School of Engineering Science
Burnaby, BC V5A 1S6
jnaorbe@sfu.ca

June 11, 2021

Dr. Craig Scratchley
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 405W/440 Requirements for OpenSpot

Dear Dr. Craig Scratchley,

In the document attached to this letter, you will find the requirements specification for our product OpenSpot as required for ENSC 405W Capstone A: Project Design, Management, and Documentation. OpenSpot uses mounted cameras combined with computer vision to allow users to view available parking spaces in a parking lot through a mobile-friendly website and observe parking density of certain areas through an LED light indicator placed on light poles throughout the lot.

The purpose of this document will be to detail both the functional and non-functional requirements of our smart parking system. Further, we will be giving details regarding the design and system of our module. An overview of our system in terms of the physical, hardware, software, backend server, and website requirements will also be presented. In addition, there will also be a discussion of sustainability, safety, an acceptance test plan, and engineering standards.

OpenSpot is a team of passionate and dedicated engineering students from both Computer Engineering and System Engineering disciplines: Justin Naorbe, Soroush Saheb-Pour-Lighvan, Curtis Lui, Darius Nadem, and Gurmesh Shergill. Our company is determined to create a product to ultimately alleviate the stress of parking in busy locations and save valuable time.

Thank you in advance for taking the time to read through this document and if there are any further questions or concerns, please do not hesitate to reach out at jnaorbe@sfu.ca.

Sincerely,

Justin Naorbe

Chief Executive Officer
OpenSpot

OpenSpot

ENSC 405W

OpenSpot Requirements Specification

Company 7

Justin Naorbe

Soroush Saheb-Pour-Lighvan

Curtis Lui

Darius Nadem

Gurmesh Shergill

Abstract

Due to increasing populations and driving accessibility, parking in outdoor lots has become increasingly troublesome - especially during peak hours. Finding parking during these times has resulted in relying on luck and being in the right place at the right time. This ends up wasting the time of many people in the search for empty parking spots which may lead to frustrated drivers and disputes. Our team has developed OpenSpot with the goal in mind to make finding an open parking spot stress free and efficient with the added bonus of some security features.

The OpenSpot smart parking system allows users to view available parking spots through a mobile-friendly website. Upon arrival, drivers can view the density of cars parked through an LED light indicator mounted on light poles, which summarizes the availability of open stalls in the area. A detailed overview of our product OpenSpot and its requirements are presented in this document.

Table of Contents

List of Figures and Tables

1 Introduction

The team at OpenSpot is developing a smart parking system that will allow for a stress-free parking experience. Through our cameras and LED lights, we will be able to provide real-time information to the driver at the parking lot or on their mobile device. At the parking lot, drivers will see the density of parked cars via our pole mounted LED lights which display various colours associated with the density/availability. In addition, our website will provide more in-depth information as to which exact parking spots are available or taken at the respective parking lot.

The differentiator of our system compared to the competition is our cost and method of implementation. One single camera covers more parking spots in comparison to having an individual sensor per spot. While providing the same functionality, our system is non-intrusive to the existing parking lot infrastructure and cost efficient for our clients. With our cameras and software application, we will be able to map out the individual parking spots and determine whether they are available. With this information, we will be able to communicate to our LED indicator system to give a real-time update to drivers in the lot. Furthermore, specific information in regard to each parking spot's availability will be viewable on our website.

We also plan to tackle two main issues: the environmental impact caused by cars and the safety and security in parking lots. This will be achieved by streamlining the process for parking. Drivers will no longer need to guess where available spots are and reduce the time driving endlessly only to find that the section, they are in is full. This will lead to a reduction in traffic, of which 30% of all traffic is caused by parking lots, and in doing so we will achieve lower carbon emissions [1]. Vehicle thefts are a growing concern across Canada, as B.C ranks as the 3rd highest province with car thefts and from 2017-2018 car thefts have increased by 16.2% [2]. Our modules will be able to detect car alarms and immediately notify security. The LED indicator lights in the area will begin flashing after detecting a car alarm and will act as a visual aid for the security team in locating the vehicle. In addition to parking availability assistance, OpenSpot aims to tackle the growing concerns in the areas of environmental impact and safety.

1.1 Background

In this section, we will provide some background knowledge on how our technology will work in identifying the parking spots and how we will relay this information. Lastly, we will go over the intended use of our system and how it will work when fully implemented.

1.1.1 Identifying Free Parking Spots

The entirety of our solution relies on our ability to recognize which parking spots are in use or empty. Our module will use a camera to periodically take pictures of the area it monitors and send it back to our server. The backend system uses a trained computer vision model that will detect and recognize when a parking spot is occupied by a vehicle. Subsequently, the result is relayed back to the module on site to update the LED indicator.

1.1.2 Intended Use

The OpenSpot system will initially be targeted towards outdoor parking lots with high traffic. Examples of where it can be applied are in universities, businesses, and supermarkets. Any area where parking is highly contested will be able to greatly benefit from the features OpenSpot offers. The ability to know where parking spots are available in the area through our LEDs and knowing ahead of time where busy areas of the lot are through our website, will decrease carbon emissions and improve the parking experience. The choice of using a computer vision-based system instead of individual sensors opens the opportunity to provide a security measure to each parking lot. In addition to the microphones, we would be able to have surveillance footage through the cameras. Lastly, OpenSpot will grant all visitors an easier time in the search for a spot which may encourage more visitors.

