



עגלת קניות חכמה

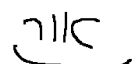
מוגש על ידי: אור בן נעים

תאריך הגשה 05/05/2024

מספר ת.ז.: 208019703

בהנחיית: ד"ר פיני זורע

חתימת המנחה: 

חתימת הסטודנט: 

הוגש לשם מילוי חלקי של הדרישות לקבלת התואר
"בוגר במדעים B.Sc. בהנדסת חשמל ואלקטרוניקה"

תקציר:

פרויקט זה בוצע במסגרת תכן הנדסי פנימי (פרויקט גמר) ב- "מכללה האקדמית להנדסה בראודה בכרמיאל".

מטרת הפרויקט היא לקצר את זמני ההמתנה בקופות התשלום בסופרמרקטים השונים ואף לצמצם את השימוש בקופאיות ו/או בעמדות תשלום עצמיות ובכך להשיג שיפור בחוויית הקנייה של הלקוח, חיסכון בזמן וחסכון כלכלי לבעלי רשתות הסופר למיניהן.

הפרויקט עוסק בתכנון ובנייה של אב-טיפוס למערכת עגלת קניות חכמה אשר תזהה את המוצרים שהלקוח הוסיף לעגלה באמצעות צילום ברקוד (ברקודים מסוג: UPC-A, EAN13) המופיע על גב המוצר ופענוח התמונה.

המערכת מסוגלת לזהות מוצרים ללא ברקוד, כגון: פירות וירקות, באמצעות משקל הממוקם בתחתית העגלה, אשר מעביר את המשקל של תכולת העגלה (משקל כלל המוצרים) בזמן אמת כנתון אל ה-Raspberry Pi.

הלקוח יתקשר עם המערכת באמצעות מסך מגע בגודל 7 אינץ', באמצעותו ניתן לבצע מגוון פעולות: הוספת מוצר, הורדת מוצר, תשלום ועוד. התשלום הלקוח יקבל Email עם חשבונית על הקנייה שביצע.

לאחר הוספת מוצר בהצלחה, תשלום בכרטיס אשראי ופעולות נוספות, המערכת תשמיע הודעה קולית באמצעות רמקול אשר יחובר אל ה-Raspberry Pi באמצעות Bluetooth.

המערכת מבוססת על בקר Raspberry Pi 4B המופעל ע"י מתח 5V.

בנוסף, העגלה החכמה תתקשר עם המחשב ראשי של מנהל הסופר באמצעות חיבור WIFI לרשת האינטרנט, ובכך מנהל הסופר יוכל לבצע עדכון מחירים למוצרים הקיימים בסופר בקובץ Excel שיש ברשותו ולשלוח אותו אל העגלה החכמה ע"י פרוטוקול תקשורת TCP.

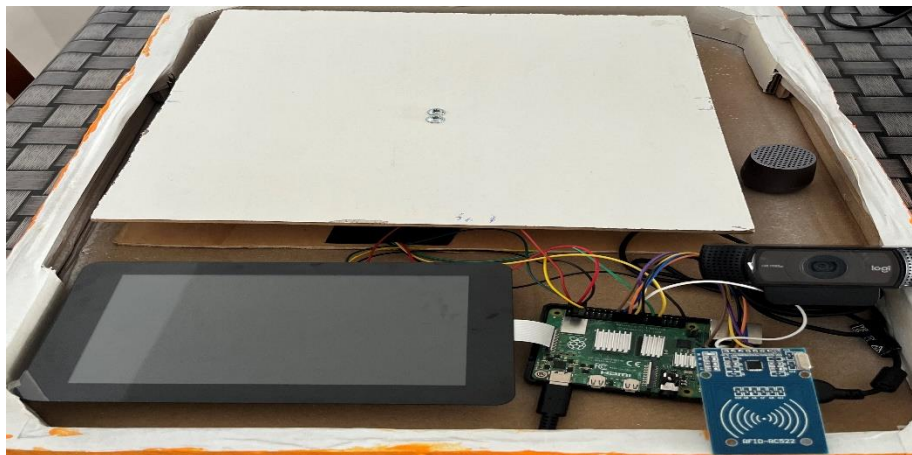
כמו-כן, על מנת לאפשר למנהל הסופר לעקוב אחר מלאי המוצרים שברשותו בזמן אמת,

על המחשב הראשי של מנהל הסופר תפעל תוכנית שיוצרת שרת (server),

העגלה החכמה תתחבר אל השרת של המחשב הראשי ותעביר לו את רשימת המוצרים שהלקוח קנה.

אותה תוכנית שפועלת על המחשב הראשי, תקבל את רשימת המוצרים ותעדכן את קובץ המלאי שנמצא במחשב של מנהל הסופר בהתאם למוצרים שהלקוח רכש.

הרעיון לפרויקט עלה מתוך צורך לייעל את תהליך הקנייה ולקצר את זמן ההמתנה בסופר.



איור 1: המחשת המערכת "עגלת קניות חכמה"

תוכן עניינים :

תקציר :

תוכן עניינים :

1. מבוא:.....1
2. תיאור המערכת:.....2
 - 2.1 מפרט פונקציונלי :.....2
 - 2.2 מפרט טכני :.....3
 - 2.2.1 טבלת פירוט מכלולי המערכת :.....3
 - 2.2.2 תרשים מלבני :.....4
 - 2.2.3 סכמה חשמלית וחלוקה למכלולים :.....5
 - 2.2.4 עקרון פעולת המערכת- העגלה :.....10
 - 2.2.4 עקרון פעולת המערכת- המחשב הראשי :.....11
3. מטלות:.....12
 - 3.1 מטלות הנדסיות:.....12
 - 3.2 שלבי התכנון:.....13
 - 3.3 תכנון אב- תרשימי זרימה של המערכות:.....16
 - 3.4 תכן מפורט- מערכת העגלה:.....18
 - 3.4.1 : תכן תוכנה :.....18
 - 3.5 תכן מפורט- מערכת המחשב הראשי:.....20
 - 3.5.1 : תכן תוכנה :.....20
 - 3.6 בעיות הנדסיות:.....21
 - 3.7 הדגמת פעולת המערכת:.....22
4. סיכום ודיון:.....24
 - 4.1 סיכום:.....24
 - 4.2 טבלת עמידה בדרישות:.....24
 - 4.3 הצעות לשיפור:.....25
5. ביבליוגרפיה:.....25
6. נספחים:.....26

רשימת האיורים :

- **איור 1:** המחשת המערכת "עגלת קניות חכמה"
- **איור 2:** תרשים מלבנים
- **איור 3:** סכמה חשמלית
- **איור 4:** חיבור המצלמה לבקר
- **איור 5:** חיבור מסך המגע לבקר
- **איור 6:** חיבור מודול RFID לבקר
- **איור 7:** חיבור חיישן משקל לבקר
- **איור 8:** תרשים זרימה של המערכות
- **איור 9:** תרשים זרימה מפורט של העגלה
- **איור 10:** מסך "הבית" בממשק ה-GUI
- **איור 11:** הוספת מוצר ללא ברקוד - חלק 1
- **איור 12:** הוספת מוצר ללא ברקוד - חלק 2
- **איור 13:** מסך "להצגת סל הקניות"
- **איור 14:** מסך "לסיום קנייה ותשלום" - חלק 1
- **איור 15:** מסך "לסיום קנייה ותשלום" - חלק 2

רשימת הטבלאות :

- **טבלה 1:** רשימת קיצורים
- **טבלה 2:** תרשים מלבנים
- **טבלה 3:** חיבור הרכיבים לפי פינים ל-Raspberry Pi
- **טבלה 4:** עמידה בדרישות

רשימת קיצורים :

| שם מלא | קיצור |
|--------------------------------------|-----------|
| European Article Number | EAN |
| General-Purpose Input Output | GPIO |
| Frames Per Second | FPS |
| Graphical User Interface | GUI |
| High-Definition Multimedia Interface | HDMI |
| Mega Pixel | MP |
| Open-Source Computer Vision | OpenCV |
| Computer Vision | CV |
| Raspberry Pi | RPI |
| Serial Peripheral Interface | SPI |
| Transmission Control Protocol | TCP |
| Universal Product Code | UPC |
| Universal Serial Bus | USB |
| Wireless Fidelity | WIFI |
| BT | Bluetooth |

טבלה 1: רשימת קיצורים

1. מבוא :

זמן ההמתנה בסופרמרקטים השונים ברחבי הארץ עומד על כ- 17 דקות בממוצע על מנת להגיע לעמדת התשלום. [1]

משך זמן ההמתנה הארוך נובע ממספר בעיות: חוסר בכוח אדם, חוסר בעמדות תשלום וכדומה. בשנים האחרונות רשתות הסופרמרקטים השונים הציבו עמדות תשלום אוטומטיות שאכן קיצרו את זמני ההמתנה אך לא באופן משמעותי.

כיום, קיימות בשוק עגלות חכמות שאינן מצריכות לעבור דרך הקופאיות.

אולם, מערכות אלו מאלצות את הלקוח להמתין בעמדות שקילה שמניחים עליהן את העגלה כולה על מנת לוודא שהלקוח אינו רימה את המערכת והוסיף לעגלה מוצרים ללא סריקתם.

בנוסף, סריקת המוצר מתבצעת ע"י סורק בדמות אקדח, ללא כל ממשק בין הלקוח והמערכת עצמה.

בדרך זו, על הלקוח לזכור האם סרק כל מוצר ומוצר לפני שהגיע לעמדת השקילה, במידה ושכח עליו למצוא את המוצר שלא סרק ולסרוק אותו- מערכת מסורבלת שלעיתים לא משפרת את חווית הלקוח ואף מזיקה לחווייתו.

פרויקט זה עוסק בפיתוח מערכת הנקראת "עגלת קניות חכמה" אשר תאפשר ללקוח לא רק לאחסן את מוצריו בתוכה, אלא גם לשלם באמצעות כרטיס אשראי בכל רגע נתון באמצעות רכיב מתאים שיותקן על העגלה עצמה ולסיים את תהליך הקנייה באופן מידי ללא המתנה בעמדות התשלום ובכך לבטל לחלוטין את זמני ההמתנה, לשפר את חווית הקנייה של הלקוח ולחסוך לבעלי הסופרמרקטים את העסקת הקופאיות.

