TracKat

Author: MeeDm Bossard, Tarush Govil, Lucas Moiseyev: Electrical and Computer Engineering, Carnegie Mellon University

Abstract— TracKat is a novel pet tracking and diagnostic system. The project uses a modular system of custom-made Bluetooth Low Energy (BLE) sensor tags to transmit pet house cat id's and eating habits to a hub device running a local web application. The web app stores and displays daily food and water intake and tracks the information over time. If the system finds any sudden changes in routine, it alerts the user and recommends a check up with a local veterinarian, based on a provided zip code. TracKat is sure to be an excellent tool for cat enthusiasts and vets alike.

***Index Terms*—Bluetooth Low Energy (BLE), Raspberry Pi (Raspi), Radio Frequency Identification (RFID)**

I. Introduction

House cats instinctively hide signs of weaknesses or sickness. One tell-tale sign that a cat is sick or injured is a change in its normal eating habits, but such behavior patterns can be difficult for owners to track. Additionally, if a household has multiple cats that eat freely throughout the day, it is extremely difficult to determine their individual food and water consumption.

TracKat will enable owners of multi-cat households to track and maintain their cat's health, sensing sudden changes in food and water consumption levels per cat. For the device to adequately inform users of their cats' health, it will be able to accurately detect and display cat feeding and drinking with a maximum of 10% false positives and 10% false negatives per day.

While there are devices on the market that are similar, there are a lack of devices that allow you to monitor a cat's eating and drinking habits. For instance, there are devices that first detect a cat's microchip number and then allow specific cats to get to their food. On the flip side, there are autonomous feeders which dispense food on a regular schedule. Both devices do not allow for the user to be able to track eating and drinking for multiple cats in a user-friendly way. To fill this gap, TracKat will be able to both track cats that eat freely and identify which cat is eating.

II. Design Requirements

Having specific design requirements will help ensure that TracKat is able to successfully track fast changes and indicate cat health in multi-cat households.

The most important requirement for TracKat is its ability to **accurately detect a specific cat's eating and drinking habits**. Not only to detect these habits but be able to catch fast changes in consumption as well. ~10% of false positives and ~10% of false negatives translates to every 10 times the cat goes to eat or drink, the value for 1 of those would be off. This way the owner can still get a great sense of the cat's health. It will be tested by attaching a microchip to a 3D printed robot cat which will "use" the device in different ways and then recording the accuracy of the device.

One main aspect is measuring its ability to have **modular configurations**. Like mentioned before, other devices on the market are not suitable for multi-cat households when it comes to regulating food intake. TracKat must be configurable by the user for different combinations, specifically the number of cats and the number of bowls. Users will be able to tailor the device to their own needs, whether it be no microchip reader for a 1-cat household, or only the water bowls, or a combination of food bowls, water bowls, and microchip readers. This will be tested by varying configurations and verifying that the system works for all variations.

To allow the most accurate information as possible about the cat's consumption, the **weight sensing granularity** must be able to detect the smallest size that the cat would eat. This would be one kernel, which weighs around 0.1 grams. Hence the weight sensing granularity must be around 0.05 grams at the least to account for some standard deviation around the minimum weight the cat would consume. Additionally, an average lick of water for a cat is 3/100 of a teaspoon[1], so for 5 grams / teaspoon, this translates to 0.15 grams of water. Hence water intake was not factored into this requirement. To test this, one kernel will be taken out of the bowl, and check that the sensor is able to capture it accurately.

Changing out batteries frequently can get tiresome and redundant for a user. Hence, **maximizing the battery life** of the components of the device will be important in ease of use for the user. Replacing batteries every 8 weeks is reasonable for most electronic devices. This will be tested by running the device for a day, measuring the battery consumption, then extrapolating this value for a year.

Lastly, being able to refer to the cat's information for an extended period can be valuable for the user to check their cat's long term consumption routines. However, any longer than a year as a period **to display the data** won't be as useful for the owner since fast changes in their habits are more likely to be of interest. Hence the data should be available to the user for 1 year.

