
# ÇANKAYA UNIVERSITY FACULTY OF ENGINEERING COMPUTER ENGINEERING DEPARTMENT

# CENG 407

## Innovative System Design and Development I

Project Report

## Team ID: 202417

## iProViS : Intelligent Product Vision System

Nursena Bitirgen-202011029

Tamer Memiş-201911210

Furkan Yamaner-202011211

Boran Gülbaşar-202011033

Advisor: Doç. Dr. Gül Tokdemir

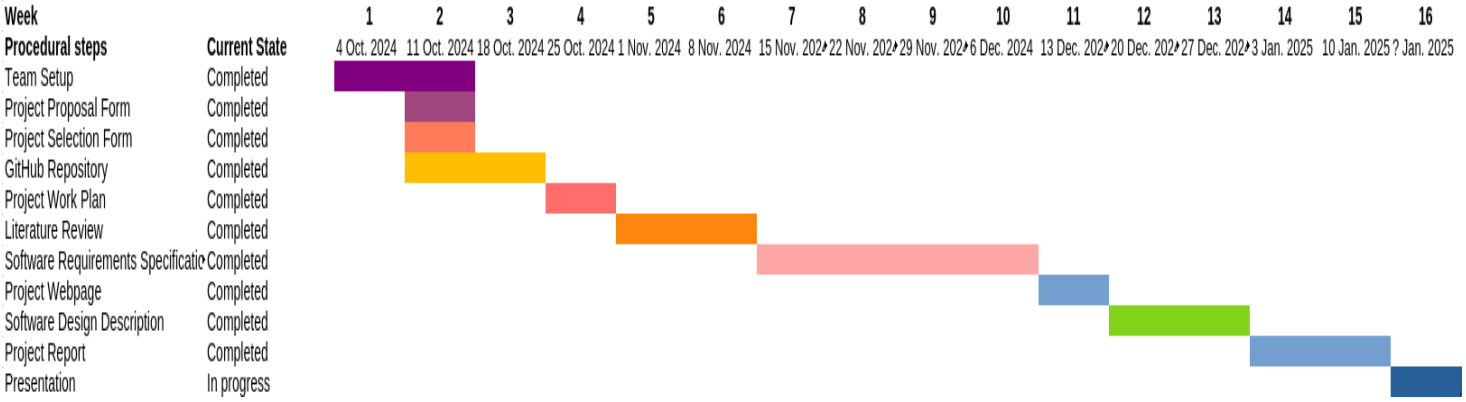
# Table of Contents

# Introduction

The Intelligent Product Vision System (iProViS) is an innovative solution designed to enhance the shopping experience by leveraging cutting-edge technologies such as computer vision, artificial intelligence, and data integration. The core objective of iProViS is to provide users with detailed product information, including pricing, stock availability, recyclability, and multilingual support, all through a simple photograph. By streamlining access to this information, iProViS empowers users to make informed decisions, save time, and embrace eco-friendly practices.

This project addresses key challenges faced by modern consumers, such as identifying products, comparing prices across multiple platforms, and understanding the environmental impact of purchases. With its modular and scalable architecture, iProViS integrates seamlessly with third-party APIs and ensures a smooth user experience. Additionally, the system includes a notification feature that alerts users to new deals and discounts, making it a valuable tool for budget-conscious shoppers. Through this project, we aim to bridge the gap between technology and consumer needs, fostering a more informed and sustainable marketplace.

# Project Work Plan




# Literature Review (LR)

## Abstract

This project proposes the development of a computer vision-based product recognition system aimed at revolutionizing the retail sector. Unlike traditional methods that rely on barcodes or QR codes, this system utilizes real-time image recognition to identify products across various retail categories such as supermarkets, clothing, food, and electronics. A key feature of the proposed system is a mobile application that enables consumers to access detailed product information, compare prices across different stores, and interact with multilingual content for a more inclusive shopping experience. In addition to enhancing consumer engagement, the project integrates an innovative intelligent recycling framework, leveraging the iProVis system to automatically recognize and sort recyclable product packaging. By providing insights on the type of packaging material and its information, the system aims to promoting environmental sustainability. This approach enhances user convenience while contributing to sustainability goals within the context of green IT strategies. Key developments features are; Shot and display, product recognition, mobile applications, and intelligent recycling, providing a comprehensive foundation for the proposed system's design and functionality!

## Introduction

With the rapid advancements in computer vision technology, there is an increasing interest in its application within the retail sector, specifically in recognizing products without the use of barcodes or QR codes. Traditionally, products in retail have been identified using barcodes or QR codes, but these methods have limitations such as the need for direct scanning and potential human error. Modern systems leverage neural networks, deep learning, and large-scale datasets to recognize products with high accuracy from photographs. The application of such systems in retail has a wide range of implications. It can assist consumers in accessing product details, comparing prices, and gaining more informed insights about the product.

## Related Work

## Methods of Product Recognition

Recent advancements in product recognition through computer vision have shown significant promise in various sectors, particularly retail. For instance, Guo et al. (2021) introduced a dataset named Fashion IQ aimed at improving image retrieval based on natural language feedback, a step towards enhancing product search capabilities in mobile applications. This development is critical for enabling users to compare products effectively, as the proposed system relies heavily on accurate and efficient image recognition algorithms. Furthermore, George and Floerkemeier (2014) explored a multi-label image classification approach, which allows for identifying multiple attributes of products within a single image, thus increasing the system's versatility in recognizing various products in supermarkets and clothing stores.

## Challenges and Limitations in Product Recognition

Despite the promising advancements, several challenges persist in developing a CV system for product recognition. One major challenge is the variability of lighting conditions that can affect image recognition accuracy. As noted in earlier studies, fallback solutions such as barcode recognition could serve as a temporary measure when CV fails to identify a product accurately. There is also multi-label image classification George and Floerkemeier (2014) key challenge is the inherent imbalance between different label classes, where some labels may appear more frequently than others, leading to biased model predictions. A solution they propose is employing techniques like weighted loss functions or data augmentation to balance the contribution of each label, improving model generalization.

## Methods of Price Comparison

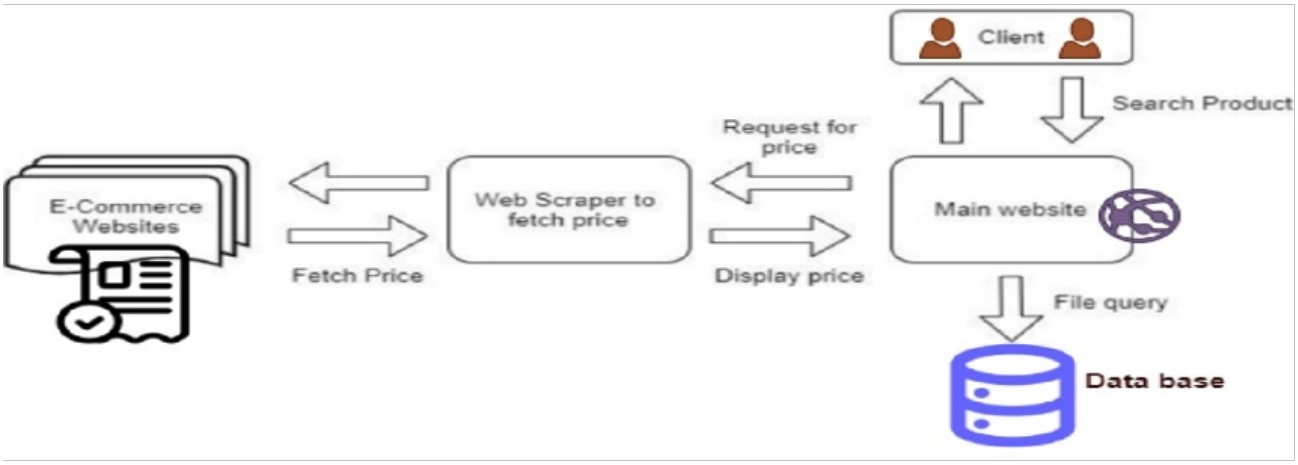
Price comparison sites are designed to compare the price of goods and services from a range of providers, which will help consumers in making decision to choose products that will save their money Arman Shaikh(2023). Price comparison systems typically rely on three primary methodologies for gathering data:

•Web Scraping: Early price comparison systems used web scraping to collect pricing information directly from retailer websites. Arman et al. (2023) outlined a typical scraping approach where automated agents (or "bots") traverse websites, extract product and price details, and store this data for comparison. However, this method presents challenges such as website structure changes, legal concerns, and the reliability of scraped data.

