Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Semester Report

Ecobin

Submitted to

Benjamin Cootner
(631) 880-2195
bcootner@bu.edu

by

Team 9
Team Members

Aditya Wikara adwikara@bu.edu
Hayato Nakamura hayaton@bu.edu
Kevin Sinaga kfsinaga@bu.edu
Charles Thao cthao19@bu.edu
Shreenidhi Jayaram shreej@bu.edu

Submitted: 12/11/2018

Executive Summary

Ecobin

Team 09

Ecobin is a smart trash can that automates recycling, making it an easier task in a modern home and therefore aiming to reduce garbage disposals to landfill and hence carbon footprint per American household. All the user has to do is throw trash into the device, and Ecobin will automatically sort it into its appropriate compartment: recyclable or non-recyclable. The logic of Ecobin lies in the embedded system powered by a Raspberry Pi and several sensors. Its software functionality relies on computer vision, image recognition and cloud computing to determine the recyclability of an object. Ecobin also comes with an iOS application with an user friendly interface. In short, it is an up-and-coming Internet-of-Things device that would change and expedite recycling in an American household

1.0 Introduction

The origin of Ecobin stems from the dilemma that Mr.Benjamin Cootner, a Boston University alumnus and software engineer, faced when trying to categorize his disposal between recyclable and general waste.As many studies and surveys suggest, many people lack the knowledge of how and what to recycle, or, in Mr. Cootner's case, hesitate due to the hassles of separating their wastes into different compartments. As a result, they put all items one trash bin. As Mr. Cootner, Ecobin's primary customer, expressed, he and many households believes in the importance of recycling, but the inconvenience or lack of knowledge often hinder them from doing so.

Ecobin exists to solve the pain point shared by Mr. Cootner as well as many Americans. It is a smart trash can which automates recycling and makes it an easier task in a modern and smart home. Ecobin acts as a replacement for existing trash cans while providing an intelligent and friendly user interface. In addition to a powerful image recognition algorithm, it comes with an iOS application as a its user interface, which tracks user recycling performance as well as remind the user when the bin is full. In order to further encourage the user, this application will also shows metrics such as the amount of carbon reduction due to the their recycling habits. In short, Ecobin attempts to promote an eco-friendly and fun way to recycle.

In a more societal sense, Ecobin's dedication to promote a more sustainable lifestyle aims to curb the effects of climate change, which had dominated many intellectual forums for its destructive effects on human lives. It has caused from torrential flooding in Asia to monstrous wildfires in California, drove 50% of Earth's marine life to extinction, and imposed many public health crises around the globe. According to many scientific studies, these traumatic events stem from the rise of industrial activities and expansion of urban life, of which garbage disposal is a major consequence. Careless and unsystematic handling of waste have caused health concerns around the world. As a response, the United Nations and governments have been promoting recycling materials in order to curb the effects of climate change. Despite various efforts to raise awareness in the media as well as in education, the results do not seemed substantial: In the United States, while 80% of materials are recyclable, some 55% of materials end up in landfills; much of which is plastic wastes that can take up to 450 years to decompose. Within the city of Boston, a city with its progressive environmental agenda, only 21% of disposed materials are recycled in 2017. Actions are required to reduce the effects of climate change.

As the world continues to embrace the advancement and introduction of Internet-of-Things devices, not only would Ecobin promote eco-friendliness, it would also expedite a smarter and automated lifestyle. First, using Ecobin decreases the amount of trash that would eventually end up in landfill, improving the sustainability of a household. The ultimate goal of Ecobin is to vouch for a greener life to achieve lower carbon footprints in American households. With the user interface, a user could easily check the current capacity of their bin or be reminded by the application when it is time to empty the bin.

Behind the interactive front end, Ecobin combines a set of modern technologies for its functionality: microprocessing, computer vision and cloud computing. Its hardware, comprising of sensors as well as a camera module, is backed by a Raspberry Pi (henceforth RasPi or Pi) which handles signal processing as well as image capturing. On the cloud, the software back end of Ecobin primarily includes an image recognition algorithm which automatically processes and categorizes an object placed in the bin as recyclable or non-recyclable. This setup renders a smooth user experience.

2.0 Concept Development

2.1 Analysis of Customer's Expectations:

The primary expectations from the customer were:

1. System can differentiate between trash and recyclable items
2. A sorting system for given item to go into the respective bin
3. An indicator denoting when the trash can is full
4. Use motion sensor technology to switch on the trash sorting system when the item is thrown in

The secondary expectations:

1. Display the timeline of decreasing carbon footprint of user
2. Data analytics providing suggestions on how to minimize waste over time
3. System does not need to be battery powered and can be connected to a power source

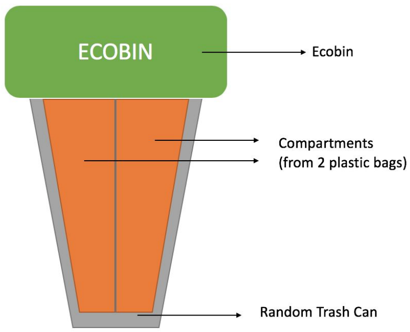
Appendix 7.1 shows the engineering requirements for the Ecobin, which incorporates both primary and secondary expectations of the customer from a more technical standpoint.

2.2 Overview of Preliminary Hardware and Software Design Considerations:

Hardware: After the client meeting, the team discussed the possibility of using a microprocessor, resulting in a decision to employ a Raspberry Pi 3B+. To detect the object, we would also need a system of sensors. The potential use of an infrared or ultrasonic sensor was discussed in order to be able to detect the object when thrown in and potentially even be placed strategically to indicate when the trash bin compartment was full. An integral part of customer requirement was to successfully and accurately distinguish between recyclables and trash.For this purpose, team members used the a Raspberry Pi as a control hub, with the aid of Pi camera modules.

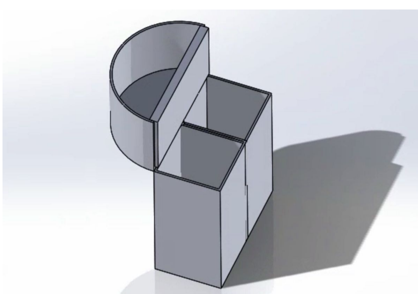
Software: To make a smart system that automatically sorts waste, the CE team thought of using machine learning and Artificial Intelligence in order to accomplish this task. Members decided on using the TensorFlow Neural Network backend with the Keras API in order to be able to accomplish this task. However, a major limitation in the development of this idea was the fact that the RasPi would not be able to process object detection. To counteract this, all software members decided to research OpenCV methods which involved using a combination of contouring, gradient analysis, shape recognition, watershed algorithm and other standard computer vision methods.

