

# הטכניון

## מכון טכנולוגי לישראל



Project in Arduino & IoT

## Auto Recycle Machine

Itay Dabran

Boris Ban Sosin

Lina Maudlej

Gal Peretz

Dor Stern

Yoav Malul

Kfir Shalev

תמצית :

המסמך הבא מפרט את הרכיבים ממנו בנוי הפרויקט , את אופן העיצוב הכולל של המוצר וכן את החלק התכנותי של הפרויקט העוסק במכונת המחזור האוטומטי.

מכונת מחזור אוטומטי שתמיין את מוצרי מחזור : זכוכית , פלסטיק , קרטון , מתכת וכו'.

הסיווג של המוצרים נעשה על ידי תוכנה בענן של azure המשתמשת ב custom vision AI , ובמספר חיישנים הבודקים תכונות פיזיות נוספות שיפורטו במסמך.

מטרת העל של הפרויקט: בנאדם המעוניין למיין את מוצרי האשפה למחזור יגיע אל המכונה יניח את הפריט במרכז הקופסא, הפריט יזוהה על ידי המערכת זיהוי תמונה וכן על ידי החיישנים ויושלך לפח המתאים.

תיאור :

מכונת המחזור האוטומטית בנויה מעץ. המכונה מורכבת מקופסת עץ שנעה על גבי מערכת מסילות שמתחתיה פזורים הפחים, מיקום הפחים נקבע באופן רנדומלי אך הוגדר בקוד בהתאם לתבנית ספציפית.

המוצרים הניתנים לזיהוי על ידי המכונה הם : זכוכית , פלסטיק , מתכת , וקרטון אך ניתן להתאים את המערכת לסוגי מוצרים נוספים , אך העניין דורש אימון נוסף למערכת זיהוי התמונה, וכן שינוי גודל המסילה.



המערכת פותחה בעזרת מכשירי Raspberry Pi 3 B + וכן שני Arduino Uno , כאשר ברספברי הקוד נכתב בשפת פייתון ועל הארדואינו נכתב קוד בשפת C++ .

רפסברי פי הוא מכשיר ממשפחת מחשבי לוח יחיד , בעלי מיקרו מעבדים.(איור 1)

ארדואינו הוא מיקרו-בקר בעל מעגל מודפס יחיד עם סביבת פיתוח משולבת אשר מטרתה ליצור סביבה נוחה וזולה לפיתוח פרויקטים המשלבים תוכנה עם רכיבי אלקטרוניקה.(איור 2)

הפלטפורמה הפיזית של הארדואינו מבוסס על מעגלים המודפסים על הלוח אשר מקשרים בין המיקרו קונטרולרים לבין היציאות output/input, הארדואינו מכיל ווסת מתח וממשק USB המאפשר תקשורת עם המחשב המשמש לתכנות. המתח עבור הארדואינו יכול להגיע דרך כבל USB המחובר למחשב וכן דרך מתח DC על ידי ספק כוח של 9 וולט.



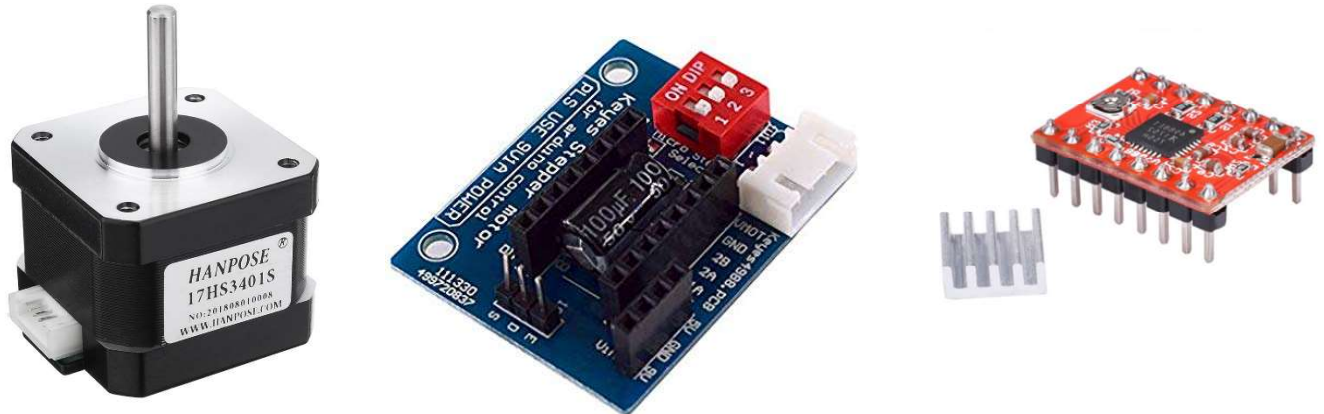
איור 2



איור 1

חומרים ורכיבים נוספים שבהם היה שימוש:

- Nema 17 – מנוע צעד שנשלט על ידי דרייבר A4988 היושב על HW4-434 controller.