III. Architecture and/or Principle of Operation

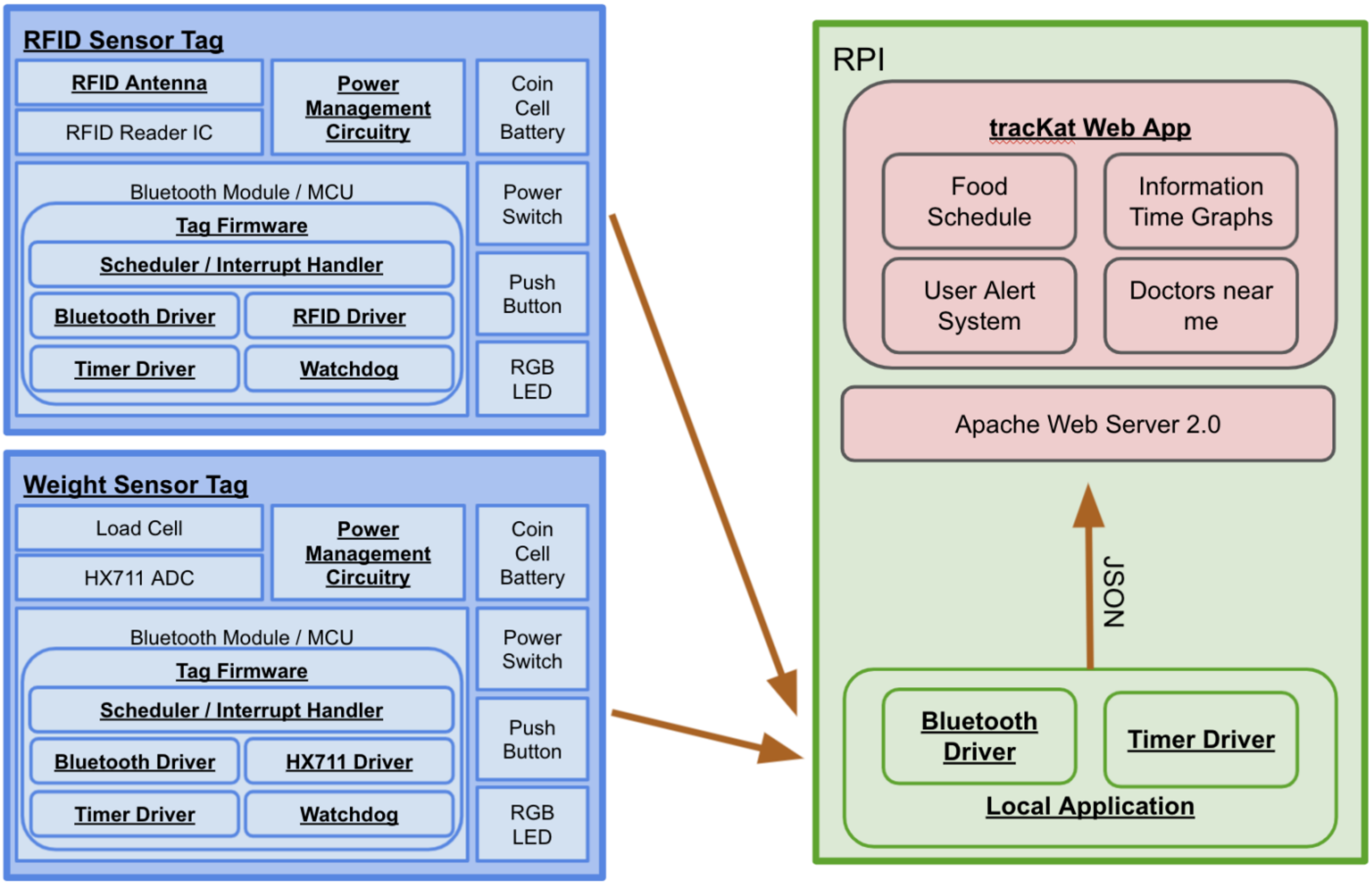

Fig. 1. Block Diagram of Overall System Architecture

The device has three main components: the tags, the hub, and the web app. Each component has an integral part in the overall system, then communicate with each other to achieve the overall device.

<u>Main Components</u>

RFID Tag

The RFID tag reads in the microchip number from the cat. It uses a coil as a sensor to read the microchip number. The tag itself polls the sensor around 1 time every 5 seconds for readings from the sensor and stores those readings. Concurrently, it waits for the hub to ask for those stored microchip number values.

