CEN 5035 Software Engineering, Fall 2019

Project 4 Project Final Report: Couch Guardian

"An evolutionary pet behavior correction solution"

Team Name: thePerfect10

Team Number: 10

Prepared By: David Baston – Product Owner

Francis Iniobong John Ifon – Individual Contributor

Alyssa Johnson – Individual Contributor

Tsz Shing Tsoi – Scrum Master

Trello: https://trello.com/b/i8VdXmye/group-10-software-engineering

YouTube Presentation: https://youtu.be/2tebSB6xQng

Course Instructor: Dr. Shihong Huang

Date: December 4, 2019

Version History:

Date	Version	Description	Approved By
09/27/2019	1.0	Initial proposal	David Baston
12/04/2019	1.2	Final project report	David Baston

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Couch Guardian is a smart home pet behavior correction device. Upon detection of bad pet behaviors, its pre-recorded audio messages are activated to immediately cease those behaviors. In addition to stopping cats from clawing furniture, Couch Guardian can also stop dogs from barking, cats from laying on a kitchen counter, pets from chewing indoor plants, and various other behaviors specified by the pet owners. Couch Guardian operates autonomously, giving new and veteran pet owners a peace of mind in protecting furniture and other properties at home while continuously correcting pet behaviors in their absence. Couch Guardian also tracks pets' behaviors through a dashboard summarizing the history of device activations and pet activities.

According to the *American Pet Products Association* in [1], 67% of U.S. households or 84.9 million households own a pet. In 2018, the total U.S. pet industry expenditures were at $72.56 billion, which is equivalent to an annual pet expense of $855 per pet-owning household. Pet expenditures had grown three-fold in the past two decades and the trend is expected to continue in the future. Additional studies further indicate that, due in part to the increase in value of smaller, portable electronic devices, pet related property damage has surpassed $3 billion per year for pet owners [7].

Pet products have a sizable market. Some existing pet behavior correction devices require manual activation while autonomous, motion sensor-activated pet deterrent devices, are unable to differentiate pets from humans or other moving objects, nor are they capable of targeting only undesirable pet behaviors. No existing products currently provide a history or log of pet behavior improvements. Couch Guardian is a complete home pet behavior correction solution, providing pet owners all the tools needed to correct pet behaviors and protect their homes from damage caused by untrained pets.

COMPETITIVE ANALYSIS

The proposed functionality of Couch Guardian can be distilled into three key purposes: detecting motion, identifying pets and non-pets or pet behaviors, as well as activating deterrent(s) on a valid entity. As one of the starting points for this project, our team examined a cross section of similar products from various competitors within this industry to better understand the rationale behind their collections and the perceptions of users, and analyze the advantages of Couch Guardian in comparison to these competitors.

Products with similar feature sets have been reviewed from notable platforms such as Amazon.com in [2], [3], [4] and [5] to determine their primary attributes. Similar to these competitors, Couch Guardian will consist of motion detectors to identify the movement of a valid entity within the detection area. But unlike these competitors, a key difference that sets Couch Guardian apart is the in-built machine learning algorithm trained to distinguish between pets and non-pets. This algorithm will further classify the pet species or their behaviors. Another key feature of Couch Guardian in comparison to the competitors is the utilization of a dashboard to monitor the pets' behavioral history. This dashboard will ensure the timeline of the device activations and pet activities are kept up-to-date with complete history for pet owners' review.

The competitive products examined in this competitive analysis are summarized below. Table 1 compares Couch Guardian's proposed features with features of these competitive products.

- **PetSafe SSSCat Cat Training Aid**: This product ensures cats are deterred from going near valuable furniture, counters and plants. It consists of an adjustable electronic motion sensor that detects

a cat approaching the restricted area, before releasing a harmless scentless spray to surprise the intruder cat.

- **PetSafe Pawz Away Pet Barriers**: This product is designed to keep dogs away from problem areas around the home. It is a cordless and wireless deterrent. It is a waterproof and battery-operated barrier that consists of a transmitter and an adjustable receiver collar worn by the pets.
- **PredatorGuard PestAway Ultrasonic Outdoor Animal & Cat Repeller**: This product consists of motion sensors that detects pets and pests from entering a specified range of protection by emitting an ultrasonic noise, which is only audible to the pets and pests, and flashes ultra-bright strobe lights to effectively scare and keep pests far away. This is only used outdoors.
- **Doggie Don't Device**: This is a handheld dog repellent and bark control device that uses a loud sound, over 100 decibels, to deter bad behavior.