מכלולי המערכת כוללים:

- בקר Raspberry Pi 4B.
- מצלמת רשת (Webcam) המתחברת ל-RPI באמצעות חיבור USB.
- מסך מגע בגודל 7 אינץ' שבאמצעותו הלקוח יתקשר עם המערכת דרך ממשק GUI שיופיע על גבי המסך.
- חיישן שקילה מבוסס: HX711 + Load Cell.
- קורא RFID-RC522 אשר יזהה כרטיס אשראי.
- רמקול BT אשר ישמיע הודעות קוליות לאחר שהלקוח ביצע פעולות מסוימות.

2. תיאור המערכת :

2.1 מפרט פונקציונלי :

- פיתוח המערכת נעשה על Raspberry Pi 4B (Linux OS), התוכנית שמפעילה את המערכת נכתבה בשפת Python
- הלקוח מזין את כתובת המייל דרך ממשק GUI שיוצג על גבי המסך
- זיהוי מוצרים בעלי ברקוד באמצעות מצלמה
- מדידת המשקל המצוי בתוך העגלה ע"י מערכת המשקל-Weight Scale
- הוספת מוצרים ללא ברקוד, כגון : פירות וירקות
- זיהוי הוספת מוצר לתוך העגלה ללא דיווח מקדים בממשק ה-GUI
- הצגת כל המוצרים הקיימים בעגלה ברשימה על גבי המסך
- הגדלת/הקטנת כמות של מוצר שכבר קיים ברשימת המוצרים ללא סריקה נוספת של אותו מוצר
- מחיקת מוצר מן העגלה באופן מוחלט (במצב של הסרתו מן העגלה)
- חישוב המחיר הכולל של הקנייה והצגתו על גבי מסך המגע בכל רגע נתון
- תשלום באמצעות כרטיס אשראי בעזרת קורא מגנטי (RFID-RC522)
- שליחת קבלה למייל הלקוח
- השמעת הודעות קוליות באמצעות רמקול BT לאחר שהלקוח ביצע פעולות מסוימות
- שחזור הנתונים (כתובת מייל הלקוח, רשימת המוצרים שהלקוח הוסיף לעגלה) במקרה שממשק ה-GUI מפסיק לפעול באופן פתאומי בזמן שהלקוח משתמש בעגלה.
- שליחת קובץ Excel של רשימת המוצרים בסופר עם מספרי הברקוד שלהם אל העגלה החכמה באמצעות פרוטוקול SSH
- שליחת רשימת המוצרים שהלקוח רכש מהעגלה החכמה אל המחשב הראשי ועדכון מלאי המוצרים הקיימים בסופר בזמן אמת

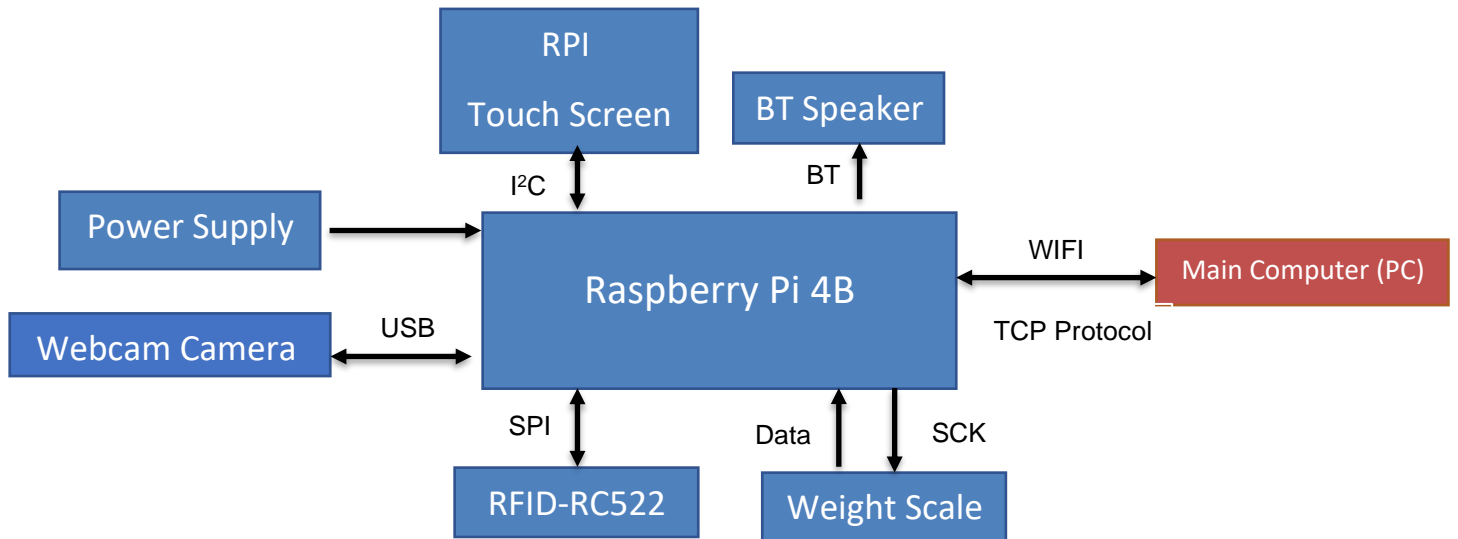
2.2 מפרט טכני :

2.2.1 טבלת פירוט מכלולי המערכת :

| מכלולי המערכת | תפקיד | פרמטרים / דרישות |
|---------------|----------------------------------|---|
| 1 | Raspberry Pi 4B | <ul style="list-style-type: none"> • שליטה ובקרה על כל המערכת. • מתח עבודה : 5V זרם : 3A • מעבד : ARM v8 64-bit • שעון : 1.5 GHz • Standard 40-pin GPIO header • אספקת מתחים 5V ו-3.3V • Micro SD card slot |
| 2 | מצלמת רשת Webcam Camera | <ul style="list-style-type: none"> • צילום הברקוד והעברת המידע לבקר • המצלמה נפתחת ונסגרת לצילום Streaming בהתאם להחלטת הבקר. • רזולוציה : 1920x1080p (2.1 MP) • פרוטוקול תקשורת : USB • Frame Rate : 30 fps |
| 3 | מקור מתח | <ul style="list-style-type: none"> • אספקת מקור מתח לבקר. • אספקת 5V וזרם 3A |
| 4 | RPI Touch Screen | <ul style="list-style-type: none"> • מאפשר תקשורת בין הלקוח והמערכת ע"י ממשק GUI קל לתפעול. • רזולוציה : 800x480p • גודל מסך : 7 Inch • DSI connector |
| 5 | רכיב למדידת משקל Weight Scale | <ul style="list-style-type: none"> • שקילת מוצרים המתומחרים לפי ק"ג ושקילת תכולת העגלה בזמן אמת. • הרכיב בנוי משני תתי מודולים : Load Cell, HX711 • שמחוברים אחד אל השני. • מתח עבודה : 5V • מדידת משקלים בטווח : 0-5 Kg |
| 6 | קורא כרטיסים מגנטי RFID-RC522 | <ul style="list-style-type: none"> • מאפשר תשלום באמצעות כרטיס אשראי ע"י זיהוי כרטיס אשראי ומספר הכרטיס. • מתח עבודה : 3.3V • פרוטוקול תקשורת : SPI |
| 7 | רמקול BT | <ul style="list-style-type: none"> • הודעות קוליות יושמעו באמצעות רמקול שיחובר אל ה-RPI ע"י BT. • פרוטוקול תקשורת : Bluetooth |
| 8 | מחשב ראשי PC | <ul style="list-style-type: none"> • שליחת קובץ אקסל של כל המוצרים הקיימים בסופר מה-PC אל ה-RPI באמצעות פרוטוקול SSH, הקובץ מכיל מספר ברקוד של כל מוצר ואת מחירו. • קבלת רשימת המוצרים שהלקוח רכש בתום הקנייה ועדכון מלאי המוצרים בסופר • על המחשב הראשי ועל ה-RPI להיות מחוברים לאותה רשת אינטרנט. • כתובת ה-IP של העגלה החכמה צריכה להיות ידועה עבור המחשב הראשי ולהפך. |

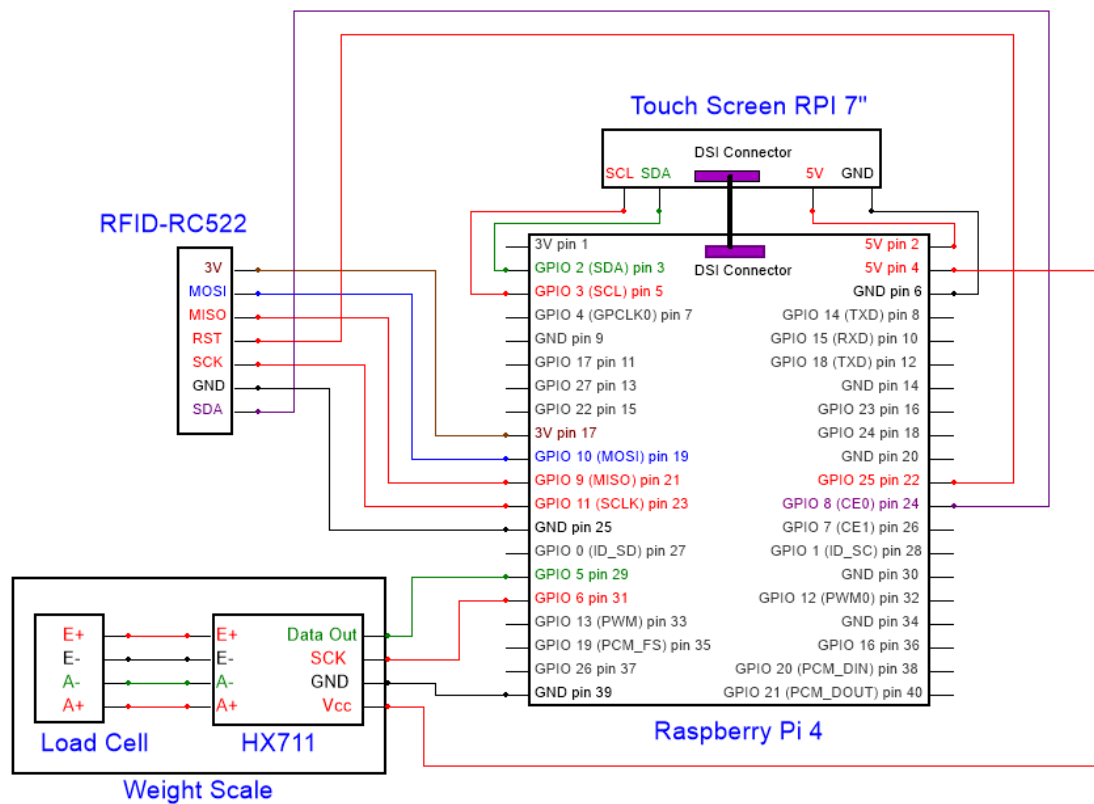
טבלה 2: תרשים מלבנים

2.2.2 תרשים מלבני:



איור 2: תרשים מלבני

2.2.3 סכמה חשמלית וחלוקה למכלולים :



איור 3: סכמה חשמלית

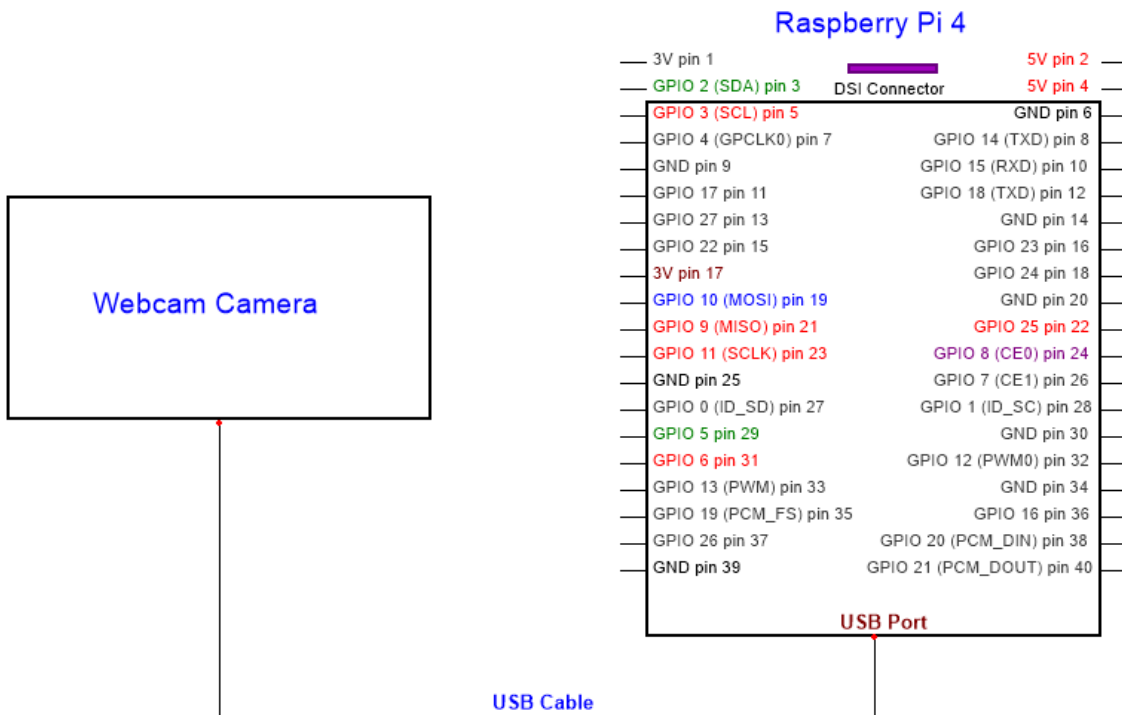
| PIN Number | Name of the PIN | In / Out | # |
|------------|---------------------------------|----------|----|
| 1 | 3.3V DC Power | Out | 1 |
| 2 | 5V DC Power | Out | 2 |
| 3 | GPIO 2 (SDA1, I ² C) | In | 3 |
| 4 | 5V DC Power | Out | 4 |
| 5 | GPIO 2 (SCL1, I ² C) | Out | 5 |
| 6 | Ground | Out | 6 |
| 17 | 3.3V DC Power | Out | 7 |
| 19 | GPIO 10 (MOSI, SPI0) | Out | 8 |
| 21 | GPIO 9 (MISO, SPI0) | In | 9 |
| 22 | GPIO 25 | Out | 10 |
| 23 | GPIO 11 (SCLK, SPI0) | Out | 11 |
| 24 | GPIO 8 (CE0 N, SPI0) | In | 12 |
| 25 | Ground | Out | 13 |
| 29 | GPIO5 | In | 14 |
| 31 | GPIO6 | Out | 15 |
| 39 | Ground | Out | 16 |

טבלה 3: טבלת חיבור הרכיבים לפי פינים ל- Raspberry Pi

• צילום ברקוד ופענוח התמונה :

מכלול זה בנוי מבקר Raspberry Pi 4 ומצלמת רשת. המצלמה מתחברת אל הבקר באמצעות חיבור USB, המצלמה עובדת ברזולוציה של 1080p.

בתהליך הזיהוי, הבקר שולח פקודה למצלמה ולהתחיל לצלם את הברקוד. הבקר מנתח את הברקוד מתוך התמונה שצולמה ומחליט האם המוצר הנוכחי הינו חלק ממוצרי הסופר או לא, במידה וכן המוצר מתווסף לרשימת המוצרים שהלקוח רכש ומחיר הקנייה הכולל מתעדכן.



איור 4: חיבור המצלמה לבקר

- מכלול מסך מגע:

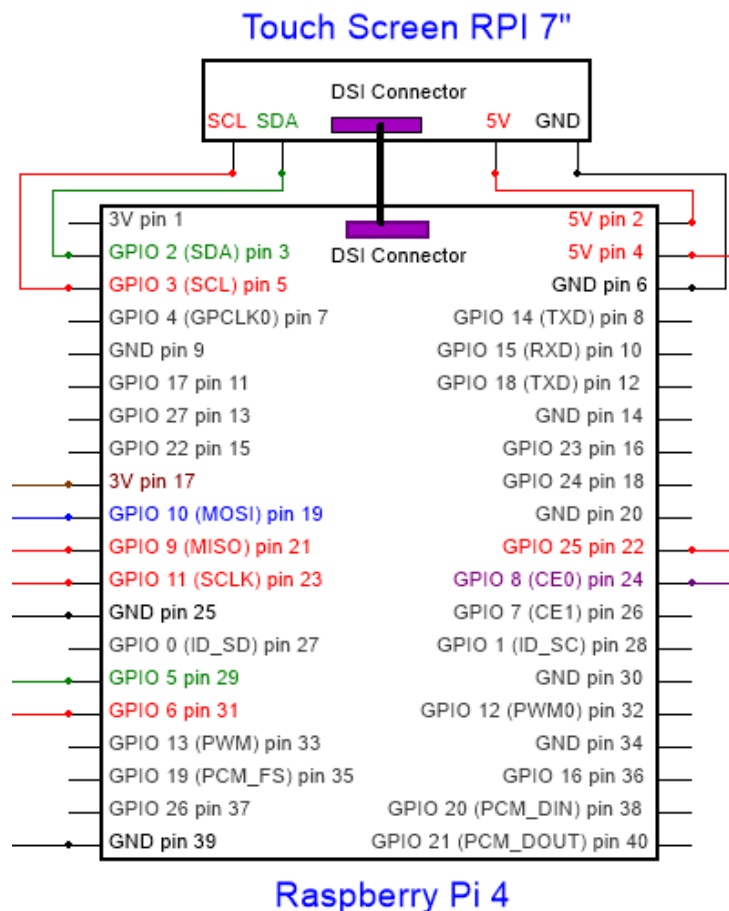
הלקוח מבצע פעולות בעגלה החכמה באמצעות מסך המגע וממשק GUI ייעודי שנכתב עבור הפרויקט.

המסך בעל רזולוציה של 800x480p ובגודל 7 אינץ'.

על גבי לוח הבקר ישנו חיבור מסך מסוג DSI, אליו מחברים קצה אחד של כבל המסך ואת הצד השני מחברים לחיבור המקביל בחלקו האחורי של המסך.

בין המסך לבקר יש שימוש בפרוטוקול תקשורת I2C.

I2C הוא אפיק תקשורת טורי סינכרוני, המאפשר למספר רכיבים לתקשר זה עם זה באמצעות שני חוטים בלבד: קו נתונים טורי (SDA) וקו שעון טורי (SCL).



איור 5: חיבור מסך המגע לבקר

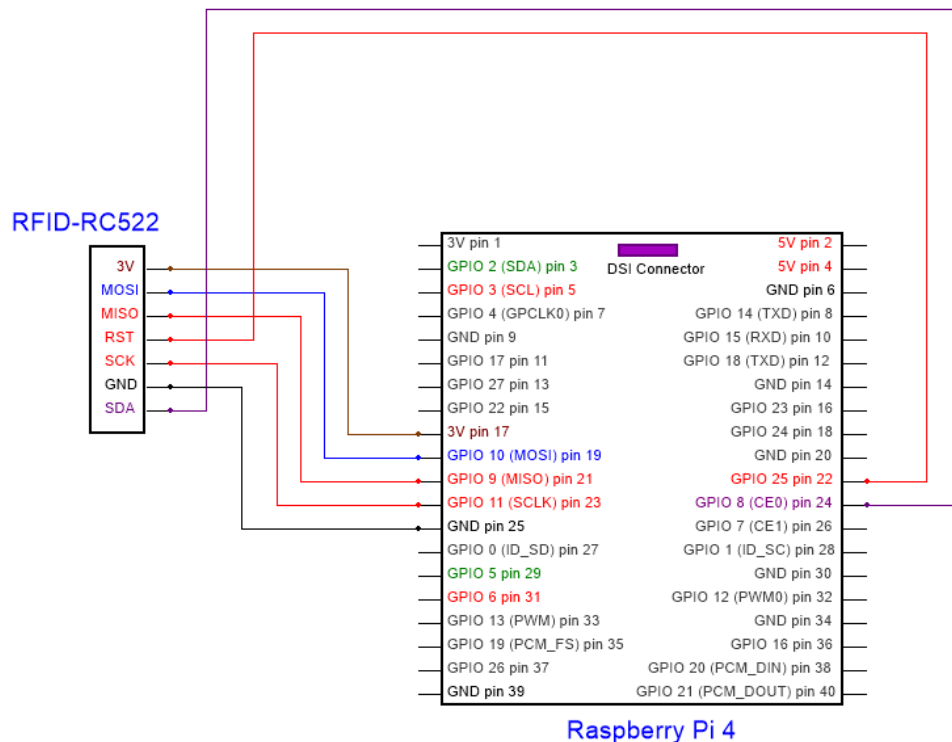
• מכלול זיהוי כרטיס אשראי:

מכלול זה מורכב מרכיב RFID-RC522 [5], רכיב זה מתוכנן ליצור שדה אלקטרו-מגנטי בתדר של 13.56 [MHz] וזהו תדר העבודה של הרכיב.

