



Software Engineering Department

Braude College

Final Project Phase A

CartGenie - A smartphone application for Smart Grocery Shopping

Project Code: 25-2-D-20

Founders:

Alex Segal

Nir Froimovich

alex.segal@e.braude.ac.il

nir.froimovich@e.braude.ac.il

Project Drive: <https://docs.google.com/document/d/1sq5ebCbRFEKJzeuBZh2HUJsYIHgoruB7oX5qDRYUujo/edit?usp=sharing>

GitHub : <https://github.com/CoderxX22/CartGenie.git>

Semester B Spring 2025

Contents

| | |
|--|-----------|
| Contents..... | 2 |
| 1. Abstract..... | 4 |
| 2. Introduction..... | 4 |
| 3. Literature review..... | 5 |
| 3.1 Grocery List Management Platforms..... | 6 |
| 3.2 Nutrition Monitoring and Dietary Advising Systems..... | 6 |
| 3.3 The Need for CartGenie in Daily Nutrition Management..... | 7 |
| 3.4 Optical Character Recognition in the Project: Using Tesseract OCR..... | 8 |
| 3.4.1 What is Tesseract OCR?..... | 9 |
| 3.4.2 How Tesseract Works..... | 9 |
| 3.4.3 Effectiveness and Limitations..... | 11 |
| 3.4.4 Relevance to Our Application..... | 12 |
| 3.4.5 Product Nutrition Table Scan..... | 13 |
| 3.4.5 Grocery Receipt Scan..... | 14 |
| 3.5 Disease and Health Condition Analysis: Using XGBoost..... | 15 |
| 3.5.1 What is XGBoost?..... | 15 |
| 3.5.2 How XGBoost Works?..... | 16 |
| 3.5.3 Relevance to Our Application..... | 17 |
| 3.5.4 Why XGBoost?..... | 17 |
| 3.6 Data Preprocessing and Feature Management: Using pandas..... | 18 |
| 3.6.1 What is pandas?..... | 18 |
| 3.6.2 How Pandas Works?..... | 19 |
| 3.6.3 Role in the Project Pipeline..... | 19 |
| 3.6.4 Benefits of Using pandas..... | 20 |
| 4. Project Goals and Expected Achievements..... | 21 |
| 4.1 Expected Features and Deliverables..... | 22 |
| 4.2 System Workflow..... | 22 |
| 4.3 Success Criteria..... | 23 |
| 5. Engineering Process..... | 24 |
| 5.1 Process – Development Approach and Methodology..... | 24 |
| 5.2 Development Methodology..... | 24 |
| 5.3 Engineering Stages..... | 25 |
| 5.3.1 Semester A – Research & Design..... | 25 |
| 5.3.2 Semester B – Development & Testing..... | 25 |

| | |
|--|----|
| 5.4 Constraints and Challenges..... | 26 |
| 5.5 Product – Algorithms, Models, and Interface..... | 26 |
| 5.5.1 Key Algorithms and Data Flow..... | 26 |
| 5.5.2 Planned Data Structures..... | 27 |
| 5.6 User Interface Features..... | 27 |
| 5.7 Requirements..... | 29 |
| 5.8 Diagrams..... | 32 |
| 5.8.1 Use case diagram..... | 32 |
| 5.8.1.1 UC01 – Login / Authenticate..... | 33 |
| 5.8.1.2 UC02 – Create / Manage Profile..... | 33 |
| 5.8.1.3 UC03 – Manage Medical Parameters..... | 33 |
| 5.8.1.4 UC04 – Upload Blood Test..... | 33 |
| 5.8.1.5 UC05 – Upload Docs for Scan..... | 33 |
| 5.8.1.6 UC06 – Scan Docs..... | 34 |
| 5.8.1.7 UC07 – View Feedback History..... | 34 |
| 5.8.1.8 UC08 – Export Feedback Report..... | 34 |
| 5.8.1.9 UC09 – Manage Report..... | 34 |
| 5.8.1.10 UC10 – Extract Text from Files..... | 34 |
| 5.8.1.11 UC11 – Parse & Structure Data..... | 35 |
| 5.8.1.12 UC12 – Analyse Medical Data..... | 35 |
| 5.8.1.13 UC13 – Generate and Send Feedback..... | 35 |
| 5.8.2 Activity diagram..... | 36 |
| 5.8.3 Workflow..... | 36 |
| 5.8.3.1. Authorization..... | 36 |
| 5.8.3.2 Profile Parameters Input..... | 37 |
| 5.8.3.3 Navigation Decision..... | 37 |
| 5.8.3.4 Manage Account..... | 38 |
| 5.8.3.5 View Feedback History..... | 38 |
| 5.8.3.6 Scan Product..... | 39 |
| 6. System User Interface..... | 40 |
| 7. Verification and Evaluation..... | 42 |
| 7.1 Evaluation..... | 42 |
| 7.2 Verification..... | 43 |
| 8. References..... | 45 |

1. Abstract

Purchasing groceries is an integral part of daily life, but the preparatory process involves a series of manual tasks—alerting household members, checking existing supplies, and drawing up lists. Despite the assistance from digital tools like phone applications, chatbots, or memo papers, users often recall essential items at times and do not synchronize food consumption with their medical needs. We create a smart grocery management platform that moves beyond traditional mechanisms. The system can predict potential diseases by scanning blood test reports from users and generating individualized shopping lists according to health-based dietary advice. It also scans supermarket receipts to check if the items purchased are by the health-based diet advice of the user. The platform reduces man-power usage, automates core processes, and optimizes the effectiveness of food shopping by involving health and nutrition advice in the purchase process.

2. Introduction

Grocery shopping is a fundamental aspect of daily life, yet the choices one makes while grocery shopping hardly overlap with one's health needs. Most individuals have ongoing or undiagnosed health conditions that could be better managed or avoided by wiser food decisions. However, deciding what food is beneficial for one's specific medical condition typically must be accomplished by a professional, which is readily accessible while everyday grocery shopping. As a result, people unknowingly purchase foods detrimental to their health and miss out on prevention and increased well-being opportunities.

Currently, after a diagnosis of illness, doctors tend to give patients generic dietary recommendations—general guidance about what foods to eat or avoid—rather than a detailed list of specific product-level items that are suitable for their condition. Without tangible, product-level information, patients find it difficult to translate medical suggestions into effective day-to-day food choices, especially in a grocery store setting.

The core issue that this project can address is the lack of a smart system that connects personal health data, expressed as blood work results, to food choices. While many applications related to health and nutrition are present, they tend to rely on manual input, general diet advice, or lack instant feedback concerning medical data. Current grocery apps focus on convenience and catalog management, and not on guiding customers toward healthier products based on individual medical histories.

We recommend a smartphone application, CartGenie, a health-based smart grocery assistant. The program reads the users' blood test reports to identify potential diseases or health risks and, therefore, generates personalized grocery recommendations according to medically recommended diet guidelines. Additionally, the website may also read and evaluate grocery receipts from grocery stores to determine whether the items purchased suit the user's health-based food requirements. The process educates users regarding the selection of health food consistent with their medical needs and facilitates long-term health benefits through diet.

The stakeholders of the solution are individuals with chronic illnesses, health-conscious consumers, caregivers, and medical professionals who want better tools to help patients. The app empowers users to control their diet in a simple, accessible way, retroactively, bridging the gap between medical diagnosis and everyday decisions in the supermarket.

In a time where preventive wellness and customized healthcare are becoming more and more prominent, CartGenie offers a fresh, data-driven approach to grocery shopping that turns health information into actionable, food-related guidance.

3. Literature review

This chapter reviews current literature in two broad fields: (1) platforms that individuals use to build and maintain grocery lists, and (2) platforms that enable the tracking of nutrition and provide dietary recommendations to users. These two fields lay the

groundwork for grasping the solutions' current shortcomings and highlighting the envisaged innovation of the proposed system, CartGenie.

3.1 Grocery List Management Platforms

Grocery list apps are generally employed to help users plan and manage their shopping habits. Applications like Out of Milk, Bring!, and AnyList provide simple features such as list sharing, categorization, barcode scanning, and synchronization among family members. These apps have gained popularity because they are easy to use and convenient. Nevertheless, they are mostly intended for logistical assistance and do not integrate with health-related or nutritional data.

Human-computer interaction and personal informatics research suggest that while such platforms improve the efficiency of shopping, they are not sensitive to the user's health profile or nutritional needs. Hence, grocery selections remain disconnected from specific health contexts, potentially resulting in nutritionally unfavorable eating habits, particularly among users with chronic ailments or specific nutritional needs. Several studies promote smarter systems that not only facilitate list generation but also help users select better, condition-appropriate food options.