Table 1: Comparison of Top Competitive Products from Amazon.com

Price / Features	Couch Guardian	PetSafe SSSCat Cat Training Aid	PetSafe Pawz Away Pet Barriers	PREDATORGUARD PestAway Ultrasonic Outdoor Animal & Cat Repeller	Doggie Don't Device
Retail Price	TBD	$35.94	$59.95	$44.97	$49.95
Deter pets	Keeps pets away from furniture and properties	Keeps pets away from furniture and properties	Keeps pets away from furniture and property	Keeps pets and animals from yards and gardens	Stops barking and bad behavior
Detect motion	Installed motion sensor that detects movement	Installed motion sensor that detects movement	Installed motion sensor that detects movement	Installed motion sensor that detects movement	*
Identify pets and non-pets	Designed to differentiate between pets and non-pets	Not configured to differentiate between pets and non-pets	Not configured to differentiate between pets and non-pets	Not configured to differentiate between pets and non-pets	*
Identify pet behaviors	Designed to classify pet behavior	Not designed to classify pet behavior	Not designed to classify pet behavior	Not designed to classify pet behavior	*
Record pet behaviors	By using its in-built dashboard, it can record pet behavior	Not designed to record pet behavior	Not designed to record pet behavior	Not designed to record pet behavior	Not designed to record pet behavior

* Manually activated by user

DATA DEFINITION

Glossary of key terms to be used throughout this project is included below:

Dashboard. Dashboard is a software interface which is used to generate plots for summarizing and displaying data.

Data storage system. Data storage system is the system which stores error logs, captured images, video recordings and device activation history. The data storage system may reside on the device or external to the device.

Detection area. Detection area refers to the entire effective area of the sensor if not specified by user, or a subset of it if specified by user.

Deterrent. Deterrent refers to any substance, physical stimulus and mechanism to repel pets. Its current capability refers to pre-recorded audio messages when used in the context of Couch Guardian version 1.0.

Device. Device refers to the physical integrated system of Couch Guardian, specifically components responsible for visual detection, data collection, and deterrent delivery.

Entity. Entity refers to any living organism and non-living object capable of being recognized by the learning algorithm, and their behaviors.

Entity type. Entity type refers to a label or classification of an entity by the learning algorithm.

Invalid entity. Invalid entity is an entity type with no associated user-defined action or with an associated action of “do nothing”.

Learning Algorithm. The software model responsible for classifying visual data from the device and prescribing a response from the system.

Pet. Pet refers to only cats and dogs when used in the context of Couch Guardian version 1.0.

Pet behavior correction. Pet behavior correction refers to the use of deterrent(s) to stop and cease any bad pet behavior.

System. The software architecture that manages the user interface responsible for maintaining user access to saved device data and the statistical dashboard.

Valid entity. Valid entity is an entity type defined by a user with an associated action of activating deterrent(s).

OVERVIEW, SCENARIOS AND USE CASES

This section discusses the project and usage scenarios of Couch Guardian through several main use cases. The actors are first summarized in Table 2, followed by the main use cases in Table 3a, Table 3b, Table 3c and Table 3d.

Table 2: Actors

Name	Type	Description
Pet Owner	Primary	The owner of the pet and person placing the device in a designated area
System Support Engineer	Supporting	The engineer or system which accesses the device remotely to perform software update without pet owner intervention
Data Storage System	Supporting	The storage system that is sent error logs, video recordings, and device activations for reporting and error handling

Table 3a: Use Case - Set-up Device

Items	Description
Use Case Name	Set-up device
Use Case ID	UC-001
Brief Description	The Pet Owner sets up the device to monitor a specific area
Actors	Pet Owner, Data Storage System
Type	[**X**] External [ ] Temporal
Preconditions	1. The Device is off
Basic Flow	1. The Pet Owner places the device front of the area it wishes to protect 2. The Pet Owner turns the device on 3. The device analyzes the sensor area and initializes the pet detection area 3.1. Optionally, the Pet Owner specifies a custom detection area 4. Optionally, the Pet Owner sets up custom entity types 4.1. The Pet Owner records individual pets with the device sensor 4.2. The Pet Owner records bad pet behaviors with the device sensor 5. Optionally, the Pet Owner configures audio deterrent 5.1. The Pet Owner select custom pre-recorded audio message files from the Data Storage System to be used as audio deterrent 6. Optionally, the Pet Owner sets up Data Storage System location 7. The device creates a notification for the Pet Owner to say the detection area is set 7.1. This is the end of the use-case
Alternative Flow	1. The device is unable to determine the area 1.1. An error should be logged in the Data Storage System and sent to the Pet Owner for review
Post Conditions	1. An area for pet detection is set

Table 3b: Use Case - Deter Pet from Area

Items	Description
Use Case Name	Deter pet from area
Use Case ID	UC-002
Brief Description	Deters a pet from being in the detection area
Actors	Pet Owner, Data Storage System
Type	[**X**] External [ ] Temporal
Preconditions	1. The Pet Owner has placed the device in the area to be monitored 2. Monitoring is currently turned on for the device 3. The Device is on and functioning 4. The deterrent is functional
Basic Flow	1. An object enters the area 1.1. The use case begins when the device senses an object in the defined area 2. Determine if pet in area 2.1. The device determines if the object is a pet 3. Log pet entry 3.1. The device logs the pet encounter to later be handled by the system 4. Record Pet Behavior 4.1. The camera on the device should activate to begin recording the pet's behavior 4.2. This recording is sent to the Data Storage System to be viewed later by the Pet Owner 4.3. The recording should continue until it is confirmed the pet has left the area, and then an additional 2 seconds 5. Activate deterrent 5.1. The deterrent is activated to scare the pet away from the area 6. Confirm the pet has left the area 6.1. The device records the time the object leaves the area 6.2. This ends the use case
Alternative Flow	1. Handle the object is not a pet 1.1. Then the device should not report non-pet objects in the area. The device should stop processing at this point if the object is not a pet 2. Handle errors with the sensors not working properly 2.1. When the sensors stop functioning during the process, an event should be logged to the system that notifies the Pet Owner to inspect the device and ensure it is operational 2.2. The device should then shut-down until reactivated by the Pet Owner 3. Handle errors with pet entry log