מודול קורא RFID או משדר משתמש בגלים אלקטרומגנטיים בתדר רדיו כדי להעביר נתונים. יחידת הבקרה וסליל האנטנה של מודול הקורא מייצרים שדה אלקטרומגנטי בתדר גבוה.

כאשר תג RFID או משדר מגיע בטווח השדה האלקטרומגנטי (טווח זיהוי) של מודול קורא RFID (משדר), עקב אינדוקציה הדדית, נוצר מתח בסליל האנטנה של התג, ומתח זה פועל כספק כוח לשבב.

כעת, התג מתחיל לשדר נתונים באופן סדרתי והקורא קורא את מידע התג.



איור 6: חיבור RFID לבקר

• מכלול שקילת מוצרים :

מכלול זה הינו חיישן שקילה שמורכב משני המודולים : Load Cell ,HX711.

: Load Cell

- Load Cell או בשמו המלא Strain Gauge Load Cell [6], הוא סוג של מתמר כוח אשר ממיר כוח/לחץ שמופעל על הרכיב לסיגנל חשמלי (מתח).

- Strain Gauge הינו חיישן שההתנגדות החשמלית שלו משתנה בהתאם לכוח המופעל עליו, הוא בנוי מחוט מוליך דק מאוד אשר מחובר בצורת 'זיגזג' בתצורה של קווים מקבילים.

כאשר החיישן נמתח, שטח הפנים שלו קטן ולכן ההתנגדות שלו גדלה לפי נוסחה :

$$R = \rho * \frac{L}{A}$$

כאשר החיישן נדחס, שטח הפנים שלו גדל וכתוצאה מכך ההתנגדותו קטנה.

Load Cell הינו פרופיל מתכת שעליו מורכבים ארבעה חיישני Strain Gauge כאשר הם

מחוברים בקונפיגורציה של Wheatstone Bridge.

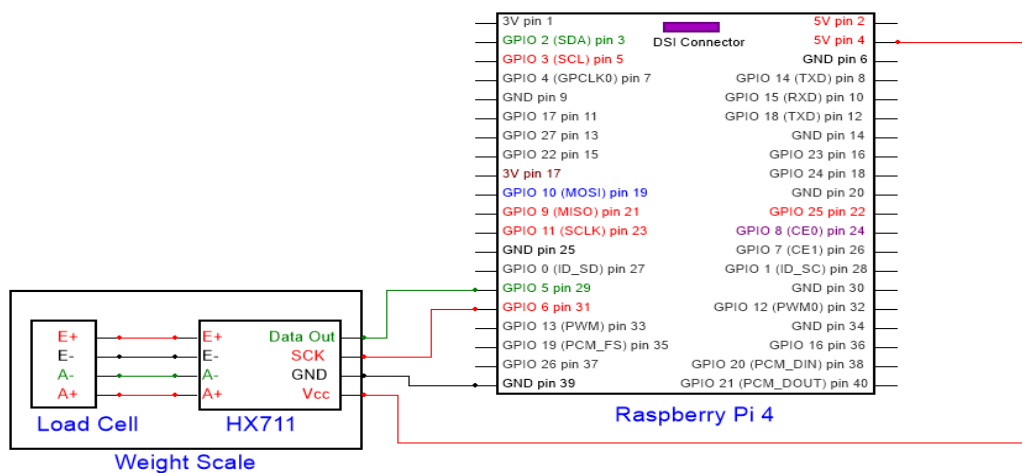
:HX711

- רכיב Load Cell מפיק אותות חשמליים קטנים מאוד, כלומר אותות חשמליים עם שינויי מתח מזעריים.

בנוסף, על מנת שהבקר יוכל לקבל את המידע שמפיק רכיב ה- Load Cell, יש לבצע המרה של האות האנלוגי לאות דיגיטלי איתו הבקר יכול לעבוד.

רכיב HX711 [7] מגביר את האות החשמלי על מנת שנוכל להבחין בשינויים קטנים במתח ובנוסף הוא ממיר את האות החשמלי לאות דיגיטלי כאשר רכיב זה מכיל A/D בעל 24 סיביות מה שמאפשר לייצג את השינויים המזעריים במתח ע"י 2²⁴ ערכים דיגיטליים שונים.

רזולוציה גבוהה זו מאפשרת מדידה מדויקת של שינויים קטנים במתח המוצא של רכיב Load Cell.



איור 7: חיבור חיישן משקל לבקר

2.2.4 עקרון פעולת המערכת - העגלה:

- המשתמש יתקשר עם המערכת באמצעות ממשק GUI שיוצג על גבי מסך המגע. המשתמש יוכל לבצע את הפעולות הבאות (ע"י לחיצה על כפתור מתאים):
 - התחלת קנייה חדשה
 - רישום למערכת באמצעות כתובת מייל
 - הוספת מוצר עם ברקוד
 - הוספת מוצר ללא ברקוד כגון פירות וירקות
 - ביטול הקנייה
 - הצגת סל הקניות
 - סיום קנייה ותשלום באמצעות כרטיס אשראי
 - השמעת הודעות קוליות באמצעות רמקול BT
- העגלה החכמה תקבל קובץ Excel מהמחשב הראשי שיכיל את המוצרים הקיימים בסופר ואת מספרי הברקוד שלהם.
- שליחת רשימת המוצרים שהלקוח רכש אל המחשב הראשי באמצעות שרת (server) הפועל על המחשב הראשי.
- **על המשתמש לנהוג באופן הבא:**
 - כל הוספת מוצר חדש לעגלה, הסרת מוצר מהעגלה, הגדלת/הקטנת כמות של מוצר קיים בתוך העגלה, תעשה כך שהמשתמש ידווח תחילה למערכת על רצונו (באמצעות לחיצה על כפתור מתאים) ורק לאחר מכן יבצע את הפעולה באופן פיזי.
- **במידה והלקוח הכניס אל העגלה מוצר שלא דווח קודם לכן במערכת**, יופיע מסך המתריע על ניסיון גניבה ולא יהיה ניתן להתקדם בממשק ה-GUI עד שהמשתמש יסיר את המוצר שהוסיף לעגלה ללא דיווח מקדים.
- לכל עגלה חכמה, יהיה קובץ Excel אשר ישמר באופן מקומי על ה-RPI של אותה עגלה, ובקובץ זה יופיעו אוסף כל המוצרים הקיימים בסופר עם מספר הברקוד של כל מוצר, שם המוצר, מחיר המוצר ומשקל פר יחידה אחת.
- לכן, כדי שלכל עגלה יהיה קובץ מחירים מעודכן במידה ובעל הסופר מחליט לשנות מחיר של מוצר מסוים, על כל עגלה חכמה להיות מחוברת לרשת ה-WIFI בסופר, כדי שקובץ המחירים המעודכן יישלח אל ה-RPI של כל עגלה ועגלה בזמן אמת. הקובץ יישלח ע"י חיבור לרשת ה-WIFI ופרוטוקול תקשורת SSH.
- מיד לאחר כל פעולה שהלקוח מבצע (רישום כתובת המייל, הוספת/מחיקת מוצר וכו'), המערכת שומרת את הנתונים בקובץ JSON בתוך ה-RPI של אותה עגלה חכמה.
- בכל פעם שהמערכת פועלת, לפני שהיא מפעילה את ממשק ה-GUI, היא בודקת האם ממשק ה-GUI "נפל" בפעם האחרונה שעבד או שנסגר בצורה תקינה.
- לדוגמא, המערכת נותקת מאספקת החשמל באופן פתאומי בזמן שהמשתמש היה בתהליך קנייה פעיל (הזין את כתובת המייל שלו, דיווח למערכת על מספר מוצרים שהכניס לעגלה וכדומה).

המערכת תבדוק האם הממשק "נפל" או שנסגר בצורה תקינה.
 במידה והממשק אכן "קריס", המערכת תשחזר את כל נתוני הקנייה ותחזיר את הלקוח
 לאותו המצב בדיוק טרם הכשל בממשק ה-GUI. כך, הלקוח אינו נדרש להוסיף את כל
 מוצריו מחדש עקב תקלה בלתי צפויה.

2.2.4 עקרון פעולת המערכת- המחשב הראשי :

- המחשב הראשי הינו מחשב PC שבו קיים קובץ Excel המחזיק את כל המוצרים הקיימים
 בסופר, מחשב זה מיועד לתפעול ע"י בעל הסופר.
 לכל מוצר קיימת עמודה עם מספר הברקוד שלו (במידה ומדובר במוצר עם ברקוד), שם
 המוצר, מחיר המוצר פר יחידה או פר ק"ג (במידה ומדובר במוצר ללא ברקוד) ומשקל של
 יחידה אחת (רלוונטי עבור מוצרים עם ברקוד בלבד).
 בנוסף, במחשב זה קיים קובץ Python שמחזיק את כתובות ה-IP של כל RPI בכל עגלה
 ועגלה אשר מחוברים לרשת ה-WIFI בסופר.
 במידה ובעל הסופר מעוניין לשנות מחיר של מוצר קיים בסופר, לשנות את שם המוצר או
 למחוק מוצר בכדי שלא יהיה יותר חלק מאוסף המוצרים הקיימים בסופר, כל מה שעליו
 לעשות זה לבצע את השינויים הנדרשים בקובץ Excel ולאחר מכן להפעיל את קובץ ה-
 Python ללא צורך בהבנה של הקוד.
 באופן פעולה זה, ע"י לחיצת כפתור אחת, כל העגלות החכמות בסופר יקבלו את קובץ
 המחירים המעודכן בבת אחת במקום לעדכן את קובץ המחירים באופן ידני בכל עגלה ועגלה
 בנפרד.
- בנוסף, במחשב הראשי קיים קובץ Excel נוסף, המכיל בתוכו את המלאי של המוצרים
 הקיימים בסופר.
 לאחר שהלקוח מבצע תשלום בעגלה החכמה, המחשב הראשי מקבל ממנה את רשימת
 המוצרים שהלקוח רכש ומעדכן את קובץ המלאי בהתאם לפירוט המוצרים שנרכשו.