3.2 Nutrition Monitoring and Dietary Advising Systems

Simultaneously, a wide range of mobile health (mHealth) applications focus on nutrition monitoring, caloric intake, and dietary counseling. These include Yazio, MyFitnessPal, and Cronometer, which support users in inputting food consumed, tracking macronutrients, and assessing how well dietary targets are being followed. These technologies are underpinned by theories of behavior change and typically involve the use of food databases to provide instant nutritional advice. They tend to be largely dependent on manual entry of information, however, which has been shown to lower long-term usage due to excessive cognitive burden.

In addition, most nutrition apps offer generic nutritional advice and typically do not incorporate clinical health data such as blood biomarkers, lab work, or diagnostic outcomes. Personalized nutrition systems have attempted to fill this gap more recently with evidence-based research. For example, Trattner and Elsweiler (2017) dealt with recipe recommender systems based on users' health goals, and Chen et al. (2020) showcased the potential of machine learning algorithms in tailoring meal plans to diabetics or cardiovascular patients. However, these models operate only within the context of recommending recipes and not within real grocery shopping scenarios.

Aside from that, although platforms like Apple HealthKit and Google Fit integrate basic health measures, they have very little potential in terms of diet personalization. Latest studies (e.g., Patel et al., 2022) recognize the potential in integrating clinical data, such as blood glucose or lipid profiles, into dietary advice. However, current consumer-facing technology rarely interprets such data into effective advice at the point of food selection.

3.3 The Need for CartGenie in Daily Nutrition Management

Recent literature highlights the primary lack of integrating personal health information into day-to-day food nutritional choices, particularly in shopping for groceries. A seminal research by Zeevi et al. (2015) displayed that reactions to postprandial glucose varied extremely between different individuals when consumed with the same meal. The variability highlights the fact that biomarker-based customized dietary recommendations would take into consideration test results, such as a blood test, rather than universal nutrition counseling.

Even with growing awareness of personalized nutrition, existing tools do not support practical users. Wang et al. (2019) conducted a systematic review evidencing that the majority of diet tracking apps register high rates of user dropout, primarily due to the cognitive burden of manual food diary entry and poor meaningful, personalized

feedback. These shortcomings suggest that automated context-aware systems are needed to reduce user effort while promoting compliance with health recommendations.

Nunes and Verdezoto (2019) continued to note the disconnection between everyday activities and self-care technologies, and noted that most grocery or meal planning apps are designed for convenience, not for enabling support for health management by medical profiles. They saw a gap in unmet need for technology that combines everyday activities, like grocery shopping, and long-term health goals.

To validate this research line, Chen et al. (2020) demonstrated that artificial intelligence can convert clinical data, such as diagnoses or laboratory tests, into action-oriented dietary recommendations. Their work validates the potential of intelligent systems to assist individuals in making food choices aligned with specific health states.

Together, these studies paint a picture of a global deficit in personalized nutrition: patients diagnosed with diseases like diabetes will typically be sent home with advice such as "stay away from sugar," and "eat whole grains," but they are not given a feasible, itemized shopping list tailored to their cases. CartGenie takes this gap away with an astute grocery management system that reads blood test results, develops medically sound grocery recommendations, and reviews purchasing history by scanning receipts. By adding clinical wisdom to ordinary shopping patterns, CartGenie allows users to make informed choices that uphold preventive wellness and regulate ongoing ailments.

3.4 Optical Character Recognition in the Project: Using Tesseract OCR

One of the core functionalities of our application is the ability to automatically extract textual information from various user-provided inputs, such as blood test reports, grocery receipts, or photos of nutritional labels. To enable this, we utilize Tesseract OCR — an open-source Optical Character Recognition engine developed by Hewlett-Packard and currently maintained by Google.

3.4.1 What is Tesseract OCR?

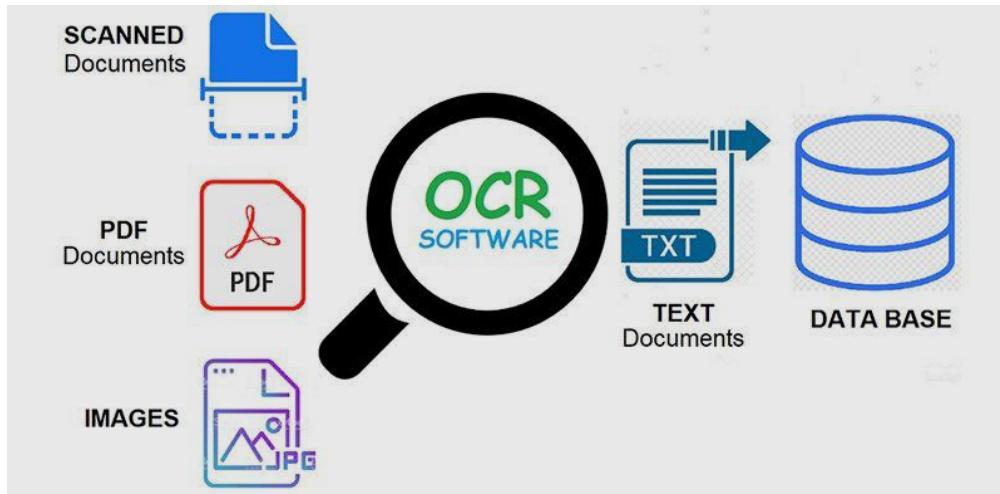
Tesseract OCR is a machine learning-based engine that converts raster images (e.g., JPG, PNG, scanned documents) into machine-readable text. Starting from version 4.0, Tesseract incorporates a Long Short-Term Memory (LSTM) based neural network, which significantly improves its ability to recognize printed text even in noisy or low-quality conditions.

Tesseract supports over 100 languages, including Hebrew, and can handle Right-to-Left (RTL) scripts. This is particularly relevant for our use case, as many Israeli users submit documents in Hebrew.

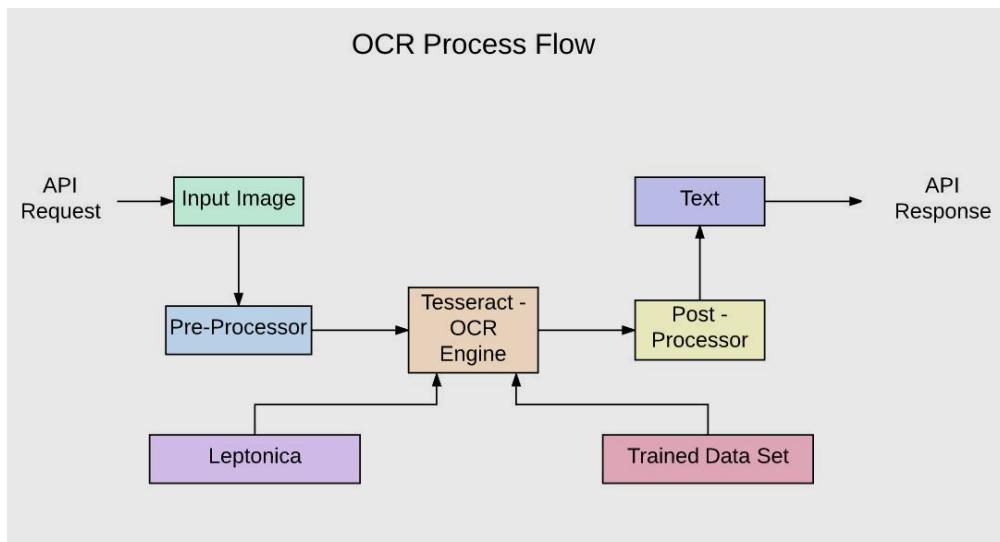
3.4.2 How Tesseract Works

The OCR process with Tesseract generally involves the following steps:

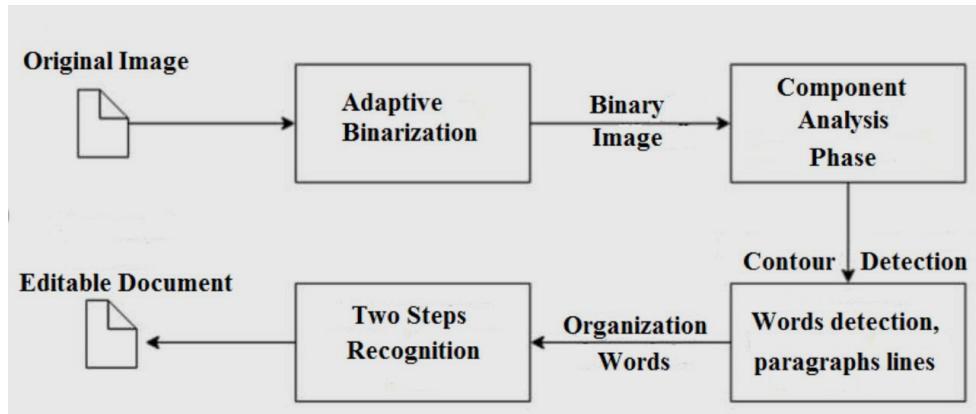
1. Preprocessing: The image is binarized (converted to black and white) and optionally deskewed to improve clarity. Tools such as OpenCV or PIL are often used to enhance the input before recognition.
2. Layout Analysis: Tesseract identifies text blocks, paragraphs, lines, and words based on spatial structure.
3. Character Recognition: The LSTM model is applied to recognize sequences of characters, allowing context-aware interpretation of letters and words.
4. Post-processing: Dictionary-based corrections and confidence scoring improve output accuracy. The output can be returned in plain text or structured formats (e.g., HOCR, ALTO XML).