	3.1. If the device is failing to send log entries for the pet being in the area, this should be noted in the event log. Processing should continue as normal otherwise 4. Handle errors with the video recording 4.1. If the device is failing to send files to the external system, an error should be recorded in the event log and the Pet Owner notified 4.2. If the camera is not functioning, an error should be recorded, the Pet Owner notified, and the device shut down 5. Handle errors with deterrent running out of air 6. Handle errors with error logging 6.1. In the event that error logs are failing to be recorded, the device should shut down 7. Handle pet not leaving the area 7.1. If after activating the deterrent, the pet fails to leave the area, it should attempt to active the deterrent again 7.2. If after a second attempt to deter the pet fails, this should be reported to the report system to notify the Pet Owner 7.3. The camera should cease recording 2 seconds after the second failed attempt
Post Conditions	1. The pet leaves the areas and the Pet Owner is notified; a log entry exists for the pet entering the area; a log entry exists for the pet leaving the area; a video exists for capturing the deterrent activation and pet leaving the area 2. The pet does not leave the area and the Pet Owner is notified; a log entry exists for the pet entering the area; a log entry exists for the pet failing to leave the area; a video exists capturing the deterrent activation and the pet not leaving the area

Table 3c: Use Case - Review Notifications

Items	Description
Use Case Name	Review notifications
Use Case ID	UC-003
Brief Description	The user reviews notifications generated by the device regarding the pet
Actors	Pet Owner
Type	[**X**] External [ ] Temporal
Preconditions	1. There are unread emails 2. The System is operational
Basic Flow	1. Pet Owner receives an email notification 2. The Pet Owner reviews the email 3. This ends the use case

Table 3d: Use Case - View Dashboard

Items	Description
Use Case Name	View Dashboard
Use Case ID	UC-004
Brief Description	The user reviews the dashboard for device activation and pet activity history
Actors	Pet Owner
Type	[**X**] External [ ] Temporal
Preconditions	1. The Pet Owner has an account for the system 2. The system is operational
Basic Flow	1. Pet Owner signs into System 2. Pet Owner navigates to dashboard portion of the system 3. The system retrieves the dashboard and any supporting video and error logs if applicable and presents them to the Pet Owner 4. The Pet Owner is presented the dashboard 4.1. This ends the use case
Alternative Flow	1. The Pet Owner is unable to sign in 1.1. Account management will be necessary for the user to reset their password if they have forgotten it
Post Conditions	1. The notifications are no longer considered unread

LIST OF HIGH-LEVEL FUNCTIONALITIES

A list of high-level functionalities implemented in Couch Guardian is provided below.

F-1. Entity Detection

1.1. Motion Detection

1.1.1. *Detect motion*. The device monitors any detectable change in the environment or movement of any entity within the detection area. Upon detection of such change or movement, the device will signal that a motion is detected.

1.2. Object Detection

1.2.1. *Identify pets and non-pets*. The device differentiates pets including cats and dogs from all other entities such as couch and countertops, and will signal in accordance with the detected entity type of pets and non-pets.

F-2. Deterrent Activation

2.1. Deterrent Activation Mode

2.1.1. *Activate deterrent on valid entity*. When a signal is received from motion or object detection, the device will analyze if the signal entity type matches the pre-defined valid entity of cats or dogs. If a match is found, the device will activate the deterrent

once and will activate the LED indicator light to confirm that the valid entity is detected.

2.1.2. *Activate deterrent based on entity types*. When a signal is received from motion or object detection, the device will analyze if the signal entity type matches any valid entity from a list of user-defined entity types of cats and dogs. If a match is found, the device will activate the deterrent with setting corresponding to each valid entity type. The list of entity types and the corresponding deterrent settings are pre-defined in the device.

F-3. Device Set-up

3.1. General System Configuration

3.1.1. *Data storage location*. The device is pre-defined to use on-device storage for temporary storage of captured images and cloud storage for detection logs.

3.2. Detection Area Configuration

3.2.1. *Detection area*. The detection area is pre-defined to be the entire sensor (camera) effective area for motion and object detections.

3.3. Entity Configuration

3.3.1. *Individual pets*. The device is pre-defined to detect pets between cats and dogs.

3.4. Deterrent Configuration

3.4.1. *Pre-recorded audio messages*. The device is pre-defined to select between a randomly generated audible noise and a pre-recorded human-audible audio based on the detected entity types of cats and dogs as deterrent.

F-4. Data Storage

4.1. Activity Logs

4.1.1. *Log activities in dashboard*. The device logs detection activities in a database stored in the cloud. The logged data include date and time of motion detection and entity type. The database is retrievable from anywhere with Internet connection. The content of the database is summarized and visually displayed in a dashboard format.