1.2 Requirement Label Scheme

This section describes the convention used for each requirement throughout the document and how to understand what each label represents. The following labelling scheme is how requirements are representing in the rest of the document:

[ReqX.Y.Z.#-S]

X – Represents the section number
Y – Represents the subsection number
Z – Represents the subsubsection number (if it exists)
– Represents the requirement number
S – References the stages for the requirement, either **A**, **B**, or **P**

- **A** – Represents Alpha Phase (Proof-of-concept Prototype)
- **B** – Represents Beta Phase (Engineering Prototype)
- **P** – Represents Production Phase

2 System Overview

This section covers a high-level view and understanding of how our smart parking system will function. The various physical and software components of our design will be discussed. The subsection will briefly cover the purpose of the component and later in the document, specific requirements are presented. This is only a high-level overview and is subject to change, as new discoveries will appear during development.

2.1 OpenSpot Module

The overview of the OpenSpot module is split into the various hardware components that comprise the hardware portion of the system.

2.1.1 Microcontroller

The microcontroller selected for the system is the Raspberry Pi. It will be responsible for facilitating data communication with our backend server as well as executing instructions programmed on it. The hardware connections to the microcontroller are shown in Figure 1. The data communication structure of our system is shown in Figure 2.

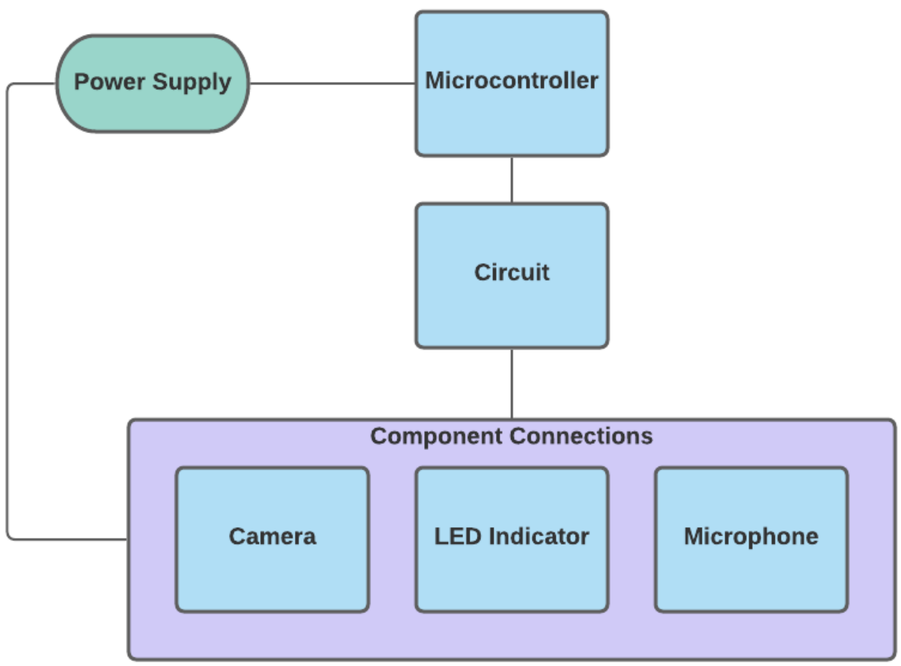

Figure 1: Hardware connections with microcontroller

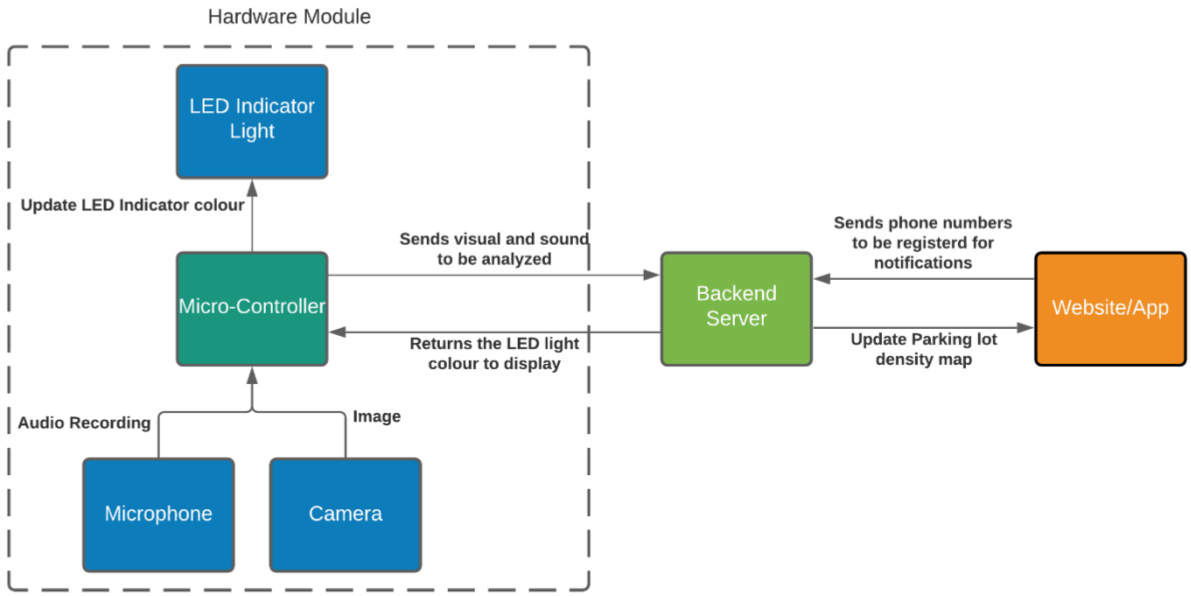

Figure 2: System Block Diagram

2.1.2 Camera

The camera is one of the most important parts of the system. It is responsible for capturing the image which is then analyzed by our program to identify available parking spots. It will take a picture every 30 seconds[1] to provide adequate parking updates. The camera will take an image and pass it to the microcontroller which then sends it to the backend server.