מנוע צעד הוא סוג של מנוע סינכרוני אשר מחלק את הצעדים שלו לרגעים. ישנם שני סוגי מנועי nema אנו נשתמש בפרויקט בסוג הדו קוטבי.

על מנת להניע את מנוע הצעד דרך הארדואינו, נדרשת תוספת הדרייבר והקונטרולר המאפשרים לווסתם את זרימת הזרם הדרושה להפעיל את המנוע עצמו. רכיב זה הוא ה A4988.

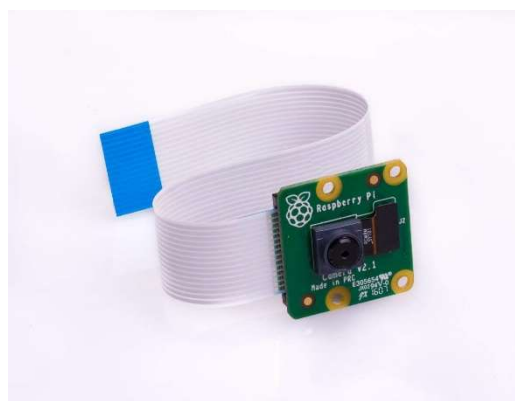
- MG 996R – סרבו מוטור

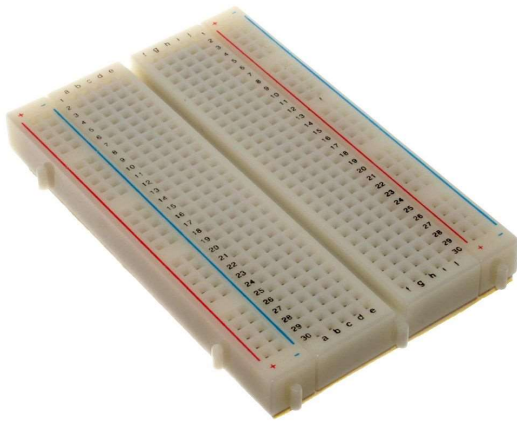
סרבו מוטורים הם סוג מסוים של מנוע בדרך כלל בעלי עוצמה קטנה. הדגם בו נשתמש הוא MG996R בעל סיבוב רציף.



- Raspberry Pi camera

מצלמת 5 מגה פיקסל שמתחברת לרספברי על ידי כבל ריבון

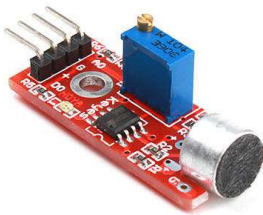




- Breadboard  
לוח בעל תשתית למעגלים חשמליים



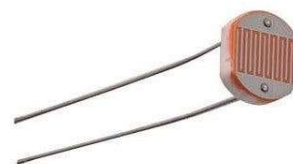
- – Micro Servo SG90  
סרבו מוטור קטן בעל יכולת תנועה של 180 מעלות.



- Uxcell sound sensor  
חיישן קול בעל רגישות גבוהה על ידי מיקרופון.



- Photoresistor and led 20mm



- סליל מחוט נחושת
- כפתור

רכיבים של המבנה :  
לוחות עץ  
מסמרים וברגים  
גומי הנעה  
4 גלגלים של ספה

## תהליך העבודה לבניית המערכת:

התהליך מחולק לשני חלקים, החלק של תכנון המבנה הפיזי והפעלתו והחלק התכנותי. נתחיל מהחלק הפיזי של חומרת המערכת.

