Project Part II Documentation

Monica Razak, Anthony Picone, Jennifer Cafiero

Project Description

Our overall project is the implementation of a “Smart Home.” Each sub-team is responsible for a different aspect of the Smart Home, and our team specifically is focused on the Echo Interface/Alexa Integration. Our main project is focused on creating an Alexa Skill that recognizes emotion, and acts accordingly. For example, if the user were to activate Alexa by saying “Alexa, I am feeling sad today,” then Alexa would respond by turning on an ice cream machine and replying back with some comforting advice or a funny joke.

User Stories

1. I am a user who is feeling sad/angry/upset when I come home. I would like my home to react to and improve my mood without me doing any work.
2. As a user, I want to connect Alexa to a central hub so that she can be activated by my voice and then connect to whatever part of the house that can solve my bad mood.
3. As a user, I want to ask alexa to pick up information from the camera at my front door and tell me who is at the door. I want to tell Alexa to either open the door, or shoot a paintball at the person. Alexa will transfer this information to whatever hardware takes care of these responses.

Use Cases

1. User comes home sad. The user wants his/her house to improve his/her mood. They tell Alexa about their mood. Alexa responds with a comforting statement, in addition to activating a part of the home that will enhance the user’s mood (i.e. the ice cream machine)
2. The engineer wants to start setting up the connection between different appliances and alexa. He/she needs a central hub to allow for all the devices to talk to each other. Alexa will then be able to connect to each device in order to process the user’s request.

Test Stories

1. In order to verify the acceptance of this story, we would need to verify that the Echo can recognize the mood keywords
2. In order to verify the acceptance of this story, we would need to verify that the Echo sends the correct signal to the central processor based on the recognized mood

Source Code

On gitlab

Testing Results

Code still doesn’t function properly completely - in the process of testing.

Hardware:

User Stories

Raspberry Pi to Alexa		
Acceptance Test: Pi-Alexa	Priority 1	Story Points 2
Description		
As a Developer I want to be able to send information between the Alexa and the raspberry pi		

Raspberrry Pi connectivity to Node Devices		
Acceptance Test: Pi-Node	Priority 1	Story Points 4
Description		
As a Developer I want to be able to send information between the final hardware nodes to control physical devices, using bluetooth/wifi connectivity		

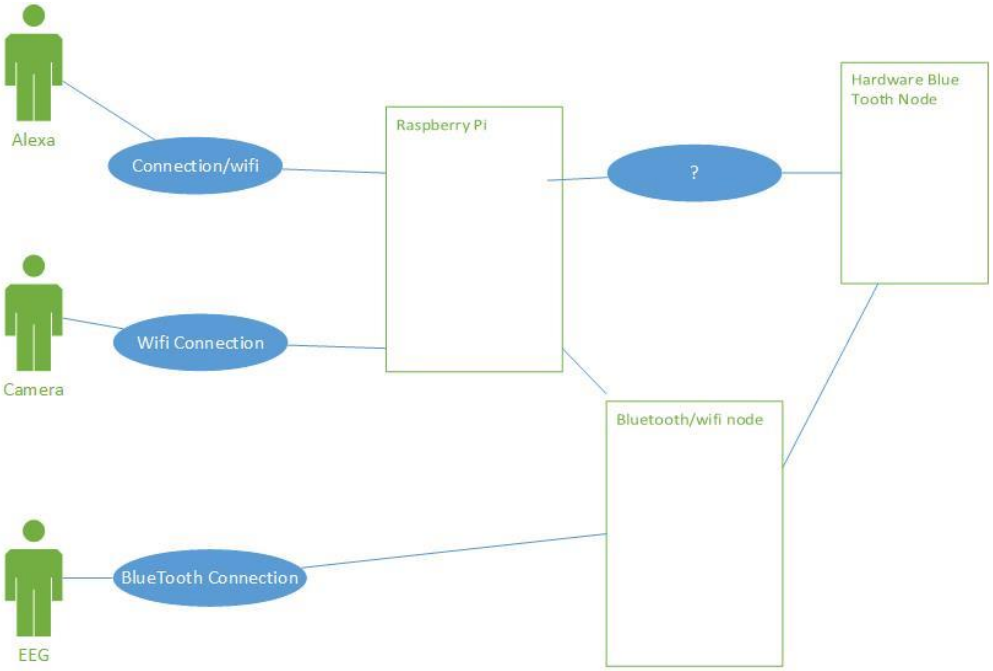
Building Hardware Connectivity		
Acceptance Test: Hardware-On-Off	Priority 2	Story Points 4
Description		
As a Developer, I want to be able to wire up physical devices to the external bluetooth nodes.		