•API Integration: More recent systems use Application Programming Interfaces (APIs) provided by retailers, manufacturers, or third-party aggregators. APIs allow for the seamless exchange of data between price comparison platforms and retailers, ensuring that the pricing information displayed is more accurate and up-to-date. APIs offer advantages over scraping, such as faster access to real-time data and reduced legal and technical challenges.

•Crowdsourcing: In some cases, users of price comparison platforms actively contribute price data. Crowdsourcing can provide more granular and accurate price comparisons in physical retail environments. This approach has also been applied to areas like grocery shopping and brick-and-mortar retail.

•Data Aggregation and Big Data Analytics: Price comparison engines increasingly leverage big data techniques to aggregate and analyze pricing information across vast amounts of products and retailers. The use of machine learning to predict optimal pricing strategies or alert users to sales and discounts is another growing trend in the field.



## Challenges and Limitations in Price Comparison Systems

•Data Accuracy and Consistency: One of the key challenges in price comparison systems is ensuring the accuracy and consistency of data.

•Dynamic Pricing: With the rise of dynamic pricing algorithms, which change product prices in real-time based on demand, competition, and other factors, it is difficult for price comparison systems to capture accurate prices.

•Inclusion of Non-Price Factors: While price is an important determinant, consumers often consider non-price factors such as shipping costs, delivery times, return policies, and product quality.

•Privacy and Ethical Issues: Price comparison systems, particularly those that rely on web scraping or crowdsourcing, may raise privacy concerns, especially in regards to consumer data and tracking behaviors.

## Amazon's Just Walk Out Technology with Computer Vision

Just Walk Out Technology was first used by Amazon Go. The Go stores, which are driven by Amazon's "Just Walk Out" technology, use deep learning, weight sensors, and overhead cameras to identify items that customers take off or put back on shelves and to record the items that are placed in a virtual cart (Amazon, 2021). Using the Amazon Go smartphone app, customers enter the store through a turnstile. The Just Walk Out technology sends a receipt to the app and debits the customer's Amazon account for the items they bought when they leave the store. Our checkout-free shopping experience is made possible by the same technologies employed in self-driving cars: deep learning, sensor fusion, and computer vision (Tillman, 2021). When items are taken off or put back on shelves, Just Walk Out Technology detects it and keeps them in a virtual cart. Customers can easily leave the store after doing their purchasing. Amazon Go charges consumers' Amazon accounts after sending them a receipt via email (Amazon, 2021). The first thing that jumps out at us is how simple it is for retailers using computer vision technologies to determine whether the items that customers add to their shopping baskets match those in their databases, and then provide the prices of the items that do.


Figure 2: Working of Just Walk Out Technology
Source: (Pavia, 2018)

## Enhancing Retail Experiences through Mobile Product Recognition Applications

Mobile apps have become indispensable for improving user experiences in a variety of industries, but especially in retail, where they provide easy-to-use interfaces for product information and shopping support. Mobile applications use real-time image processing and deep learning algorithms to recognize products quickly and accurately. The capacity of mobile app-based product identification systems to do image-based searches, offer pricing information, and facilitate product comparisons in a matter of seconds was highlighted in a recent study by Chen et al. (2021). Furthermore, linguistic support improvements have made it possible for apps to serve a variety of users, improving accessibility and inclusion in the shopping experience. By combining these capabilities, the iProViS mobile application sets itself apart as a user-centered solution that puts convenience first while enabling smooth access to product details, multilingual content, and intelligent comparison tools.

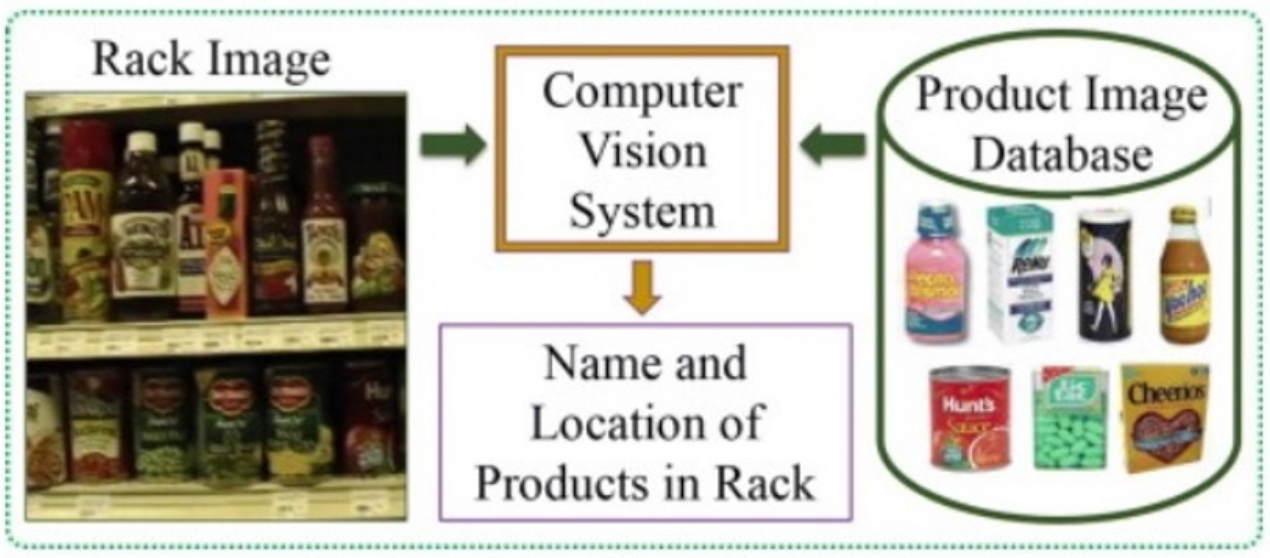
## Data Science and Machine Learning Inventory Tracking System

We concentrated on a smooth user experience, effective functionality, and technical viability in order to create a retail inventory tracking application that is both efficient and easy to use.

•Application Design and Functionality User navigation and basic application access were prioritized, and essential browsers were used for access. was given with little effort. Several layout choices were tested during the iterative design process until the ideal one was discovered. All associated stores were shown on individual pages with up-to-date information on the home screen. To allow customers to keep them for later inspection, camera features were incorporated into the product page. This design strategy guarantees that the right product information is directed and that the taken snapshot may be readily examined.

•Supporting Different Items in the Dataset A supermarket's shelves are stocked with a variety of unusual merchandise. The items' types, brands, and features vary widely, and these aspects have a significant impact on customer choices. A sizable dataset comprising images of every product had to be used to train the computer vision model so that it could accurately identify and classify these goods. However, this led to significant difficulties in gathering data and processing power. The amount of data needed and the higher chance of error made it impossible to train a model to identify each unique product. The solution used the strategy of classifying products into broad groups in order to mitigate this issue.

•Improving Accuracy and Model Performance Improving the computer vision model's accuracy and performance was another significant hurdle. Different lighting situations, crowded backgrounds, and product placement presented further hurdles for the model, which had to process photos of varying quality and perspectives. To guarantee precise object detection and accurate counts, the model underwent extensive testing in real-world scenarios. Photographs of different lighting situations, resolutions, and angles were added to the training data in order to overcome this problem.



## Conclusion

Product recognition through computer vision and machine learning is transforming retail by making product identification more efficient, user-friendly, and sustainable. Combining this technology with price comparison, multilingual support, and recycling initiatives can enhance consumer experiences and promote sustainability. However, challenges such as noisy data, low-light conditions, and data accuracy remain areas for improvement. Amazon Go exemplifies the ambitious potential of cashier-less stores, yet the high costs and privacy concerns present obstacles for wider adoption. Continued advancements in model robustness, data quality, and infrastructure are essential to create a seamless, future-ready retail experience.