Optical Character Recognition process



OCR Process Flow to build an API with Tesseract from a blog post



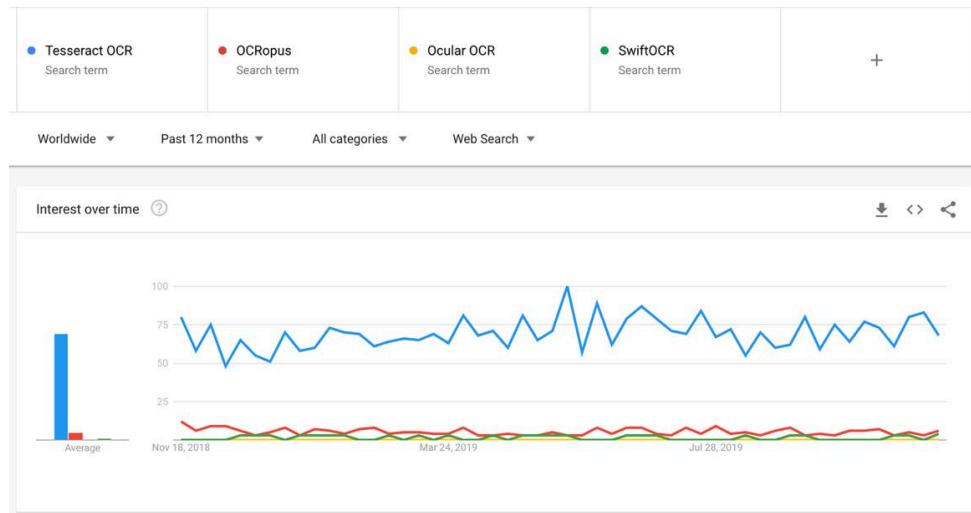
Tesseract 3 OCR process from paper

3.4.3 Effectiveness and Limitations

For high-quality printed documents, Tesseract can reach over 95% accuracy. However, in cases of:

- Low resolution or poor lighting
- Skewed or rotated images
- Complex layouts (e.g., tables or multiple columns)
- Hebrew text with vowels or mixed scripts

The accuracy may drop to 80–90% or lower, depending on preprocessing quality and correct configuration (e.g., using the Hebrew language pack and appropriate Page Segmentation Mode). Tesseract is not well-suited for handwritten text, but it performs well with printed Hebrew, assuming clear fonts and good contrast.



3.4.4 Relevance to Our Application

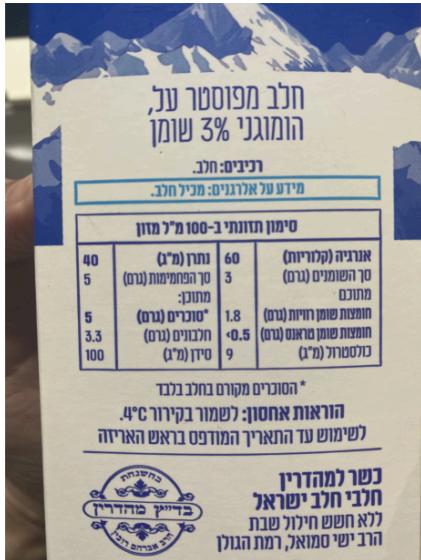
In our project, Tesseract OCR enables us to process three types of user inputs:

1. Blood Test Screenshots (PDFs or images): Extracting numerical values (e.g., cholesterol, glucose) and identifying medical terms to interpret the user's health profile.
2. Grocery Receipts: Parsing product names and quantities to evaluate shopping behavior.
3. Nutrition Labels: Capturing key data such as calories, sugar, sodium, saturated fat, which are then cross-validated against the user's health constraints.

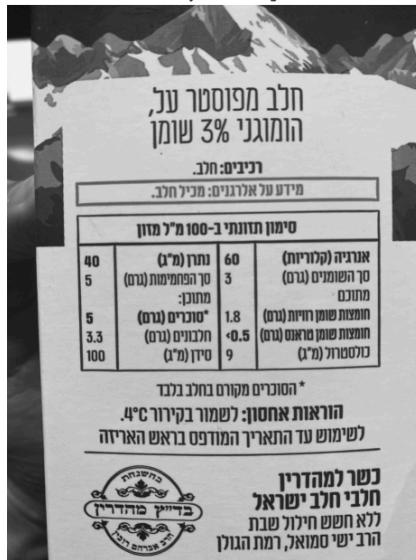
Once extracted, the textual data is analyzed using custom classification rules and a health-aware decision engine to determine whether the product is recommended or potentially harmful.

3.4.5 Product Nutrition Table Scan

1. Original Image (BGR)



2. Grayscale Image



3. Thresholded Image (Text should be white)



4. Dilated Image (Text connected)



5. Contours Detected



3.4.5 Grocery Receipt Scan

| 1. Original Image (BGR) | | | | 2. Grayscale Image | | | | 3. Thresholded Image (Text should be white) | | | |
|-------------------------|--|------------------------|-------|--------------------|--|------------------------|-------|---|--|------------------------|-------|
| לתשולם | תאור | קוד | תפקיד | לתשולם | תאור | קוד | תפקיד | לתשולם | תאור | קוד | תפקיד |
| | שופרסל בע"מ הגהנה, 2 כרמיאל, נ.צ. 04-6276788 520022732 מושך מושחה 0557652476 02226046399086 שחבוני / קבלת | 1867037 | | | שופרסל בע"מ הגהנה, 2 כרמיאל, נ.צ. 04-6276788 520022732 מושך מושחה 0557652476 02226046399086 שחבוני / קבלת | 1867037 | | | שופרסל בע"מ הגהנה, 2 כרמיאל, נ.צ. 04-6276788 520022732 מושך מושחה 0557652476 02226046399086 שחבוני / קבלת | 1867037 | |
| 16.10 | תפוח אדמה לבן ק"ג 4.90 x 3.285 | 4030810 | | 16.10 | תפוח אדמה לבן ק"ג 4.90 x 3.285 | 4030810 | | 16.10 | תפוח אדמה לבן ק"ג 4.90 x 3.285 | 4030810 | |
| 19.00 | מיץ תפוז בנהנה פס'פ | | | 19.00 | מיץ תפוז בנהנה פס'פ | | | 19.00 | מיץ תפוז בנהנה פס'פ | | |
| 0.30 | המזרע כל פיקודן אריזה | | | 0.30 | המזרע כל פיקודן אריזה | | | 0.30 | המזרע כל פיקודן אריזה | | |
| | פטל בון | | | | פטל בון | | | | פטל בון | | |
| 12.90 | פירות&וגו אנטס תפוח המזרע כל פיקודן אריזה | 7290018471180 | | 12.90 | פירות&וגו אנטס תפוח המזרע כל פיקודן אריזה | 7290018471180 | | 12.90 | פירות&וגו אנטס תפוח המזרע כל פיקודן אריזה | 7290018471180 | |
| 0.30 | 2. בון | | | 0.30 | 2. בון | | | 0.30 | 2. בון | | |
| 29.90 | מאנדרת מסקרופונה תפוציסט בברזיון וDOBש | 7290018652923 | | 29.90 | מאנדרת מסקרופונה תפוציסט בברזיון וDOBש | 7290018652923 | | 29.90 | מאנדרת מסקרופונה תפוציסט בברזיון וDOBש | 7290018652923 | |
| 19.60 | 4.90 x 4 ליחידה | 7622201139278 | | 19.60 | 4.90 x 4 ליחידה | 7622201139278 | | 19.60 | 4.90 x 4 ליחידה | 7622201139278 | |
| 31.80 | גבינה פילוליה ליחידה | 7290116932033 | | 31.80 | גבינה פילוליה ליחידה | 7290116932033 | | 31.80 | גבינה פילוליה ליחידה | 7290116932033 | |
| 20.20 | חמאה מהדרין 200 גרם | 7290116932033 | | 20.20 | חמאה מהדרין 200 גרם | 7290116932033 | | 20.20 | חמאה מהדרין 200 גרם | 7290116932033 | |
| 33.90 | פילה הרינג 400 גרם | 4810168057435 | | 33.90 | פילה הרינג 400 גרם | 4810168057435 | | 33.90 | פילה הרינג 400 גרם | 4810168057435 | |
| -4.00 | 400 גרם דג הרינג 29.90 | 400 גרם דג הרינג 29.90 | | -4.00 | 400 גרם דג הרינג 29.90 | 400 גרם דג הרינג 29.90 | | -4.00 | 400 גרם דג הרינג 29.90 | 400 גרם דג הרינג 29.90 | |