ראשית השתמשנו ב-4 חתיכות עץ לטובת יצירת הקופסה המרכזית ועטפנו אותו בנייר עטיפה. בתוך הקופסה מונח הפריט שאמור להיות מזוהה על ידי לקיחת תמונה ממצלמת ה-PI camera (פריט 7) את המצלמה מיקמנו בזווית אופטימלית על חתיכת עץ המוצמדת לדופן הקופסה לטובת זיהוי כל הפריטים בצורה המיטבית ביותר ואת הזווית המדויקת השגנו על ידי זווית מפלסטיק. על אותה חתיכת עץ גם יושבים הרספברי (פריט 6) וכן מוט עץ נוסף (פריט 8) שבקצהו חוט שמאפשר את הפתיחה והסגירה של דלת הקופסה.

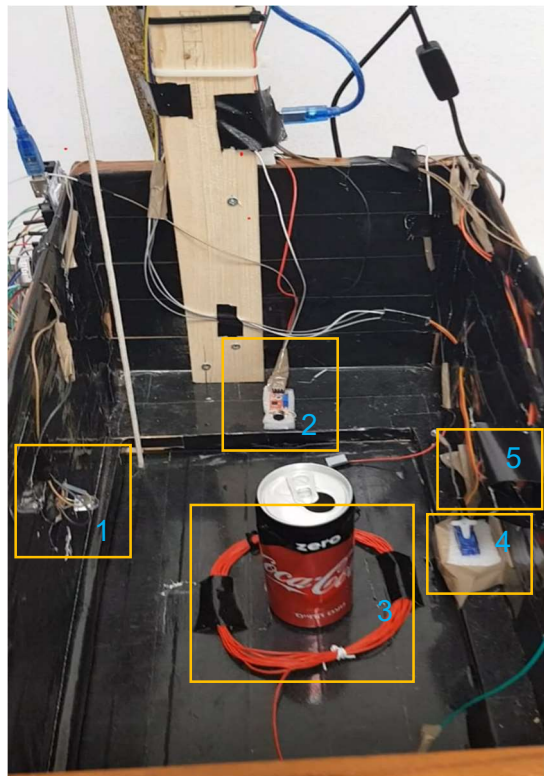
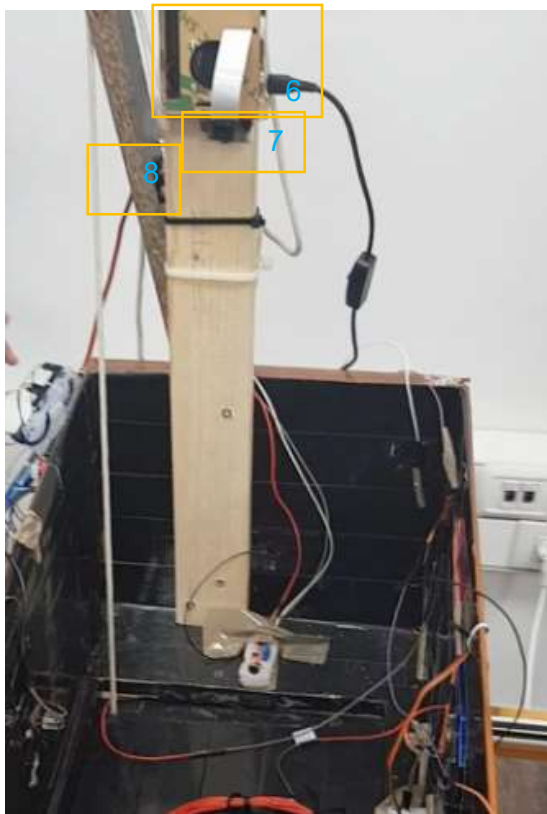
בתחתית הקופסה ישנה דלת נפתחת שדרכה המוצר נופל את פח המחזור. לאחר שסיימנו לבנות ולתכנן את הקופסה הראשית בנינו את המסילה והתקנו את הגלגלים (פריט 9) עבור הקופסה בצורה שיאפשרו תנועה על גבי המסילה.

הקופסה וזה על ידי מנוע הצעד שמחובר לרצועת הגומי (פריט 10), הרצועה הזאת מחוברת בנוסף גם לקופסה ועל ידי כך המנוע מאפשרת את תנועתה על גבי המסילה.