2.1.3 LED Indicator Light

The LED indicators provides drivers real-time information about the availability of spaces around the light pole without having to access our website. The light changes colour according to the percentage of available spots determined by the server. A green light will indicate that there are more than 50% spots available, a yellow light will signify less than 50% spots are available, and lastly red will signify less than 5% of spots are available[2].

2.1.4 Microphone

The microphone is used for the added security feature of detecting car-alarms. This will be done by utilizing a trained audio recognition model for car alarms. We can then interact with the system further by flashing the lights that correspond to the area of the break-in which should not only alert security but also captivate the attention of bystanders so that there may be an increase of eyewitnesses.

2.2 Backend Server System

The backend server system is responsible for all the communication between the module and webpage. The server receives images and audio recordings as a request from the module and performs the necessary computation to generate a response. The server replies to the request with a response

[1] Period of images subject to change
[2] Lighting scheme and values are subject to change as the development continues

indicating what state the LED light should be in. The server also updates a database that contains information on all the spots in the parking lot and sets the occupancy of the respective spots from the analyzed image. The webpage requests data from the server to update the live parking lot floor plan with real-time information.

2.3 Website

The website system will act as a way to relay parking lot information to the users[3] and clients[4]. Based on which one you are, different options will be available to you. Clients can use the website as regular users as well. The website communicates with our backend servers to fetch real-time information stored on the database. The interface for the website has to be pleasing to the eye and easy to use, as it will be the main application that clients and users will interact with when using our Smart Parking System. In the following subsections, we will cover a high-level overview of what each type of user can do on the website.

2.3.1 Client

Clients who use the web interface can register with our portal to gain access to extra features such as receiving notifications if a car alarm is going off. This will allow their security team to respond promptly. In addition, they will be able to view the status of the modules that are out in their parking lots such as: checking if the camera is obstructed, LED lights are damaged, and power loss. This will allow immediate feedback for any maintenance needed. In addition, they can request to have their parking lots set up with our system and contact customer support.

2.3.2 Users

Through the web interface, users will be able to look at a parking lot layout and check which spots are available. Users can also sign up for text notifications via a text message to get updates on parking lots they subscribe to. Unsubscribing will be provided through the website as well as by replying through text.

3 Physical Requirements

The module consists of the camera, LED, microphone, and microcontroller. The physical requirements outline the overall device structure which includes device durability. The requirements are based off the fact that our module is mounted on light poles in an outdoor environment. Making sure the system meets these physical requirements are essential for its functionality and safety.

3.1 Durability

[Req3.1.1-B]: The module surface must be resistant to corrosion and degradation from weather

[Req3.1.2-B]: The internal electronic components must be protected from moisture and condensation build up

[Req3.1.3-P]: The module must be at least IP64 rated

[Req3.1.4-P]: The module must be able to withstand temperatures from -10°C to 50°C

3.2 Wiring

[Req3.2.1-B]: The wires must be insulated and have a protected casing

[3] Everyday drivers who are looking for parking spots

[4] A person or organization using the services and products of OpenSpot E.g., parking lot owners

[Req3.2.2-B]: The wires must be concealed in a wall mount or junction box

3.3 Mounting Bracket

[Req3.3.1-P]: The mounting bracket must be rust proof
[Req3.3.2-P]: The mounting bracket clamps must be adjustable to fit different sized poles

3.4 Weight

[Req3.4.1-B]: The entire module must not exceed 20lbs
[Req3.4.2-P]: The weight of the bracket must not damage the existing light poles

3.5 Electricity

[Req3.5.1-B]: The module must have at least 12V input
[Req3.5.2-P]: The transformer must be able to step down 120V line input to 12V

4 Hardware Requirements

The hardware requirements specified in this section are needed to effectively communicate the data being captured by our system while being able to withstand harsh outdoor conditions. All components mentioned will need to relay their information to the microcontroller so that data can be sent to and from our servers. We have elected to use the Raspberry Pi since it is reliable and supports a wide variety of component interfaces which makes it a suitable for our system. Some key considerations need to be made when selecting the camera such as focal length, aperture and quality [3]. Camera quality is important in getting accurate results from our computer vision model. We will be using LED lights because they are bright, change colour, have a long lifespan and are energy efficient [4]. A reliable microphone is also needed since causing false alarms or not recognizing a break in would devalue the security aspect of OpenSpot.