# SRS Document

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to present the simulation known as iProVis: Intelligent Product Vision System. This artificial intelligence-based vision system is designed to provide users with the ability to easily identify the cheapest products and locate the nearest availability of these products. The document provides detailed information about the project requirements, addressing identified constraints, and the proposed software functionalities. It also outlines the software technologies utilized in the project's execution, highlighting the positive impacts these technologies have had on overall system performance. Additionally, this Software Requirements Specification (SRS) defines how participants interact with the model-based computer vision system.

### 1.2 Scope of the Project

Today, customers have numerous shopping options, including online and physical stores, each with various shopping methods. The rapid growth of online shopping has made life easier for customers, thanks to software solutions tailored for this space. However, this convenience has also led to challenges such as "getting away from the product itself," with numerous sellers offering the same product. As a result, while e-commerce platforms offer various opportunities for sellers, the process of choosing the best product at the most affordable price has become increasingly difficult for the end user. The price differences across sellers on multiple platforms are often minimal, which complicates the decision-making process.

Despite the advancements made by developed online marketplaces, customers still struggle with finding the "most preferred product at the most affordable price." This challenge becomes more pronounced when shopping in physical stores. Our system is designed to address these difficulties. With our solution, customers can easily determine the best price and the nearest store location for a product they are considering. If the product is unavailable in a physical store, the system will recommend the most suitable online marketplace. The key insight behind this system is that customers instinctively seek "better prices" when they find products in physical stores that seem overpriced. Our AI-powered software, equipped with advanced artificial intelligence models, will provide real-time information on the best prices and locations for these products, ensuring that end users can make purchases under the most optimal conditions.

## 1.3 Glossary

- End User: The person or group who uses a product or service at the final stage.
- E-commerce: The buying and selling of products or services over the internet.
- Physical Stores: Brick-and-mortar locations where products are sold directly to customers
- face-to-face.
- Artificial Intelligence (AI): Computer systems and software designed to mimic human
- intelligence.
- AI Model: An algorithm trained to perform a specific task or make predictions.
- Smartphone: A mobile device that combines phone functions with advanced features like
- internet access and application use.
- API (Application Programming Interface) : Protocols for software interaction, enabling
- communication between systems.
- Camera Interface : Feature allowing users to capture product images for information
- retrieval.
- Database : Structured data storage for user profiles, product data, and transactions.
- Hardware Interfaces : Physical components the application interacts with, like cameras and
- sensors.

## 1.4 Overview of the Document

The second section of this document describes the functionalities of iProVis: Intelligent Product Vision System. The Overall Description provides an overview of the project's objectives, areas of application, and its intended purpose. It explains in detail how the project will assist customers and highlights the features that underscore its importance for the end user. One of the key sections, Development Methodology, further explains the project's construction process and the methods used to guide its development.

# 2. Overall Description

This project aims to revolutionize the retail experience by leveraging computer vision and deep learning technologies to replace traditional barcodes and QR codes for product recognition in sectors such as supermarkets, clothing, food, and electronics. The developed system addresses multiple consumer needs, including effortless access to product price information, comparative price analysis across stores, and multilingual support for accessing detailed product content information.

The product is designed as a standalone mobile application integrated with the iPRoVis system's API for advanced functionality. This integration enables not only product identification but also facilitates an intelligent recycling process by recognizing recyclable packaging types, determining whether the packaging is empty or full, and providing weight details. This functionality ensures efficient sorting and recognition by smart recycling machines, thereby reducing the time and effort required for waste disposal and recycling.

The system is developed with the dual purpose of enhancing the consumer shopping experience and contributing to sustainability goals under a green IT strategy. By promoting environmentally friendly practices and providing advanced features like multilingual support, the product caters to a diverse user base from various cultural and linguistic backgrounds. The project also proposes an innovative framework for recycling by connecting product recognition capabilities to smart recycling systems, encouraging sustainable behaviors among users. This holistic approach not only solves immediate user needs but also aligns with long-term sustainability objectives, making it a transformative solution for both the retail and environmental sectors.

### 2.1.1. Development Methodology

For the development of this project, we have adopted the Agile methodology, specifically leveraging the Scrum framework, which is ideal for dynamic and iterative project needs. Scrum divides the work into smaller, manageable cycles called sprints, each lasting a fixed duration of 2-4 weeks and delivering a potentially shippable product increment. This iterative approach enables flexibility and continuous improvement throughout the development process. In our Scrum-based workflow, tasks are organized and tracked through a Scrum Board, which consists of six distinct phases: Project Backlog, containing all planned tasks; To Do, listing prioritized tasks ready for the sprint; In Progress, showing tasks currently under development; In Review, for tasks being tested and validated; To Deploy, indicating modules ready for integration; and Done, where completed and verified tasks are marked. These phases ensure a structured and transparent workflow.

| Sprint | Phase | Activities | Expected Outcome |
|---|---|---|---|
| Sprint1 | Phase 1 | Planning | Conduct requirement gathering and finalize project scope.<br>Define project backlog, breaking down tasks into epics and user stories.<br>Prioritize tasks in the backlog based on stakeholder input and project goals.<br>Prepare initial wireframes and prototypes for the mobile application.<br>Establish the development environment, tools, and repository structure. |
| Sprint 2 | | | Develop the detailed architecture for the system, including computer vision and API integrations.<br>Finalize UI/UX designs for the mobile application.<br>Plan data acquisition strategies for training the computer vision models.<br>Conduct risk assessments and create mitigation strategies. |
| Sprint 3 | Phase 2 | Training & Testing | Acquire and preprocess data for the computer vision model.<br>Begin training the machine learning models for product recognition.<br>Implement basic functionality for mobile app features, such as user registration<br>and login.<br>Conduct initial unit testing for developed modules. |
| Sprint 4 | | | Optimize and fine-tune the machine learning models to improve accuracy.<br>Develop price comparison and multilingual product detail features.<br>Test API integration with iPRoVis for smart recycling functionality.<br>Perform integration testing for the mobile application. |
| Sprint 5 | | | Implement error handling and security measures in the application.<br>Conduct performance testing for computer vision recognition under varying conditions.<br>Execute user acceptance testing (UAT) with a select group of participants.<br>Collect feedback from UAT and prioritize improvements in the backlog. |
| Sprint 6 | Phase 3 | Go-Live | Finalize all features and resolve outstanding bugs identified during UAT.<br>Deploy the application to a staging environment for final validation.<br>Conduct end-to-end testing to ensure system reliability and scalability.<br>Train administrators on using the system for product data management. |
| Sprint 7 | | | Prepare deployment plans and set up production servers.<br>Launch the application to the public.<br>Monitor system performance post-launch, addressing any issues promptly.<br>Collect initial user feedback and plan for post-launch improvements. |

Each sprint begins with detailed planning, including story points and risk assessments for tasks. The team conducts daily stand-up meetings, limited to 15 minutes, to communicate progress, address potential blockers, and ensure alignment toward sprint goals. Scrum roles are clearly defined to streamline this process: the Product Owner defines and prioritizes project requirements, the Scrum Master ensures adherence to the Scrum framework and facilitates team efficiency, and the Development Team collaboratively works to achieve sprint objectives. Scrum's iterative nature allows the team to adapt to evolving requirements and incorporate stakeholder feedback after each sprint. Delivering a working increment at the end of every sprint ensures early validation and continuous improvement of the product. This adaptability is particularly advantageous for the project, as it accommodates changes while maintaining high- quality outcomes.

| Phase | Backlog | To Do | In Progress | In Review | To Deploy | Done |
|---|---|---|---|---|---|---|
| Planning | Requirement gathering pending | Create prototypes | Set up development environment | | | |
| Planning | Finalize UI/UX designs | Plan data acquisition strategies | Finalize system architecture | | | |
| Training & Testing | Define data preprocessing steps | Acquire and preprocess data | Train machine learning model | | | |
| Training & Testing | Implement error handling | Test API integration | Develop price comparison feature | | | |
| Go-Live | Resolve final UAT feedback | End-to-end testing | Deploy to staging environment | | | |
| Go-Live | Plan for post-launch improvements | Training & Testing | Launch application | | | |

This project aims to transform the retail industry by replacing traditional barcode and QR code scanning with computer vision technology for product recognition. Through its mobile application, users can access product price information, compare prices across stores, and obtain multilingual product details. Additionally, the system integrates with smart recycling through the iPRoVis API, supporting sustainability by enabling automated sorting of recyclable packaging. By incorporating Agile principles and leveraging Scrum, this project ensures a structured yet flexible approach. The methodology facilitates timely, high-quality delivery while addressing innovative and sustainable goals, ultimately enhancing consumer experiences and contributing to environmental responsibility.