Lighting Node		
Acceptance Test: Lights-On-Off-Adaptive	Priority 2	Story Points 1
Description		
As a customer, I want to be able to change the lights as I desire.		

Fridge Node		
Acceptance Test: FridgeInteraction	Priority 2	Story Points 2
Description		
As a customer, I want to be able to interact with the fridge from afar.		

Creation of Bluetooth Server		
Acceptance Test: CompileAndRunBluetooth	Priority 1	Story Points 2
Description		
As a Developer I want to be able to create a working Bluetooth server on a local host		

Use Cases:



Test Stories:

Raspberry Pi to Alexa		
Acceptance Test: Pi-Alexa	Priority 1	Story Points 1
Description		
Given a connection exists between the Pi and Alexa And there is an Alexa skill to connect the two **When** information is sent by one **Then** ensure that the information is received and handled correctly		

Compiling and Running Test Bluetooth Code		
Acceptance Test: CompileAndRunBluetooth	Priority 1	Story Points 1
Description		
Given code for a Bluetooth server to allow for incoming messages and processing, **When** it is essential to have a working server to progress in the project **Then** ensure that the server is able to compile and run on a local machine.		

Testing Results:

Currently, we have finished the Creation of Bluetooth Server user story, and have completed the CompileAndRunBluetooth test story. The test did work properly, as the server code was able to compile and run on a computer.

The next step is to finish the setup of the raspberry pi, and to have the server running on it.

Christian Bouwense, Alex Massenzio, Justin Zauderer
EEG Retrieval – Project Part II
4/18/17

Description:

We are the sub team responsible for retrieving the data from the MindWave Mobile unit. We will be giving the stream of data to the EEG Processing sub team. This is done through the use of a Java class, which is used by EEG Processing.

User Stories:

Title: How to Get Thoughts from EEG		
Acceptance Test: getThoughts	**Priority: 1**	**Story Points: 7**
Description		
As a programmer		
I want to be able to find out how data is represented and given from the Mindwave		
So that we can interface with the device and store its raw data		

Use Cases:

Use Case 1:

- **Name**: New Mood
- **Description:** User puts on headset and is in a mood.
- **Actors:** Customer
- **Flow:** The customer comes home, and puts on the headset. They turns it on, and it connects via Bluetooth to the central Raspberry Pi of their house. The headset begins to stream data about their mood to the processing unit.

Test Stories:

Title: How to Get Thoughts from EEG		
Acceptance Test: getThoughts	**Priority: 1**	**Story Points: 7**
Description		
Given a user has the headset on properly and all the hardware is turned on correctly		
When the user thinks something		
Then read their thoughts		

Testing Results: We have not fully implemented the Bluetooth connectivity with the central Raspberry Pi. Once we do, we will be able to test connectivity at differing ranges times. We will have to also test different moods and see if there is any loss of accuracy in the data transmissions.

EEG Processing Sub Team

Information

Hello, we are a team of four people dedicated to interpreting the data and running code to interpret the users mood. Our names are Matt Colozzo, Khanh Nguyen, Jonathan Pavlik, and Zeke Zhao. Some goals of our team is to be able to identify the brain wave data given to us by the EEG Retrieval team's code. With that data with then intend to do some tasks such as dispense ice cream if the user was sad.

Some limiting factors of our team are that our code cannot be 100% tested till the EEG Retrieval teams has their setup ready to go for us. As such we have communicated closely with that team to understand how we will be receiving data from them and getting as close as we possibly can. We can still manually send data to test, and hopefully this allows us to mitigate the amount of changes we need to make at a later date.

User Stories

Ice Cream Mood

Acceptance Test:
IceCreamDispenseMoodTest

Priority: 3
Because this was a customer requested feature

Story Points: 1
As this is a relatively easy feature to implement

User: Is in a sad mood
Given: The data from the EEG Reading
Do: Determine the mood is sad

Use Cases

Use Case 1

- **Name:** Sad Mood(dispense ice cream)
- **Description:** User is in a sad mood
- **Actors:** Customer
- **Flow:** The customer is wearing the EEG sensor which is connected to the smart home who receives the reading then runs our code to determine mood, then sends signal to dispense ice cream.
- **Alt Flow:** The user is no longer sad and no longer needs ice cream or more ice cream

Test Stories

Ice Cream Dispense

Acceptance Test: IceCreamDispenseTest

Priority: 3
Because this was a customer requested feature

Story Points: 3
As this requires several pieces of hardware to properly communicate to get the ice cream dispensed