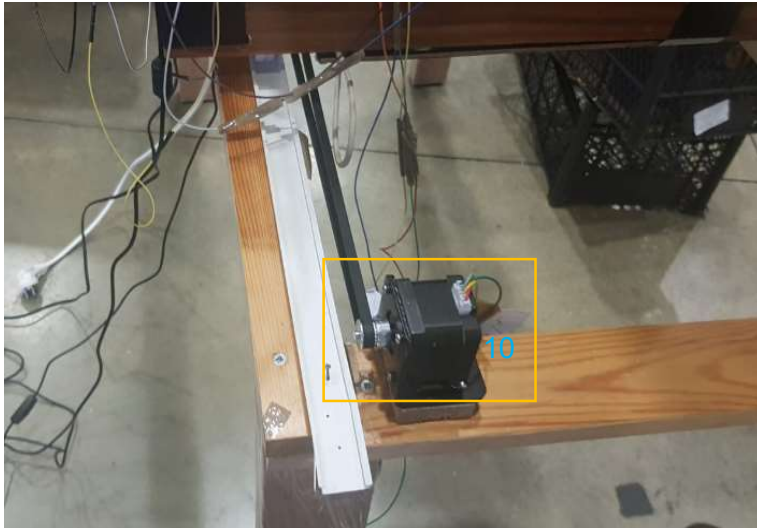
לאחר שסיימנו את הבנייה הפיזית של המערכת, החלנו בבדיקת הניסויים עבור החיישנים ומיקומם. חיישן גילוי המתכות המורכב מסליל של חוט נחושת מוקם במרכז הדלת הנפתחת ויוצרת מעיין תחום שבו צריך לשים את הפריטים.

חיישן הקול – מוצב בצד הקופסה ובצידה האנכי מוצב הסרבו הקטן שתפקידו להכות על מוט ברזל על הפריט בכדי לספק אינדקציה לחיישן הקול עבור זכוכית או פלסטיק.

חיישן מדידת חדירות האור – מורכב מחיישן הפוטורסיסטור וכן נורת הLED שמורכבים בדפנות הקופסה. ולבסוף הוספנו את הכפתור שתפקידו לאותת למערכת כי הוכנס פריט חדש.







לאחר שסיימנו את כל הבנייה הפיזית של המערכת חיברנו את המעגלים החשמליים להפעלת כל החיישנים והמנועים באופן המתואר בשרטוט בעמוד המצורף

החלק התכנותי :

מימוש וניהול החיישנים והמנועים על ידי התוכנה בוצע על ידי קוד ברספברי וכן קוד על הארדואינו.  
חלוקת האחריות ביניהם מוגדרת באופן הבא:

ארדואינו -

- מנוע צעד - אחראי על תזוזת הקופסה
- גלאי המתכות - כל המעגל החשמלי שעליו מבוסס גלאי המתכות מנוהל על ידי הארדואינו
- שולח פולסים חשמליים למעגל ומודד את הזמן שלוקח לקבל להיטען ולפרוק.
- גלאי קול
- פוטורסיטור והלד - הפעולה של הדלקת הלד ומדידת עוצמת האור העובר
- הסרבו שאחראי על סגירת הדלת.

רספברי -

- כפתור - שתפקידו לסמן למערכת שפריט חדש הוכנס
- צילום התמונה - הרספברי לוקח תמונה באמצעות pi camera ושולח אותה לענן שבתורו מחזיר את הסיווג המתאים.

החבילות הנדרשות להתקנה על הרספברי הן :

RPI.GPIO  
Picamera  
Pyserial  
Camera  
Azure-cognitiveservices-vision-customvision  
Python-firebase