## 2.2 User Characteristics

This section defines the distinct user groups who will interact with the system. Users are classified based on their technical knowledge, expectations, and roles, providing clarity on how the system is designed to meet their needs.

### 2.2.1 Participants

Participants represent the general consumer audience who use the mobile application to access product information. Their interaction with the system includes learning about product prices, comparing prices across stores, and accessing detailed product content information.

#### 2.2.1.1. General Consumer Representation

Participants are typical consumers who rely on the system to make informed purchasing decisions by comparing prices and analyzing product details through the mobile application.

#### 2.2.1.2. Ease of Use

Participants require no prior technical knowledge to use the system. A user- friendly interface ensures they can navigate and complete tasks effortlessly.

#### 2.2.1.3. Multilingual Support

Participants can benefit from multilingual features that enable them to view product information in multiple languages, ensuring accessibility across diverse regions.

#### 2.2.1.4. Recycling Integration

The system supports participants in streamlining recycling processes by integrating with smart recycling systems for products designed with sustainable practices.

### 2.2.2 Admin

Admins are the authorized personnel responsible for managing and overseeing the system’s functionality. Their primary roles include updating product data, maintaining system integrations, and ensuring overall performance.

#### 2.2.2.1. System Oversight

Admins monitor and manage the system's operations, ensuring smooth functionality and addressing user feedback effectively.

#### 2.2.2.2. Data Management

Admins are responsible for updating product information, including pricing and descriptions, and addressing user-submitted feedback.

#### 2.2.2.3. Technical Competence

Admins require basic computer skills to operate the system and perform their duties efficiently.

#### 2.2.2.4. System Integration Management

Admins oversee the integration of the iPRoVis API and smart recycling systems, ensuring seamless functionality and accurate data flow.

#### 2.2.2.5. Multilingual Capabilities

Admins must ensure the accuracy and consistency of content across multiple languages and may require familiarity with different languages to manage multilingual features effectively.

# 3. REQUIREMENTS SPECIFICATION

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

**Mobile Application Interface**

- The design should be user-friendly, enabling smooth navigation.
- A camera interface should be integrated for product image capture.
- Display detailed product information, including price, stock status, and multilingual
- content.
- Include a price comparison feature that shows a list of stores and their prices.
- Provide a user profile management section to save preferences and viewing history.

### 3.1.2 Hardware Interfaces

**Camera**

- Integration with the device's camera for image capture.

**Sensors**

- Utilize ambient light sensors to optimize image capture conditions.

**Internet Connectivity**

- Support for Wi-Fi or mobile data connections to facilitate server communication.

### 3.1.3 Software Interfaces

**Operating Systems**

- The app should be compatible with Android platforms.

**APIs**

- Integration with third-party APIs for price comparison and retrieving product information.

**Database**

- The application should connect to a backend database to store user profiles, product data, and transaction history.

### 3.1.4 Communications Interfaces

**Network Protocols**

1. HTTPS should be used for secure communication with the iProViS server.

**Data Formats** *JSON or CSV formats should be employed for data exchange between the mobile app and the server.

## 3.2 Functional Requirements

### 3.2.1 Login/Register Use Case

**Primary Actor:** Customer

**Goal:** Allow users to log into their account or create a new one for personalized features.

**Preconditions:** The app is installed, and the user is either logging in for the first time or after a session has ended.

**Postconditions:** The user gains access to personalized features, including saved searches and activity history.

**Main Flow:**

**1) Open the app:** The user launches the app for the first time or after logging out.

**2) Login/Register prompt:** The app offers options to log in or create a new account (e.g., “Log In” or “Create Account”).

**3) Login:** If the user has an existing account, they enter their credentials (e.g., email or username and password). The app verifies these and grants access to personalized features.

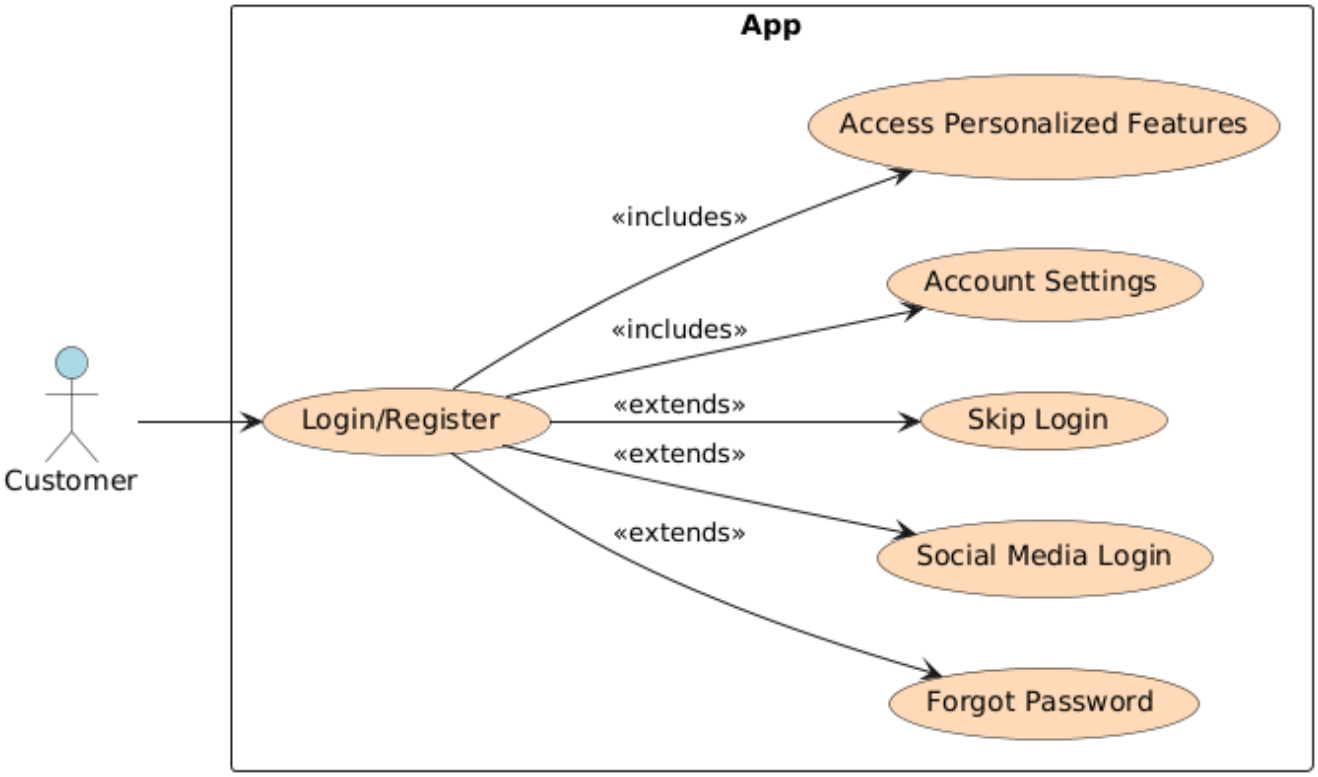
**4) Registration (for new users):** When selecting “Create Account,” the user provides necessary details (email, password, name, shipping address, etc.). Once submitted, the account is created, and the user is logged in.

**5) Account settings:** Users can modify preferences like language or notification settings.

**6) Access personalized features:** After logging in, users can scan products, view saved preferences, and receive targeted offers.

**Alternative Flow:**

- Forgot Password: If the user forgets their password, a reset link is sent via email.
- Social Media Login: Users can log in via social media accounts (e.g., Google, Facebook, or Apple ID).
- Skip Login: Users can skip the login and use the app in limited mode, with an option to log in for a personalized experience.