2.3 Overview of Preliminary Mechanical Design Considerations:



a. Initial design: With a minimalistic knowledge in Mechanical Engineering design, Ecobin planned to build a smart lid for current trash cans. Instead of designing a fully functional trash can, the initial design will receive trash by means of a conveyor belt technology and sort it to the desired compartment. There will be compartments partitioned by means of two plastic bags attached to the bottom face of the Ecobin.

Figure 2.1. Ecobin first design



b. Second design: The second design was completed along with the 4 additional Mechanical Engineering members who joined the project after the preliminary design. At that point, both ME and ECE members concurred to design an entire bin instead of just a lid. Figure 2.2 is a CAD diagram for the second design. The semicircle structure is where the trash is placed. The straight wall of that structure will rotate due to a motor and mechanically push the trash to one of the two compartments.

Figure 2.2. Ecobin second design with Mechanical Engineering members

c. Current design: Refer to Appendix 7.3

2.4 Conceptual Approach:

In order to fulfill the engineering requirements, the team first developed a user story after considering several dilemmas in recycling. First, when a piece of paper is not so clean and crumpled, the user might think of it as trash. However, it might be recyclable since paper is recyclable. For an additional example, when someone throws away a half-full cup of coffee, they might think it is recyclable since the cup is recyclable but it should actually be sorted into the trash compartment as it has liquid material inside. These types of dilemmas are issues to which the Ecobin should be able to solve.

After researching further into concepts in recycling, the team determined that having false negatives is preferable as opposed to false positives (False negatives are recyclable objects that are detected as trash while false positives are trash that are detected as recyclable). Having false positives poses an issue because these non-recyclable items could damage the recyclable items in the recycling plant, or even damage the system's machinery. In addition, having false positives also waste time and resources within the recycling facility. Therefore, while it is most ideal to have true positives and true negatives, we decided to have false negatives rather than false positives. To be able to eliminate false positives, the team decided to have the default label for the detected object as trash, and to only label it as recyclable when the algorithm allows it.

In order to further satisfy the client's requirements, as well as the preliminary design considerations, the proposed conceptual design will utilize a cloud-based neural network model. On the hardware side, Ecobin uses a RasPi, a camera, a motion sensor, a ultrasonic sensor, and a lighting system. This concept will be implemented into an actual full size trash can, which will consist of an automatic lid, a sorting mechanism, and two compartments for trash and recyclable. A need for cloud computing rose in the days leading up to the prototype testing, when the Raspberry Pi 3 failed to run a deep learning model. Due to the facts that it only has 1.4Hz quad core CPU and 1GB memory, the team increased the swap memory to 2GB. However, the model still failed. After realizing the computational limitations in the RasPi, a cloud instance was used in order to allow more fast and powerful processing.

Furthermore, the team considered alternative solutions to the engineering requirements. First, the team initially decided to use OpenCV for the image recognition process by analyzing shapes and contours. OpenCV aids in image recognition and identification without having to use the processing power of machine learning and neural network libraries. However, the software team found it more time and resource efficient to implement Keras as members decided to go with deep learning to detect and classify a wider variety of items. Second, the team initially wanted to use a LED and a diffuser for the lighting in the object detection system. However, members realized the inefficiency in building a diffuser, and decided that using an LED strip would be more efficient. Last, for the sorting mechanism of the project, initial approaches were to use a conveyor belt type of design. However, there were issues with keeping the conveyor belt hygienic and configuring it to move in both ways (front and back). Therefore, Ecobin ended up using a simple motor for the sorting mechanism, details can be found in section 7.3.

3.0 System Description

3.1 System Overview

There are two major components within the Ecobin: an object detection system and a sorting mechanism. The sorting mechanism is a collaborative effort between the mechanical engineering team and the ECE team to deliver a fully integrated electro-mechanical system. The object detection system utilizes both hardware and software. The hardware aspects incorporate a Raspberry Pi, camera, motion sensor, ultrasonic sensor, and a lighting system. The software aspects include the sorting algorithm, a cloud instance, and a mobile application. Section 3.2 and 3.3 gives a more detailed discussion on the hardware and software system respectively.

3.2 Hardware System

The hardware system integrates several components to enable the object detection functionality. The main components include:

1. RasPi acts as the controlling unit
2. Pi Camera module
3. Passive Infrared (PIR) motion sensor for motion detection
4. Ultrasonic sensor
5. LED strip lighting system
6. Indicator LEDs

The first prototype incorporates all hardware components of Ecobin, except for the ultrasonic sensor. The RasPi functions as a relay station of Ecobin: it receives and transmits signals to the camera, the motion sensor, and the lighting system. Additionally, when the cloud relays a signal to the Raspberry Pi, the red and green indicators LEDs will announce whether the object is recyclable. Noticeably, the product will utilize two power supplies, because the RasPi draws 5V, while the LED strip draws 12V, which is controlled by a BJT transistor. Figure 3.1 shows a detailed hardware schematic of the Ecobin encompassing all major functionalities.

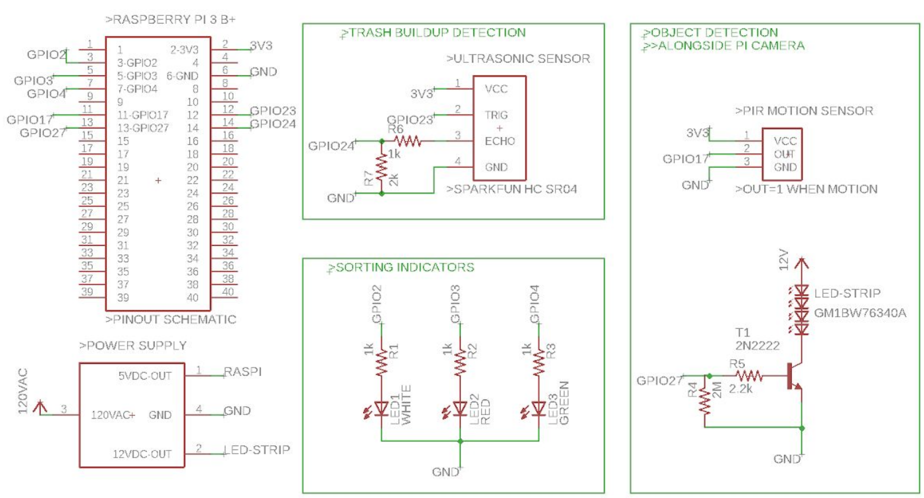


Figure 3.1. Hardware Schematic

First, the top left represents a pinout diagram of the RasPi and shows where the ports are connected to. The GPIO ports are essential for transmitting and receiving signals with the sensors, while the remaining ports, like 3V3 or GND, control power supply of the sensors. For instance, the rightmost represents the motion sensor. The 3V3 port from the RasPi is connected towards the VCC of the motion sensor, GND is connected to GND, and a GPIO port is connected towards the output of the motion sensor in order to send information whenever a motion is detected. The LED indicators, LED strip (including the BJT transistor), and ultrasonic sensor all utilize similar mechanisms. The ultrasonic sensor schematic reflects a planned design since it belongs to the second semester's plan.