Weight Sensor Tag

The weight sensor tag senses the weight of the food or water bowl. It uses a load cell as a sensor to determine the weight. The tag itself polls 1 time every 5 minutes to record the current weight. Concurrently, it waits for the hub to ask for those stored weight values.

Hub

The hub serves as the connector between the sensors and the web app. It first decodes the information that is transmitted via BLE, then packages it into a readable format, namely JSON, to allow for easy parsing on the web app side which then displays parsed data in a graphical format.

Web App

For the web app, the user must first register or login (if not already logged in) to their account in order to access the rest of the application. They are then redirected to their cat's profile page displaying the eating and drinking trends of their cat, as well as having the option to view nearby vets and pet stores based on their address.

<u>Connections</u>

Sensor -> BLE

The Raspberry Pi will work as a hub to ask and collect data from the tags via BLE 5.0. It will first poll the tags either twice a day or when the user requests it, then receive the data that has been saved on the tags up till that point.

BLE -> RasPi

The raspberry pi will be receiving packets of information through BLE 5.0 using the bluepy python library. It first scans and connects to the correct Bluetooth device. Afterwards, we wait for the packets of information from the sensors to arrive by specifying fixed time intervals at which we request data.

RasPi -> Web App
After the RPi receives data via BLE, it interprets the sensor data by scanning for the headers and reading in the corresponding values. It then packages the data in a standardized JSON file which can be easily read by the web application, which goes on to display the packaged data in a graphical manner.

IV. Design Trade Studies

A. Web App

AWS: EC2 vs S3
The most common ways to deploy to the cloud are Amazon's EC2 and S3 instances. For deciding which one to move forward with, we looked at the primary functionality of each of these services and settled on EC2. We did not have a use case for needing to store large static files on the cloud, which usually is the main purpose of using S3, and hence determined that EC2 would satisfy our need for running and hosting our application.

Web Frameworks: Django vs Flask
We were pretty certain that we wanted to proceed with a Python based web application. Given that, Django and Flask are the two most popular frameworks as of 2021.One of the reasons why Django turned out to be an obvious choice was its built-in object-relational mapping system, which would make querying from the database much easier. Another reason was that Django is much stronger than Flask in terms of security, offering protection against cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection attacks. The final nail in the coffin was Django's built-in authentication system, making it easier to verify users wishing to access information about their cat.

B. Tags

BLE/MCU: DA14531
This chip was chosen for several reasons. Most importantly, it supports BLE 5.1 so it has the capabilities that we need. However, when comparing to other chips on the market, we found that it had every peripheral we needed without being bloated, great documentation, and readily available in stock. It also has extremely low power consumption due to the Arm Cortex M0+ and has 270nA hibernation and uses 3.5mA for the TX line and 2.2mA for the RX line. In general it had all the capabilities we needed without being too much.

Load Cell Amplifier/ADC: HX711
This chip is the industry standard for a load cell amplifier; however it is currently out of stock. We will have to buy the breakout board and de-solder the them to use this chip in our custom PCB.

RFID Reader IC: TMS3705 vs. HTRC11001T
The TMS3705 was our main choice since it is the pet standard and uses 132.4kHz RFID, the international standard for pets. However, it was out of stock, so we had to go with the HTRC11001T, which is similar but instead uses the 125kHz RFID. This is still a pet standard but isn't the international standard which shouldn't be an issue for the scope of our project.

C. Hub

Communication between Components: BLE 5 vs. Wifi
To determine how we would have the sensor communicate with the RPi, we focused on factors like power and flexibility. With respect to power, Bluetooth served to be a better tool as it takes up significantly less power in comparison to WiFi and can be powered via battery for approximately two years, as opposed to WiFi needing to be hooked up to an outlet. Regarding flexibility, Bluetooth allows us to be flexible with the number of bowls we use and if we end up using an RFID chip reader or not.

Bluetooth Version: Bluetooth 4 vs 5 Standard
When looking at speed, bluetooth 5 has twice the speed as compared to bluetooth 4 due to the much larger bandwidth provided in the newer version. With regards to range, bluetooth 5 serves up to 4 times the distance to which it should allow connectivity. When looking at power, the bluetooth 5 tends to use up less, thus allowing us to keep it on for a larger period of time. Finally when looking at support for IoT devices, Bluetooth 5 comes out on top primarily because of its increased range and speed.