App
Access Personalized Features
«includes»
Account Settings
«includes»
Customer
Login/Register
«extends»
Skip Login
«extends»
«extends»
Social Media Login
Forgot Password

### 3.2.2 Capture Product Photo for Information Retrieval Use Case

**Primary Actor:** Customer

**Goal:** Capture a product photo to retrieve detailed information.

**Preconditions:** The app is installed, and the product is visible to the camera.

**Postconditions:** The user receives detailed information, including price, stock status, price comparisons, recyclability, and multilingual options.

**Main Flow:**

1. The user opens the app and selects "Scan Product" or "Take Photo."
2. The app prompts the user to take a clear photo of the product.
3. The user captures the photo.
4. The app uses computer vision to identify the product.
5. Product details are displayed:

-Price: The product's current price.

-Stock Availability: Whether the product is in stock or not.

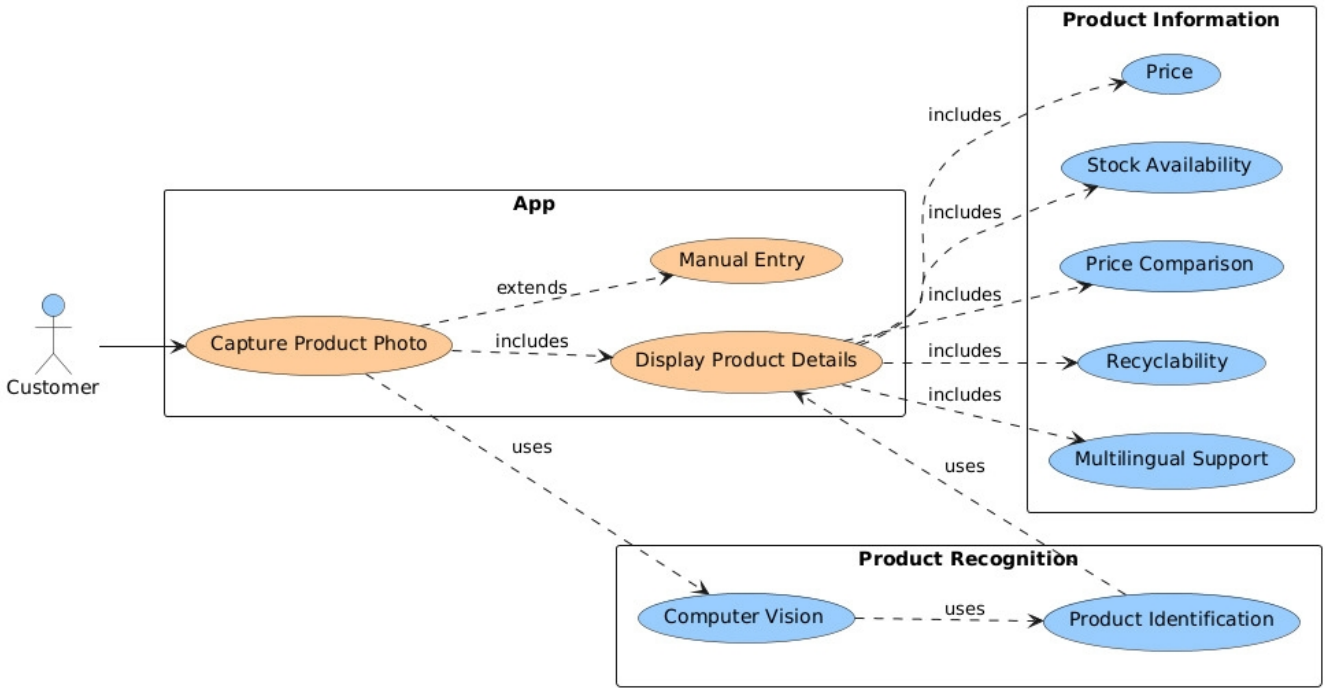
-Price Comparison: Comparisons across stores (online or offline).

-Recyclability: Information about whether the product or packaging is recyclable.

-Multilingual Support: Option to view details in different languages.

**Alternative Flow:**

-If the product cannot be recognized, the app notifies the user and offers manual entry or retry options.



### 3.2.3 Price and Stock Availability Display Use Case

**Primary Actor:** Customer

**Goal:** View price and stock status for a product.

**Preconditions:** The user has taken a product photo, and the app has recognized it.

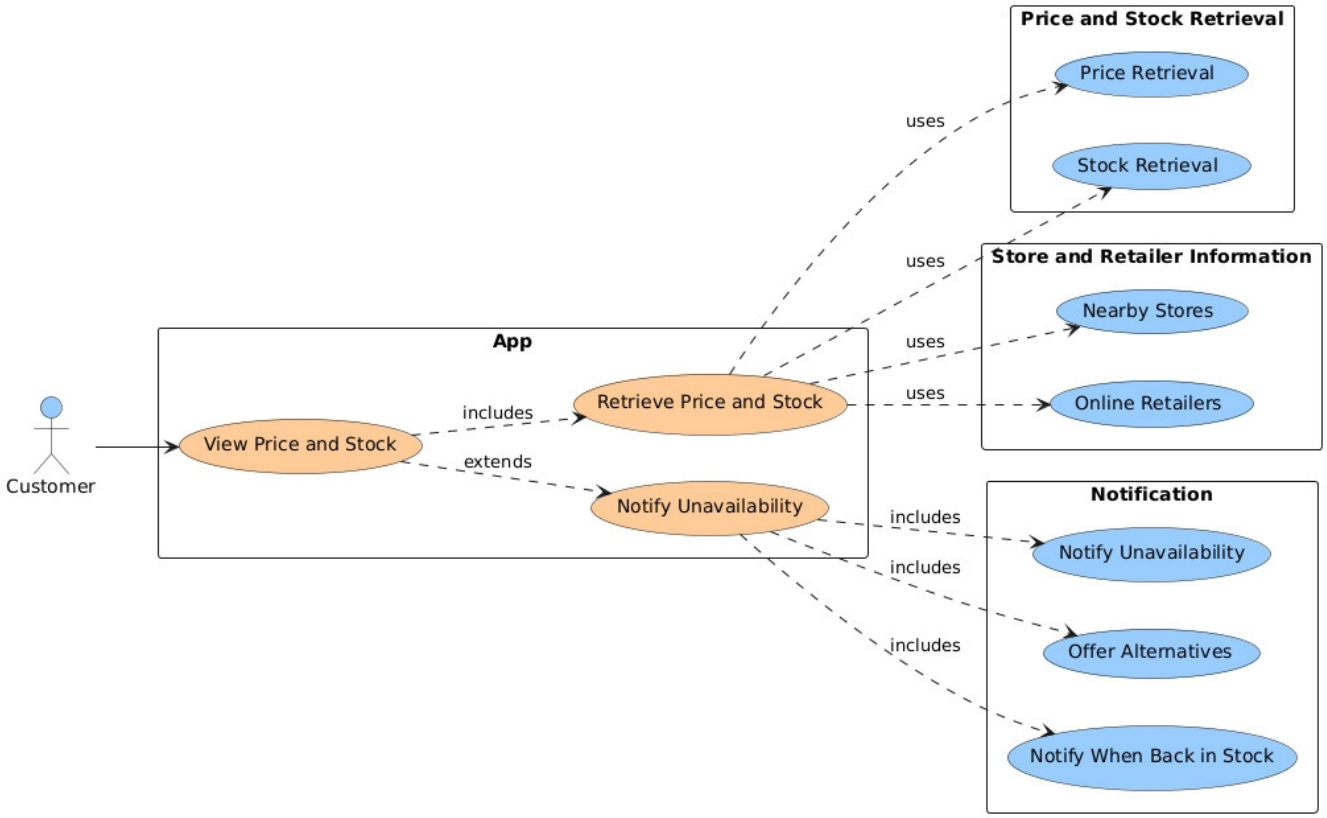
**Postconditions:** The user views current pricing and stock information.