4.2. Sensor Data

4.2.1. *Email detected pet images to user*. The device will automatically email an image of the detected pets to users through a cloud email server. The email service and sent and recipient email addresses are defined in each device.

F-5. User Interface

5.1. *Detection confirmation*. The user will receive positive confirmation of the detection status through the on-device LED indicator light. The user can also confirm the detection activities and detected entity types of cats and dogs through the online dashboard from a Web browser.

LIST OF NON-FUNCTIONAL REQUIREMENTS

An initial list of high-level non-functional requirements is provided below.

NF-1. Usability

1.1. Humanitarian factors

1.1.1. The system shall conform to all standards in regards to the ethical treatment of animals. The system should serve as an uncomfortable but non-harmful deterrent to pets while protecting pet-based damage to property.

1.2. Aesthetics

1.2.1. The device shall inhabit an unobtrusive space, easily portable within a common living space. The system should not be a distracting feature in the room or become a possible tripping hazard to human occupants.

1.3. Area of Effect

1.3.1. The device shall be capable of monitoring an area within 5 feet of the device with a field of view encompassing 62.2 x 48.8 degrees (Pi Camera standard capability).

1.3.2. The device shall be capable of producing a sound-based deterrent audible within its field of operation.

NF-2. Reliability

2.1. Frequency and severity of failure

2.1.1. The device shall fail to respond no more than once per week.

2.1.2. The system shall fail to provide access to saved data resources no more than once per week. This refers to Watson Database services and user email notifications.

2.2. Accuracy

2.2.1. The system shall correctly distinguish between human and pet with 95% accuracy under normal lighting conditions.

2.2.2. The system shall detect bad pet behavior and initiate an appropriate response with 90% accuracy when a pet is detected by the device.

2.2.3. The device shall not initiate a deterrent action without the presence of an animate entity.

2.3. Durability

2.3.1. The device should be able to withstand an attack from an average sized housecat under 20 lbs.

2.3.2. The device shall allow repeated uses of speaker system without crash or fail.

2.3.3. The device shall not be damaged if it is overturned in the course of operation.

NF-3. Performance

3.1. Response Time

3.1.1. The device shall detect an animate entity within 5 seconds of entering its field of operation.

3.1.2. The device shall detect bad behavior within 3 seconds from the start of such an occurrence.

3.1.3. The device shall initiate a deterrent response within 2 seconds of detection of bad behavior.

3.2. Efficiency

3.2.1. The device shall not sustain a deterrent alarm for longer than 3 seconds per activation.

3.3. Throughput

3.3.1. The device shall be able to identify the presence of one pet at a time and judge the appropriateness of its behavior.

3.3.2. The data repository shall be accessible by at least 5 simultaneous users.

NF-4. Supportability

4.1. Maintainability

4.1.1. There is a preference for systems that will include the use of source code version control as provided in systems like Git.

4.2. Compatibility

4.2.1. There is a preference for Linux based systems using Python and Python language libraries and tools designed for use with the Raspberry Pi 3B+ Microcontroller.

4.3. Configurability

4.3.1. No further requirements anticipated at this time.

4.4. Installability

4.4.1. The device should be recognizable to an established network for internet connectivity when powered on.

4.4.2. The data repository system should be accessible from a standard web browser such as Google Chrome or Firefox.

4.4.3. At this time, the device should be functional at the time of plug-in without further customization on the part of the user.