V. System Description

A. Weight Sensor Tag

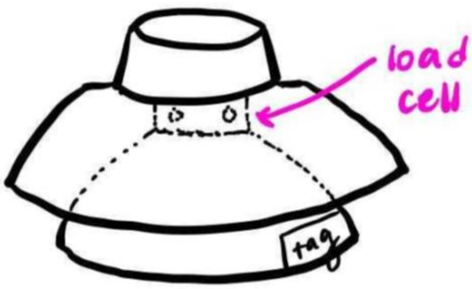

Fig. 2. Diagram of Weight Sensor Tag

The load cell is the actual sensor for the weight. It uses differences in strain in the material to create a differential resistance.

The main consideration for the design of the weight sensor was that it made sure the electronic components were not exposed to the food or water in the bowl or spilled out of the bowl. To ensure this did not happen, we decided on an umbrella shaped design for the 3D printed parts that are below and on top of the load cell.

The umbrella on top will make sure that if a spill were to happen, it would fall down the shape and avoid the underneath area where the tag will be. However, we don't want the top to touch the ground since that would corrupt the weight value coming from the load cell.

The umbrella shape under the load cell will be flush to whatever surface it is on and gives the load cell a stable surface to be on while still letting the top portion to still be off the ground.

The actual tag will be powered with a coin cell and use the HX711 as the ADC chip, which is connected to the load cell. It has an MCU, the DA14531, that then polls the ADC for values and sends via BLE 5 when prompted by the hub.

B. RFID Reader Tag

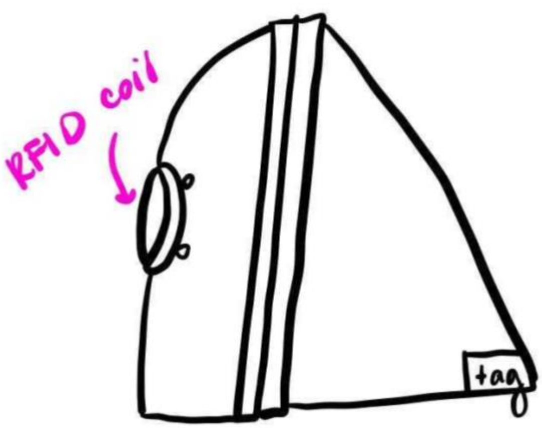

Fig. 3. Diagram of RFID Reader Tag

The sensor that can read microchip numbers is the RFID coil highlighted in the diagram. It is essentially wire that is wrapped around the entrance for the cat's head. When the cat puts their head through the entrance, the reader should be able to read the microchip located in the back of the cat's neck.

The sensor is powered by the HTRC11001T chip, which then sends the microchip number of the specific cat to the DA14531, similarly to the weight sensor tag. The MCU then sends the microchip numbers recorded via BLE 5.0 when prompted by the hub.

C. Hub

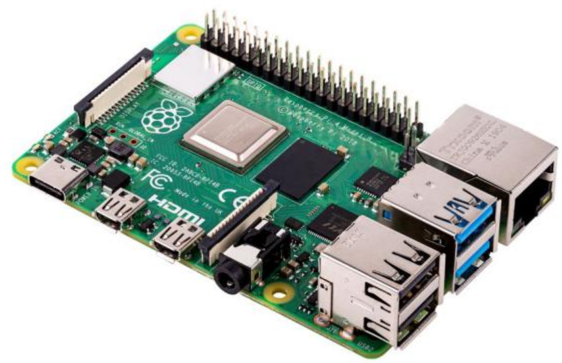

Fig. 4. Raspberry Pi 4 as a Hub

The hub itself is a Raspberry Pi 4. The hub itself sends communications to each of the tags for values every twice per day or on the request of a user. It then is able to send the values that it gets from the tags via a JSON file to the web app.

