for the Shark Vacuum Robot. They can be viewed below.

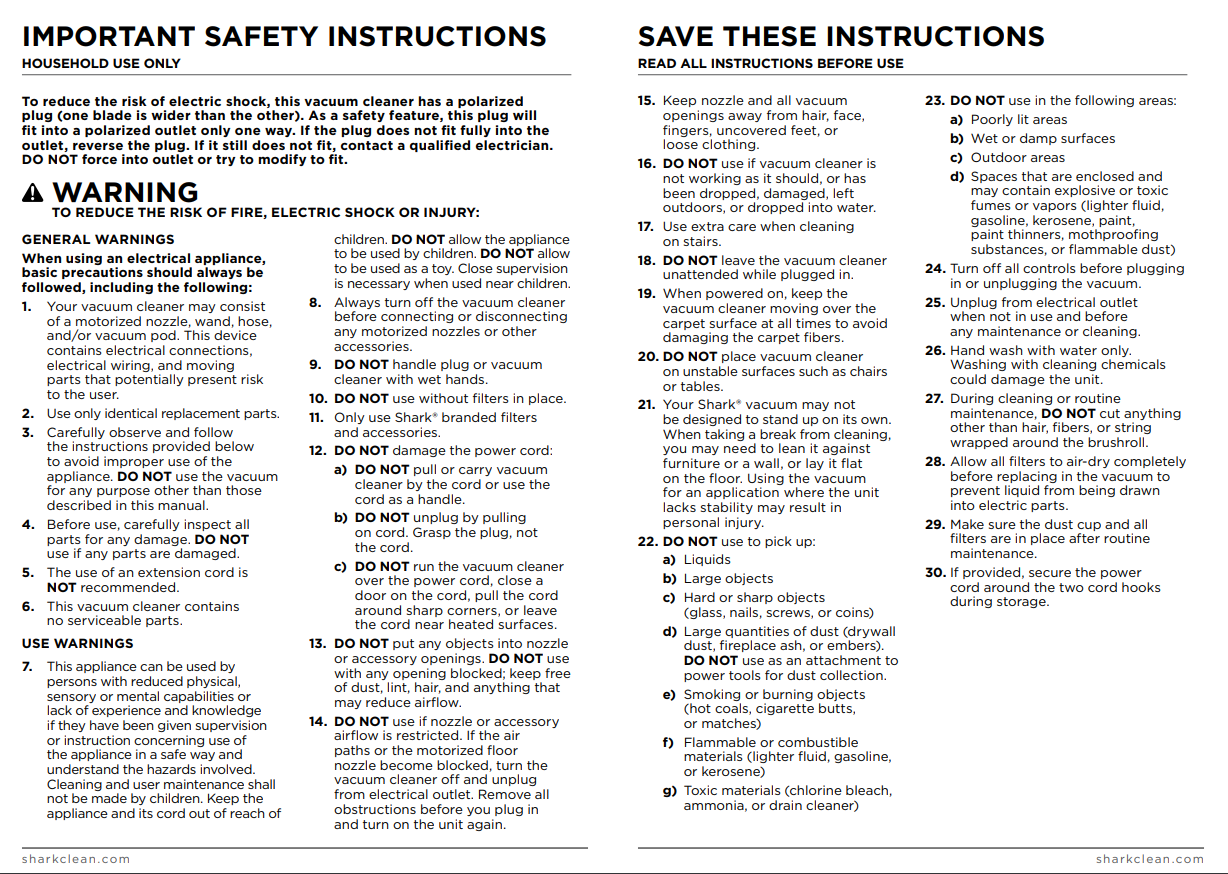


Figure 4: Shark Vacuum Robot Safety Instructions

# Technical Background

The SharkCam utilizes the technology of the Shark Vacuum robot for its navigation. The robot relies on 360 degree LIDAR and bumpers to map its surroundings as well as detect and avoid obstacles in real time. LIDAR works by emitting a laser which bounces off of surfaces near the robot. A sensor then measures the time it takes for the reflected light to return. These measurements are used to build a 2D map. Since the SharkCam adds significant height to the Shark Vacuum, it is necessary to implement a secondary means of collision detection and obstacle avoidance for the added height.

A TOF sensor is positioned at the bottom of the SharkCam and points towards the top plate. TOF sensors work by emitting a laser and measuring the distance based on the time required for it to return. The TOF sensor is therefore pointed towards the top plate so that it can constantly measure the distance between the top and bottom of the SharkCam to create an invisible curtain. If there is an obstacle above the Shark Vacuum’s bumpers and LIDAR dome, the curtain will be tripped as the TOF sensor will detect a change in height, and know to enter into its built in rerouting procedure in order to avoid the obstacle. This can be seen in the Figure below.