User: Is in a sad mood
Given: The data from the EEG Reading
Do: Determine the mood is sad and send a signal to hardware side who would make sure ice cream is dispensed

Source Code

```
public class EEGProcessing {

  //create array of booleans for raspberry pi to read
  static int[] sendData(boolean sad, boolean happy) {
    int s = (sad) ? 1 : 0;
    int b = (happy) ? 1 : 0;
    int[] emotionArray = {s, b};
    return emotionArray;
  }

  //check if sad
  boolean dispenseIceCream(int brainwaves) {
    if(brainwaves < 38 && brainwaves > 30) {
      return true;
    }
    return false;
  }

  public static void main(string[] args) {
    //class object from other team
    EEGRetrieval data = new EEGRetrieval();
    //initialization don't assume emotions
    boolean sad = false, happy = false;
    //check if user is wearing
    while(data.waves != 0) {
      //update waves(it might do this automatically)
      data = new EEGRetrieval();
      sad = dispenseIceCream(data.waves);

      //at end send info
      sendData(sad, happy);
    }
  }
}
```

Testing Results

Test Code

```
import org.junit.Test;
import static org.junit.Assert.assertEquals;
import org.junit.runner.JUnitCore;
```

```
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestEEG {

   String emotionalBaggage = [true, false];
   int low_alpha_waves = 35;
   EEGProcessing emotional = new EEGProcessing(emotionalBaggage);

   @Test
   public void testEmotion() {
      assertEquals(emotionalBaggage,emotional.sendData());
      assertEquals(low_alpha_waves, emotional.dispenseIceCream());

   }
}

public class TestRunner {
   public static void main(String[] args) {
      Result result = JUnitCore.runClasses(TestEEG.class);

      for (Failure failure : result.getFailures()) {
         System.out.println(failure.toString());
      }

      System.out.println(result.wasSuccessful());
   }
}
```

Test Results

Issue: Cannot actually test full operation. This is due to needing other teams to be finished before we can fully implement and test our features.

Matt McCreesh, Eric Zhen, Louis Rozencwajg-Hays, Ayal Ciobotaru

Facial Recognition

Description

This project is to recognize faces at the door of the home owner. It will text him if the person at the door matches someone it has in its database. We started by writing this in Java with JavaFaces, but that solution was not viable. Now we are doing it in Python with OpenCV, an open source project written in C++ that can work with Python.

User Stories

1.

Title: Camera turns on and takes picture		
Acceptance test: CameraTest	Priority: 1	Story Points: 2
User walks to front door/gate which causes motion sensor to turn on camera. Camera auto focuses and takes picture of person.		

Status: Pending having camera. Right now we are running project on Windows and using built in laptop camera.

2.

Title: Match Picture to Database		
Acceptance Test: MatchTest	Priority: 1	Story Points: 2
Facial recognition software attempts to match picture of user's face to a database of homeowner's Facebook friends (or other database).		

Status: Complete and tested. Still trying to get accuracy higher though. Uses local database.

3.

Title: Match Found		
Acceptance Test: KnownPerson	Priority: 1	Story Points: 1
Facial recognition software on local microprocessor finds a match and sends the homeowner a message with the picture of and who is at the front door/ gate. If match is not found, homeowner still gets a message that somebody unknown is at front door		

Status: Complete. Text message of persons name is sent to number of file if face is recognized.

4.

Title: Denying access into House		
Acceptance Test:AccessDenied	Priority:3	Story Points: 1
Homeowner has a feature to block people from entering the house even though they have previously been allowed to enter with dual protection		

Status: Pending and may get scrapped.

5.

Title: Unlocked		
Acceptance Test: FingerUnlock	Priority: 3	Story Points: 3
Facial Recognition software as well as FingerPrint sensing will be able to allow a user to enter a house without the use of a key.		

Status: Pending considering security concerns

Test Cases

Title: Camera Test		
Acceptance test: CameraTest	Priority: 1	Story Points: 2
Given that a person has walked up to the door/gate, test that the camera focuses properly, takes a picture, and only takes a picture when the camera is properly focused.		

Status: Pending hardware, but works on laptop camera.

Title: Match Test		
Acceptance test: MatchTest	Priority: 1	Story Points: 2

Given a picture and a database of faces, the program matches the picture taken by the camera to the person's face in the database if there is a match.

Status: succeeded

Use Cases

Home owner gets text message to tell him who is at the door when his friends come over.

Friends faces can easily be added to the database and let into the house if they are at the door.

Owner can choose to lock out certain people when they come to his door.