D. Web App

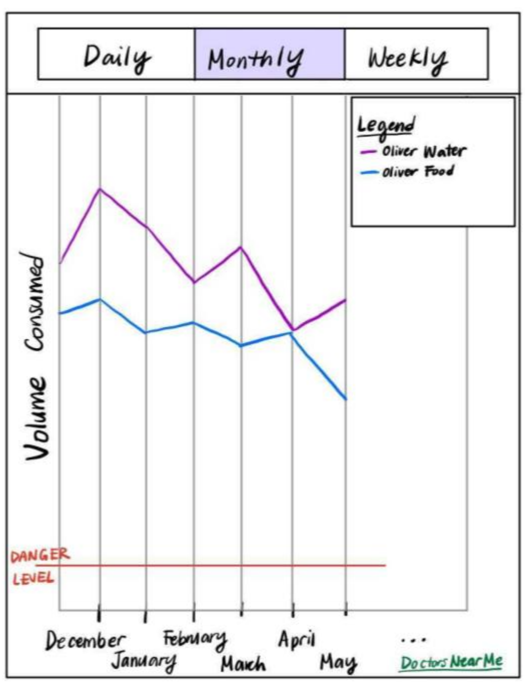

Fig. 5. Diagram of Screenshot of Web Application

The web application will be built using the Django framework and will be deployed on the cloud using an Amazon EC2 instance. For authentication, Django provides us with a user authentication system which will handle user accounts, permissions, and groups. Every time a user visits the web app, they will be met with a login page where they have to enter their username and password. If they do not have an account, there will be an option to go to the registration page where they can register themselves and their cat. Later on, whenever they visit the application, if they are already logged in, they will be able to directly head to the cat's profile page.

On the cat' profile page, ther user will get a graphical display of their cat's eating and drinking habits. The user will be able to view this information in different timeframes, ranging from daily to weekly to monthly. On the same page the user will have the optionality of clicking on a URL titled "Vets near me" which will redirect them to a page showing nearby vets and pet stores based on their address. This will be accomplished by integrating the Google Places API with Django.

We also will be sending SMS notifications via the Twilio SMS API whenever the sensors report back their bowl is empty or there has been a sudden drop in the cat's eating and drinking habits. This SMS will contain a description of what the problem is as well as the link to the "Vets near me" page.

VI. Project Management

A. Schedule

For the most part, there haven't been any major changes in our schedule. By the middle of the week of 10/17, we hope to have a basic demo setup where we can show communication between the BLE to the RPi and the RPi to the web application. This will serve as the basic framework for future work leading up to the interim demo scheduled for November. We hope to be wrapping up the project in the week of 11/21, taking into consideration approximately 2-3 weeks of slack time at the end. Included in Page 8 in Appendix.

B. Team Member Responsibilities

We split responsibilities for each team member based on their strengths. Lucas will primarily work on the hardware side and will get the BLE module setup along with the Load Cell and RFID IC. MeeDm will be establishing communication between the BLE module and RPi hub as well as decoding the packets of information sent from the sensors, and Tarush will be working on building the web application for displaying the data related to each individual cat as well as sending information from the RPi to the web app.

C. Budget

Budget is included in Appendix. Page 7.

D. Risk Management

As of now we hope to be able to finish the development of Shelley, our robocat to demonstrate final testing of our project, however, if time does not permit we plan on using a stuffed animal embedded with a microchip. If we are unable to send and receive back the implemented design of our PCBs, Lucas will be able to develop them out of his house. If we are unable to successfully establish communication between the tags and the hub, we plan on increasing the number of hub devices or potentially re-specing the device. If our cat feeding detection isn't as accurate as intended, we will increase the polling frequency and recalibration routine to weight tags. Risk mitigation for sufficient RFID range involves making improvements to the current existing enclosure design we have planned out and improving weight sensing granularity would involve re-specing our load cell.

VII. Ethical Issues

If TracKat is being relied upon to give an accurate picture of the cat's health and fails, there can be two cases. First, the user may be notified of a potential problem where there is no problem. This wouldn't be too devastating of an impact, but eventually it could get annoying if it continued to happen. The second is that the user may not be notified of a problem when there is a problem with the cat. Since the user is relying on the device to govern the condition of the cat, it could lead to the cat's health declining. This could possibly lead to illness, injury, or even death of the cat. In order to avoid this, a disclaimer will be added to the web application to inform the user to not solely rely on the device to monitor their cat. This device will be within the bounds of 5% false positives and 5% false negatives to further mitigate this issue.