Overall, the hardware system starts with the motion sensor when there is hand motion, this signal triggers the lighting system as well as the camera where an image is then taken. The image is then passed over to the cloud instance where it is processed as trash or recyclable. Once the processing is done, the RasPi received this signal and outputs it into the electro-mechanical tool for the mechanical sorting to take place. For the prototype, these signals were driven to a green LED for recyclable, and a red LED for trash.

3.3 Software System

The software portion of Ecobin requires a multitude of dependencies in which it successfully handles image processing and deep learning. All of the deep learning as well as the sensor control scripts are developed in Python, which include the following dependencies:

1. keras.preprocessing.image
2. GPIO Processing
3. OS

A critical functionality of the project is to correctly identify the category of trash with a success rate of 70%. To maximize this percentage, the first semester prototype uses a 16-layer convolutional VGG network that utilizes a pre-trained network of over 2 million photos. These photos include not only trash but regular object images, which help increase the accuracy of the detection algorithm. A constraint of this system is that this Keras network with a Tensorflow backend may only output a string. This specific string describes what the network thinks the object in the image should be. However, for the purpose of the first semester prototype, a simpler output would be in binary format. In order to alleviate this issue, the system incorporates a *dictionary-search* method of which it would search the recyclable object dictionary for any matches.

Through ImageNet, Ecobin had amassed over 5000 photos of plastic recyclable trash, and ran all images through the Keras deep learning network explained in the section above. The output string of each image is then stored into a text file which acts a dictionary. The general flow of the software system is that for every new waste photo taken, the Keras script will output a string that describes this image. This string is then searched through the text file of key-words related to recyclable plastic. If and when a match is found, the python script will output a '1' back to the Raspberry Pi, and when there is no match it will output a '0'. From this method, the software system will generate a binary output, making the sorting function simpler, and more efficient.

Image processing tasks have been resource intensive, so data-preprocessing is necessary for fast and efficient recognition. While Raspberry Pi can take high-quality 5 megapixel photos, the Keras script does not require such heavy and computationally exhaustive arrays of pixels. Therefore, before the image is sent to the VGG model, it is resized to a lower resolution. At the moment, the team found out that images with a 500 x 500 pixels resolution perform well with the VGG model without losing output quality.

As mentioned, the image recognition back end was originally planned to run on the Raspberry Pi, but later migrated to the cloud for more robust and powerful processing capability. Currently, the image recognition script is running constantly on Boston University Shared Computing Cluster. With low traffic, it offers nearly limitless computing powers for the project, with 256GB memory, 2 fourteen-core Intel Xeon CPUs and 10Gbps Internet. The desired communication channel between the cloud and the Raspberry Pi will take place as a RESTful API for stable use. So far, it takes 0.85 second for a round trip (starting from the Raspi sending the image to the cloud to the cloud relaying back an output from the classification Python script).

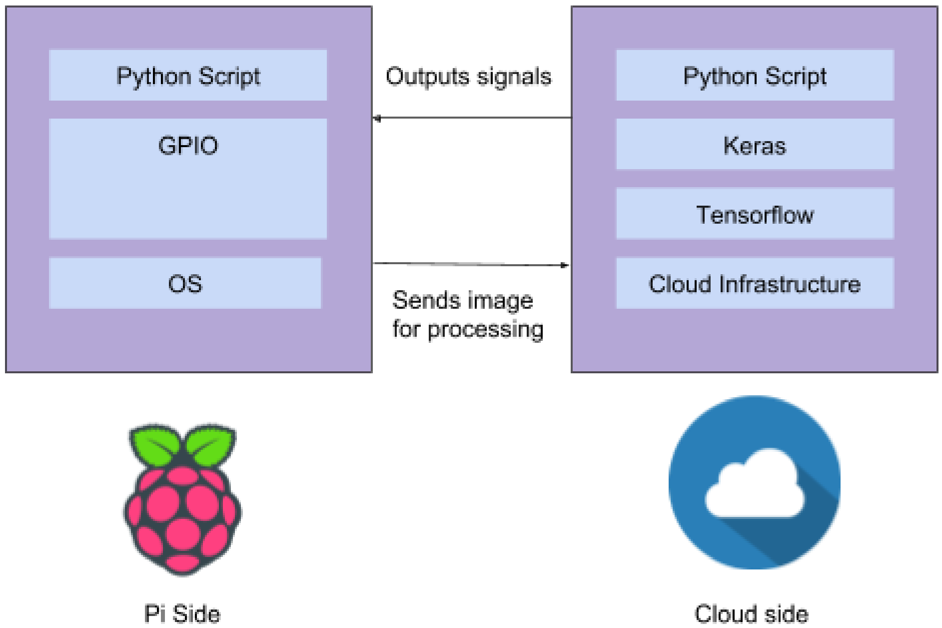


Figure 3.2. Infrastructural design for the connection between the cloud and Raspberry Pi

With fast cloud computing capability, the team will develop a cloud-based Mongo database. The technicalities behind the database are as follows:

1. NoSQL database concept allows dynamic schema for unstructured data. This means that data can be stored in multiple formats: column-oriented, document-oriented, graph-based or key value stores. This kind of database also allows for storage and processing of large volumes of structured and unstructured data that Ecobin's database will eventually amass.
2. Preliminary database design_: The team will be using MongoDB to integrate with the existing Python scripts. Establishing a connection with the MongoDB on the cloud requires us to create a MongoClient [9] to run the MongoDB instance . To create this database, a MongoClient instance is used to specify a database name (db = client['ecobinuserdb']). The data in the MongoDB will be stored using JSON styled documents. Computer engineering students in the team will be creating two collections within this instance - one for the machine learning processing (Application Data) and the other for the user data for the iOS application. The initial collections schematic as well as how the cloud databases interact with the overall system is represented below:

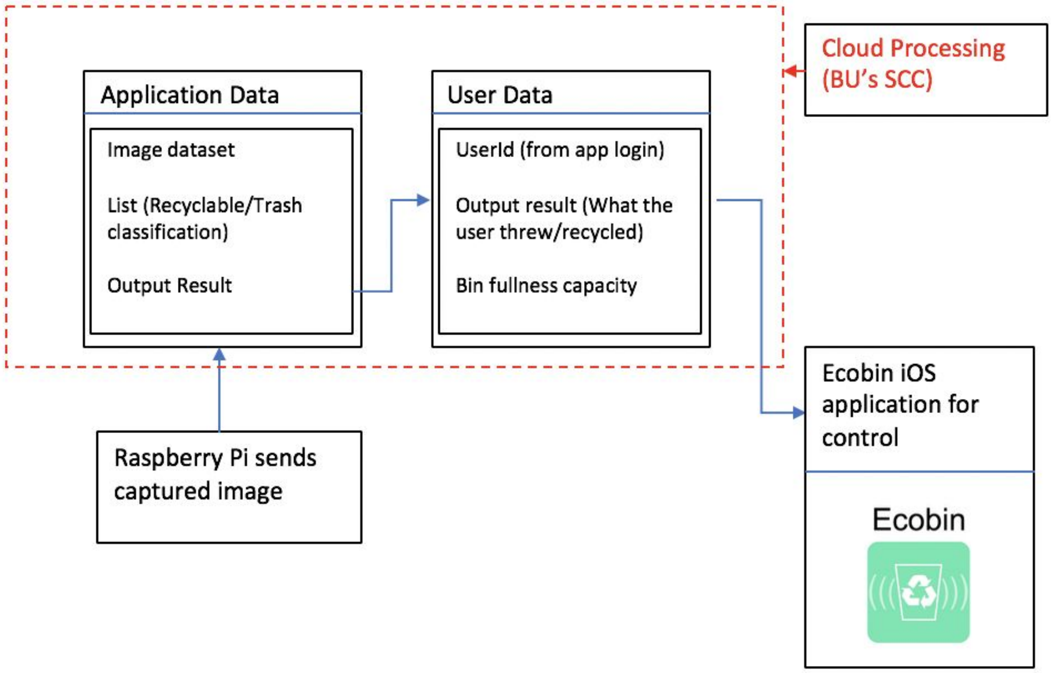


Figure 3.3: Preliminary Database Schematic

For the user to have a more personal and interactive experience with Ecobin, the team will develop an iOS application using Swift on XCode. The application wireframe design currently has three views: home, current status, and user performance. The home view will have information about the most recently disposed item and current capacity. The current status view will display detailed data about recyclable and trash compartments. More interestingly, the user performance view will inform the user of their recycling habits. Here, the team will display metrics that will encourage continuous user interactions like carbon emission reduction and achievements.

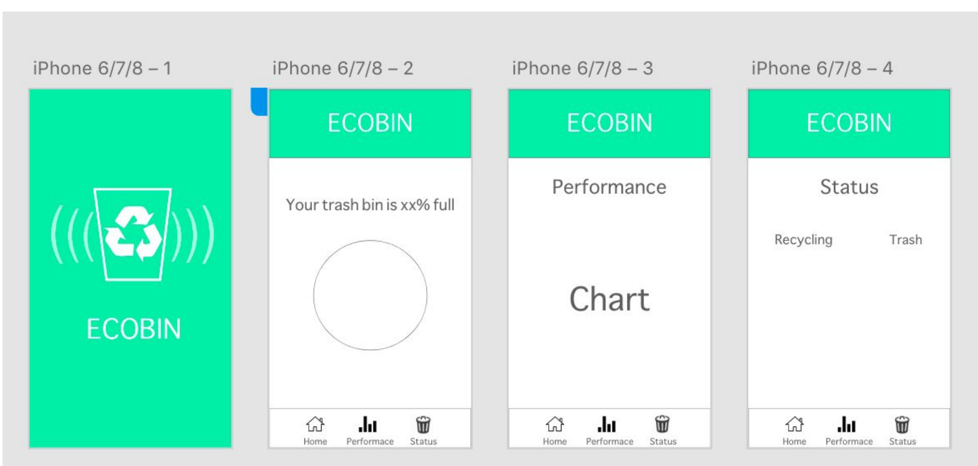


Figure 3.4. Mobile Application User Interface

3.4 System Block Diagram

The system block diagram (Figure 3.5) demonstrates five major sub-system blocks, comprised of input, output, object detection, user interaction, and capacity build-up detection. Within each major block, there are several subsystems and the arrows illustrate communications and interactions within subsystems. For instance, the PIR motion sensor receives signals from the hand motion, and drives this output to the RasPi. Similarly, the PiCamera receives a signal from the RasPi to take an image and saves it back into the RasPi.

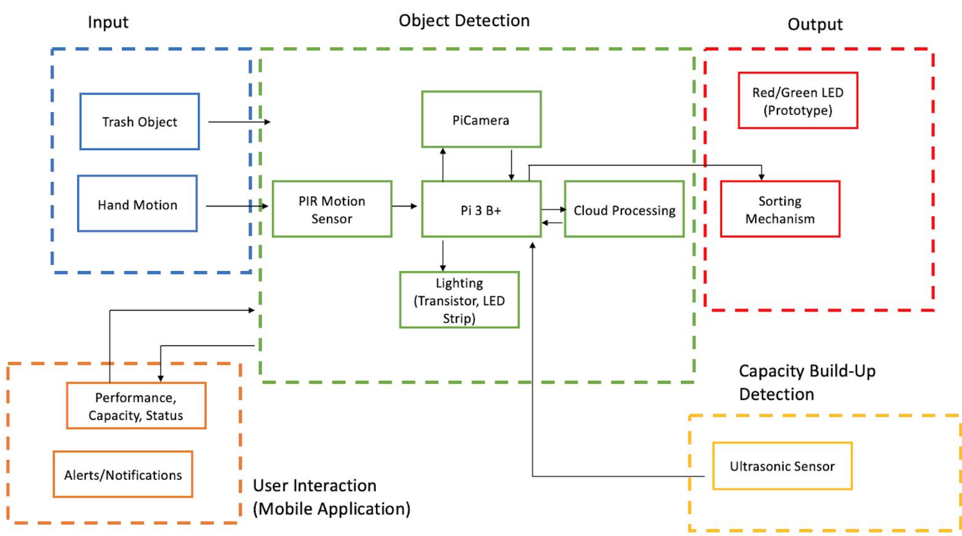


Figure 3.5. System Block Diagram

There are two main types of inputs: the disposal item and hand motion. Once the system detects motion, its lid will open and the user will place the item in. Then, the object detection subsystem, the core and largest subsystem, processes and analyzes whether this item is recyclable or not. This subsystem block consists of a PIR motion sensor, camera module, RasPi, lighting system, and the cloud-based image recognition algorithm.