והקוד שביצע הרספברי הוא :

```
from picamera import PiCamera
from clarifai.rest import ClarifaiApp
import RPi.GPIO as GPIO
import json
import time
import serial
import datetime
import urllib2, urllib, httplib
from time import sleep
from serial import Serial
from firebase import firebase
from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
ENDPOINT = "https://recyclevision2019-prediction.cognitiveservices.azure.com/"
prediction_key = "*****"
predictor = CustomVisionPredictionClient(prediction_key, endpoint=ENDPOINT)
firebase = firebase.FirebaseApplication('*****', None)
button = 12
#setup camera
camera = PiCamera()
camera.resolution = (648, 484)
camera.rotation = 180
#setup arduino serial
arduino = Serial('/dev/ttyACM1', 9600)
#clean serial
time.sleep(2)
arduino.flushInput()
time.sleep(0.5)
#board setup
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(button, GPIO.IN, pull_up_down = GPIO.PUD_UP)
#inner stepper setup
GPIO.setup(37, GPIO.OUT)
p = GPIO.PWM(37, 50)
```



```

# #-----start the all operation-----
try:
    while True:
        # #-----update the state in firebase before picture-----
        firebase.put('/state',"state","before")
        firebase.put('/item',"item","no")
        print("start")
        channel = GPIO.wait_for_edge(button,GPIO.FALLING)
        if channel == button:
            print(channel)
            # #-----take picture-----
            camera.start_preview()
            time.sleep(1)
            camera.capture('/home/pi/image-recognition.jpg')
            camera.stop_preview()
            # #-----check the prediction in azure-----
            best_pred_label = ''
            best_pred_prob = 0
            with open("/home/pi/image-recognition.jpg", "rb") as image_contents:
                results = predictor.classify_image(
                    "9a61cf61-dff8-44e6-9ee2-13962b8c8fdd", "checkone", image_contents.read())
                # Display the results.
                for prediction in results.predictions:
                    if prediction.probability * 100 > best_pred_prob:
                        best_pred_prob=prediction.probability * 100
                        best_pred_label= prediction.tag_name
                    print("\t" + prediction.tag_name +
                        "\t: {0:.2f}%".format(prediction.probability * 100))
            print(best_pred_label)
            print(best_pred_prob)
            # #-----check if metal-----
            metal = False;
            cardbord = False;
            arduino.write('6') #call arduino function to metal detect
            r = arduino.readline()
            print(r)
            r = arduino.readline()
            print(r)
            if r.startswith( 'METAL' ):
                metal= True
            arduino.write('0')
            # #-----check if cardboard-----
            arduino.write('7')
            r = arduino.readline()
            print(r)
            r = arduino.readline()
            print(r)
            if r.startswith( 'CARDBORD' ):
                cardbord = True;
            arduino.write('0')
            # start operation of servo sound sensor
            i = 0
            while i < 3:
                p.start(0)
                p.ChangeDutyCycle(3)
                sleep(0.5)
                p.ChangeDutyCycle(7)
                sleep(0.5)
                p.ChangeDutyCycle(3)
                sleep(0.5)
                p.stop
                i = i + 1
            # move according the item and update the firebase that added item

```

```

if metal == True :
    arduino.write('1')
    firebase.put('/item',"item","add")
elif best_pred_label == "plastic":
    firebase.put('/item',"item","add")
    arduino.write('3')
elif best_pred_label == "glass":
    firebase.put('/item',"item","add")
    arduino.write('2')
elif cardbord == True and best_pred_label == "cardboard" :
    arduino.write('4')
elif best_pred_label == "cardboard" and best_pred_prob==100:
    firebase.put('/item',"item","add")
    arduino.write('4')
else:
    # we get here if we dont recognize the item and let the user answer
    firebase.put('/predict',"label","unknown")
    firebase.put('/state',"state","after")
    label = firebase.get('/predict',"label")
    time.sleep(15)
    label = firebase.get('/predict',"label")
    state = firebase.get('/state',"state")
    if label == "can":
        arduino.write('1')
    elif label == "cardboard":
        arduino.write('4')
    elif label == "plastic":
        arduino.write('3')
    elif label == "glass":
        arduino.write('2')
    print("ask user")
    arduino.write('0')
    print("end")
except KeyboardInterrupt:
    GPIO.cleanup()
GPIO.cleanup()

```

```

#include <Stepper.h>
#include <Servo.h>
#define STEPS 200
#define pulsePin A4
#define capPin A5
#define ledLight 6
#define pResistor A0
#define photoval 240
#define metalval -26500
Servo myservo;
Stepper stepper(STEPS, 5, 4);
int soundDigital = 2;

void setup() {
    // delay(1500);
    Serial.begin(9600);

    // setup servo
    myservo.attach(9);
    myservo.write(180);

```

הקוד שמבוצע בארדואינו :

```

// setup photo resistor
pinMode(ledLight, OUTPUT);
pinMode(pResistor, INPUT);

//setup metal detect
pinMode(pulsePin, OUTPUT);
digitalWrite(pulsePin, LOW);
pinMode(capPin, INPUT);

//reset metal detect
pinMode(capPin, OUTPUT);
digitalWrite(capPin, LOW);
delayMicroseconds(20);
pinMode(capPin, INPUT);
applyPulses();

//setup motor
stepper.setSpeed(350);

//soundsensor
pinMode(soundDigital, INPUT);
}
int attuale ;
int SensorDataDigital;
int counter = 0;

void loop() {
// read num from rasbarry
if(Serial.available()){
    delay(100);
    attuale = Serial.read()-'0';
//    Serial.println(attuale);
}

// do nothing
if(attuale == 0){

}

// move first bin - can
if(attuale == 1){
    move_first_bin();
}