3. מטלות:

3.1 מטלות הנדסיות:

- כתיבת הצעת פרויקט
- למידת התחום Computer Vision (CV)
- כתיבת פונקציה המצלמת ומפענחת את מספר הברקוד מתוך תמונה
- למידת הרכיב RFID-RC522 וכתיבת פונקציה אשר מפענחת את מספר כרטיס האשראי
- למידת אופן הפעולה של החיישן Strain Gauge
- למידת הרכיב HX711 ו- Load Cell
- חיבור הרכיבים : HX711, Load Cell ובניית מערכת המשקל- Weight Scale
- כיול חיישן המשקל- Weight Scale
- הכרת ממשק מסך המגע וחיבורו אל ה- RPI
- כתיבת ממשק GUI
- אינטגרציה של כלל המערכות
- בדיקת המערכת בתנאים שונים
- איתור תקלות ותיקון
- כתיבת ספר פרויקט + מצגת

3.2 שלבי התכנון:

- **בחירת סביבת הפיתוח:**
סביבת הפיתוח שנבחרה היא Python, הסיבה לכך היא שבשפת Python קיימות ספריות שימושיות בתחום של עיבוד תמונה ו- Computer Vision.
- **בחירת ממשק המשתמש:**
ממשק המשתמש נבנה ע"י ספריית Tkinter [3].
ספרייה זו מאפשרת בניית אפליקציה בצורה נוחה ופשוטה לצד פונקציונליות רבה שהספרייה מציעה.
- **Raspberry Pi:**
 - בקר זה מנהל את מערכת העגלה כולה, בין היתר אחראי על: עיבוד תמונה ופענוח הברקוד, יצירת GUI וניהול האפליקציה בזמן אמת.
 - ניהול הרכיבים השונים אשר מתחברים אל הבקר: מסך מגע, RFID-MFRC522, מערכת המשקל- (Load Cell + HX711) Weight Scale.
 - הבקר שולח למחשב הראשי את רשימת המוצרים שהלקוח רכש כדי שהמחשב הראשי יוכל לעדכן את מלאי המוצרים בסופר לאחר הקנייה.
 - בנוסף, המחשב הראשי שולח לבקר קובץ Excel שמכיל את רשימת המוצרים הקיימים בסופר ומספרי הברקוד שלהם.
- **בחירת סוג התקשורת:**
 - פרוטוקול התקשורת שנבחר בין מערכת העגלה והמחשב הראשי הוא TCP [4], הסיבות לבחירה בתקשורת זו הן:
 1. תקשורת בין מחשבים באמצעות שיטה זו, מבטיחה שכל המידע שנשלח ממחשב א' מגיע אל מחשב ב'.
 2. שיטת תקשורת זו מבוססת על מודל Client-Server בה יש 'לקוח' שמבקש גישה אל השרת שמחליט אם לאפשר את הגישה ואת התחלת התקשורת בניהם.
 3. פרוטוקול TCP מאפשר שיטות שיתוף מידע P2P כמו SSH - זהו פרוטוקול רשת שמאפשר גישה ממחשב אחד למחשב אחר בצורה בטוחה ואמינה על גבי רשת לא מאובטחת.
- **רכיב RFID-MFRC522:**
 - RFID-MFRC522 Reader [5] הוא התקן קורא/כותב המסוגל לקרוא/לכתוב נתונים מ/אל משדר RFID. הוא מורכב מ-3 רכיבי מפתח, אלה הם MFRC522 IC, מתנד קריסטל 27.12 מגה-הרץ ואנטנה, האנטנה גל אלקטרומגנטי בתדירות של 13.56 MHz.
 - בנוסף, קיים רכיב נלווה בשם RFID Card שמתפקד כרכיב פאסיבי, כלומר אין לו אספקת מתח משלו.
 - הוא מורכב ממיקרו-שבב וסליל (אנטנה). המיקרו-שבב מיועד לאחסון הנתונים, וסליל האנטנה מיועד להעברת הנתונים למודול ה- RFID Reader.
 - מודול קורא RFID או משדר משתמש בגלים אלקטרומגנטיים בתדר רדיו כדי להעביר נתונים. יחידת הבקרה וסליל האנטנה של מודול הקורא מייצרים שדה אלקטרומגנטי בתדר גבוה.

כאשר תג RFID או משדר מגיע בטווח השדה האלקטרומגנטי (טווח זיהוי) של מודול קורא RFID (משדר). עקב אינדוקציה הדדית, נוצר מתח בסליל האנטנה של התג, ומתח זה פועל כספק כוח לשבב. התג מתחיל לשדר נתונים באופן סדרתי והקורא קורא את מידע התג.

• מסך מגע 7 אינץ' :

מסך זה בעל רזולוציה 800x480p ובגודל 7 אינץ'. מאפשר למשתמש ללחוץ על המסך כדי לתקשר עם ממשק ה-GUI שמוצג על גבי המסך.

• רכיב המשקל - Weight Scale :

רכיב המשקל בנוי משני רכיבים אשר נקראים Load Cell ו-HX711.

: Load Cell

- Load Cell או בשמו המלא Strain Gauge Load Cell [6], הוא סוג של מתמר כוח אשר ממיר כוח/לחץ שמופעל על הרכיב לסיגנל חשמלי (מתח).

- Strain Gauge הינו חיישן שההתנגדות החשמלית שלו משתנה בהתאם לכוח המופעל עליו, הוא בנוי מחוט מוליך דק מאוד אשר מחובר בצורת 'זיגזג' בתצורה של קווים מקבילים.

כאשר החיישן נמתח, שטח הפנים שלו קטן ולכן ההתנגדות שלו גדלה לפי נוסחה :

$$R = \rho * \frac{L}{A}$$

כאשר החיישן נדחס, שטח הפנים שלו גדל וכתוצאה מכך התנגדותו קטנה.

- Load Cell הינו פרופיל מתכת שעליו מורכבים ארבעה חיישני Strain Gauge כאשר הם מחוברים בקונפיגורציה של Wheatstone Bridge.

:HX711

- רכיב Load Cell מפיק אותות חשמליים קטנים מאוד, כלומר אותות חשמליים עם שינויי מתח מזעריים. בנוסף, על מנת שהבקר יוכל לקבל את המידע שמפיק רכיב ה-Load Cell, יש לבצע המרה של האות האנלוגי לאות דיגיטלי איתו הבקר יכול לעבוד.

רכיב HX711 [7] מבצע הגברה של האות החשמלי על מנת שנוכל להבחין בשינויים קטנים במתח ובנוסף הוא ממיר את האות החשמלי לאות דיגיטלי כאשר רכיב זה מכיל A/D בעל 24 סיביות מה שמאפשר לייצג את השינויים המזעריים במתח ע"י 2²⁴ ערכים דיגיטליים שונים.

רזולוציה גבוהה זו מאפשרת מדידה מדויקת של שינויים קטנים במתח המוצא של רכיב Load Cell.

רמקול BT:

לבקר Raspberry Pi קיים ממשק BT מובנה המאפשר חיבור אלחוטי לרכיבים תומכים ברדיוס של עד כ- 10 מטרים.

תפקידו של הרמקול הוא להשמיע הודעות קוליות לאחר שהלקוח מבצע פעולות שונות בממשק ה-GUI.

• מחשב ראשי:

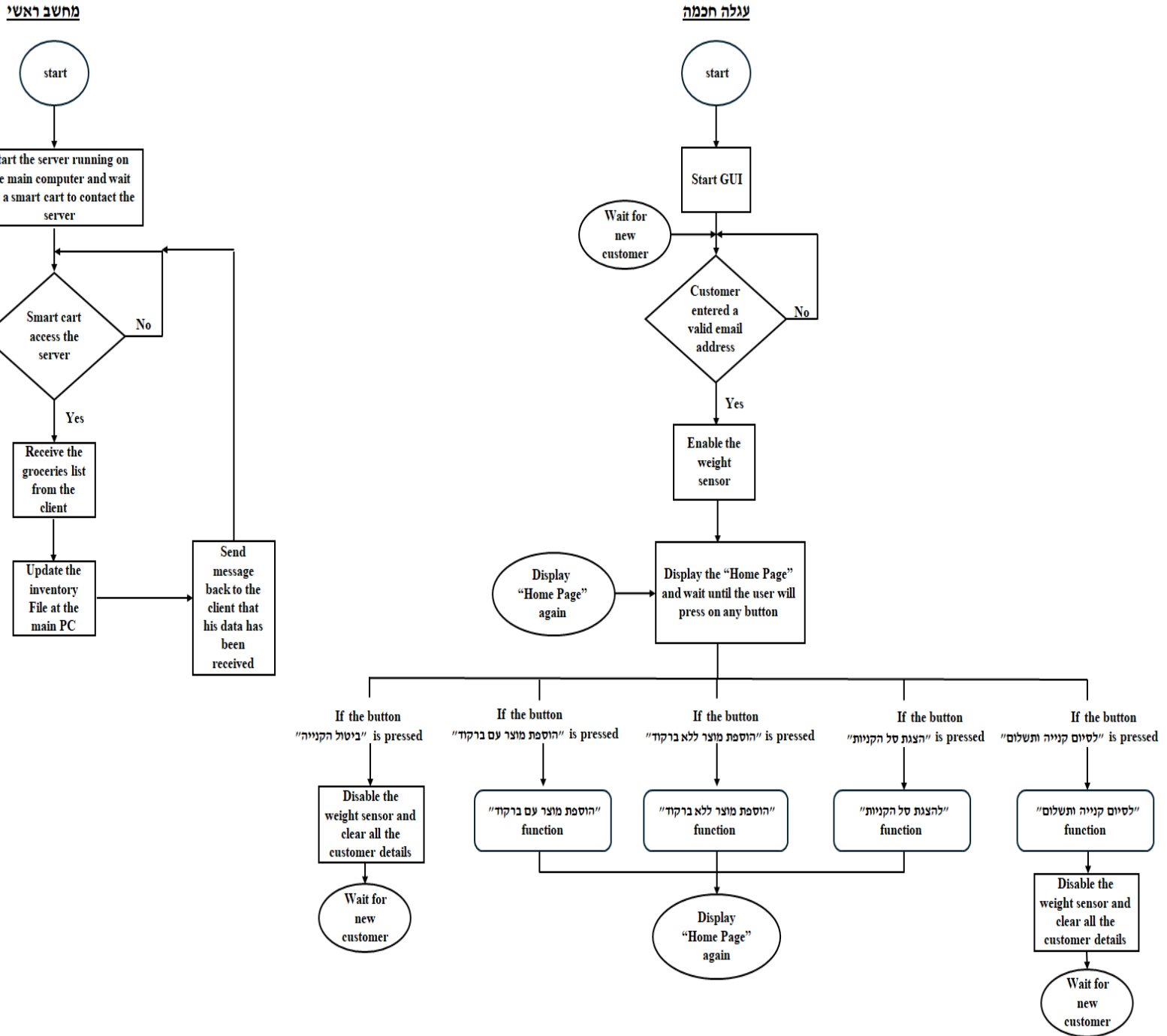
המחשב הראשי הינו מחשב PC שמשמש בפרויקט זה כמחשב של מנהל הסופר. במחשב קיימים שני קבצי Excel: "Groceries_Info.xlsx", הוא קובץ שמכיל את כל המוצרים שניתנים לרכישה בסופר ומספרי הברקוד שלהם, שמות המוצרים, מחיר לפי יחידה או לפי ק"ג בהתאם לסוג המוצר.