Similar configurations were made for the lighting system and the cloud. For the final design, the output signals will be driven to an electro-mechanical sorting mechanism, to place the item into a trash or recyclable compartment. However, for the first prototype, the outputs were 2 indicator LEDs: green for recyclable and red for trash.

The last two subsystems support user interaction and capacity build-up detection. They remain active regardless of the state of Ecobin. The iOS application acts as a user platform and syncs with the physical parts of Ecobin and provide alerts, information, and notifications about the bin’s performance, capacity, and status. The capacity build-up detection subsystem consists of a ultrasonic sensor which constantly checks and reports the capacity of the trash compartment and the recyclable compartment to the user.

4.0 First Semester Progress

For the first semester, Ecobin prototype aimed to detect simple objects, such as plastic bottles, and classify it into two general categories: recyclable and non-recyclable. The team designed and implemented an embedded system with Raspberry Pi which controls a variety of different hardware components such as, but not limited to, the PIR motion detector, transistor, and the Raspberry Pi Camera module. This prototype also explored concepts of cloud computing by processing image recognition on the cloud.

As listed in *system description*, the hardware portion of the Ecobin project contains a variety of sensors and hardware add-ons that create the interconnected system. As a result, the hardware accomplishments of the first semester includes having a Raspberry Pi capable of taking a picture when motion is detected by the PIR motion sensor, and triggered a start in the python script on the software end. At the same time, an LED strip, powered by an external power source, is turned on for 1 second in order to provide the waste with sufficient light sources, improving the quality of the picture taken. Additionally, there are three different LEDs; red, green, and white, that are illuminated at different times. The white LED is triggered when the motion sensor detects movement, while the red/green LED shows the result of the object recognition, red meaning it is non-recyclable, and green meaning that it is recyclable. Following the example of a plastic bottle that is thrown into the ecobin, the green LED will be illuminated. This was a critical addition to the purpose of the prototype, as it assisted fast prototyping and displayed what the mobile IOS application will present in the future. Originally, an ultrasonic sensor was being considered as the main sensor that detects both motion and trash build-up, however for the purpose of the prototype, a PIR motion detector was used. Since this project involves a mechanical counterpart, the mechanical design is being completed in parallel by the ME team while the ECE team works on integration of software code and hardware communication.

The software portion of the Ecobin project successfully identified a given waste object as either recyclable plastic, or non-recyclable waste using a pre-trained VGG network from the Keras library with a Tensorflow backend. Keras is an open source high level neural network library which allows for efficient and fast prototyping and experimentation. Towards the beginning of the project, the team was exploring a variety of different computer vision programs such as OpenCV or Google Cloud Vision. However from testing and deploying the different CV systems, it was evident that Keras and Tensorflow was the most advanced and customizable, which was why it was an attractive option for a project that depended on high processing quality. As most of the team members did not receive prior coursework on advanced topics such as machine learning and deep learning, completing much of the object detection in the first semester was deemed optimistic. However, using the pretrained VGG network, the team could identify waste with a threshold accuracy of 75%. Ultimately the interlinked system between the hardware and software sections fit steadily as the first semester prototype and will act as the basis when branching out to a heavily customized software model that the team will complete in the second semester prototype.

5.0 Technical Plan

5.1. Phase 1: Planning and Conceptual Design

The team plans to finalize the design choices for the hardware, software and mechanical structures before February in order to begin construction within the first two weeks of the spring semester. Deliverables from this phase include final bills of materials, preliminary designs for mechanical parts, PCB circuit diagrams, a general framework for the Ecobin mobile app and general design of user recycling performance tracking algorithm.

Task 1. ME/EE: Finalize Ecobin Design
Finalize the design of the physical structure of the device, i.e. the bin with two receptacles, along with the stage and housing for the hardware components all with the correct dimensions. To accomplish this task, the team will design an electro-mechanical sorting mechanism to complement the automatic lid.
Deliverables: bill of materials and CAD design for product's physical structure and all required components, receipts for orders to file reimbursement request.
ME PIC: Jiaqian (sorting mechanism), Qianqian (automatic lid); ECE PIC: Kevin.

Task 2. CE: Finalize Software Components
Set up algorithm to track user recycling performance, review software backend design and check for possible improvements, begin designing mobile app and develop the initial framework for the Ecobin mobile app.
Deliverables: framework for user recycling stats algorithm, initial framework for mobile app.
PIC Lead: Shree; Assisting: Hayato

Task 3. EE: Finalize Hardware Components
Review sensors for use in the final product, make measurements for ideal conditions for image capture, design PCB circuitry, update bill of materials for PCB development, and look into power supply units for use in the final product.
Deliverables: updated bill of materials with all components set to arrive before the first day of classes, receipts for reimbursement requests, PCB schematic, dimensions for ideal picture taking conditions submitted to ME team before break.
PIC: Lead: Kevin; Assisting: Aditya.

5. 2. Phase 2: Construction

The first four weeks of the spring semester will be devoted to building the Ecobin device. The team expects to have developed an initial version of the fully functional device as for installation in users' homes with all engineering requirements met, along with the mobile app through which users can interact with the device. At the end of this phase, the team aims to have developed a product offering the full user experience.

Task 4. ME/EE: Build Sorting Mechanism
Build electro-mechanical sorting component that disposes object into either the recyclable or non-recyclable receptacle. Must be able to move a 1.0 kg or 2.2 lb. object into either receptacle, low power consumption. Test using a full 1L water bottle for maximum load scenario.
Deliverables: Electro-mechanical sorting component as part of the top compartment housing the hardware components, fully functional and tested using the 1.0 kg maximum load scenario.
ME PIC: Yuzi; ECE PIC: Aditya.