| 4. Dilated Image (Text connected) | | | | 5. Contours Detected | | | |
|---|--|--|--|---|--|--|--|
| <p>שופרסל בע"מ הגהנה, 2 כרמיאל, נ.צ. 04-6276788 520022732 מושך מושחה 0557652476 02226046399086 שחבוני / קבלת</p> <p>לתשולם</p> <p>תאור</p> <p>קוד</p> <p>תפקיד</p> <p>16.10 19.00 0.30 29.90 19.60 31.80 20.20 33.90 -4.00</p> <p>תפוח אדמה לבן ק"ג 4.90 x 3.285 מיץ תפוז בנהנה פס'פ המזרע כל פיקודן אריזה פטל בון פירות&וגו אנטס תפוח המזרע כל פיקודן אריזה 2. בון</p> <p>7290018471180 7290018652923 7622201139278 7290116932033 7290116932033 7290116932033 4810168057435 400 גרם דג הרינג 29.90</p> | | | | <p>שופרסל בע"מ הגהנה, 2 כרמיאל, נ.צ. 04-6276788 520022732 מושך מושחה 0557652476 02226046399086 שחבוני / קבלת</p> <p>לתשולם</p> <p>תאור</p> <p>קוד</p> <p>תפקיד</p> <p>16.10 19.00 0.30 29.90 19.60 31.80 20.20 33.90 -4.00</p> <p>תפוח אדמה לבן ק"ג 4.90 x 3.285 מיץ תפוז בנהנה פס'פ המזרע כל פיקודן אריזה פטל בון פירות&וגו אנטס תפוח המזרע כל פיקודן אריזה 2. בון</p> <p>7290018471180 7290018652923 7622201139278 7290116932033 7290116932033 7290116932033 4810168057435 400 גרם דג הרינג 29.90</p> | | | |

3.5 Disease and Health Condition Analysis: Using XGBoost

Once text has been extracted from documents using Optical Character Recognition (OCR), the next critical task in our application is interpreting this data to assess the user's health risks and dietary compatibility. To accomplish this, we employ XGBoost (Extreme Gradient Boosting), a powerful machine learning algorithm known for its high accuracy, robustness, and efficiency in classification problems.

3.5.1 What is XGBoost?

XGBoost is an open-source machine learning library that implements optimized gradient boosting algorithms designed for speed and performance. Originally developed by Tianqi Chen, it has become one of the most widely used methods in structured/tabular data tasks in both industry and academia.

XGBoost is based on the gradient boosting framework, where multiple weak learners (typically decision trees) are trained sequentially. Each new tree corrects the mistakes of the previous ones by minimizing a specific loss function using gradient descent.

Key advantages of XGBoost include:

- High predictive accuracy
- Fast training time
- Built-in regularization to prevent overfitting
- Support for missing data
- Feature importance analysis

3.5.2 How XGBoost Works?

The core idea of XGBoost involves:

1. Initializing a model (starting with a constant prediction, e.g., average)
2. Iteratively training new decision trees to minimize the residual error of the previous predictions
3. Combining the outputs of all trees into a final prediction through weighted aggregation

XGBoost optimizes the following objective:

$$L(\theta) = \sum l(y_i, \hat{y}_i) + \Omega(f)$$

Where:

- L is the total loss
- $l(y_i, \hat{y}_i)$ is the error between the prediction and the true label
- $\Omega(f)$ is the regularization term penalizing complexity

XGBoost supports classification, regression, ranking, and custom loss functions, making it highly adaptable.

3.5.3 Relevance to Our Application

In the context of our health recommendation app, XGBoost is used to classify the user's risk level for specific conditions based on blood test results and other features. The process includes:

1. Feature Engineering: Extracted data (e.g., LDL cholesterol, glucose, hemoglobin levels) is converted into numerical vectors.
2. Labeling: Training data is annotated with known diagnoses (e.g., prediabetes, hypertension, anemia) based on clinical thresholds or historical medical records.
3. Model Training: An XGBoost classifier is trained to recognize patterns in the input features that correspond to specific diseases or health risks.
4. Prediction: For each new user input, the model predicts the likelihood of various health conditions and flags relevant risks.

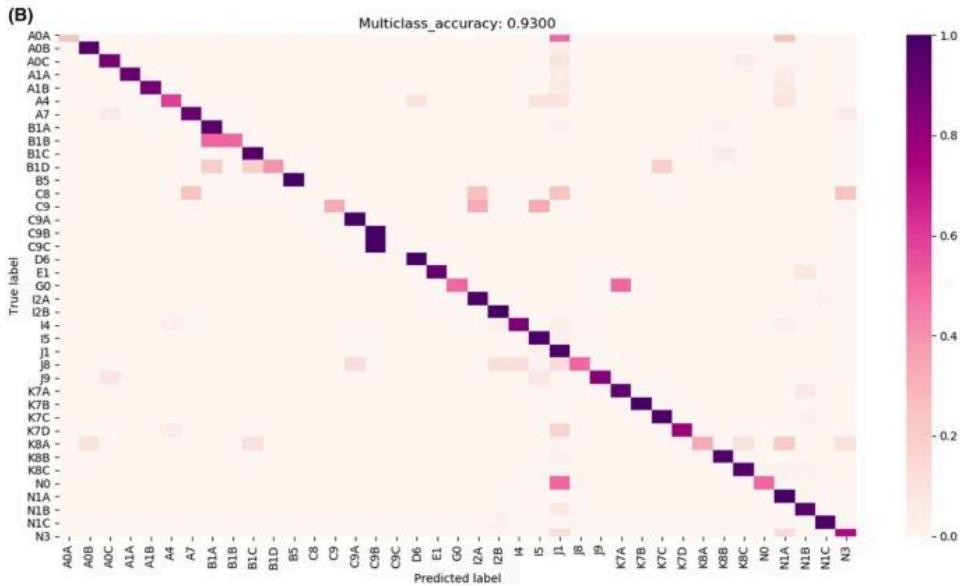
For example:

- If the extracted values include fasting glucose = 130 mg/dL and HbA1c = 6.5%, XGBoost may output a high probability for Type 2 Diabetes.
- If sodium intake from grocery products is high, and the user has hypertension indicators, the model can flag this food as a risk.

3.5.4 Why XGBoost?

XGBoost was selected due to:

- Its excellent performance on tabular health data
- Ability to deal with sparse or partially missing information
- Clear explainability via feature importance, which allows us to justify recommendations
- Compatibility with both numerical and categorical data from OCR outputs



Confusion matrix of the ensemble model (optimal accuracy model): (A) The predictive power (accuracy) of TOP1 (representing the most likely disease) result, (B) TOP5 (the five most likely diseases) result.

3.6 Data Preprocessing and Feature Management: Using pandas

To support our machine learning–based health evaluation pipeline, we require a robust tool to clean, structure, and manipulate data extracted from OCR. For this purpose, we utilize pandas, a powerful open-source data analysis library in Python. Pandas plays a critical role in bridging the gap between unstructured text (e.g., blood test results or product nutrition data) and structured machine learning input vectors.

3.6.1 What is pandas?

Pandas is a data analysis library that provides fast, flexible, and expressive data structures designed to work with "relational" or "labeled" data, particularly tabular data. The core component is the DataFrame, a 2-dimensional labeled data structure with columns that can contain values of different types (e.g., integers, floats, strings).

Originally developed by Wes McKinney in 2008, Pandas is now an essential part of the Python data science ecosystem and is widely used in data engineering, machine learning, and analytics applications.

3.6.2 How Pandas Works?

Pandas supports a wide range of operations, such as:

- Reading data from various sources (CSV, Excel, JSON, SQL)
- Cleaning and formatting data (handling missing values, renaming columns)
- Transforming and filtering rows and columns
- Aggregating and summarizing statistics
- Exporting data to machine learning pipelines (e.g., scikit-learn, XGBoost)

3.6.3 Role in the Project Pipeline

In our React-based smartphone application, the user provides screenshots or photos of medical or nutritional documents. These are processed by OCR (Tesseract), resulting in raw text. That text is parsed into values, which pandas then helps to manage and convert into a clean tabular format. These steps are crucial before passing the data to the XGBoost model.

Key tasks where pandas is used in our project include:

1. Data Cleaning:

- Remove non-numeric values or outliers from OCR output

- Normalize measurement units (e.g., mg/dL vs mmol/L)

2. Feature Extraction:

- Convert values into feature vectors based on medical relevance
- Derive additional features (e.g., BMI from weight and height)

3. Dataset Preparation:

- Store input data in a consistent schema
- Feed pandas DataFrames directly into ML models

4. Debugging and Logging:

- Log each user input and model decision in a structured format
- Export data for later analysis or auditing

3.6.4 Benefits of Using pandas

One of the major benefits of using pandas is its speed and efficiency in handling large amounts of data. It supports vectorized operations, which means it can perform transformations and calculations across entire columns at once, making it highly scalable. Another advantage is its seamless integration with machine learning libraries such as XGBoost, scikit-learn, and NumPy, allowing smooth transitions from data preparation to model training and inference. Pandas also ensures data consistency by enforcing a structured format, which is essential when feeding inputs into predictive models. Lastly, it improves visibility into the data by making it easy to inspect, debug, and validate the

parsed information that was extracted from OCR — an important step in ensuring that the final predictions are based on accurate and reliable input.