בכל פעם שמנהל הסופר רוצה לבצע שינוי כלשהו בקובץ זה, כגון: עדכון מחיר המוצרים, הוספת מוצר חדש, הורדת מוצר קיים,

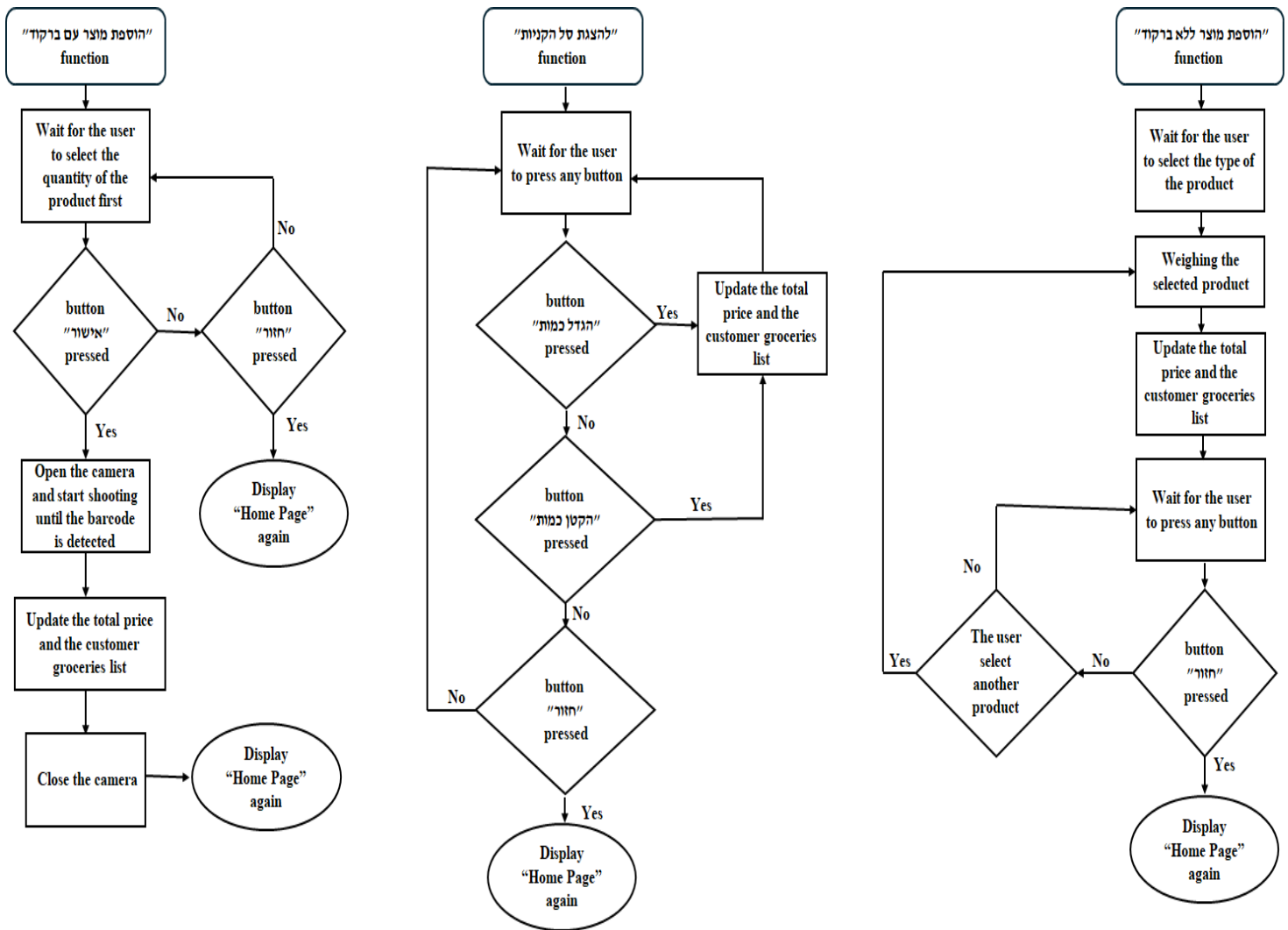
עליו להפעיל תוכנית פייטון (אין צורך בידע בתכנות אלא רק הפעלת הקובץ) ששולח לכל עגלה חכמה בסופר את קובץ המחירים המעודכן באמצעות חיבור WIFI לרשת האינטרנט ושימוש בפרוטוקול תקשורת SSH שמבוסס על TCP. בקובץ פייטון שמורים מספרי ה-IP של כל בקר ובקר שנמצא בכל עגלה ועגלה.

הקובץ השני "Inventory_Groceries.xlsx", הוא קובץ מלאי של המוצרים בסופר. כאשר לקוח ביצע בהצלחה תשלום בעגלה החכמה, העגלה שולחת למחשב הראשי את רשימת הקניות של הלקוח באמצעות פרוטוקול תקשורת TCP. כדי שהעגלה תוכל לעזות זאת, במחשב הראשי הוקם שרת שמופעל ע"י תוכנית פייטון אחרת (גם כאן אין צורך בידע בתכנות אלא רק להפעיל את הקובץ) ותוכנית זאת צריכה לעבוד כל הזמן ללא הפסקה בשעות הפעילות של הסופר על מנת לאפשר לעגלה חכמה לשלוח מידע אודות הקנייה של הלקוח.

3.3 תכנון אב- תרשימי זרימה של המערכות:



איור 8: תרשימי זרימה של המערכות



איור 9: תרשים זרימה מפורט של העגלה

3.4 תכן מפורט- מערכת העגלה:

3.4.1 : תכן תוכנה :

- שלב א' – הפעלת המצלמה וזיהוי מספר ברקוד ע"י פקודה מהבקר :

בשפת Python קיימת ספרייה בשם cv2 (Open CV) [9] שבה יש מתודה העונה לשם: `cv2.VideoCapture()`, בעזרתה ניתן להגדיר את רזולוציית הצילום הרצויה (במקרה שלנו מספיקה רזולוציה של 640x360 פיקסלים כדי לזהות את הברקוד. באמצעות מתודה זו, נפתח על המסך חלון ובו רואים את מה שהמצלמה מצלמת.

קריאת כל frame מהמצלמה נעשית ע"י המתודה: `cv2.VideoCapture().read()` בשיטה זו, ניתן לקרוא את כל frames שהמצלמה מצלמת.

כדי לפענח את הברקוד מתוך קריאת ה-frames, יש לייבא את המתודה `decode` ע"י הפקודה: `from pyzbar.pyzbar import decode` [10].

בעזרת המתודה `decode` ניתן לחלץ את מספר הברקוד ולבדוק האם המספר שהתקבל תואם לאחד ממוצרי הסופר הקיימים.

אם הברקוד הנקלט תואם לאחד מהמוצרים הקיימים בסופר, הבקר נותן פקודה לסגור את המצלמה.

- שלב ב' – שקילת מוצר :

שקילת המוצר מתבצעת ע"י הרכיבים HX711 + Load Cell. ראשית יש לכייל את חיישן המשקל, ה-Load Cell שנבחר לצורך הדגמת פעולת המערכת הוא כזה שיכול למדוד מסה מקסימלית של 5 ק"ג. לכן לטובת כיול המערכת, בוצע כיול ראשוני עם מוצר שמסתו 1 ק"ג. כדי לעבוד עם רכיב HX711 יש להשתמש בספריה המתאימה ע"י הפקודה: `from hx711 import HX711` [11].

אתחול חיישן המשקל מתבצע פעם אחת בלבד כאשר התוכנית שמפעילה את מערכת העגלה מופעלת בפעם הראשונה ולצורך כך נבנתה פונקציה: `init_weight_scale()`

התחלת מדידה נעשית ע"י הפקודה:

`hx.get_weight_mean(readings=15) / 1000`

הפרמטר `readings = 15` מגדיר את מספר הקריאות שהחיישן יבצע, ככל שהערך המועבר לפרמטר זה יהיה גדול יותר כך המדידה תהיה מדויקת יותר אך מנגד משך זמן המדידה יהיה ארוך יותר.

ע"י בדיקות נמצא כי הערך `readings = 15` מספק תוצאות טובות. הערך שחוזר מהפקודה הנ"ל הוא ביחידות מידה של גרם ולכן יש לחלק ב 1000 על מנת לקבל ערכים בק"ג.