Task 5. ME: Build Structure
Build the physical bin with two receptacles that can each accommodate a tall kitchen size trash bag, a top compartment housing the stage, sorting mechanism and all the hardware components, along with the automatic motion activated lid.
Deliverables: physical structure of the product.
Lead/PIC: Qianqian; Assisting: rest of ME team.

Task 6. EE: Hardware Optimization
PCB procurement, hardware component installation into the hardware housing in the top compartment of the device, wire management and space optimization.
Deliverables: PCB, integration of all hardware components into top compartment of device, all hardware components fully functional within the new housing.
Lead/PIC: Aditya, Assisting: Kevin.

Task 7. EE: Power Supply
Procurement/ordering, installation and integration of power supply unit with single 120VAC input and three outputs: one for Raspberry Pi (5VDC), one for the LED strip (12VDC), and another output for electro-mechanical components.
Deliverables: power supply unit fully integrated into the device, with a single power cord and adapter connecting the device to a wall outlet.
Lead: Kevin.

Task 8. CE: Finalize Software Backend
Implement and finalize algorithm that tracks user recycling statistics, run through object detection and sorting algorithm again and make further improvements as required.
Deliverables: object detection and sorting algorithm at specified accuracy, database of user recycling habits, and fully functional backend.
Lead: Charles; Assisting: Shree

Task 9. CE: Mobile App Development
Mobile app development and integration, users should be able to set up the device, get notified when the device is full or needs user input to sort certain objects it cannot recognize, and let the user know of their recycling statistics via the mobile app.
Deliverables: fully functional mobile app.

PIC: Hayato

Task 10. ALL: System Integration and UX Simulation
Integrate software, hardware, and structural and mechanical components of the Ecobin device, ensure that all the components are working together properly, simulate end user experience and get feedback from team members.
Deliverables: fully functional Ecobin device, offering full end user experience.
ECE PIC: Aditya; ME PIC: Esther

5. 3. Phase 3: Optimization and Debugging

The next month will be devoted to debugging and further optimization of the system. A functional product is expected to be ready by March 20, 2019 with the initial version of the physical device finalized and the mobile app generally functional. The team aims to complete the final product for customer review at the end of this phase.

Task 11. CE: Software Debugging and Optimization
Debugging software components, further optimization of sorting algorithm to ensure maximum accuracy, improvements and optimization of user recycling habit tracking algorithm, debugging and further improving the mobile app, integration with hardware and structural/mechanical components.
Deliverables: Software works smoothly on both front and back end, guaranteed presentability to customer, maximum accuracy of sorting algorithm, final integrated system.
Lead: Hayato; Assisting: Charles

Task 12. ME/EE: Further Improvements and Optimization
Ensure all components fully functional, make necessary improvements as required, optimize hardware and structural/mechanical components to ensure consistently smooth operations, integration with software.
Deliverables: final integrated product for functional testing.
ECE PIC: Aditya, ME PIC: Esther.

5. 4. Phase 4: Functional Testing and Customer Installation

Submission to customer is finalized by the beginning of this phase. Customer reviews final product and the team makes further improvements on the product as required for product demonstrations during ECE Day and MechE Day.

Task 13. All: Functional Testing
Product should be fully functional during the week of functional testing. By this time, the physical device and mobile app are fully integrated with all engineering requirements met, i.e. object detection and sorting to the desired accuracy, fully

functional trash build up detection and user alert service, easy to use user interface, and energy use within desirable limit.
Deliverables: Device runs well during functional testing, feedback from customers and professors.
ECE/ME: All Hands on Deck

Task 14. All: Customer Installation
The whole month of April is dedicated to reviewing the final Ecobin product with the customer and making further improvements if necessary.
Deliverables: Fully functional and optimized Ecobin device with all engineering requirements met and ready for demonstration at ECE Day and MechE Day.
Customer Liaison: Charles

5. 5. Phase 5: Product Demo and ECE/MechE Day

This phase entails project completion and termination. The product must be fully functional and have required customer approval for presentations and demonstrations on ECE Day and MechE Senior Capstone Day. This phase ends with project termination.

Task 15. ALL: Product Demonstrations
All team members present and professionally dressed for presentations and product demonstrations on ECE Day and MechE Senior Capstone Day. Product fully developed and installed according to customer specifications. Project terminates at the end of ECE Day and MechE Senior Capstone Day.

6.0 Budget Estimate

The budget estimate accounts for both hardware and software components. For hardware, the major items were the Raspberry Pi, LED strips, motion sensors, and the camera. For software, all of the components including access to a cloud instance, have been negligible. Budget from the mechanical engineering partners will cover the remaining components, which includes the electro-mechanical tool and the exterior of the Ecobin.

Mr. Cootner, the primary customer, did not specify any components. There were also no items donated by the customer or any professors. The team was given flexibility with components as long as they fulfill the engineering requirements as specified in Appendix 7.1.

Item	Description	Donated	Cost/Unit	Quantity	Total Cost
	Major Items				
1	Raspberry Pi 3 B+ (1.4 Ghz Cortex-A53 with 1 GB RAM)	No	$46.90	1	$46.90
2	DC 12V LED Strips	No	$8.95	1	$8.95

3	Raspberry Pi Camera Module V2-8 Megapixel, 1080p	No	$24.90	2	$49.80
4	HC-SR501 PIR Motion Sensor	No	$12.99	1	$12.99
Minor Items					
1	Raspberry Pi 3 Case (Black)	No	$7.99	1	$7.99
2	USB 3.0 to 10/100/1000 Gigatbit Ethernet Adapter	No	$14	1	$14.00
3	RJ45 Cat-6 Ethernet Patch Cable	No	$3.58	1	$3.58
4	12W USB Power Adapter	Yes	$19.00	1	$19.00
5	USB Data Transfer Cable	Yes	$10.00	1	$10.00
6	LED Pack (White Diodes, 5mm, 3V, 20mA, 100 pcs)	No	$6.67	1	$6.67
7	2N2222 npn-BJT	Yes	$0.45	1	$0.45
	Total Cost (excluding donations)	-	-	-	$151.33

Table 6.1. Budget Estimate of Ecobin

The budget estimate comprises of two sections: major and minor items. Major items indicate essential components to fulfill the engineering requirements while minor items are parts that may only be used for the prototype. The team spent a total of $150.33, 30% of the $500 budget.