4.5. Localizability (Internationalization)

4.5.1. At this time, only support for U.S. English is anticipated.

NF-5. Security

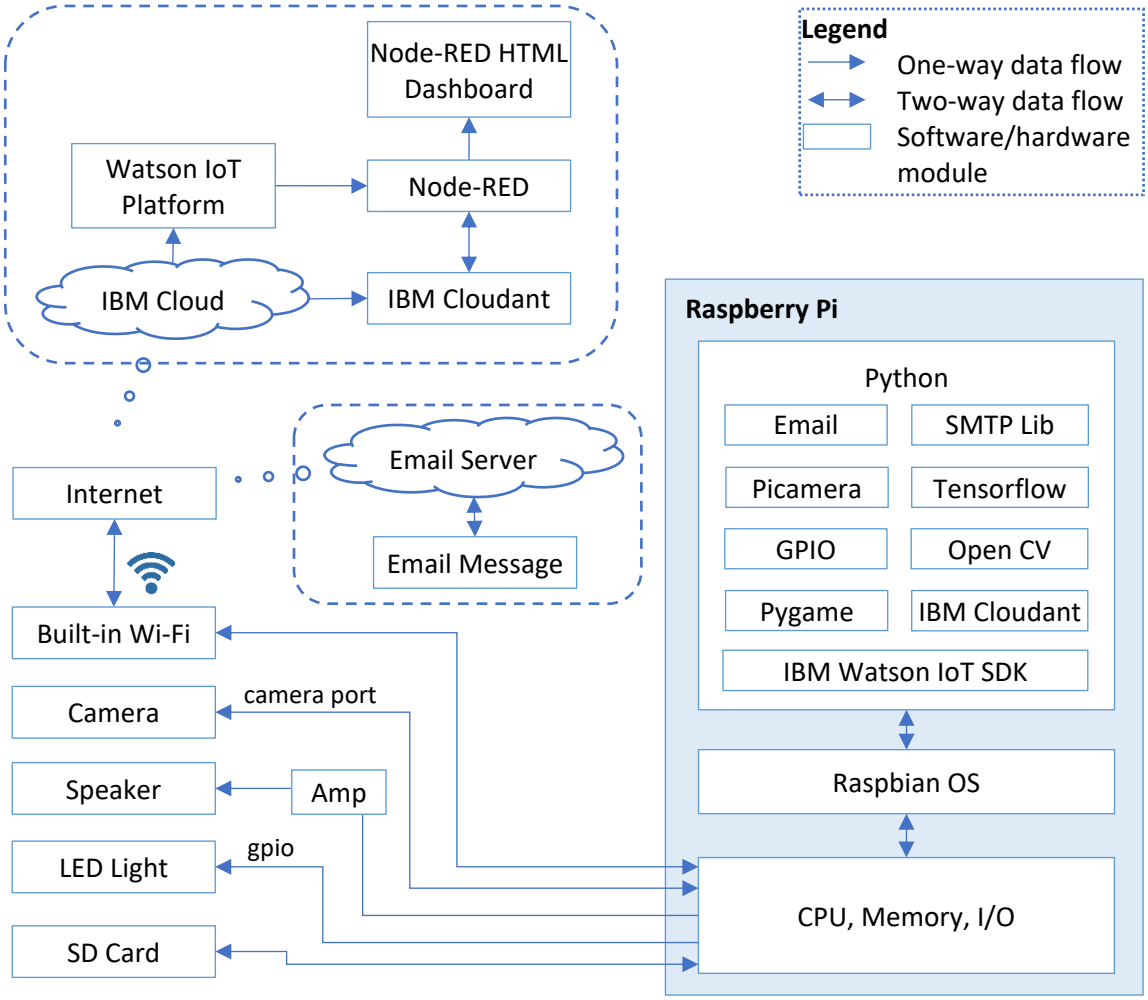
5.1. Access Control

5.1.1. The system shall limit access to data storage from the device to registered users of the device.

HIGH-LEVEL SYSTEM ARCHITECTURE

The high-level system architecture and the relationship between the core components for Couch Guardian is shown in Figure 1.

Figure 1: Couch Guardian High-level System Architecture



The three main parts of the system architecture include the Raspberry Pi running the software programs locally, the physical connections connected to the Raspberry Pi as peripherical components, inputs and outputs, and the cloud services used for Internet of Things (IoT) integration. Key components, including the software products, tools, languages and systems, are briefly discussed below. Detailed discussion of these components is included in the next section in project implementation details.

Raspberry Pi / Software Programs

- **Raspberry Pi**: The microcontroller used for the device.
- **Raspbian OS**: The operating system for the **Raspberry Pi**, which comes pre-installed with several programming languages including **Python**.
- **Python**: The main language used, which comes pre-installed with **Raspbian OS**. The following key Python libraries are used:

- **GPIO**: Library module to interact with general purpose input/output (GPIO) pins.
- **Picamera**: Library module to interact with the camera module.
- **Pygame**: Library module to control audio playback in the speaker.
- **Open CV**: Library module to process images.
- **Tensorflow**: Library module to perform pet and object classifications with learning algorithm.
- **Email**: Library module to compile email messages.
- **SMTP Lib**: Library module to connect to email server through Simple Mail Transfer Protocol (SMTP) to send email messages.
- **IBM Cloudant**: Library module to connect to IBM Cloudant service and create JSON objects containing detection logs in IBM Cloudant for storage.
- **IBM Watson IoT SDK**: Library module to connect to IBM Watson Internet of Things (IoT) Platform and send JSON objects containing activation status signal to the IoT Platform.

Physical Connections

- **Camera**: Connected to the **Raspberry Pi** for capturing images and videos of the environment.
- **Speaker**: Used to play pre-recorded audio messages to serve as the pet deterrent and is connected to the 3.5mm audio jack of the **Raspberry Pi** through an **amplifier**.
- **LED Light**: Connected to the **Raspberry Pi** to provide visual confirmation when a valid entity of cats or dogs are detected.
- **SD Card**: Local storage solution for image files captured from the device.
- **Built-in Wi-Fi**: Used to communicate with **IBM Cloud** and email services.

IoT Integration

- **IBM Cloud**: A collection of cloud services provided by IBM. The following IBM Cloud services are used:
  - **Watson IoT Platform**: A cloud platform for connecting with IoT devices.
  - **IBM Cloudant**: A cloud NoSQL database provided by IBM which is built based on Apache CouchDB for storing detection logs in the format of JSON objects.
  - **Node-RED Application**: A flow-based development tool hosted in IBM Cloud to communicate with **Watson IoT Platform** and **IBM Cloudant**.
  - **Node-RED HTML Dashboard**: A live dashboard hosted in **IBM Cloud** to display detection logs from the device.
- **Email Server**: An email server supporting SMTP email transmission.

PROJECT IMPLEMENTATION DETAILS

This section discusses the project implementation details including inputs, outputs and cloud services used in Couch Guardian, and the system integration between these components and Raspberry Pi.