```

```

// move second bin - cardboard
if(attuale == 2){
    move_second_bin();
}

// move third bin - plastic
if(attuale == 3){
    move_third_bin();
}

// move forth bin - glass
if(attuale == 4){
    move_forth_bin();
}

//metal detect
if(attuale == 6){

// METAL DETECTOR
int val2 = 0;

for(int i = 0; i < 256; i++){
    pinMode(capPin, OUTPUT);
    digitalWrite(capPin, LOW);
    delayMicroseconds(20);
    pinMode(capPin, INPUT);
    applyPulses();
    val2 += analogRead(capPin);
    delayMicroseconds(104);
}

Serial.println(val2);
if(val2 <= metalval) {
    Serial.println("METAL");
}
else{
    Serial.println("NOT");
}
}

//END metal detect

//photo resistor check
if(attuale == 7){
    digitalWrite(ledLight, HIGH);
    int minv = 400;

```



```

    delay(1000);
    for(int i = 0; i < 4096; i++){
        int val5 = analogRead(pResistor);
        // Serial.println(val5);
        if(minv > val5){
            minv = val5;
        }
    }
    digitalWrite(ledLight, LOW);

    //          delay( 300 );
    Serial.println(minv);
    if(minv < photoval){
        Serial.println("CARDBORD");
    }
    else{
        Serial.println("NOTC");
    }
}

//soundservo
if(attuale == 8){
    SensorDataDigital=digitalRead(soundDigital);
    counter = counter + SensorDataDigital;
}
if(attuale == 9){
    Serial.println(counter);
    counter = 0 ;
}
delay(500);
}

void applyPulses()
{
    for (int i=0;i<3;i++)
    {
        digitalWrite(pulsePin,HIGH); //take 3.5 uS
        delayMicroseconds(3);
        digitalWrite(pulsePin,LOW); //take 3.5 uS
        delayMicroseconds(3);
    }
}

```

```

void move_first_bin()
{
    myservo.write(15);
    delay(5000);
    myservo.write(180);
}

void move_second_bin()
{
    delay(4000);
    stepper.step(5250);
    delay(4000);
    myservo.write(15);
    delay(5000);
    myservo.write(180);
    delay(5000);
    stepper.step(-5250);
}

void move_third_bin()
{
    delay(4000);
    stepper.step(10800);
    delay(4000);
    myservo.write(15);
    delay(5000);
    myservo.write(180);
    delay(5000);
    stepper.step(-10800);
}

void move_forth_bin()
{
    delay(4000);
    stepper.step(17000);
    delay(4000);
    myservo.write(15);
    delay(5000);
    myservo.write(180);
    delay(5000);
    stepper.step(-17000);
}