There are several implications and constraints to the budget estimate. First of all, the team still have 70% budget remaining, which is desirable since it may need to purchase additional sensors for the final design. For instance, Table 6.1 shows a purchase of two cameras, because the first camera module malfunctioned during the early development of the prototype. Second, the major items account for 78% of the spent budget, which is good because it signifies that the team is utilizing the majority budget to complete the engineering requirements of the Ecobin. However, the major items have relatively cheap prices, which poses some constraints. For example, the purchased motion sensor was one of the cheapest models with relatively low quality. However, it was sufficient for the prototype. Computing powers of Raspberry Pi 3B+ indicate another constraints, since it is a cheaper embedded unit compared to an Intel NUC. Fortunately, the team could operate a cloud instance free of charge to do the data processing.

7.0 Attachments

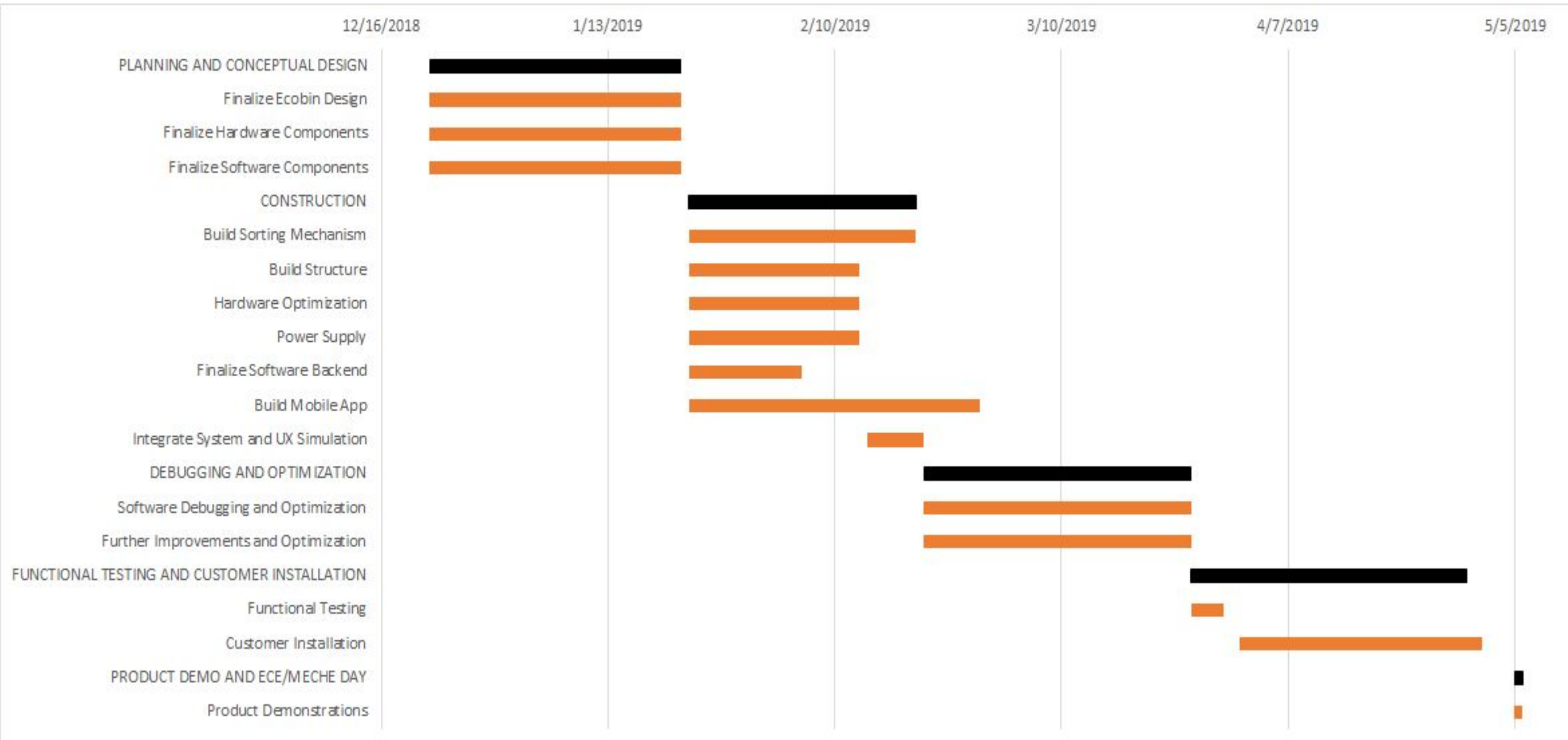
7.1 Appendix 1 – Engineering Requirements

Team 9
Team Name: Ecobin
Project Name: Ecobin

Requirement	**Value, range, tolerance, units**
Case Dimensions	1.5m x 0.6m x 0.6m
Power Supply	Single 120VAC wall outlet connection with outputs for ● 5V, 12W for Raspi ● 12V, 0.9W for LED strip ● As required for sorting mechanism and lid
Energy Use	<0.7 kWh/day (about the same as Energy Star rated compact refrigerators)
Object Detection	● Phase 1 - Successfully identify and sort two separate types of waste, recyclable plastic and general trash, at a 90% success rate. ● Phase 2 - Perform sorting tasks on three types of waste: Combustible, plastic, and general trash, all at a 90% success rate. ● Phase 3 - Completion of edge cases with a 90% success rate which include the following but are not limited to: ○ *Scenario A*: Plastic within a non-recyclable plastic waste (ie. Plastic bottle inside a paper bag) ○ *Scenario B*: User throws away waste that must be treated with caution (ie. lithium ion batteries, toxic waste [1]).
Electro-Mechanical Sorting Tool	● Sustain waste with the weight up to 1.0kg (2.2lbs). ● Move object to the desired compartment within 5s.
Practical	The overall cost of the final Ecobin design should be < $150.
Cloud Integration	● Data processing for object classification via a cloud instance. Processing time < 2s. ● Desired specifications: 8 CPU cores, 16GB RAM, 10 GB storage. ● Current specifications (per node on BU Shared Computing Cluster): 2 14-core CPUs, 256 GB RAM, 886 GB storage.
User Interaction	● Mobile app sync time of < 2s. ● Keeps track of user disposal history for the past 30 days.
Internet Connection	● Minimum upload speed of 500 kbps required

7.2 Appendix 2 – Gantt Chart

Below is a Gantt Chart, which is a projection of how Ecobin plan to complete the remaining tasks of the Ecobin. The tasks are classified into several major phases: Planning and Conceptual Design, Construction, Optimization and Debugging, Functional Testing, Customer Installation, and finally it ends with the Product Demonstration during ECE/MechE Day.