- שלב ג' – קריאת כרטיסי אשראי

לצורך קריאת כרטיס חכם באמצעות רכיב RFID-RC522 יש לעבוד עם הספרייה :
[12] from mfrc522 import SimpleMFRC522

הפונקציה שנבנתה משתמשת במתודה SimpleMFRC522().read() אשר מחזירה את המספר הסריאלי שניתן מראש לכרטיס החכם.

- בשלבים ב' ו- ג' בוצע שימוש בספריית GPIO של Raspberry Pi ע"י הפקודה :
import RPi.GPIO as GPIO

בכל פונקציה שבה יש צורך לעבוד עם סיגנלים חיצוניים (I / O) בוצע שימוש בספרייה זו.

- שלב ד' – יצירת GUI

ממשק המשתמש נבנה ע"י ספריית Tkinter [3].
 ספרייה זו מאפשרת בניית אפליקציה בצורה נוחה ופשוטה לצד פונקציונליות רבה שהספרייה מציעה.
 האפליקציה נבנתה בצורה כזו כך שכל "דף" שמוצג למשתמש הוא מחלקה בפני עצמו שירשת מהטיפוס tk.Frame.
 לכל דף (מחלקה שירשת מאובייקט tk.Frame) יש את השדות שלו עם הפונקציונליות הייחודית לאותו הדף.
 כאמור, שם המחלקה הוא אובייקט מסוג tk.Frame ולכן כדי להציג ברגע כלשהו דף מסוים יש להשתמש בפקודה :
 Name_of_Page.tkraise()

כדי לשפר ביצועים, הוחלט להשתמש בספרייה threading שמאפשרת לבצע 2 פקודות במקביל.

שתי הפעולות שנבחרו הן :
 (1) הפעלת ה-GUI וניהולו.
 (2) מדידת המשקל המצוי בעגלה בכל רגע ורגע.
 פעולה זו הינה פעולה שדורשת זמן חישוב בלתי מבוטל.

3.5 תכן מפורט- מערכת המחשב הראשי:

3.5.1 : תכן תוכנה :

על המחשב הראשי לבצע שתי פעולות עיקריות :

- (1) שליחת קובץ Excel של מוצרי הסופר עם מחיריהם המעודכנים.
- (2) הקמת שרת על גבי המחשב הראשי אשר תפקידו להמתין עד שעגלה חכמה פונה אל אותו שרת ושולחת לו את רשימת המוצרים שהלקוח רכש על מנת שיתבצע עדכון של מלאי המוצרים בסופר.

(1) פעולה מספר 1 בוצעה ע"י שימוש בפרוטוקול SSH שהוא אמצעי לתקשר בין שני מחשבים באופן מאובטח על גבי רשת האינטרנט שהיא אינה אמינה.

כדי לבצע זאת, בוצע שימוש בספרייה **Paramiko [13]** אשר מאפשרת חיבור למחשב/בקר מרוחק באמצעות SSH.

(2) השרת שהוקם הוא מסוג **Flask Web Server [14]**. שרת זה מאפשר ללקוח לגשת אליו באמצעות פרוטוקול HTTP ולנהל בקשות HTTP. לצורך ביצוע פעולה מספר 2 נדרש שימוש מסוג **POST Request**.

ראשית יש ליצור אובייקט של שרת מסוג Flask ע"י הפקודה :
`server = Flask(__name__)`, כדי שהשרת יתחיל לפעול יש להשתמש במתודה :
`server.run(host=Host_IP, port=Port_Number)`

נבנתה פונקציה ייעודית שמטרתה היא להמתין לעגלה חכמה שתשלח את רשימת המוצרים שהלקוח רכש באמצעות **POST Request**. הפונקציה מקבלת את הרשימה ופונה לקובץ ה- Excel של מלאי המוצרים בסופר ומעדכנת אותו בהתאם.

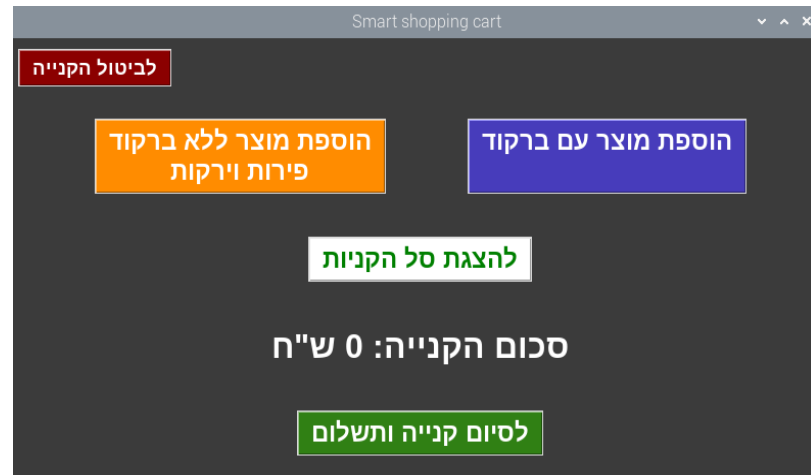
- על מנת לפתוח את קובץ ה- Excel ולערוך בו חיפוש של כל מוצר ומוצר ולבצע עדכון של כמות המלאי בעמודה הרלוונטית, נעשה שימוש בספרייה **openpyxl** אשר מאפשרת לעבוד עם קבצי Excel בצורה נוחה.

3.6 בעיות הנדסיות:

- חיישן המשקל חישב ערכים עם שגיאות מדידה גדולות ולכן נוצרו סטיות משמעותיות.
הפתרון עליו חשבתי הוא להגדיל את כמות הקריאות של החיישן כדי להגדיל את דיוק המדידה, עקב הגדלת הקריאות שהחיישן מבצע זמן החישוב עלה ומכיוון שהחישוב מתבצע בזמן אמת כתבתי את החישוב של חיישן המשקל בתור Thread כחלק מספריית Threading [16] ב-Python על מנת לייעל את החישוב של חיישן המשקל.
- העגלה החכמה מיועדת להיות מערכת ניידת ועל כן אינה יכולה לקבל אספקת חשמל ע"י מטען נייד שמחובר לרשת החשמל.
לכן הפתרון הוא להשתמש בסוללות שמספקות את אותם הנתונים של המטען של הבקר RPI שהם: זרם 3A ומתח 5V.
- תחילה השתמשתי בקבצי CSV כדי לאחסן מידע אודות המוצרים הקיימים בסופר, הרצון להשתמש בקבצים מסוג זה נבע מהיכרות מוקדמת עם סוג הקובץ במסגרת קורס אקדמי שנלמד במסגרת התואר הראשון ובשל הנוחות שבעבודה עם קובץ מסוג זה.
הבעיה שנוצרה היא שבקובץ מסוג זה ישנן קשיים בשמירת מספרים ארוכים מאוד כגון מספרי ברקוד מה שגורם לתאים בעמודה הרלוונטית של מספרי הברקוד להימחק.
הפתרון לבעיה היה מעבר לשימוש בקבצי Excel ולכן למדתי בעצמי כיצד יש לעבוד עם קבצים מסוג זה.
- לאחר שהמצלמה קראה מספר ברקוד, חלון המצלמה על גבי המסך נסגר מיד לאחר מכן והמצלמה מעבירה את הערך שקראה לבקר.
לעיתים המצלמה קראה בצורה שגויה את מספר הברקוד שעל גבי המוצר, כדי לפתור את הבעיה יצרתי מנגנון שבודק את תקינות הברקוד טרם סגירת חלון המצלמה, במידה והקריאה שבוצעה שגויה, חלון המצלמה נשאר פתוח והמצלמה ממשיכה לצלם כדי לאפשר קריאה תקינה של מספר הברקוד.

3.7 הדגמת פעולת המערכת:

- במידה והמשתמש לחץ על כפתור "הוספת מוצר עם ברקוד", המסך הבא שיופיע יהיה מסך לבחירת כמות יחידות רצויה מהמוצר שהמשתמש רוצה להוסיף לעגלה.
- לאחר שהלקוח בחר כמות רצויה, יופיע על גבי המסך חלון ובו רואים את מה שהמצלמה מצלמת בשידור חי, בדומה לשידור Streaming.
- על הלקוח לכוון את המוצר כך שהברקוד פונה אל המצלמה, לאחר צילום הברקוד בהצלחה מחיר הקנייה הנוכחי יתעדכן והמשתמש יחזור אל מסך "דף הבית".



איור 10: מסך "הבית" בממשק ה-GUI

- לאחר לחיצה על הכפתור "הוספת מוצר ללא ברקוד" פירות וירקות" ייפתח המסך שמוצג באיור 11. הלקוח יבחר את סוג המוצר שברצונו להוסיף ומיד לאחר מכן תופיע הודעה שמבקשת מהלקוח להניח את המוצר שבחר על גבי המשקל.
- לאחר שהלקוח מניח את המוצר על חיישן המשקל, יש ללחוץ על כפתור "סיום".
- יש להקפיד על סדר הפעולות הנ"ל.
- לאחר שהמשתמש לחץ על כפתור "סיום", המערכת מתחילה לשקול את המוצר ללא הברקוד שהתווסף לעגלה זה עתה, המערכת מחשבת את מחיר המוצר ע"י הנוסחה הבאה: מחיר לפי ק"ג x משקל המוצר שנמדד.
- בסיום השקילה מחיר הקנייה הכולל יתעדכן בהתאם.