Inputs

- **Camera**: A Pi camera is connected to the camera port of **Raspberry Pi** and is used as the sensor in Couch Guardian to capture visual data of the environment. The captured visual data are continuously processed through the learning algorithm implemented in **Raspberry Pi** with **Tensorflow** for object recognition to identify pets.

Outputs

- **Speaker**: A speaker is used to output audio as deterrent to deter pets. When a valid entity of cats or dogs is identified from the **Tensorflow** learning algorithm, the audio data would be sent to the speaker through an **amplifier**. Different pre-defined audio outputs would be selected based on the types of pets detected.
- **LED Light**: The **Raspberry Pi** will turn on the LED light when a valid entity of cats or dogs is identified and is used as a visual confirmation for the user.

Cloud Services

- **Watson IoT Platform**: The **Raspberry Pi** will send a detection signal in the format of JSON object containing the IP address of the **Raspberry Pi** to Watson IoT Platform. This signal is used to trigger updates to the **Node-RED HTML Dashboard** through the **Node-RED Application**.
- **IBM Cloudant**: Upon each detection of a valid entity of cats or dogs, the **Raspberry Pi** will send the detection logs to IBM Cloudant for storage. IBM Cloudant allows various types of indexes to be configured, and views (MapReduce) and search indexes were set up for executing queries from the **NodeRed Application**.
- **Node-RED Application**: The Node-RED Application monitors the **Watson IoT Platform** for any detection signal received from the **Raspberry Pi**. When a signal is detected, the Node-RED Application would query the latest detection logs from **IBM Cloudant** and display on the **Node-RED HTML Dashboard**. The application URL is: https://cen5035group10.mybluemix.net/red (password protected)
- **Node-RED HTML Dashboard**: Detection logs and historical data are displayed on the dashboard. The dashboard is fed live from **IBM Cloudant** through **the Node-RED Application**. The URL for a demonstration of the dashboard is : https://cen5035group10.mybluemix.net/ui
- **Email Server**: Upon each detection of a valid entity of cats or dogs, the **Raspberry Pi** will send an email message with a picture of the detected entity, through an email server, to the pre-defined email address. The email server used for demonstration is Gmail at `smtp.gmail.com`.

System Integration

The core component of Couch Guardian is the Raspberry Pi which runs the core software programs including image capture, object recognition and deterrent activation, locally. This is a significant advantage to Couch Guardian that the core functions of pet training and deterrents would still be operational in case of lost or spotty Internet connection. All software programs are implemented with Python using available Python libraries. The detection begins with video streaming from the Pi camera using the `picamera` library. The video stream is then processed with the `cv2` (Open CV) library to extract each frame for processing. Each frame is fed into a Tensorflow Lite learning algorithm using the `tensorflow` library for object recognition. A minimum waiting time of 5 seconds (default) is set between each deterrent activation to avoid overly frequent activation. When a valid entity of cats or dogs is detected and the minimum waiting time is exceeded, several functions described below are executed:

1. *Activate deterrent*: Raspberry Pi represents analog audio signals in digital format using pulse width modulation (PWM). When a cat is detected, a PWM signal with frequencies randomly chosen between 200 Hz and 5,000 Hz (default) is output to GPIO40, which is an inaccessible GPIO pin connected to the stereo right channel in the 3.5 mm audio jack, using the `GPIO` library. When a dog is detected, a pre-recorded MP3 audio file is played using the `pygame` library.
2. *Activate LED light*: A HIGH signal is output to the GPIO pin connected to the LED using the `GPIO` library. Otherwise, a LOW signal is output to the same GPIO pin to turn off the LED if no cats or dogs are detected.
3. *Send detection logs to IBM Cloudant*: A JSON object containing the current date and time and the types of valid entity detected is created. The Raspberry Pi then connects to the IBM Cloudant database using the `cloudant` library with the pre-defined credentials, creates a new document in the IBM Cloudant database with the JSON object, and disconnects from the database. Error messages, if any, are received for diagnostic purpose.
4. *Send signal to Watson IoT Platform*: After the detection log is created in the IBM Cloudant database, another JSON object containing the IP address of the Raspberry Pi would be created. The Raspberry Pi then connects to the Watson IoT Platform using the `wiotp.sdk` (IBM Watson IoT SDK) library with the pre-defined credentials, publishes the JSON object to Watson IoT, and disconnects from Watson IoT. The IP address in JSON object is used for diagnostic purpose; however, any JSON object could also be used. Similarly, error messages from Watson IoT Platform, if any, are received for diagnostic purpose.
5. *Send email to user*: An email message containing the image frame is created using the `email` library. The Raspberry Pi then logs in to the SMTP email server using the `smtplib` library with the pre-defined credentials, and sends the email message to the user through the email server.