7.3 Appendix 3 – Ecobin Drawings & Schematics

Below are several diagrams representing our latest design for the Ecobin:

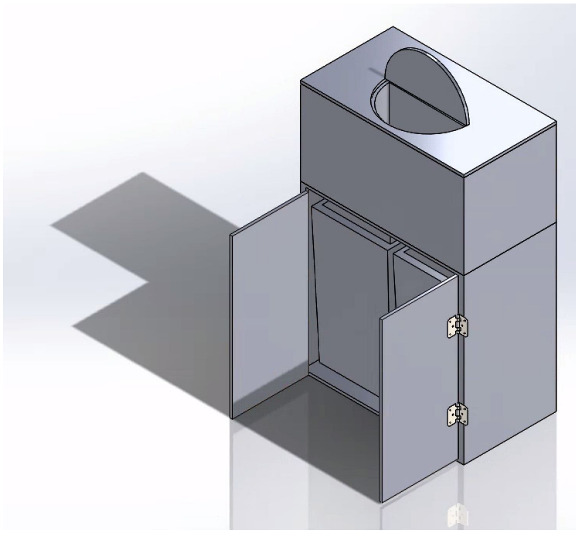


Figure 7.1. Ecobin Exterior (CAD): Shows the exterior of the Ecobin. The diagram displays the automatic lid, and the two compartments for trash and recyclable.

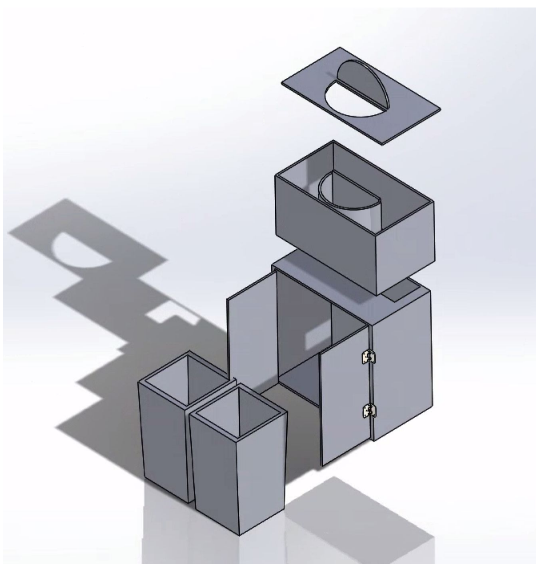


Figure 7.2. Ecobin Interior (CAD): Shows the interior of the Ecobin along with the different structures within the Ecobin. The design shows where the waste will be placed along with potential areas where the hardware can be setup.

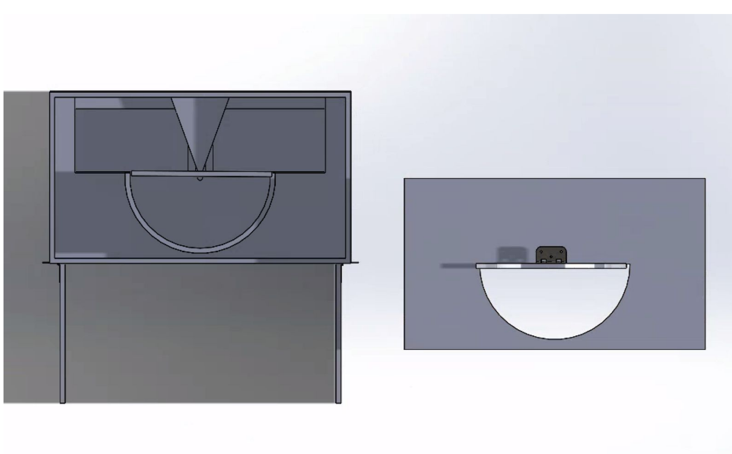


Figure 7.3. Ecobin Top View (CAD): Shows what the Ecobin looks like from a top or bird's eye view. The diagram on the right is the automatic lid. The diagram on the left is what the bin would look like if the lid is removed. The semicircle platform is where the trash will be placed. The horizontal part of the semicircle platform will rotate due to a motor which will push the trash into the desired compartment.

7.4 Appendix 4 – Technical References

[1] United States Environmental Protection Agency. (2018). *Facts and Figures about Materials, Waste and Recycling | US EPA*. [online] Available at: https://www.epa.gov/facts-and-figures-about-materials-waste-and-recycling [Accessed 8 Oct. 2018].

[2] Boston.gov. (2018). *Boston Zero Waste Planning*. [online] Available at: https://www.boston.gov/sites/default/files/zwac_presentation.pdf [Accessed 8 Oct. 2018].

[3] Live Science (2018). *7 Everyday Toxic Things You Shouldn't Toss in the Trash*. [online] Available at: https://www.livescience.com/13840-7-everyday-toxic-items-recycle.html [Accessed 8 Oct. 2018].

[4] *Boston.gov*, 2018. [Online]. Available: https://www.boston.gov/sites/default/files/zwac_presentation.pdf. [Accessed: 08 Oct. 2018].

[5] "Getting Started with PiCamera", *RaspberryPi.org*, 2018. [Online]. Available:https://projects.raspberrypi.org/en/projects/getting-started-with-picamera. [Accessed: 01 Nov. 2018].

[6] "Python code to detect if camera is available.", *RaspberryPi.org*, 2018. [Online]. Available: https://www.raspberrypi.org/forums/viewtopic.php?t=46113. [Accessed: 28 Nov. 2018].

[7] "How To Interface a PIR Motion Sensor with Raspberry Pi GPIO", *Maker Pro*, 2018. [Online]. https://maker.pro/raspberry-pi/tutorial/how-to-interface-a-pir-motion-sensor-with-raspberry-pi-gpio. [Accessed: 18 Nov. 2018].

[8] "PIR Motion Sensor with Raspberry Pi", *hackster.io*, 2018. [Online]. Available: https://www.hackster.io/hardikrathod/pir-motion-sensor-with-raspberry-pi-415c04.[Accessed: 08 Nov. 2018].

[9] Real Python, "Introduction to MongoDB and Python – Real Python," *Real Python*, 11 Jun. 2018. [Online]. Available: https://realpython.com/introduction-to-mongodb-and-python/. [Accessed: 12 Dec. 2018].