איור 11: הוספת מוצר ללא ברקוד - חלק 1

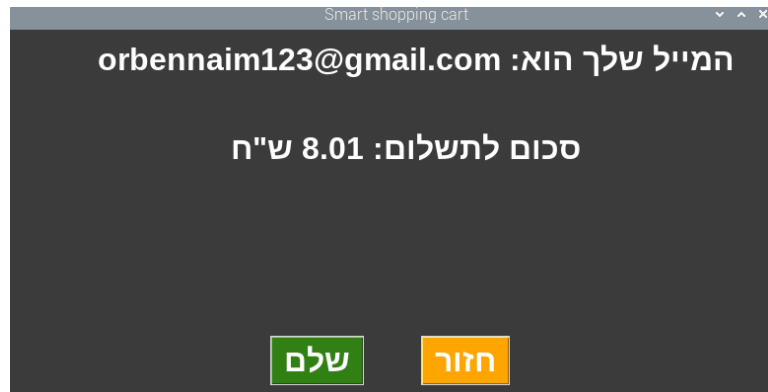


איור 12: הוספת מוצר ללא ברקוד - חלק 2

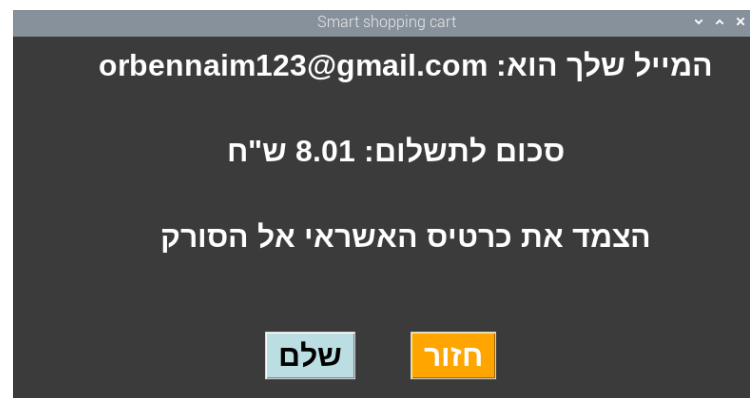
- במידה והמשתמש לחץ על כפתור "**להצגת סל הקניות**" במסך "הבית", יופיע מסך (איור 13) שמציג את כל המוצרים שהלקוח הוסיף לעגלה עד כה, כמות יחידות עבור כל מוצר ומחיר כולל של כל המוצר (מחיר מוצר בודד x כמות היחידות במקרה של מוצר עם ברקוד).
 - בנוסף, בחלון זה יופיעו כפתורים שיאפשרו ללקוח להגדיל / להקטין כמות של מוצר נבחר (בחירה מתבצעת ע"י לחיצה על השורה של המוצר הרצוי) מן הרשימה.
 - במקרה של מוצר עם ברקוד, כמות המוצר תגדל / תקטן ביחידה אחת.
- גם במידה ומדובר במוצר ללא ברקוד, הלקוח יכול להשתמש בכפתורים הללו.
- למשל, בהנחה וקיימים בעגלה כמות כלשהי של תפוחים והלקוח מעוניין להגדיל את כמות התפוחים, הלקוח ילחץ על כפתור "**הגדל כמות**" ולאחר לחיצת הכפתור המשתמש יוסיף את כמות התפוחים.
- לאחר ההוספה, הלקוח ימתין למערכת שתשקול את ה-ק"ג התפוחים שהתווסף ותעדכן את מחיר הקנייה הכולל.
- יש לפעול באופן דומה עם כפתור "**הקטן כמות**".
- במידה והמשתמש לחץ על כפתור "**לביטול הקנייה**" במסך "הבית", כתובת המייל שהלקוח הזין למערכת תמחק, רשימת כל המוצרים שהלקוח הכניס לעגלה תתאפס והאפליקציה תחזור למסך הפתיחה, מוכנה ללקוח חדש.
- במידה והמשתמש לחץ על כפתור "**לסיום הקנייה ותשלום**" במסך "הבית", יופיע מסך (איור 14) בו תוצג כתובת המייל של הלקוח ומחיר לתשלום.
- לאחר לחיצה על כפתור "**שלם**", תופיע על המסך הודעה שמנחה את המשתמש להצמיד את כרטיס האשראי אל הרכיב RFID-RC522 אשר קורא כרטיס מגנטי ומפענח את מספר הכרטיס.
- לאחר שכרטיס האשראי נסרק בהצלחה, תופיע הודעה שהתשלום בוצע בהצלחה וקבלה נשלחה לכתובת המייל שהלקוח הזין.



איור 13: מסך להצגת סל הקניות



איור 14: מסך "לסיום קנייה ותשלום" - חלק 1



איור 15: מסך "לסיום קנייה ותשלום" - חלק 2

4. סיכום ודיון:

4.1 סיכום:

בפרויקט המוצג בספר זה מתואר תהליך התכנון והפיתוח של מערכת עגלת קניות חכמה. מטרת ביצוע הפרויקט הוא ייעול תהליך הקנייה עבור לקוחות הסופר (ביטול זמני המתנה בעמדות התשלום למיניהן) וחיסכון כלכלי לבעלי הסופר (חיסכון בהעסקת קופאיות). הפרויקט גרם לי לחשוב בצורה מערכתית היות והמערכת מורכבת משתי תתי מערכות שמתקשרות ביניהם ומכיוון שהעגלה החכמה עצמה הינה מערכת שנדרשת לבצע פעולות בזמן אמת.

בפרויקט זה השתמשתי בשפת Python וברכיבים שונים, לצורך בנייתו חקרתי על כל רכיב ורכיב כאשר למדתי את אופן הפעולה שלהם ללא היכרות מוקדמת. כתוצאה מכך נדרשתי להפגין יכולות למידה עצמית, התמודדות עם איתור בעיות ופתרון. הידע שרכשתי בקורסים אקדמאים קודמים היווה בסיס לפרויקט זו ויחד עם זאת רוב הפרויקט דרש התמודדות עם בעיות חדשות שהובילו לרכישת כלים הנדסיים נוספים ורבים אותם רכשתי במסגרת פרויקט הגמר.

לסיכום,

אני סבור ובטוח שאעשה שימוש עתידי בכלים ובידע אותם רכשתי במהלך תכנון ופיתוח הפרויקט, הלוא הם כלים מעשיים ופרקטיים. לכן אני רואה בהתמודדות עם פרויקט מסוג זה אבן דרך משמעותית בהפיכתי למהנדס חשמל ואלקטרוניקה.

4.2 טבלת עמידה בדרישות:

| # | דרישות | ביצוע |
|----|---|-------|
| 1 | כתיבת הצעת פרויקט | בוצע |
| 2 | שיקולי תכנון | בוצע |
| 3 | למידה על זיהוי ברקוד ופיענוחו | בוצע |
| 4 | כתיבת אלגוריתם לזיהוי מספר ברקוד | בוצע |
| 5 | הכרת מסך המגע ב-RPI | בוצע |
| 6 | מימוש חיישן משקל | בוצע |
| 7 | כתיבת ממשק GUI למשתמש | בוצע |
| 8 | תכנון התקשורת בין המערכות ומימושה | בוצע |
| 9 | למידה על רכיב RFID | בוצע |
| 10 | תכנון מערכת השמעה ע"י רמקול BT | בוצע |
| 11 | אינטגרציה של כל המערכות | בוצע |
| 12 | בדיקת המערכת בתנאים שונים, איתור תקלות ותיקון | בוצע |
| 13 | כתיבת ספר פרויקט | בוצע |

טבלה 4: עמידה בדרישות

4.3 הצעות לשיפור:

- לאפשר למערכת לזהות היכן נמצאת העגלה בכל רגע נתון ובהתאם למיקומה הנוכחי בסופר להציע ללקוח מבצעים על המוצרים שקרובים אליו.
- רישום של לקוח חדש באמצעות מספר טלפון במקום כתובת המייל שלו כאשר לקוח חדש מתחיל קנייה חדשה בעגלה.
- זיהוי מוצרים באמצעות למידת מכונה במקום שימוש בזיהוי מספר ברקוד.

5. ביבליוגרפיה:

- [1] "מחכים לקופאית: כמה זמן תעמדו בתור בסופרמרקט?" - וואלה! כסף. (2016, April 19). <https://finance.walla.co.il/item/2954481>
- [2] (2020) "Mastering Correct Email Address Syntax for Delivery." captainverify.com/blog/email-addresses-format.html
- [3] (2019) Python Software Foundation. "Tkinter — Python Interface to Tcl/Tk — Python 3.7.2 Documentation." *Python.org*, 2019. docs.python.org/3/library/tkinter.html
- [4] (2021) Fortinet. "What Is TCP/IP and How Does It Work?" <https://www.fortinet.com/resources/cyberglossary/tcp-ip>
- [5] (2022, April 9) "What Is RFID RC522 Module | How Does It Work» ElectroDuino." <http://www.electroduino.com/what-is-rfid-rc522-module-how-does-it-work/>.
- [6] (2017) Trent, Dara. "How Does a Strain Gauge Load Cell Work? | Load Cell Central." <http://www.800loadcel.com/load-cell-and-strain-gauge-basics.html>.
- [7] (2018) "Load Cell Click | Mikroelektronika." *MIKROE*. <http://www.mikroe.com/load-cell-click>
- [8] (1996) Ylonen, Tatu. "SSH Protocol Is the Standard for Strong Authentication, Secure Connection, and Encrypted File Transfers. We Developed It." <http://www.ssh.com/academy/ssh/protocol>
- [9] (2024, March 4) OpenCV: OpenCV modules. (n.d.). Docs.opencv.org. <https://docs.opencv.org/4.x/index.html>

[10] Pyzbar [Online]. Available: <https://pypi.org/project/pyzbar>

[11] Roose, M. (n.d.). hx711: A library to drive a HX711 load cell amplifier on a Raspberry Pi. PyPI.

<https://pypi.org/project/hx711/>

[12] mfrc522-python: The mfrc522-python library is used to interact with RFID readers that use the MFRC522 chip interfaced with a Raspberry Pi. (n.d.). PyPI. Retrieved March 4, 2024, from <https://pypi.org/project/mfrc522-python/>

[13] Use Paramiko and Python to SSH into a Server. (n.d.). Linode Guides & Tutorials.

<https://www.linode.com/docs/guides/use-paramiko-python-to-ssh-into-a-server/>

[14] What is Flask Python - Python Tutorial. (2021). Pythonbasics.org.

<https://pythonbasics.org/what-is-flask-python/>

[15] AUTHORS, S. (n.d.). openpyxl: A Python library to read/write Excel 2010 xlsx/xlsm files. PyPI.

<https://pypi.org/project/openpyxl/>

[16] threading — Thread-based parallelism — Python 3.9.0 documentation. (n.d.).

Docs.python.org.

<https://docs.python.org/3/library/threading.html>

6. נספחים :

בקובץ ZIP מצורפים הקבצים הבאים :

א. הצעת הפרויקט בפורמט pdf.

ב. דפי נתונים של הרכיבים העיקריים בפרויקט.

ג. הסבר כיצד להפעיל את ממשק המשתמש (GUI).