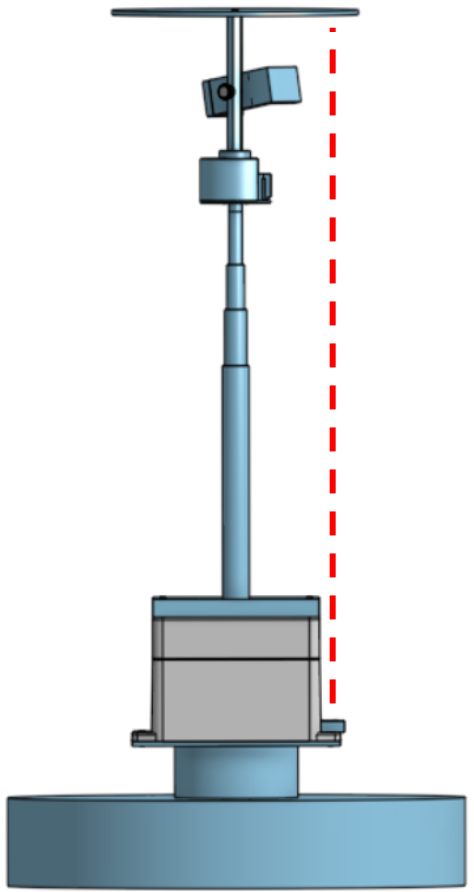


Figure 5: Illustration of the TOF distance sensor for obstacle avoidance

When designing the mechanical system of the SharkCam, it was important to understand the loads that the product would experience given the environments that it would be subjected to.

To control the robot during the photo capture routine, a python script using a SharkNinja provided API is utilized. The multiprocessing package of Python is used to allow for the obstacle avoidance routine to run in parallel with the photo capture code. When an obstacle is detected, the photo capture subprocess can immediately be suspended and alternate obstacle avoidance code can be executed. Once the detection is cleared, the capture process is resumed.

The photo capture routine works by sending a “clean” command to the robot over MQTT. The vacuum power is set to zero for quieter operation, as actually cleaning is not desired. In order to ensure that the robot is in the desired state (cleaning, pausing, docking, turning, etc.), it is necessary to check that the robot has actually received the command. When the robot’s bumper gets triggered, its own obstacle avoidance routine is run which overrides any commands from MQTT clients. To account for this, the python script will continuously send a command until the robot has verified that it is in the correct state.

To facilitate the communication between the Pi and the app, a node server is run on the Pi. Using Websockets, commands can be sent from the app to the node server. The node server can then spawn a python process that executes the robot control code. The mobile app is powered by React Native. React Native is an open source app development framework that allows javascript apps to utilize native platform features. This means the app can run on both Android and iOS, and access features like the camera without needing to go through a web browser.

# Relevant Engineering Standards

The SharkCam system complies with and relies on a number of engineering standards. A selection of relevant standards are detailed below.

Table 1: Engineering Standards

| Standard | Description | Usage in SharkCam | Identifier |
| --- | --- | --- | --- |
| Message Queuing Telemetry Transport (MQTT) | MQTT uses a publish/subscribe messaging model in which publishers send messages to a broker, and subscribers receive messages from the broker. MQTT is highly efficient as it uses small packet sizes and minimal network bandwidth. Therefore, MQTT has become a popular protocol for IoT applications. | Communicate with robot. | (ISO/IEC 20922) |
| WebSockets | WebSockets is a protocol for two-way, live communication between a client and server. This is therefore often used on websites. Unlike traditional HTTP connections which require connections to be established for each request, WebSockets use continuous connections allowing the server to push data to the client in real time. WebSockets are supported by all modern web browsers and can be implemented using languages such as Node.js, Java, and Python. WebSockets main practical advantages include lower latency, reduced bandwidth usage, and improved server stability. | Communication between app and node server on Pi. | (RFC 6455) |
| Universal Serial Bus (USB) C & USB Power Delivery | USB is a standard interface to connect devices to a computer. USB C is the latest USB standard connector type. The USB C connector supports multiple transfer protocols including USB 2.0, 3.X and Thunderbolt. One of the standards supported by the USB port in the Raspberry Pi Battery Hat is Power Deliver, which allows devices to communicate power specifications to allow for fast charging. | Fast charging for battery/Pi hat. | (IEC 62680-1-3) |
| Inter-Integrated Circuit (I2C) | I2C is a communication protocol that enables multiple electronic devices to communicate with each other using just two wires, a clock line (used to synchronize communication) and a data line (used to transmit and receive data). This protocol uses a master-slave architecture, where the master device controls the communication and initiates the transactions, while the slave device responds to its requests. I2C is therefore a popular standard for communication between chips on circuit boards. | Communication between ToF sensor and Pi. |  |
| Wireless Fidelity (WiFi) | WiFi is a family of wireless communication protocols that are used for wireless local area networks (WLANs). This protocol uses radio waves to transmit and receive data using an antenna on the device. The IEEE 802.11 standard defines several variations of the WiFi protocol, each of which offers different speeds, ranges, and features. These protocols use security measures to protect the wireless network from unauthorized access, making it safe for secure data transmission. | Wireless communication standard utilized by Pi, smartphone, and robot. | (IEEE 802.11 family) |
| American National Standards Institute (ANSI) | All fasteners on the SharkCam meet ANSI standards, specifying dimensions, materials, and mechanical properties for all fasteners. This ensures that each is safe and can be interchanged with any other fasteners of the same type and size. |  |  |
| American Society for Testing Materials (ASTM) and International Organization for Standardization (ISO) | ASTM and ISO have developed standards for testing the mechanical properties of various materials, such as tensile strength, compressive strength, shear strength, ultimate strength, etc. These metals include metals, plastics, composites, etc., and have been evaluated in numerous applications. These protocols are used to ensure that products meet minimum performance requirements and can withstand the loads of use. Compliance of these standards is therefore required by regulatory agencies. |  |  |