```

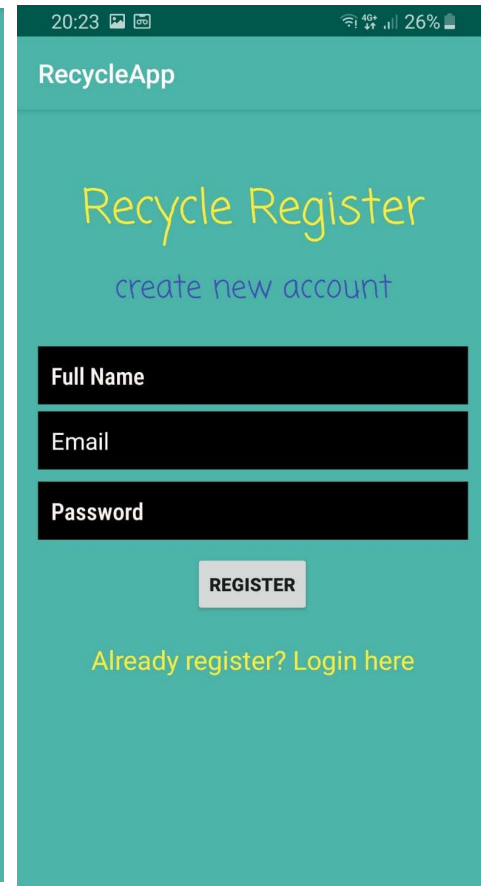
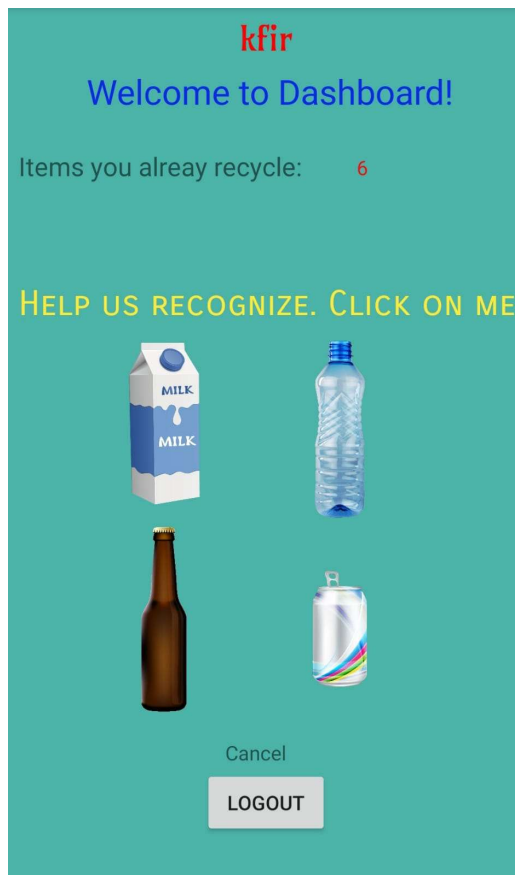
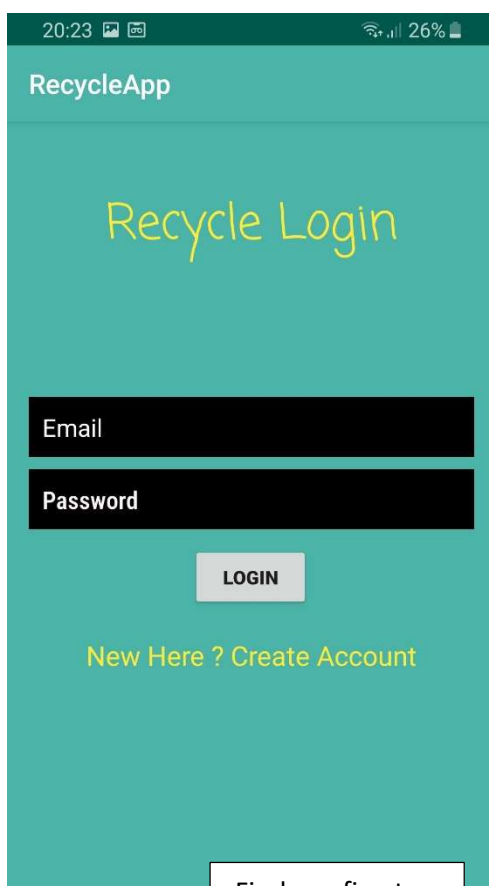
## אפליקציה :

האפליקציה נועדה כדי לאפשר מעקב אחר משתמשי המערכת ולדעת כמה כל משתמש מחזר ומאיזה סוג של פריטים.

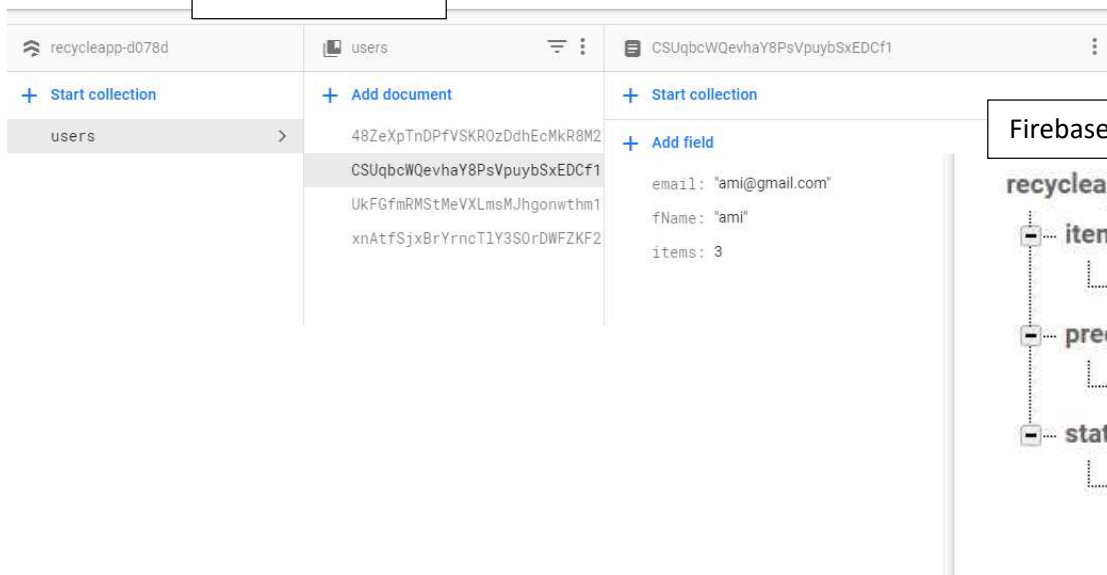
שימוש נוסף באפליקציה הוא כאשר המערכת לא מזהה את הפריט, המשתמש מקבל אופציה לבחירה ידנית של הפריט ובכך מסייע למערכת להגיע לפח המתאים.