**Main Flow:**

1. After capturing the product photo, the app retrieves and displays the price.
2. It shows stock availability at nearby stores or online retailers.
3. The user can click on a store or retailer to view more details.

**Alternative Flow:**

-If the price or stock is unavailable, the app notifies the user and suggests alternatives or offers to notify them when the product is back in stock.



### 3.2.4 Price Comparison Use Case

**Primary Actor:** Customer

**Goal:** Compare prices for the product across different retailers.

**Preconditions:** The user has taken a product photo, and the app has identified it.

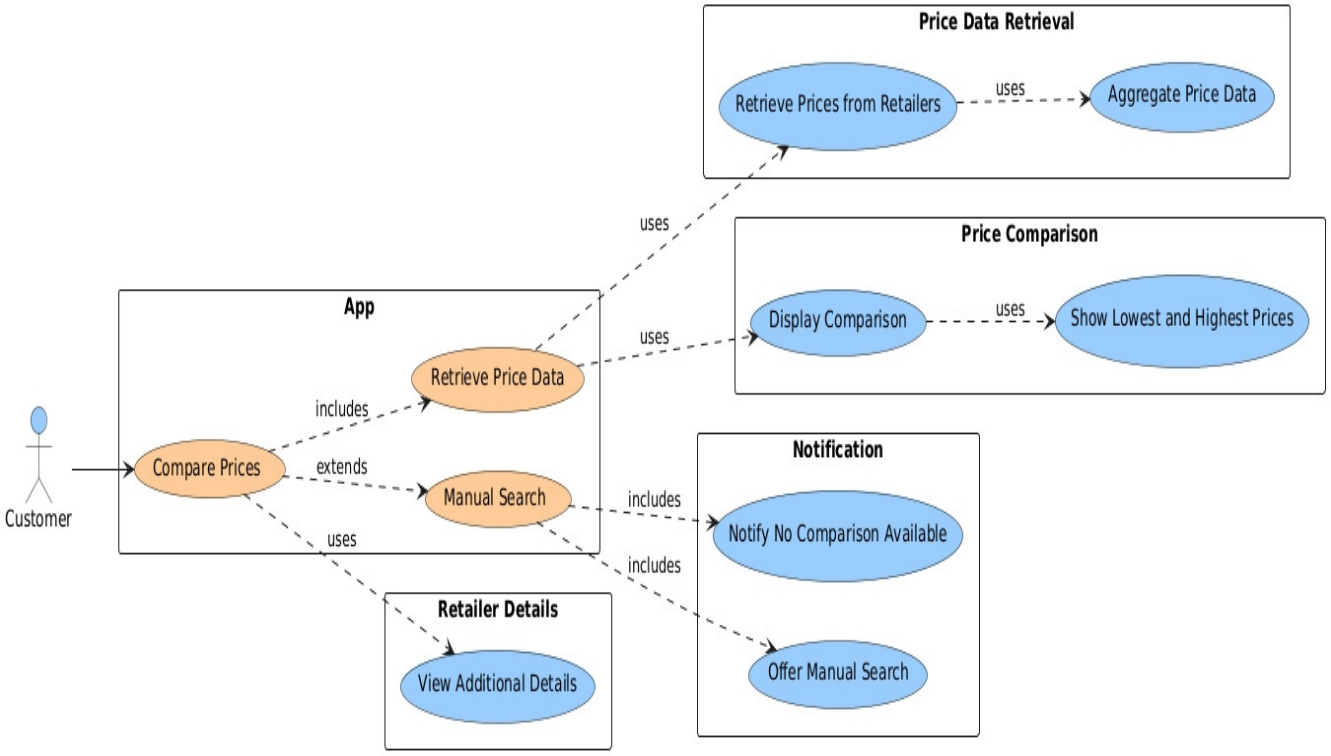
**Postconditions:** The user sees a comparison of prices from various stores.

**Main Flow:**

1. After the product is recognized, the app retrieves price data from multiple retailers.
2. A comparison is displayed, showing the lowest and highest prices.
3. The user can select a retailer to view additional details.

**Alternative Flow:**

-If no comparison is available, the app notifies the user and offers an option for manual search.



### 3.2.5 Recyclability Information Display Use Case

**Primary Actor:** Customer

**Goal:** Display recyclability information about the product.

**Preconditions:** The user has taken a product photo, and the app has recognized it.

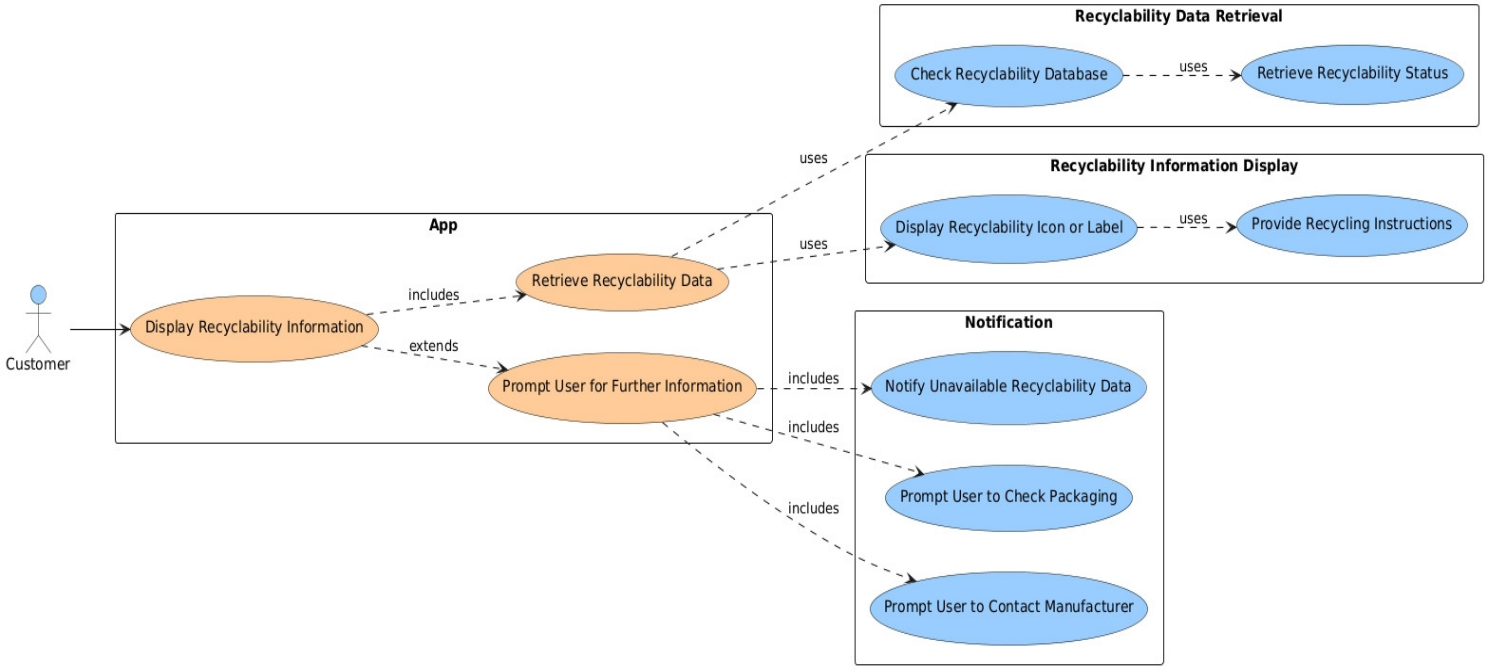
**Postconditions:** The user learns if the product or packaging is recyclable.

**Main Flow:**

1. After product recognition, the app displays recyclability information.
2. An icon or label indicates whether the product is recyclable (e.g., "Recyclable," "Not Recyclable," or "Check local guidelines").
3. If recyclable, the app provides additional recycling instructions.

**Alternative Flow:**

-If recyclability data is unavailable, the app prompts the user to check the packaging or contact the manufacturer for further information.