Storing location data of users can be an ethical issue due to privacy concerns. If the stored location somehow gets exposed, it can not only become inconvenient, but dangerous for the user. Having random people know personal information is never good. To mitigate this risk, the location will only be asked for when "Vets near me" is clicked, and not stored after this.

Lastly, some parts of the device, with enough wear and tear, could be ripped off and become a choking hazard. This could affect the cats or small children in the household. Care will have to be given to package the device when handing it to a user so that they know this can happen and to have caution if they have a particularly destructive pet or child.

VIII. Related Work

There are other products out there that use the concept of having an RFID tag to distinguish between an owner's different pets. Products like the SureFeed Microchip Small Cat & Dog Feeder and PetLibro Automatic Cat & Dog Feeder store sealed food and have lids which only open for the designated pet. You can also choose to allot specific timings and frequency of when you want to be feeding your pet. Our project takes this a step further and allows owners to develop a better understanding of what their cats' eating and drinking habits are on a regular basis. We also can immediately detect and notify the owner of sudden changes as opposed to the current products more so functioning as a black box and performing the mundane tasks necessary for feeding their pet, thus removing responsibility from the owners from having to regularly check up on their pet's well-being.

Glossary of Acronyms

BLE – Bluetooth Low Energy
RFID – Radio Frequency Identification
RPi – Raspberry Pi

References

[1] https://www.newportri.com/ZZ/sponsored/20200811/top-5-ways-to-encourage-your-cat-to-drink-more-water?template=ampart
[2] User Authentication in Django https://docs.djangoproject.com/en/3.2/topics/auth/
[3] How to Broadcast SMS text messages from Python 3 Django Web Applications https://www.twilio.com/blog/broadcast-sms-text-messages-python-3-django
[4] SureFeed Microchip Small Dog & Cat Feeder https://www.chewy.com/surefeed-microchip-small-dog-cat/dp/157145?utm_source=google-product&utm_medium=cpc&utm_campaign=hg&utm_content=SureFeed&utm_term=&gclid=CjwKCAjwh5qLBhALEiwAioods4xazbHapEA-djE2OMsKp41S2hRiWLkrfjCjDPrCWgDGxfhLpycVnBoCkE8QAvD_BwE
[5] Petlibro Automatic Dog & Cat Feeder https://www.chewy.com/petlibro-automatic-dog-cat-feeder/dp/303874?utm_source=google-product&utm_medium=cpc&utm_campaign=hg&utm_content=Petlibro&utm_term=&gclid=CjwKCAjwh5qLBhALEiwAioods2d54A1dJfONJXJw9mINDlFYEbLkd4qePIbqOQSyDtPekD1fiauxPhoCj5kQAvD_BwE

Table 1: Budget

Item	Description	Model #	Manufacturer	Qty.	Value
USBC to USB	Used to connect our Raspberry Pi 4 to power	2.0-CM-AM-6FT	Amazon Basics	1	$6.99
Micro HDMI to HDMI Adaptor	Used to connect our Raspberry Pi 4 to display	40506	UGREEN GROUP UNITED	1	$7.99
Cat Bowls	Cat bowls to test our device	none	YASMA	3	$14.99
DA14531-00FXD EVKT-U	DA14531 series Transceiver; Bluetooth® 5 Evaluation Board	DA14531-00FX DEVKT-U	Dialog Semiconductor GmbH	2	$60.00
DA14531-00000 FX2	IC RF TxRx + MCU Bluetooth Bluetooth v5.1 2.4GHz 24-WFQFN, FC	DA14531-00000 FX2	Dialog Semiconductor GmbH	10	$19.94
MMSS8550-H-T P	Bipolar (BJT) Transistor PNP 25 V 1.5 A 100MHz 625 mW Surface Mount SOT-23	MMSS8550-H-T P	Micro Commercial Co	5	$1.00
W25X20CLUXIG TR	FLASH Memory IC 2Mb (256K x 8) SPI 104 MHz 8-USON (2x3)	W25X20CLUXI G TR	Winbond Electronics	10	$5.38
2450AT18D0100 E	RF ANT 2.4GHZ CHIP SOLDER SMD	2450AT18D0100 E	Johanson Technology Inc.	10	$5.61
AWS Credits	Deployment of web app on cloud	EC2	Amazon	1	$50.00
Total Cost					$171.90

Figure: Gantt Chart