4.1 Functional

4.1.1 Microcontroller

[Req4.1.1.1-A]: The microcontroller must support a MIPI CSI camera port
[Req4.1.1.2-A]: The microcontroller must be able to update LED lights in real-time
[Req4.1.1.3-A]: The microcontroller must be compatible with our LED light, camera and microphone

4.1.2 Camera

[Req4.1.2.1-A]: The camera must be able to capture an image
[Req4.1.2.2-A]: The camera must be at least 10 Mega-Pixels
[Req4.1.2.3-A]: The camera must be able to capture colored images
[Req4.1.2.4-B]: The camera must be able to capture images during the day and night
[Req4.1.2.5-B]: The camera is able to capture clear low noise image
[Req4.1.2.6-P]: The camera must be able to record at least 1080 p video @240 frame/s
[Req4.1.2.7-P]: The camera's aperture must not exceed f/4
[Req4.1.2.8-P]: The camera's focal length must be between 2- 12 mm for wide angle viewing

4.1.3 LED Indicator

[Req4.1.3.1-A]: The light must be bright enough such that it is visible in direct sunlight
[Req4.1.3.2-A]: The light must be able to change colors
[Req4.1.3.3-B]: The light must be able to flash its colors

4.1.4 Microphone

[Req4.1.4.1-A]: The microphone must have anti-interference technology
[Req4.1.4.2-B]: The microphone must have adequate sound quality distinctly detect sound withing a 10 meters radius
[Req4.1.4.3-B]: The microphone must be able to reduce ambience noise
[Req4.1.4.4-B]: The microphone must be able to connect to a microcontroller

4.2 Non-Functional

4.2.1 Microcontroller

[Req4.2.1.1-A]: The number of input/outputs pins the microcontroller must exceed 14
[Req4.2.1.2-A]: The microcontroller must be powered with an AC-to-DC adapter

4.2.2 Camera

[Req4.2.2.1-A]: The camera must be taking a picture within 5 seconds
[Req4.2.2.2-P]: The camera must be able to function in temperatures ranging from -10˚C to 50˚C

4.2.3 LED Indicator

[Req4.2.3.1-P]: The light must be able to withstand harsh conditions without a protective covering
[Req4.2.3.2-P]: The light must be able to function in temperatures ranging from -10˚C to 50˚C

4.2.4 Microphone

[Req4.2.4.1-P]: The microphone must be able withstand harsh conditions without the need of a protective covering
[Req4.2.4.2-P]: The microphone must be able to function in temperatures ranging from -10˚C to 40˚C

5 Software Requirements

The software hosted on the microcontroller needs to be simple and run efficiently. It is responsible for communicating with the camera, microphone, LED indicator and server. Images and audio recordings will be compressed before sending a request to the server to ensure minimal data being transferred. These images and recordings will not be kept locally on the module to ensure data privacy and security. Figure 3 shows the typical flow of the software running on the module.

5.1 Functional

[Req5.1.1-A]: The software must be able to turn on and off the camera
[Req5.1.2-A]: The software must be able to take pictures with the camera
[Req5.1.3-A]: The software must be able to change the colour of the LED indicator light
[Req5.1.4-A]: The software must be able to send data to the server
[Req5.1.5-A]: The software must be able to receive data from the server
[Req5.1.6-B]: The software must be able to turn on and off the microphone
[Req5.1.7-B]: The software must be able to must be able to record audio clips from the microphone
[Req5.1.8-B]: The software must be able to compress image and audio data
[Req5.1.9-B]: The software must be able to set different light patterns for the LED indicator light
[Req5.1.10-B]: The software must be able to delete data off of local storage

5.2 Non-Functional

[Req5.2.1-A]: The software must change the LED light within 1 second of receiving a response from the server

[Req5.2.2-B]: The software must be able to send data to the server within 5 seconds