4. Project Goals and Expected Achievements

The primary goal of the **CartGenie** project is to develop a **smartphone application** that functions as a health-aware grocery shopping assistant. The application aims to analyze the user's medical and physiological profile and provide feedback on their grocery shopping habits, based on:

- **Blood test results**
(e.g., glucose levels, cholesterol, iron, vitamin deficiencies, kidney/liver markers)
- **BMI (Body Mass Index)**
Used to evaluate weight-related health risks and metabolic needs
- **Gender**
Important for adjusting nutritional recommendations (e.g., iron intake for women)
- **Age**
Nutritional needs and health risks vary significantly with age
- **Known medical conditions**
(e.g., diabetes, hypertension, anemia) — either self-reported or inferred from a blood test
- **Activity level**
(e.g., sedentary, moderate, active) — influences caloric and macronutrient requirements
- **Dietary preferences or restrictions**
(e.g., vegetarian, vegan, gluten-free, lactose intolerance)

4.1 Expected Features and Deliverables

1. Blood Test Analysis Module

Ability to parse blood test reports (in text or digital formats such as PDF or JSON), extract relevant biomarkers (e.g., glucose, cholesterol, iron, vitamin levels), and identify health risks or medical conditions.

2. User Profiling

Collection of additional user attributes (BMI, gender) to improve the personalization and accuracy of feedback.

3. Receipt Scanning and Review

Users upload supermarket receipts (manually or via image/PDF). The system identifies purchased items and compares them against medical dietary recommendations.

4. Health-Based Feedback Engine

Based on the combined profile (bloodwork, BMI, gender), the app evaluates how well the purchased groceries align with the user's health needs, highlighting problematic items (e.g., "too much sugar for a diabetic patient").

5. User Interface

An intuitive, mobile-friendly interface for uploading documents, reviewing feedback, and tracking dietary compliance over time.

4.2 System Workflow

1. Input phase:

- The user inputs their latest blood test results, BMI, and gender.
- The user uploads a receipt from a recent grocery shopping trip.

2. Analysis phase:

- The system extracts relevant health indicators from the blood test.
- The receipt is processed to identify and classify food items.

3. Evaluation phase:

- A rule-based engine (possibly enhanced with a basic machine learning layer) cross-references the food items with personalized dietary restrictions.
- Feedback is generated, indicating which items align with or violate health recommendations.

4. Output phase:

- The user receives a clear, itemized report indicating potential issues and suggestions (e.g., “high-sugar item not recommended due to elevated glucose levels”).

4.3 Success Criteria

To determine the success of the project, we define the following measurable criteria:

1. Accurate parsing of medical data

The system successfully identifies and interprets relevant biomarkers.

2. Effective item recognition from receipts

The application correctly extracts and classifies **at least 85% of food items** from scanned or uploaded receipts.

3. Relevant and medically sound feedback

At least **90% of the provided recommendations** match recognized dietary advice for users with the identified medical conditions.

4. Positive user feedback in the testing phase

During testing, **over 80% of participants** reported that the system provided helpful, understandable, and actionable feedback.

5. Stable application performance

The application runs reliably on smartphones with at least **95% uptime**, and responds within **2 seconds** to primary user actions.

5. Engineering Process

5.1 Process – Development Approach and Methodology

The development of **CartGenie** is structured over two academic semesters:

- **Semester A – Theoretical Phase**

Focused on research, use case analysis, system architecture design, and documentation. Deliverables include a comprehensive project book and validated technical design.

- **Semester B – Practical Phase**

Covers actual implementation, testing, and deployment of a Kotlin-based mobile app integrated with Flask (backend), MongoDB (storage), and ML Kit/Tesseract OCR modules.

5.2 Development Methodology

We adopted an **Agile-inspired iterative approach**, emphasizing flexibility, continuous validation, and incremental improvements. The motivation for this approach stems from:

- Frequent user input is required for tuning feedback mechanisms
- Variability in document formats (blood tests, receipts)
- Sensitivity of medical data and the need for reliability

Each development sprint in Semester B includes:

- Focused implementation of individual components (e.g., text extraction, profile creation, feedback logic)
- Unit testing and integration into the broader pipeline
- Early-stage user testing and feedback integration

5.3 Engineering Stages

5.3.1 Semester A – Research & Design

- Review of health-tech and nutrition tracking applications
- Study of OCR libraries (ML Kit, Tesseract) and their suitability for receipts and lab reports
- Mapping blood markers (e.g., glucose, LDL) to dietary implications
- Design of use case diagrams, system flow, and class structure
- UI/UX mockups for onboarding, receipt upload, and feedback display
- Documentation of success criteria and privacy assumptions

5.3.2 Semester B – Development & Testing

- **Frontend:** Development of a mobile app in Flutter
 - User registration and profile setup (BMI, age, gender, medical data)
 - File upload interface for blood tests and receipts (PDF/images)
- **Backend (Flask):**
 - Parsing and interpreting blood test data
 - Item-level nutritional evaluation from receipts
 - Logic for mapping health risks to shopping behavior
- **ML Kit / Tess-Two:**
 - Text extraction from receipts and lab reports
 - Preprocessing for improved OCR performance
- **MongoDB:**
 - Storing user profiles, past receipts, and feedback history
- **Testing:**
 - Unit tests for OCR and parsing modules
 - Pilot phase with synthetic and real user data
 - Performance tuning for mobile constraints

5.4 Constraints and Challenges

- **Data Privacy:** Personal health data is handled locally or through secure transmission with minimal external dependencies.
- **OCR Accuracy:** Receipt formats vary greatly; robust text recognition and preprocessing (noise reduction, cropping) are essential.
- **Health Recommendation Accuracy:** Dietary logic is based on medical guidelines and must be both conservative and explainable.
- **Mobile Resource Limits:** The app targets mid-range Android devices, requiring efficient memory and power use.

5.5 Product – Algorithms, Models, and Interface

5.5.1 Key Algorithms and Data Flow

1. **OCR Engine (ML Kit / Tess-Two)**
 - Extracts text from receipts and blood test documents (PDF/image)
 - Preprocessing: binarization, noise reduction, alignment
2. **Blood Test Analyzer (Flask)**
 - Parses biomarkers such as glucose, cholesterol, and hemoglobin
 - Flag abnormalities using reference ranges and thresholds
3. **Health Rule Engine**
 - Cross-references grocery items with the user's medical profile
 - Tag items as *risky*, *neutral*, or *beneficial*
4. **Feedback Generator**
 - Compiles a summary of flagged items
 - Provides contextual explanations for each risk (e.g., "High sugar not recommended due to elevated glucose")

5.5.2 Planned Data Structures

```
UserProfile: {  
    age: Int,  
    gender: String,  
    BMI: Float,  
    healthConditions: List<String>,  
    bloodValues: Map<String, Float>  
}  
  
ReceiptItem: {  
    name: String,  
    category: String,  
    nutritionalTags: List<String>,  
    riskLevel: String  
}  
  
AnalysisResult: {  
    item: ReceiptItem,  
    issueDetected: String,  
    explanation: String,  
    severity: String  
}
```

5.6 User Interface Features

- **Onboarding & Profile Setup**
 - Enter age, gender, BMI, and medical history
 - Upload blood test reports
- **Receipt Upload Screen**
 - Image or PDF input

- Instant OCR preview and confirmation
- **Feedback Dashboard**
 - Color-coded flags (green/yellow/red) per item
 - Recommendations and risks explained in simple language
- **History & Trends**
 - Access past uploads and analysis
 - Export reports in PDF format

The UI is designed with a **mobile-first approach**, focusing on accessibility, clarity, and simplicity for users with varying levels of health literacy.