האפליקציה נכתבה בjavascript בandroid studio ולשם שמירת הנתונים שם המשתמשים השתמשנו בFirebase של גוגל.

באפליקציה ישנם 3 מסכים, מסך הרשמה, מסך התחברות ומסך המציג את הנתונים עבור המשתמש



Firestore



Realtime

```
recycleapp-d078d
├── item
│   └── item: "no"
├── predict
│   └── label: "can"
└── state
    └── state: "before"
```

## סדר הפעולות בהפעלת המערכת :

1. המשתמש מניס את הפריט במרכז הקופסה ולוחץ על הכפתור
2. המצלמה לוקחת תמונה והתמונה נשלחת לענן
3. הענן מחזיר את הזיהוי המשוער של הפריט
4. הגלאי מתכות מבצע את פעולתו ומזהה אם הפריט מתכתי
5. האור נדלק ומוודד את כמות האור המתקבלת בחיישן
6. הסרבו הקטן מקיש על הפריט כדי לזהות את הקול
7. התוכנית מחליטה באיזה פריט מדובר
8. אם התוכנית לא הצליחה להסיק את סוג הפריט מופיע בקשה באפליקציה לזיהוי ידני
9. המנוע צעד מתחיל לפעול ומזיז את הקופסה למיקום הנכון
10. הסרבו מוטור מופעל וגורם לפתיחת הדלת
11. הסרבו מוטור מופעל וסוגר את הדלת
12. מנוע הצעד מחזיר את הקופסה למיקום ההתחלתי



## **מסקנות והצעות לשיפור :**

האב טיפוס הנוכחי בוצע על ידי חומרים אלמנטריים וחתיכות עץ שנמצאו , אבל הבנייה יכולה להתבצע על ידי רכיבים איכותיים יותר ובהרכה איכותית יותר, ואף לעצב את המערכת כך שיום אחד יוכלו לשים אותה במקומות פרטיים וציבוריים כמו פח רגיל לכל דבר.

כדי לשפר את המבנה יש צורך להשתמש במסילות מותאמות יותר שבהם הקופסא תזוז ללא הפרעה. הצעות נוספות לשיפור :

1. ניתן להוסיף למערכת מתקן כלשהו שיבצע דחיסה לאשפה לפני זריקתה לפחים, כדי לחסוך מקום.
2. חיישני משקל שימוקמו בתחתית הפחים ויהיו מחוברים לצד כלשהי שבו יהיה חיווי על מצבם מבחינת כמות ותפוסה
3. הוספת חיישני NFC אשר יסמנו לקופסה מתי היא מעל הפח המתאים וכך לאפשר למערכת להיות דינמית מבחינת מיקום הפחים ולא יהיה צורך להגדיר מראש את מספר הצעדים של המנוע. בעיות שקיימות במערכת כרגע :
1. תנועה – כאשר מזיזים את הקופסה לתחנה האחרונה כלומר לפח האחרון, התזוזה בחלק האחרון של המסילה לא מבוצעת בצורה חלקה כפי שקורה בשאר המסילה
2. חיישן קול – קושי בלמדוד באופן מספק את הרעש שמגיע מהנקישה על הזכוכית ולהבדילו מהרעש שמגיע מנקישה על פלסטיק (ככל הנראה בגלל רעש חיצוני וכן רעש המנוע)
3. מבנה – המבנה אינו חזק מספיק
4. זיהוי התמונה בענן – המאגר שלנו לא גדול מספיק בשביל למקסם את יכולות המערכת ולכן יש צורך באגירת מספר גדול יותר של תמונות.