[Req5.2.3-B]: The software size must be less than 1GB

[Req5.2.4-B]: Images and audio recordings will not be kept on local storage

[Req5.2.5-B]: Time to restart after failure will be less than 10 seconds

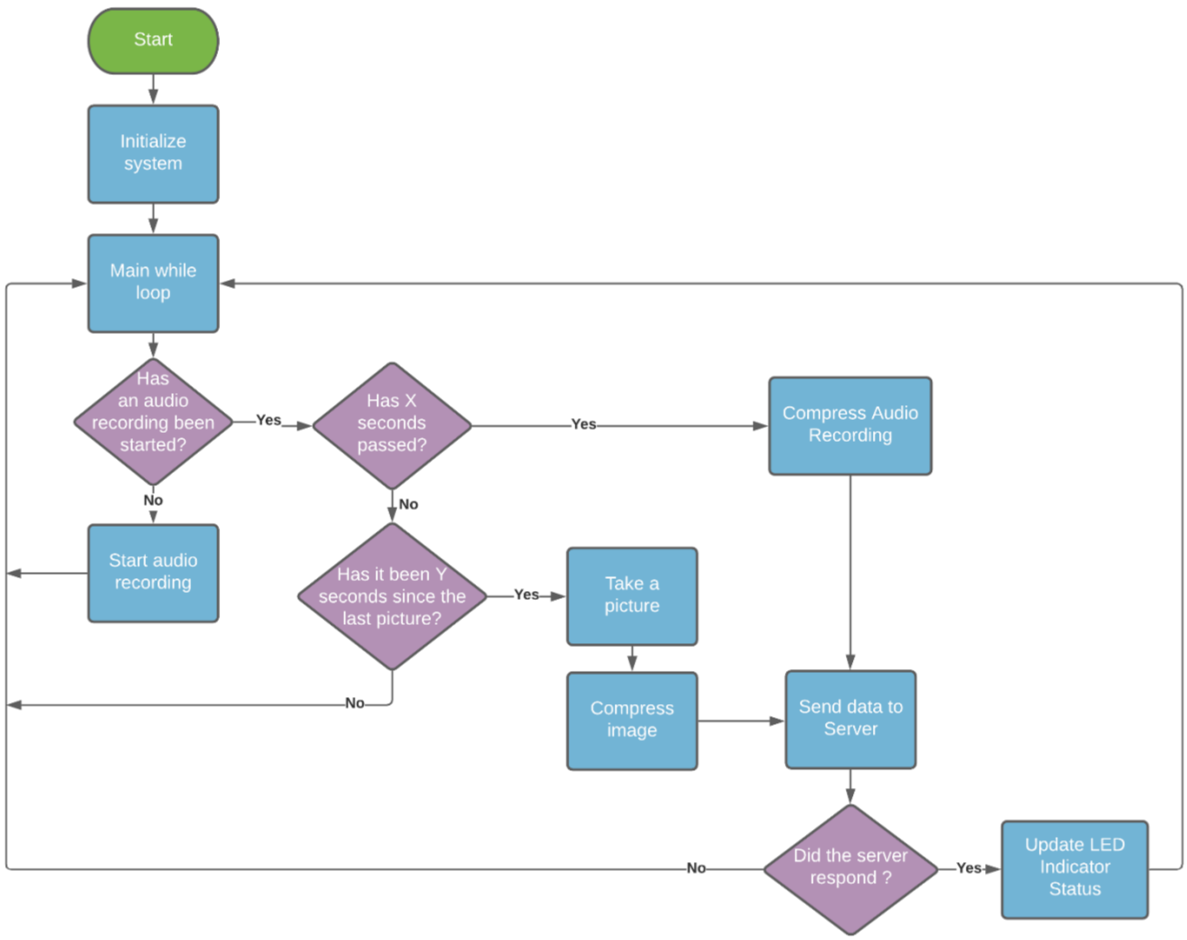

Figure 3: Module Software Flowchart

6 Backend Server Computing Requirements

The backend server is the brains of the operations as it handles all the requests and handles the artificial intelligence to generate accurate responses. Images are received from the module and are processed through the networks to determine the number of cars in the available parking spots. Audio is analyzed for indication of car alarms blaring. After computation, the server sends the appropriate status signal for the module to display on the LED lights. Parking spot availability information is also updated in the database. The backend also responds to website requests to supply real-time parking spot availability. The server will need to respond quickly and efficiently to the requests from the modules and website to provide real-time feedback to drivers.

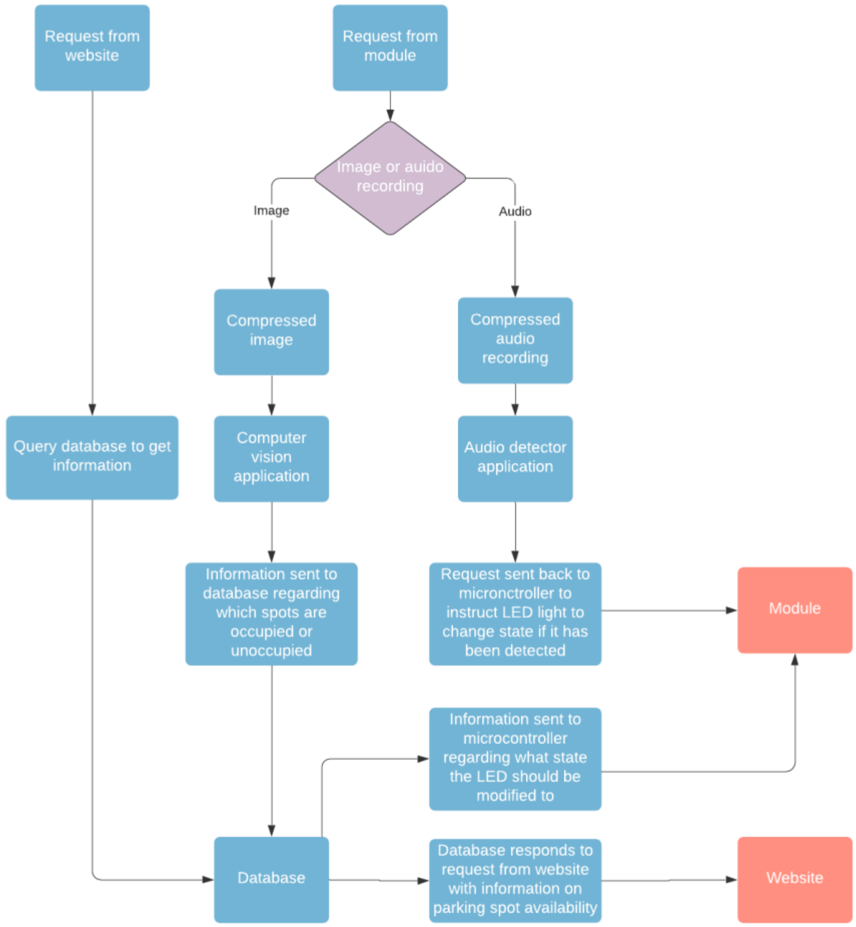

Figure 4: Server Flowchart

6.1 Functional

[Req6.1.1-A]: The server must be able to analyze images and determine the number of cars in stalls

[Req6.1.2-B]: The server must be able to store the occupancy state of each stall in the database