5.7 Requirements

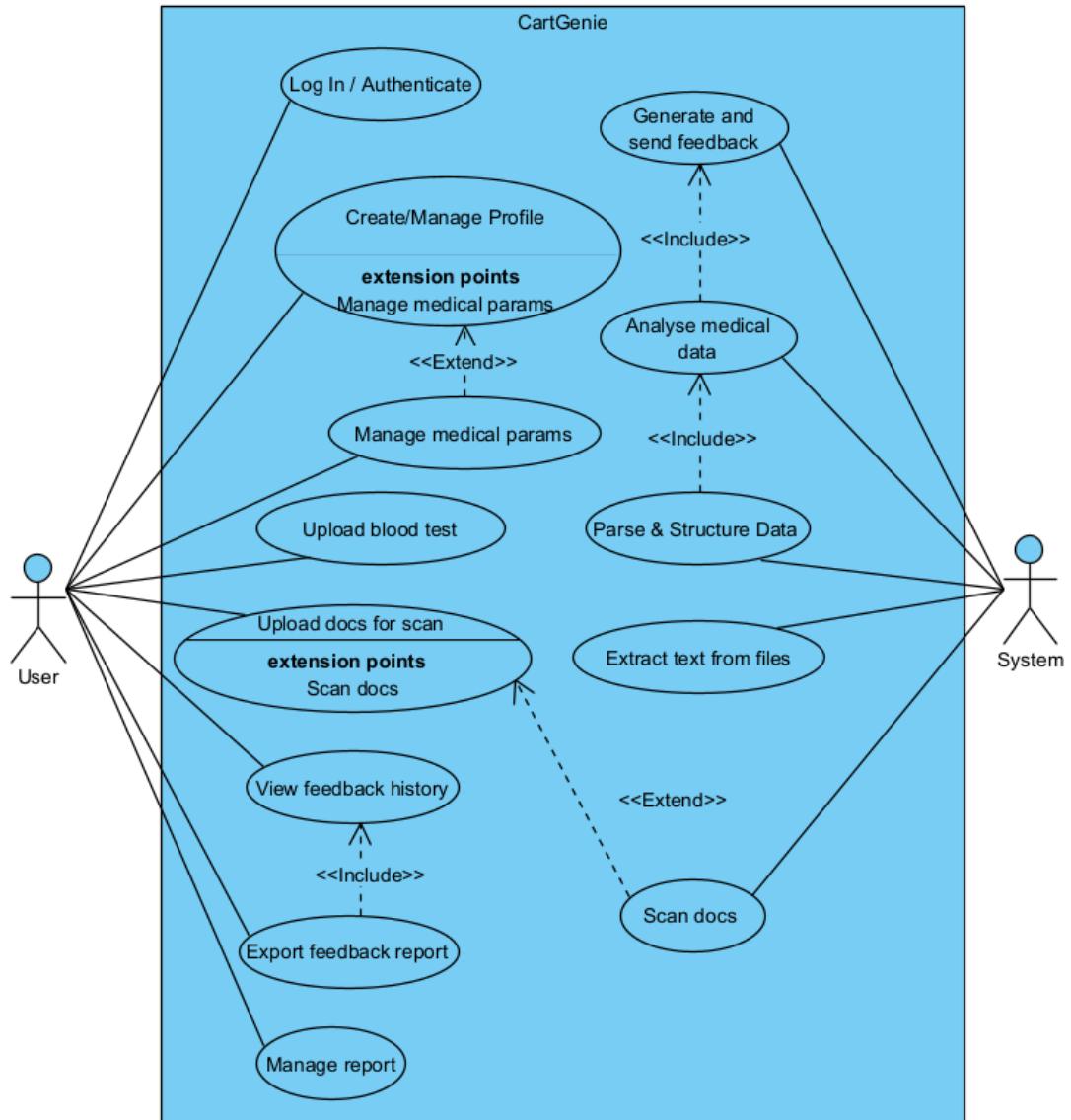
| # | Requirement Name | Description | Type |
|-----|-------------------------|---|------|
| 1 | User Profile Management | The system allows users to create and edit profiles | FR |
| 1.1 | User Profile Management | The parameters to edit are: age, gender, BMI, and health conditions | NFR |
| 2 | Upload Blood Test | The system allows users to upload blood tests | FR |
| 2.1 | Upload Blood Test | The blood tests will be in image or PDF format. | NFR |
| 3 | Upload Grocery Receipt | The system allows users to upload grocery receipts | FR |
| 3.1 | Upload Grocery Receipt | The grocery receipts will be in image or PDF format. | NFR |
| 4 | Text Extraction | The system extracts text from uploaded files | FR |
| 4.1 | Text Extraction | Text extracts using Tesseract OCR. | NFR |
| 5 | Scan Receipt | The system uses the device's camera to scan a receipt. | FR |
| 6 | Analyze Blood Test | The system parses and analyzes blood tests to identify abnormal values. | FR |
| 6.1 | Analyze Blood Test | Identification by using XGBoost model | NFR |
| 7 | Parse & Structure Data | The system converts extracted text into structured data | FR |
| 7.1 | Parse & Structure Data | Conversion of text is done by using parsing logic and pandas. | NFR |

| | | | |
|------|-----------------------|---|-----|
| 8 | Analyze Grocery Items | System matches grocery items with the user's health profile | FR |
| 8.1 | Analyze Grocery Items | Matching is done by using XGBoost to identify risks. | NFR |
| 9 | Generate Feedback | The system generates health-based feedback | FR |
| 9.1 | Generate Feedback | Feedback is based on user data and receipt items after XGBoost analysis. | NFR |
| 10 | View History | The system allows users to view past feedback reports. | FR |
| 11 | Manage Reports | Allow users to delete or modify saved reports. | FR |
| 12 | Export Report | Allow users to export feedback | FR |
| 12.1 | Export Report | Feedback report exported as a PDF. | NFR |
| 13 | Processing Speed | The system must process documents within 5 seconds (up to 2MB). | NFR |
| 14 | Uptime Guarantee | Maintain at least 95% system uptime. | NFR |
| 15 | Mobile UI Design | UI should be intuitive and optimized for mobile devices. | NFR |
| 16 | Dark Mode Support | Allow users to toggle between light and dark themes. | NFR |
| 17 | PDF Export Formatting | Exported reports must be formatted and styled correctly. | NFR |
| 18 | OCR Language Support | The system must support OCR in at least English and Hebrew for text extraction. | NFR |

| | | | |
|------|-----------------------------|---|-----|
| 19 | Blood Test Parsing Coverage | The system must accurately extract and interpret at least 10 common biomarkers (e.g., glucose, LDL, HDL, iron, etc.). | FR |
| 20 | Receipt Item Matching | The system must match at least 80% of receipt items to known food/nutrition categories. | FR |
| 21 | Item Risk Classification | Each grocery item must be classified | FR |
| 21.1 | Item Risk Classification | Classification is divided into 'healthy', 'neutral', or 'risky' categories. | NFR |
| 22 | Explain Feedback | Feedback must include a short explanation based on health logic (e.g., 'High sugar - not suitable for elevated glucose'). | FR |
| 23 | Profile Validation | The system must validate user input on the profile form | FR |
| 23.1 | Profile Validation | Validation is: age range 0-120, BMI between 10-60. | NFR |
| 24 | Data Retention Control | Users must be able to delete stored medical and receipt data at any time. | NFR |
| 25 | Battery Optimization | The app must be optimized to consume minimal battery during OCR and analysis. | NFR |

5.8 Diagrams

5.8.1 Use case diagram



5.8.1.1 UC01 – Login / Authenticate

The user logs in to the application using a secure authentication method (e.g., Google Sign-In). This step ensures that the user has access to personalized data, profile information, and private feedback history.

5.8.1.2 UC02 – Create / Manage Profile

The user creates or updates their personal profile, including entering details like age, gender, body mass index (BMI), and existing health conditions.

Extension Point:

→ **UC03 – Manage Medical Parameters:** Allows the user to edit specific medical values separately (e.g., blood pressure, glucose thresholds).

5.8.1.3 UC03 – Manage Medical Parameters

(Extends UC02)

The user inputs or updates individual medical data fields relevant to analysis, such as known conditions or risk thresholds (e.g., "Pre-diabetic range", "Iron deficiency"). These values are essential for accurate analysis later.

5.8.1.4 UC04 – Upload Blood Test

The user uploads a file (PDF/image) containing medical test results. These documents are typically structured lab reports with numerical values that the system will later parse.

5.8.1.5 UC05 – Upload Docs for Scan

The user uploads documents for scanning. These documents could be either grocery receipts, blood tests, or barcodes.

Extension Point:

→UC06 – Scan Docs: If the user prefers not to upload a file, they may instead scan the document directly using their device's camera.

5.8.1.6 UC06 – Scan Docs

(Extends UC05)

The system uses the mobile device's camera to scan a physical grocery receipt or printed blood test. The image is then sent to the system for text extraction and further analysis.

5.8.1.7 UC07 – View Feedback History

The user views previously generated health feedback reports. This functionality provides tracking and comparison over time.

Includes:

→ UC08 – Export Feedback Report

5.8.1.8 UC08 – Export Feedback Report

The user exports a selected feedback report to a PDF or shareable format or downloads it to the device. This is often used for medical consultations or personal record-keeping.

5.8.1.9 UC09 – Manage Report

The user can edit report names, delete outdated results, or categorize saved feedback. This enhances report organization and usability.

5.8.1.10 UC10 – Extract Text from Files

The system uses OCR (e.g., XGBoost) to extract raw text from uploaded documents (receipts, lab reports, products, or barcodes).

Extends UC05

The result is unstructured textual content for processing – string.