### 3.2.6 Multilingual Support for Product Information Use Case

**Primary Actor:** Customer

**Goal:** View product information in the user's preferred language.

**Preconditions:** The user has taken a product photo, and the app has identified it.

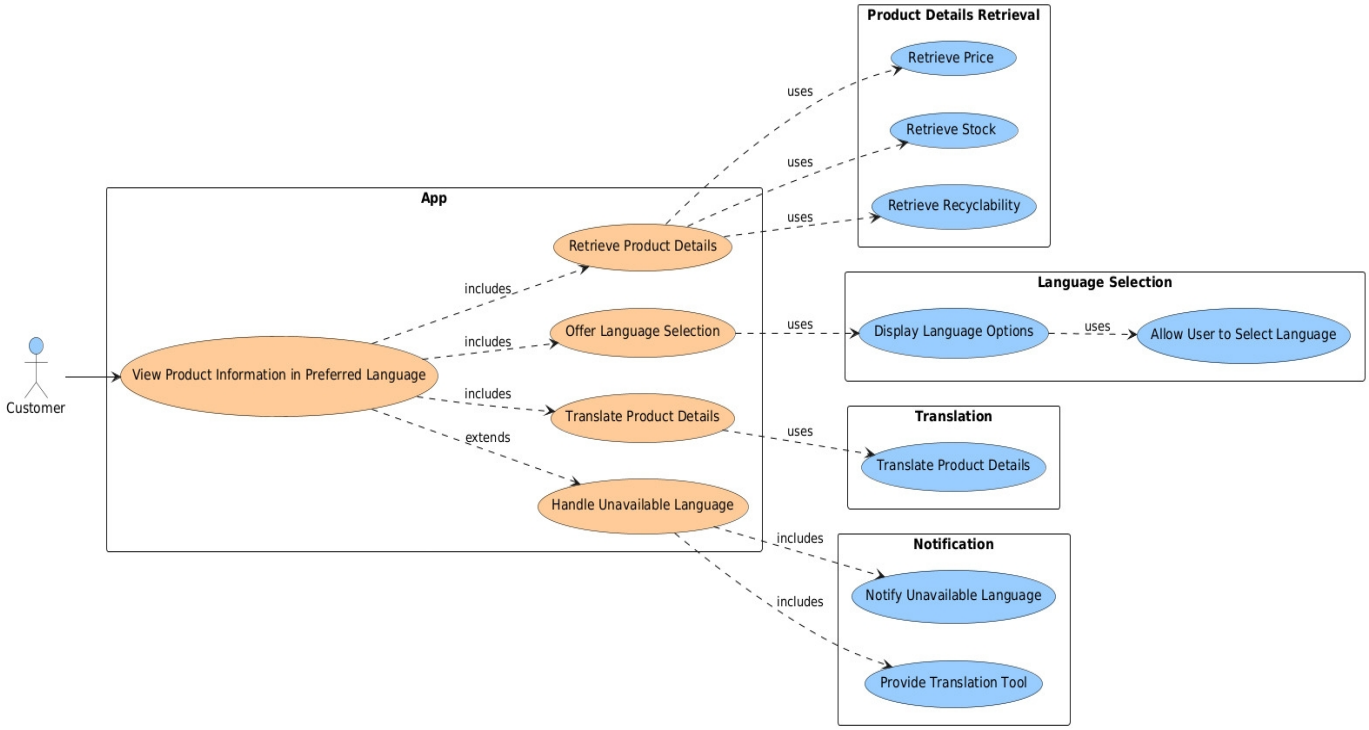
**Postconditions:** The product details are displayed in the selected language.

**Main Flow:**

1. Once product details (e.g., price, stock, recyclability) are shown, the app offers an option to switch to another language.
2. The user selects a preferred language (e.g., English, Spanish, French).
3. The app translates all details into the chosen language.

**Alternative Flow:**

-If the selected language is unavailable, the app informs the user or provides a translation tool.



### 3.2.7 Feedback and Additional Product Information Use Case

**Primary Actor:** Customer

**Goal:** Access additional information, such as reviews and ratings.

**Preconditions:** The user has taken a product photo, and the app has recognized it.

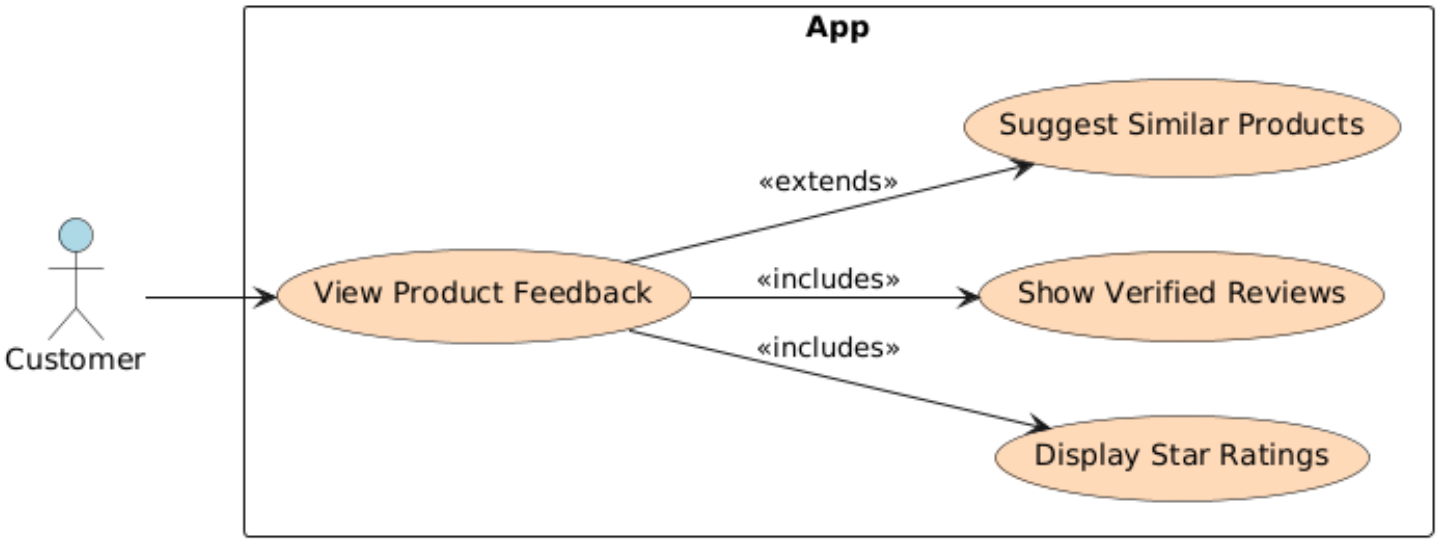
**Postconditions:** The user gains insights into the product via reviews and ratings.

**Main Flow:**

1. After displaying product details, the app shows options to view customer reviews and ratings.
2. The app may display star ratings, verified reviews, and product demonstration videos.
3. The user can scroll through reviews or click on ratings for more details.

**Alternative Flow:**

-If reviews or ratings are unavailable, the app notifies the user and suggests similar products.



### 3.2.8 Notification of New Deals or Discounts Use Case

**Primary Actor:** Customer

**Goal:** Receive notifications about available deals or discounts.

**Preconditions:** The user has taken a product photo, and the app has recognized it.

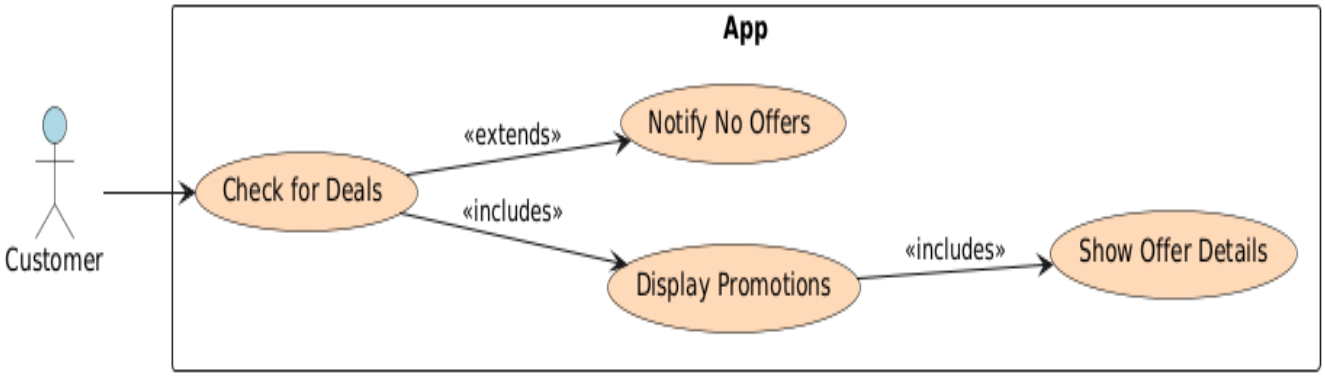
**Postconditions:** The user is notified of any available deals.