[Req6.1.3-A]: The server must be able to determine the colour that the LED indicator based on the number of cars in stalls

[Req6.1.4-A]: The server must be able to send data to the module

[Req6.1.5-A]: The server must be able to receive data from the module

[Req6.1.6-B]: The server must be able to must be able to analyze audio recordings to detect car alarms

[Req6.1.7-B]: The server must be able to respond to requests from the website

[Req6.1.8-B]: The server must be able to save data about how many cars throughout the week have occupied a particular stall

[Req6.1.9-B]: The server must be able to send data to a database

[Req6.1.10-B]: The server must be able to receive data from a database

6.2 Non-Functional

[Req6.2.1-B]: The server must be able to analyze images within 10 seconds

[Req6.2.2-B]: The server must be able to analyze audio recordings within 10 seconds
[Req6.2.3-B]: The server must be able to respond to website requests within 1 second
[Req6.2.4-B]: The server must have an availability of 90%
[Req6.2.5-B]: The server must keep backup data for a period of 60 days

7 Website Requirements

The website plays a critical role in our Smart Parking System. It will be our main form of communication between the system and clients or users. Requiring the website to be simple to use by all users of varying technical levels. Furthermore, the website will need a fast response rate in order to provide the users with accurate information. The website will be hosted on our backend server and will access our cloud database for user info and camera photos. It is important to know that a client will have the same functionalities as a user and more.

7.1 Functional: User

[Req7.1.1-B]: Website will need to have a response rate of 5 seconds or less
[Req7.1.2-B]: Website will display a visual of the parking lot
[Req7.1.3-B]: Website will display empty and full parking spots of a parking lot
[Req7.1.4-B]: Website will display the parking spot status at a response rate of 30 seconds
[Req7.1.5-B]: Website will allow for phone number input
[Req7.1.6-B]: Website will display a list of parking lots available with the system

7.2 Functional: Client

[Req7.2.1-B]: Client will be able to register an account
[Req7.2.2-B]: Client will be able to log in with the account they created
[Req7.2.3-B]: Client will be able to contact customer support for assistance
[Req7.2.4-B]: Client will receive notifications once registered if a car alarm is ringing at a specific lot
[Req7.2.5-B]: Client will receive notifications if the module is obstructed, or a failure is occurring

7.3 Non-Functional

[Req7.3.1-B]: 'Busy time' information will be listed for each parking lot
[Req7.3.2-P]: Pricing information for a Smart Parking System setup will be listed as a tab
[Req7.3.3-P]: Clients will be able to access the video footage from the camera

8 Sustainability and Safety

The module that will be used for our smart parking system is designed with two aspects kept as priority, safety and sustainability. The module will be packaged and installed with Canadian Electrical Standards (CSA C22) [5] to maintain a safe electrical environment. Furthermore, the various components comprising our system will be made up with plastic, metal, and electrical components that will be recyclable at recycling depots or electronic waste depots. The following subsections discuss how each of our components contribute to making our system as close as possible to an ideal cradle-to-cradle design.

8.1 Sustainability

The team at OpenSpot has made sustainability a number one priority when designing our system. The various physical components will be either recycled or be reusable as a stand-alone device. The physical system consists of a microphone, Raspberry Pi, camera, breadboard, and a LED light. Each of these components when separated will be functional as a stand-alone device, increasing the reusability of our system design as a whole.

8.1.1 Microcontroller

The microcontroller that we will most likely use will be the Raspberry Pi. The Raspberry Pi comes with a free service called PICYCLE, in which you ship the device to the address FREEPOST PICYCLE [6]. The device will be recycled appropriately by the manufacturers. In addition, the device can be re-used and sold as a second-hand microcontroller.

8.1.2 Camera

Major E-Waste companies accept security cameras of any nature to be recycled [7]. Before giving the device away if it comes with any local memory, we will need to wipe it appropriately to prevent any sensitive data from being leaked. Three safe methods to clean the drive would be degaussing, punching, and finally shredding [8]. After a clean wipe, the camera could also be sold as a second-hand device, increasing the sustainability of our module.

8.1.3 Lighting

LED lights are easily recyclable as they are composed of 88% glass, 5% metals, and 7% other materials. Glass can be recycled at any local recycling depot alongside the metal used. The other materials are composed of electronic parts, which are recyclable at any electronic depot [9].

8.1.4 Microphone

Microphones are generally comprised of a coil, coil connectors, magnets, and plastic shell casing with elastic connections. Making the various components recyclable; however, it will need to be broken down to be recycled appropriately. Microphones are very popular second-hand as well, allowing this component to have a high reusability rate.

8.2 Safety Requirements

[Req8.2.1-B]: The exterior of any physical components must not contain any sharp edges that could potentially damage wildlife or humans

[Req8.2.2-B]: All wires must be properly insulated

[Req8.2.3-B]: Power source must be through a safe connection so that it is tamper proof and protected from being unplugged

[Req8.2.4-B]: The attachment of our physical system on a light pole (or any pole with significant height) must be securely locked to prevent any component from falling

[Req8.2.5-P]: The data we gather from clients and users must be stored on secure databases following Canadian guidelines for cybersecurity

[Req8.2.6-P]: Cameras must be on a protected network to prevent hacking

[Req8.2.7-P]: Brightness of the light source must be safe enough for human vision to prevent any sign damage

9 Acceptance Test Plan

For the proof-of-concept phase, we would be attempting to test and evaluate OpenSpot's core functionalities. The main focus is helping the user acquire an available parking spot that is out of view.