5.8.1.11 UC11 – Parse & Structure Data

This use case organizes and converts raw text into a structured, machine-readable form. It identifies medical fields (“Hemoglobin: 13.2”) and grocery items (“Coca-Cola 2L”), normalizing data for analysis.

5.8.1.12 UC12 – Analyse Medical Data

This use case includes both **UC10 – Extract Text** and **UC11 – Parse Data**.

The system:

- Combines profile info, blood test data, and receipt content
- Identifies medical risks or nutrition mismatches
- Uses rules or ML models (e.g., XGBoost)

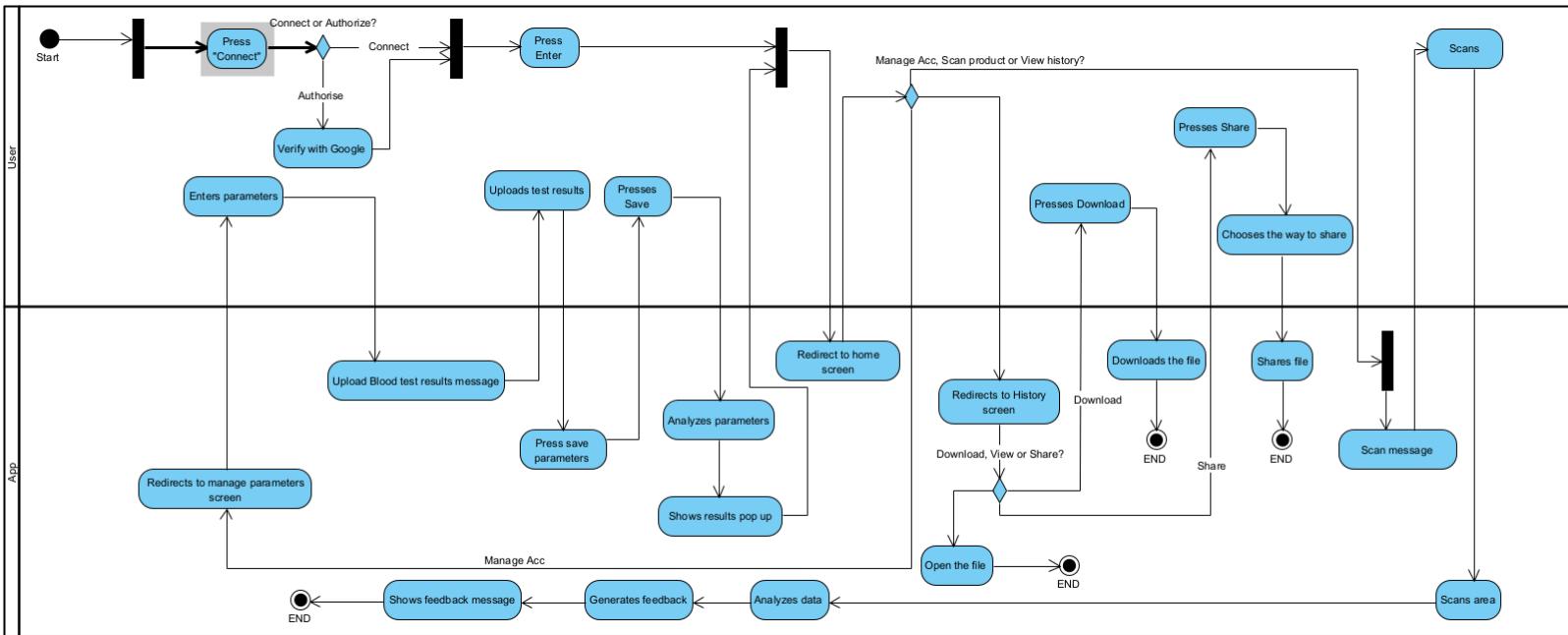
5.8.1.13 UC13 – Generate and Send Feedback

(Includes UC12)

Based on the analysis, the system produces a personalized report:

- Highlights problematic items
- Suggests alternatives
- Stores feedback in history
- Sends output to the user interface for display

5.8.2 Activity diagram



5.8.3 Workflow

5.8.3.1. Authorization

1. The user launches the app.
2. The user presses "Connect".
3. The system checks: **Connect or Authorize?**
 - If **Authorize** → system initiates **Google verification**.
 - If **Connect** → go to parameter input.

5.8.3.2 Profile Parameters Input

4. The user **enters profile parameters** (e.g., age, gender, BMI).
5. The system displays an "**Upload Blood Test Results**" message.
6. The user uploads their **blood test results**.
7. The user presses "**Save Parameters**".
8. The system **analyzes the entered parameters**.
9. The system displays a **results pop-up**.
10. The user presses "**Save**".
11. The system redirects to "**Manage Parameters Screen**".
12. The system **generates feedback**.
13. The system **analyzes data** (blood + receipt + profile).
14. The app shows a **feedback message**.

5.8.3.3 Navigation Decision

15. The user is prompted:
Manage account, Scan product, or View history?

5.8.3.4 Manage Account

16. If the user selects **Manage Account**:

- Redirect to **Home screen**
- **End**

5.8.3.5 View Feedback History

17. If the user selects **View history**:

- Redirect to **History screen**

18. User is prompted:

Download, View, or Share?

Options:

- **Download:**

19. User presses "**Download**"
20. The file is **downloaded**
21. **End**

- **View:**

22. User chooses "**Open the file**"
23. The file is **opened**
24. **End**

- **Share:**

25. User presses "**Share**"

26. Chooses sharing method

27. The file is **shared**

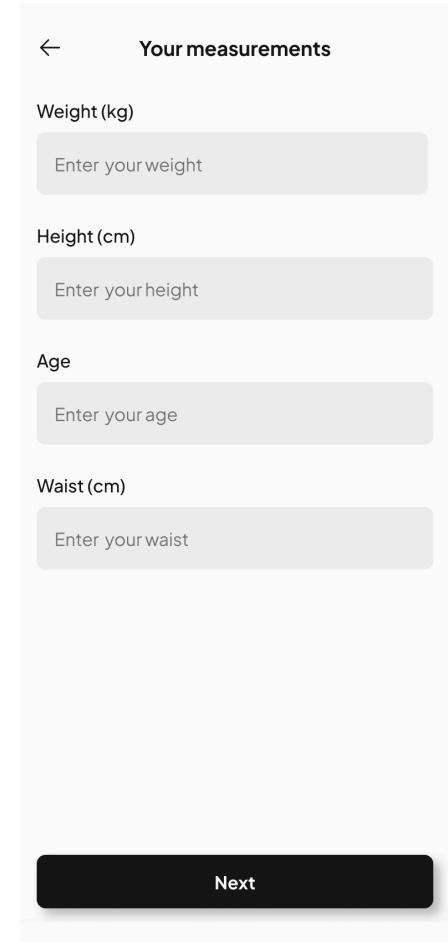
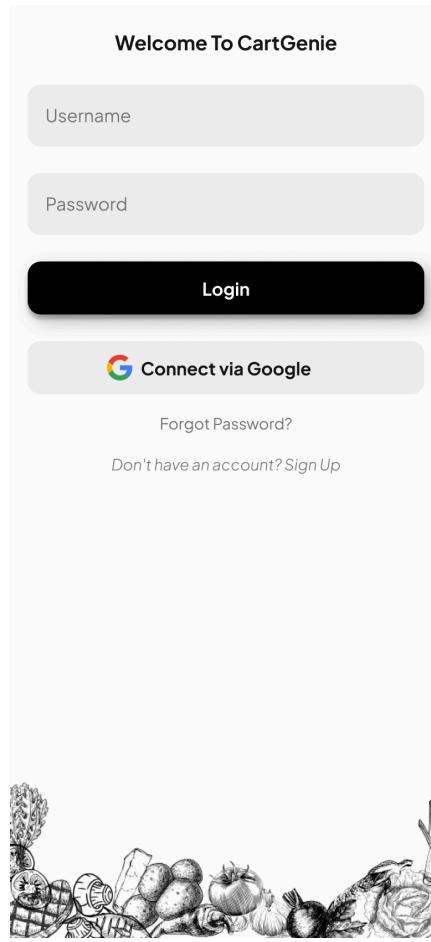
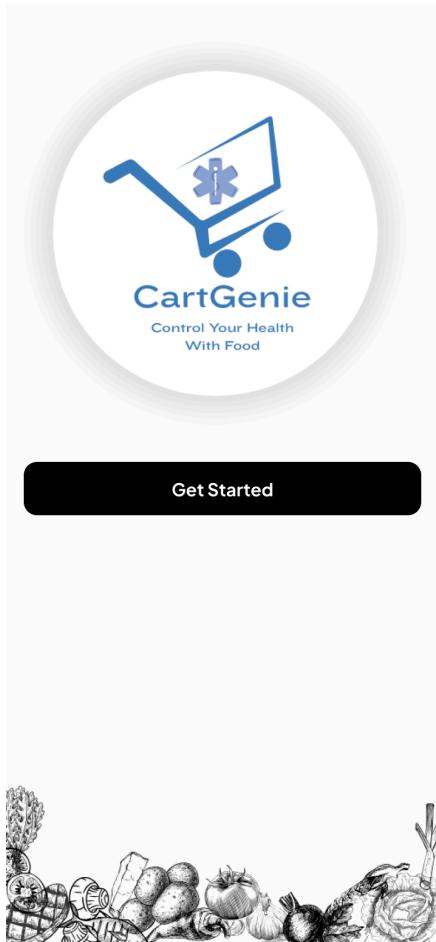
28. **End**

5.8.3.6 Scan Product

29. If the user selects **Scan Product**:

- The system activates the **Camera**
- A user scans a **receipt or document**
- The system displays a "**Scan message.**"
- The system scans the **area**
- **End**

6. System User Interface



← **Upload Blood Test Results**

Upload your recent blood test results to personalize your grocery recommendations. We accept PDF and JPG formats.