In the cloud, the Node-RED HTML Dashboard is implemented through the Node-RED Application implemented in IBM Cloud. The Node-RED Application connects to both the Watson IoT Platform and IBM Cloudant database. When a signal is received in the Watson IoT Platform, it would trigger Node-RED to query the IBM Cloudant database and update the dashboard. The dashboard will also be updated every hour regardless of any detection signals from Watson IoT. The following three queries are configured in Node-RED:

1. *10 most recent detections*: Details of the 10 most recent detections, including the timestamp and detected entity type, would be queried through the search indexes configured in the IBM Cloudant database.
2. *Detection history over the past 30 days*: Number of detections, grouped by days and entity types, over the past 30 days would be queried through views (MapReduce) configured in the IBM Cloudant database.
3. *Detection history over the past 24 hours*: Number of detections, grouped by hours and entity types, over the past 24 hours would be queried through views (MapReduce) configured in the IBM Cloudant database.

HTML webpage is created using the retrieved data to display the 10 most recent detections as a widget in the Node-RED HTML Dashboard. The historical statistics are summarized using JavaScript into a format required by the Chart node to display in the Node-RED HTML Dashboard.

Figure 2 illustrated a prototype of the Couch Guardian. Figure 3 illustrates the configurations of the Node-RED Application.

Figure 2: Couch Guardian Prototype

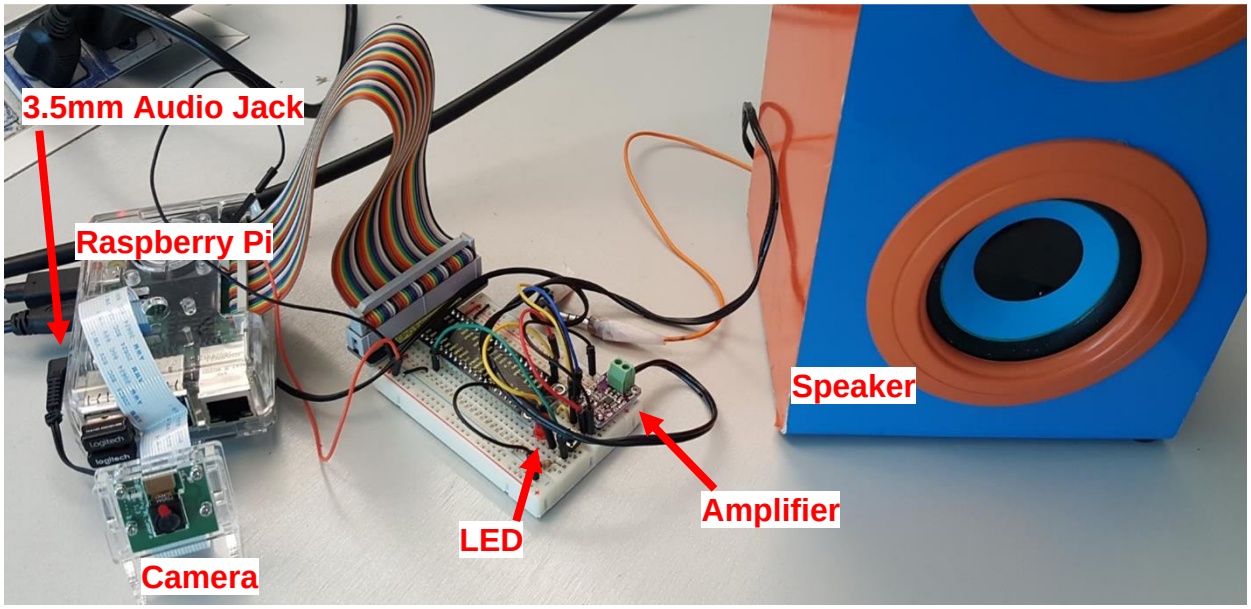
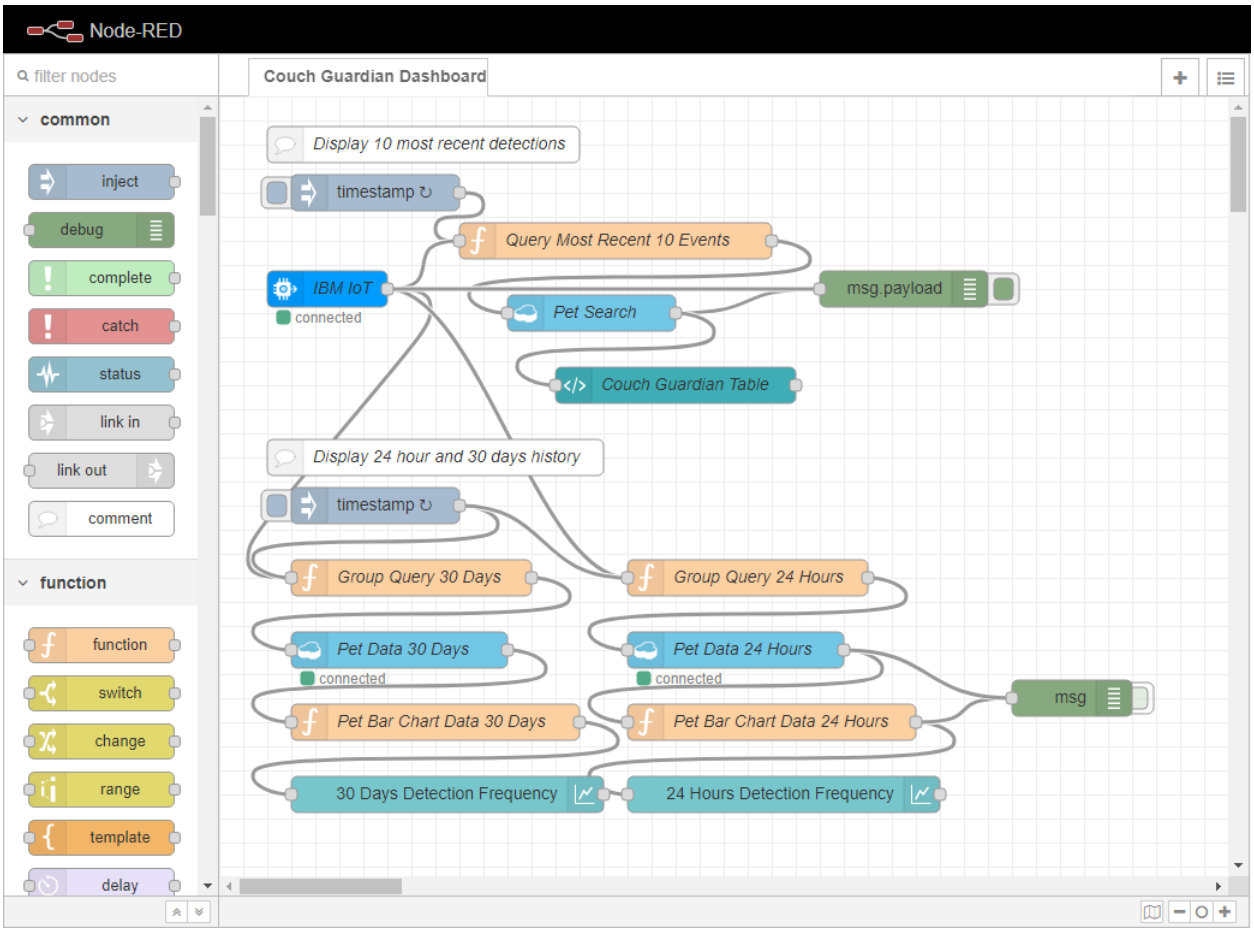