This would be accomplished if the core components of the OpenSpot module are able to directly communicate with one another and successfully connect to the backend server.

The main hardware component to be tested is the microcontroller since it directly facilitates the communication between the other hardware components and the server. We will also test the server's ability to recognize available and taken parking spots. The connectivity between these two would need to prove to be consistent as our main goal is to provide real time information for the clients.

During the proof-of-concept phase we will be testing the fundamental features of our system. The table listed in the appendix, describes the components we plan on testing and the various methods of how we will evaluate the success.

The following is a list of features that will be presented during the 405W demo:

- Module can take pictures and send images to a local server (running on laptop)
- Local server receives image and puts it through the computer vision model
- Computer vision computes number of cars in parking spots
- Local server responds to module with the light colour
- Module receives response and turns LED light to specified colour

Specific requirements that will be ready for the demo are listed in the Requirement Label Scheme – which are prefixed with an '**A**'.

10 Engineering Standards and Laws

Following engineering standards are important to our system as it requires tapping into the existing power within light poles, taking pictures, recording audio, and dealing with sensitive data. The following standards OpenSpot will follow are published by IEEE, ISO, and CSA. This covers smart parking systems, electrical, security, quality, and software standards.

10.1 Smart Parking Standards

ISO 37163:2020	Smart parking community infrastructure. Procedures for installing solutions to help drivers quickly find parking lots with available spaces [10]

10.2 Physical Standards

IEEE C57.12.31-2020	Tests and requirements for pole-mounted equipment [11]
IEC 60529:1989	Classifications of degrees of protection provided by enclosures for electrical equipment (IP Rating) [12]

10.3 Electrical Standards

IEEE/ANSI C57.1, C57.2 C57.3-1942	Standards for transformers and stepping down power from the light poles [13]
CSA C22.1-18	Safety standards for electrical installations [5]

10.4 Software Standards

ISO/IEC/IEEE 12207-1996	Outlines the software life cycle processes. Framework for developing and managing software [14]

IEEE 829-1983	Software test documentation associated with execution of procedures and code [15]
C/SAB/IEEE P2671	General requirements of online detection based on machine vision [16]

10.5 Security Standards

ISO/IEC 27000:2018	Definitions for information security management and security techniques [17]
ISO/IEC 19566-4:2020	Privacy and security features for JPEG box-based file format systems [18]

10.6 Privacy Laws and Acts

Although the following set of privacy protection laws and acts are not considered "Engineering Standards", they have been added to acknowledge personal privacy and our usage of potentially sensitive data. OpenSpot has considered these and plans to uphold it to the standard of the law.

PIPEDA S.C, c.5	Protection of personal information in the private sector [19]
PIPA S.B.C, c.63	Collection and usage of personal information [20]

11 Conclusion

OpenSpot provides consumers with a system that makes parking in busy areas efficient and effortless. Our camera and LED light system provides drivers with the information they need to identify open parking stalls and prevent potential frustration caused by the inability to find an available spot. The mobile-friendly website will allow users the ability to view open parking stalls in advance of their arrival. In addition, they will be able to view the parking density in a particular area of the lot through the LED light indicator to provide a summary of how many stalls are occupied at that moment of time. The microphone placed on our cameras will be able to detect car alarms that alert security in that lot while also providing flashing lights through the LED indicator to scare off any criminals.

The requirement specifications provided in our document for OpenSpot address areas including system, physical, hardware, firmware, back-end server, and website requirements. We have also included a discussion of sustainability and safety, an acceptance test plan, and engineering standards. The alpha phase of our product will be completed by August 2021 with the beta phase finished by December 2021.

For any situation you may be in, we will always be able to help you find an open spot.

12 Glossary

AC	Alternating current
ANSI	American National Standards Institute
Anti-Interference	Equipment, process, or technique used to reduce effect of natural and man-made noise
B.C.	British Columbia, province in Canada
Clients	A person or organization using the services and products of OpenSpot E.g., parking lot owners
CSA	Canadian Standards Association
DC	Direct current
GB	Unit symbol for gigabyte
IEC	International Electrotechnical Commission
IEEE	The Institute of Electrical and Electronics Engineers
IP Rating	Ingress protection (IP) ratings – grading of the resistance of an enclosure against the intrusion of dust or liquids.
ISO	International Organization for Standardization
LED	Light Emitting Diode
MIPI CSI	Mobile Industry Processor Interface – Camera Serial Interface
OpenSpot	Capstone Company 7 who are Developing and creating the smart parking solution
PICYCLE	Raspberry Pi recycling program
PIPA	Personal Information Protection Act – British Columbia
PIPEDA	Personal Information Protection and Electronic Documents Act – Canada
Raspberry Pi	Small programmable single board computer
SAB	Standards Activities Board
Users	Everyday drivers who are looking for parking spots

13 References

[1] IBM, "IBM Global Parking Survey: Drivers Share Worldwide Parking Woes," IBM, 2011. [Online]. Available: https://newsroom.ibm.com/2011-09-28-IBM-Global-Parking-Survey-Drivers-Share-Worldwide-Parking-Woes-1.