Drag and drop or browse
Accepted formats: PDF, JPG

Browse Files

Uploading...

Submit

Home **Browse** **Lists** **Account**

 **Grocery** 

Search for products

Your Feedback



Great selection of organic produce



Quick and easy checkout process



Helpful aisle

Scan Nutrition **Scan Receipt**



Home **Recipes** **Grocery** **Profile**

Scan

Align the nutrition table or receipt within the frame to scan.

Home **Shop** **Scan** **Recipes** **Profile**

Blood test Upload

Home Screen

Scan, Receipt, or Nutrition

7. Verification and Evaluation

7.1 Evaluation

The primary mission of CartGenie is to retrieve health-related information from documents (e.g., reports of blood tests, receipts of food) using OCR and machine learning and provide end-users with personalized feedback. The assessment will be based on the following criteria:

OCR Accuracy: Strong text recognition accuracy from various document types (blood reports, receipts, labels) in English and Hebrew, including RTL text.

- Data Processing Reliability: Accurate structuring and cleaning of OCR data, e.g., unit normalization and derived feature calculation.
- ML Classification Performance: Strong prediction accuracy of health conditions and food classes, measured by precision, recall, and F1-score.
- User Interface Quality: Simple and responsive user interface with smooth uploading, profile administration, result presentations, and report exporting.
- System Robustness: Robust operation across different devices, light conditions, file formats, and edge cases.
- Efficiency: Up to 2MB document processing should finish within 5 seconds with minimal battery consumption.
- Security & Privacy: Secure data storage and transmission with full user control over personal data.
- Testing will be conducted through actual test cases and edge cases. User feedback will be collected to make usability decisions, and results will be tracked for everyone.

7.2 Verification

The CartGenie system is structured into six modules: OCR, Data Processing (Pandas), ML Model (XGBoost), Web App (UI/UX), Security, and Integration. Each module undergoes unit and integration testing to ensure reliability and performance.

| No. | Module | Tested Function | Expected Result |
|-----|--------------------|-----------------------------------|--|
| 1 | OCR Module | Text Extraction Accuracy | Accurate recognition of medical/food data in English and Hebrew |
| 2 | OCR Module | RTL & Multilingual Support | Correct handling of Hebrew and mixed-language documents |
| 3 | OCR Module | Format Handling | Successful parsing of PDFs, JPGs, and PNGs |
| 4 | OCR Module | Noise/Skew Handling | Handle skewed/complex layouts with acceptable error tolerance |
| 5 | Data Processing | Text-to-DataFrame Conversion | Correctly structured Pandas DataFrame output |
| 6 | Data Processing | Missing/Invalid Data Handling | Resilient to incomplete or inconsistent OCR output |
| 7 | Data Processing | Unit Normalization | Consistent measurement unit conversion |
| 8 | Data Processing | Derived Feature Accuracy | Correct computation of BMI, other features |
| 9 | Data Processing | Data Cleaning | Effective removal of OCR artifacts and noise |
| 10 | ML Model (XGBoost) | Condition Classification Accuracy | $\geq 90\%$ accuracy for conditions (e.g., diabetes, hypertension) |
| 11 | ML Model (XGBoost) | Food Risk Prediction | Correct classification of food products as healthy/neutral/hazardous |

| | | | |
|----|--------------------|-----------------------------|---|
| 12 | ML Model (XGBoost) | Edge Case Handling | Accurate predictions near risk thresholds |
| 13 | ML Model (XGBoost) | Explainability | Feature importance is displayed correctly |
| 14 | Mobile App (UI/UX) | Profile Management | Users can edit age, BMI, and medical history |
| 15 | Mobile App (UI/UX) | Upload/Scan Functionality | Image/PDF upload and camera scan work reliably |
| 16 | Mobile App (UI/UX) | Report Generation | Reports generated and exported as PDF without formatting errors |
| 17 | Mobile App (UI/UX) | Feedback Visualization | Results are clearly shown with color-coded alerts and explanations |
| 18 | Mobile App (UI/UX) | Usability and Accessibility | UI is intuitive for all users and meets accessibility standards |
| 19 | Mobile App (UI/UX) | Performance | Page load <2s, document processed <5s (<2MB) |
| 20 | Mobile App (UI/UX) | Device Compatibility | Works on multiple Android versions, with dark mode support if enabled |
| 21 | Integration | End-to-End Workflow | Full process from upload to feedback works without issues |
| 22 | Integration | Error Handling | Graceful errors for unsupported or corrupted files |
| 23 | Integration | Stress Handling | Efficient performance on large documents and low-light scanned inputs |

8. References

Google Developers. (n.d.). *ML Kit text recognition*. Google. Retrieved June 9, 2025, from <https://developers.google.com/ml-kit/vision/text-recognition>

Tesseract OCR. (n.d.). *Tesseract open source OCR engine*. GitHub. Retrieved June 9, 2025, from <https://github.com/tesseract-ocr/tesseract>

Flask Documentation. (n.d.). *Flask: Web development with Python*. Pallets Projects. Retrieved June 9, 2025, from <https://flask.palletsprojects.com/>

MongoDB Inc. (n.d.). *MongoDB manual*. MongoDB. Retrieved June 9, 2025, from <https://www.mongodb.com/docs/>

U.S. Department of Agriculture. (n.d.). *FoodData Central*. USDA. Retrieved June 9, 2025, from <https://fdc.nal.usda.gov/>

World Health Organization. (2023). *Healthy diet*. Retrieved June 9, 2025, from <https://www.who.int/news-room/fact-sheets/detail/healthy-diet>

Mayo Clinic. (n.d.). *Blood tests – Overview*. Mayo Clinic. Retrieved June 9, 2025, from <https://www.mayoclinic.org/tests-procedures/blood-tests/about/pac-20384836>

Android Developers. (n.d.). *PdfRenderer | Android developers*. Google. Retrieved June 9, 2025, from <https://developer.android.com/reference/android/graphics/pdf/PdfRenderer>

Agile Alliance. (n.d.). *Agile 101*. Agile Alliance. Retrieved June 9, 2025, from

<https://www.agilealliance.org/agile101/>

GDPR.EU. (n.d.). *What is GDPR, the EU's new data protection law?*. Retrieved June 9, 2025,

from <https://gdpr.eu/what-is-gdpr/>

Zeevi, D., Korem, T., Zmora, N., Israeli, D., Rothschild, D., Weinberger, A., Ben-Yacov, O., Lador, D., Avnit-Sagi, T., Lotan-Pompan, M., Suez, J., Mahdi, J. A., Matot, E., Malka, G., Kosower, N., Rein, M., Zilberman-Schapira, G., Dohnalová, L., Pevsner-Fischer, M., ... Segal, E. (2015). Personalized nutrition by prediction of glycemic responses. *Cell*, 163(5), 1079–1094. <https://doi.org/10.1016/j.cell.2015.11.001>

Wang, Y., Xue, H., Huang, Y., & Huang, L. (2019). Effectiveness of mobile health interventions on diabetes and obesity treatment and management: Systematic review of systematic reviews. *JMIR mHealth and uHealth*, 7(4), e10299. <https://doi.org/10.2196/10299>

Nunes, F., & Verdezoto, N. (2019). Self-care technologies in HCI: Trends, tensions, and opportunities. *ACM Transactions on Computer-Human Interaction*, 26(4), 1–45.

<https://doi.org/10.1145/3332169>

Chen, J., Xu, H., Wang, Y., & Wang, F. (2020). Smart health dietary recommendation system using artificial intelligence: A review. *IEEE Journal of Biomedical and Health Informatics*, 24(6), 1634–1641. <https://doi.org/10.1109/JBHI.2020.2980226>

Sinha, A. (2020, April). OCR with Tesseract, OpenCV and Python. *Nanonets Blog*.

<https://nanonets.com/blog/ocr-with-tesseract/>

Suhara, A., Xu, Y., & Kwok, J. T. (2021). Personalized interpretations of lifestyle for health prediction. *Scientific Reports*, 11(1), 8134. <https://doi.org/10.1038/s41598-021-87171-5>