Figure 3: Node-RED Application Configuration



TEAM MEMBERS CONTRIBUTIONS

Peer evaluation is an equal split of 25 points to each member. The final team member contributions are summarized in Table 4.

Table 4: Final Team Member Contributions

Product Name: Couch Guardian	
Scrum Team Name	thePerfect10
Product Owner	David Baston
Final Contributions	▪ Created, owned and managed product backlog on Trello ▪ Software development lead for main Python code ▪ Added object detection functionality ▪ Added personal email functionality ▪ Github repository contributor ▪ Defined Non-Functional Requirements
Scrum Master	Tsz Shing Tsoi
Final Contributions	▪ Created and managed Node-RED HTML Dashboard functionality with Node-RED Application, Watson IoT and IBM Cloudant database ▪ Updated and maintained high level system architecture map ▪ Added sound deterrent functionality to software ▪ Implemented headless Raspberry Pi functionality ▪ Github repository contributor ▪ Presented final prototype with live animal testing supported ▪ Defined Functional Requirements
Individual Contributor	Francis Iniobong John Ifon
Final Contributions	▪ Investigated Competitive Analysis for product market
Individual Contributor	Alyssa Johnson
Final Contributions	▪ Created, edited, updated Final Demonstration video and PowerPoint Presentation ▪ Collaborated with live product dashboard functionality ▪ Defined product Use Cases

REFERENCES

[1] "Pet Industry Market Size & Ownership Statistics", *American Pet Products Association*, 2019. [Online]. Available: https://americanpetproducts.org/press_industrytrends.asp. [Accessed: 21- Sep- 2019].

[2] "PetSafe SSSCat Cat Training Aid", *Amazon.com*, 2019. [Online]. Available: https://www.amazon.com/PetSafe-PDT00-13914-SSSCAT-Spray-Deterrent/dp/B000RIA95G. [Accessed: 26- Sep- 2019].

[3] "PetSafe Pawz Away Pet Barriers", *Amazon.com*, 2019. [Online]. Available: https://www.amazon.com/PetSafe-Barriers-Adjustable-Proofing-Stimulation/dp/B000A76ZYS. [Accessed: 26- Sep- 2019].

[4] "PREDATORGUARD PestAway Ultrasonic Outdoor Animal & Cat Repeller", *Amazon.com*, 2019. [Online]. Available: https://www.amazon.com/PREDATORGUARD-PestAway-Ultrasonic-Repeller-Destroying/dp/B00WXIJHHO. [Accessed: 26- Sep- 2019].

[5] "Doggie Don't Device", *Amazon.com*, 2019. [Online]. Available: https://www.amazon.com/Doggie-Dont-Device-Repellent-Behaviors/dp/B01FE2FYTA. [Accessed: 26- Sep- 2019].

[6] Roppolo, M. (2019). *Pet-damage pricetag reaches $3 billion per year, study claims*. [online] Fox News. Available at: https://www.foxnews.com/tech/pet-damage-pricetag-reaches-3-billion-per-year-study-claims [Accessed 27 Sep. 2019].