# Cost Breakdown

The SharkCam prototype developed by Team 28 features a number of redundancies that would be eliminated in a SharkNinja produced implementation of the product. The “beta” version of the SharkCam would be able to roll all the features of our separate app into the official Shark Clean app. This would negate the need to use a full fledged computer like the Raspberry Pi to act as an intermediary between the app and the robot. The Shark Clean app can communicate directly with the robot over a network connection. In place of a Raspberry Pi, a cheaper microcontroller like an ESP32 and a smaller battery could be used to power and control the distance sensor.

SharkNinja’s access to the lower-level firmware of the robot could also allow them to eliminate the stepper motor, instead rotating the robot in place to capture photos in all directions. A stepper motor was used in the “alpha” prototype due to the robot’s slow rotation speed when issued a turn command over the MQTT API.

Costs listed are retail prices without bulk discounts included. Real materials costs for a mass produced product would be significantly cheaper.

Table 2: Project Costs

| Project Costs for Production of Beta Version (Next Unit after Prototype) | | | | |
| --- | --- | --- | --- | --- |
| Item | Quantity | Description | Unit Cost | Extended Cost |
| Telescoping Column | 1 | Modified Testrite Aluminum telescopic tube assembly | $17.42 | $17.42 |
| 3M VHB Adhesive | 1 | 3M VHB 100x200mm Double Sided Foam Adhesive Sheet Tape 5952 | $7.70 | $7.70 |
| Phone Holder | 1 | Manfrotto Universal Smartphone Clamp with ¼ thread connections | $11.95 | $11.95 |
| Distance Sensor | 1 | Adafruit VL53L4CD Time of Flight Distance Sensor | $14.95 | $14.95 |
| Microcontroller | 1 | ESP32 | $19.95 | $19.95 |
| Battery | 1 | Lithium polymer battery | $12.50 | $12.50 |
| Misc plastic components |  | Semi-permanent base, removable base, motor mount, and distance sensor target | ~$10 | $10 |
| **Beta Version-Total Cost** | | | | **$94.47** |

# Appendices

## Appendix A - Specifications

Table 3: SharkCam Specifications

| Battery Life | ~1 hr depending on usage |
| --- | --- |
| Fully Extended Height | 5 ft |
| Weight | 5 lbs |
| Maximum Supported Phone Width | 3.2 in |
| Photo Capture Coverage | 360° |

## Appendix B – Team Information

Sophia Delia is a Computer Engineering major at Boston University College of Engineering. She will be joining Textron Systems Air as a Software Engineer.

*E-mail:* sgdelia@bu.edu.

Liron Dubin is a Mechanical Engineering major at Boston University College of Engineering. He will be completing a master’s in engineering at BU, after which he plans to pursue a PhD.

*E-mail:* ldubin@bu.edu.

Alex Hureaux-Perron is an Electrical and Computer Engineering major at Boston University College of Engineering.

*E-mail:* alexchp@bu.edu.

Brett Senders is a Mechanical Engineering major at Boston University College of Engineering. He will be attending Ecole Polytechnique in France to pursue a master’s in Sustainable Energy Engineering and Management.

*E-mail:* bsenders@bu.edu.

Kevin Vasquez is a Mechanical Engineering major at Boston University College of Engineering.

*E-mail:* vasquezk@bu.edu.



Figure 6: Team poster