[2] P.Cline, "Theft from vehicles hits highest rate in 13 years, violent crime down: VPD," CTV, February 2019. [Online]. Available: https://bc.ctvnews.ca/theft-from-vehicles-hits-highest-rate-in-13-years-violent-crime-down-vpd-1.4308828. [Accessed June 2021].

[3] Yolanda, "reolink," 2018. [Online]. Available: https://reolink.com/how-to-choose-best-security-camera-lens/. [Accessed 10 June 2021].

[4] "BENEFITS OF OUTDOOR LED LIGHTING," March 2020. [Online]. Available: https://sitlersledsupplies.com/benefits-outdoor-led-lighting/. [Accessed June 2021].

[5] "C22.1-18 Canadian Electrical Code, Part I (24th Edition), Safety Standard for Electrical Installations," 2015. [Online]. Available: https://www.csagroup.org/store/product/C22.1-18/. [Accessed June 2021].

[6] Picycle, "Picycle," Picycle, [Online]. Available: https://picycle.org. [Accessed June 2021].

[7] Recycle My Electornics, "What Can I Recycle," EPRA, [Online]. Available: https://www.recyclemyelectronics.ca/bc/what-can-i-recycle/. [Accessed June 2021].

[8] AVA Recycling Blog, "Security Camera Recycling," Ava Recycling, [Online]. Available: https://www.avarecycling.com/blog/security-camera-recycling. [Accessed June 2021].

[9] Any-Lamp, "Can LED light be recycled?," Any-Lamp, 2021. [Online]. Available: https://www.any-lamp.com/blog/can-led-light-recycled. [Accessed 12 June 2021].

[10] "ISO 37163:2020 Smart community infrastructures — Smart transportation for parking lot allocation in cities," ISO, September 2020. [Online]. Available: https://www.iso.org/standard/69249.html. [Accessed June 2021].

[11] "C57.12.31-2020 - IEEE Standard for Pole-Mounted Equipment--Enclosure Integrity," Feb 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9366654. [Accessed June 2021].

[12] "IEC 60529:1989+AMD1:1999+AMD2:2013 CSV Consolidated version," IEC , August 2013. [Online]. Available: https://webstore.iec.ch/publication/2452. [Accessed June 2021].

[13] "ANSI C57.1, C57.2 C57.3-1942 - ANSI American Standards for transformers, Regulators, and Reactors," 11 1942. [Online]. Available: https://standards.ieee.org/standard/C57_1,C57_2C57_3-1942.html#Standard. [Accessed June 2021].

[14] "IEEE 12207-1996 - ISO/IEC International Standard - Information Technology - Software Life Cycle Processes," December 1996. [Online]. Available: https://standards.ieee.org/standard/12207-1996.html. [Accessed June 2021].

[15] "829-1983 - IEEE Standard for Software Test Documentation," Feb 1983. [Online]. Available: https://ieeexplore.ieee.org/document/573169. [Accessed June 2021].

[16] "P2671 - Standard for General Requirements of Online Detection Based on Machine Vision in Intelligent Manufacturing," December 2017. [Online]. Available: https://standards.ieee.org/project/2671.html. [Accessed June 2021].

[17] "ISO/IEC 27000:2018 Information technology — Security techniques — Information security management systems — Overview and vocabulary," February 2018. [Online]. Available: https://www.iso.org/standard/73906.html. [Accessed 2021 June].

[18] "ISO/IEC 19566-4:2020 Information technologies — JPEG systems — Part 4: Privacy and security," Mar 2020. [Online]. Available: https://www.iso.org/standard/73607.html. [Accessed June 2021].

[19] "Personal Information Protection and Electronic Documents Act S.C. 2000, c. 5," April 2003. [Online]. Available: https://laws-lois.justice.gc.ca/eng/acts/p-8.6/page-1.html#h-416888. [Accessed June 2021].

[20] "PERSONAL INFORMATION PROTECTION ACT [SBC 2003] CHAPTER 63," October 2003. [Online]. Available: https://www.bclaws.gov.bc.ca/civix/document/id/complete/statreg/00_03063_01. [Accessed June 2021].

[21] SeeedStudio, "SeeedStudio The IoT Hardware Enabler," [Online]. Available: https://www.seeedstudio.com/Raspberry-Pi-4-Computer-Model-B-4GB-p-4077.html. [Accessed 11 June 2021].

14 Appendix

14.1 Acceptance Test Plan

Table 1: Prototype Acceptance Test Plan

OpenSpot

ACCEPTANCE TEST PLAN

Stage *Alpha Prototype*

Purpose	Test Description	Acceptance Criteria
Validate connection between microcontroller and peripherals	Microcontroller takes a picture using the camera. Microcontroller changes the colour of the LED.	Image is viewable as a local file on the microcontroller. Microcontroller must be able to change the colors of the connected lights.
Validate local server connection with microcontroller	Apply tests of sending various data to and from the microcontroller to the backend server. Server can respond with what the LED colour should be.	The local backend server receives the file. The local server can tell the microcontroller to change the LED colour.
Validate system can correctly detect empty and taken parking spots	Input various images of parking lots into the system. Images should range in size of parking lots and amount of cars.	The system should be able to correctly identify where the parking spaces are, which objects are cars, and which parking spots are empty or taken.
Complete system test	Set up a module in a parking lot. Have the module take a picture and run it through the system once.	Module takes a picture and successfully sends the image to a local server. The local server is able to compute how many cars are in the image. Local server can tell the microcontroller what colour to set the LED.