**Main Flow:**

1. The app checks for any ongoing promotions on the product.
2. It displays any available deals (e.g., discounts, coupon codes, or bundle offers).
3. The user can click on the offer for more details or apply it to their purchase.

**Alternative Flow:**

-If no discounts are available, the app informs the user that there are no current offers.



### 3.2.9 Handling Unrecognized Products Use Case

Primary Actor: Customer

Goal: Assist users when the app fails to recognize a product.

Preconditions: The user has taken a product photo, but the app can't identify it.

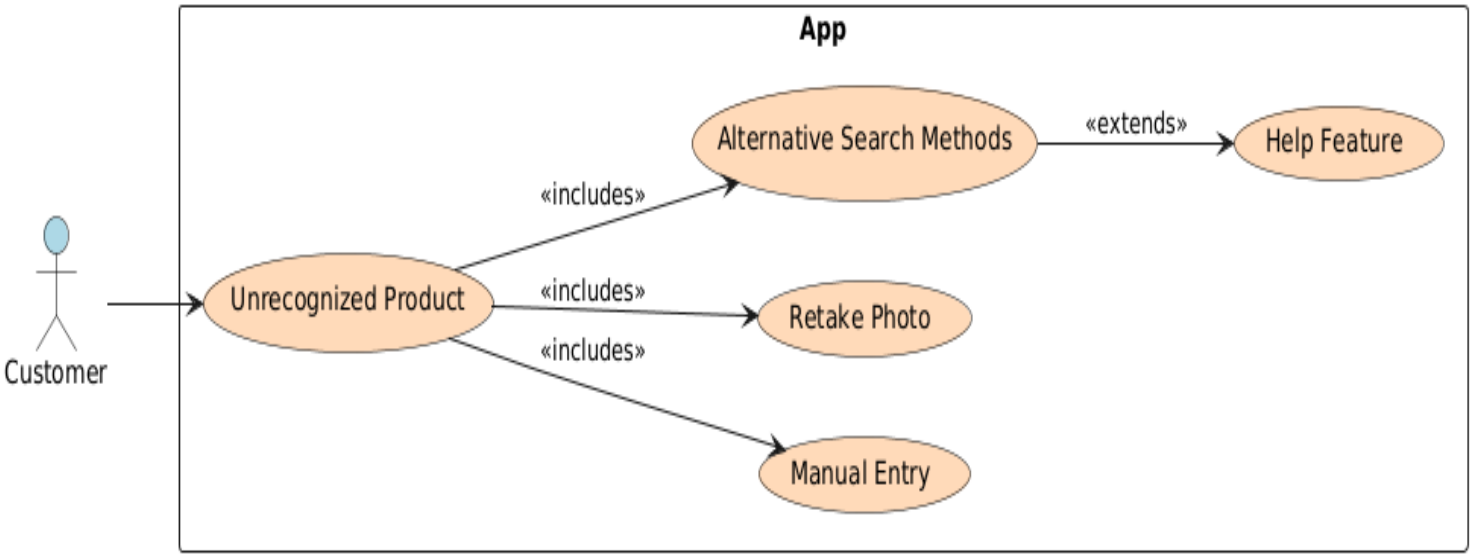
Postconditions: The user is given an alternative method to find the product information.

**Main Flow:**

1. If the app fails to recognize the product, it notifies the user.
2. The user can manually enter the product name or retake the photo.
3. Other search methods (e.g., barcode scanning or uploading an image) are provided.

**Alternative Flow:**

-If manual search options fail, the app offers a "Help" feature.



## 3.3 Performance Requirements

The application must deliver an efficient, reliable, and responsive user experience, even under heavy usage. Here are the key performance considerations:

### 3.3.1 Response Time

**Product Recognition:** The app should recognize a product within 2 seconds, assuming optimal lighting and focus conditions.

**Price and Stock Retrieval:** Product price and stock data should appear within 3 seconds of recognition.

**Price Comparison:** The comparison data should load within 5 seconds of retrieving product details.

### 3.3.2 System Throughput

**Concurrent Users:** The app must support at least 100 concurrent users without significant performance issues.

**Data Handling:** The backend should handle up to 10,000 transactions per hour without noticeable degradation.

### 3.3.3 Availability and Reliability

**Service Availability:** The app must be available 99.9% of the time, with downtime only for planned maintenance.

**Error Handling:** The system should display appropriate error messages and automatically retry failed processes when needed.

### 3.3.4 Scalability

**User Growth:** The backend should scale horizontally to accommodate growing user numbers and app traffic.

**Database Scaling:** The database must scale efficiently as the number of products, transactions, and user profiles increases.

### 3.3.5 Latency

**Network Latency:** Average network latency should remain under 1 second for most operations.

**Image Processing Latency:** Product recognition should occur within 2 seconds after image capture.

### 3.3.6 Resource Utilization

**Memory Usage:** The app should not exceed 200MB of RAM during normal operations.

**Battery Consumption:** The app should consume no more than 10% battery per hour during regular use.

**Data Consumption:** The app should use no more than 50MB of data per session for product searches and comparisons.

### 3.3.7 Load Balancing and Redundancy

**Server Load Balancing:** Load balancing mechanisms must ensure even traffic distribution.

**Failover Mechanism:** Critical services should have a failover mechanism to maintain high availability.

### 3.3.8 Caching

**Frequent Data Caching:** Frequently requested data should be cached to improve performance. The cache should refresh at regular intervals to ensure up-to-date information.

These performance specifications are designed to ensure the app delivers a fast, reliable, and efficient experience under varying conditions, while being scalable, resilient, and responsive to user demands.

## 3.4 Software System Attributes

### 3.4.1 Portability

The app should be usable across various devices with different screen sizes and resolutions.

### 3.4.2 Performance

Product recognition should happen within 2 seconds under optimal conditions.

The app should support 100 concurrent users without degradation.

### 3.4.3 Usability

The app should have an intuitive interface with minimal learning required. Accessibility features must also be provided.

### 3.4.4 Adaptability

The app should adapt to different lighting conditions to ensure optimal image recognition.

### 3.4.5 Scalability

The backend should be able to scale as the user base and product data grow.

## 3.5 Safety Requirements

The app must comply with data protection regulations (e.g., GDPR) and ensure user data privacy. Mechanisms should be in place to prevent unauthorized access to sensitive information.

# Software Design Description

## 1. INTRODUCTION

### 1.1 Purpose

The primary purpose of this Software Design Description (SDD) document is to define and detail the design, structure, and implementation framework of the iProVis: Intelligent Product Vision System. This system is an innovative, artificial intelligence-powered solution developed to assist users in identifying the most cost-effective products and locating their nearest availability, whether through physical retail outlets or online marketplaces. By addressing challenges faced by modern consumers in navigating the complexities of both online and offline shopping, the iProVis system offers a seamless and user-centric approach to optimizing purchasing decisions.

The document serves as a formal and detailed guide for all stakeholders involved in the project, including system architects, developers, quality assurance teams, and end-users. It articulates the objectives and scope of the iProVis system, ensuring alignment with the overall vision of providing a cutting-edge, real-time shopping assistant. Furthermore, it delineates the technical and functional requirements, highlighting the system's capabilities to enhance user convenience and satisfaction through advanced features like real-time price comparison, store location identification, and product availability verification.

The iProVis system aims to address the growing difficulties faced by consumers in the modern retail landscape. While e-commerce has revolutionized the way people shop by providing diverse options and convenience, it has also introduced complexities, such as the inability to directly interact with products and the challenge of comparing multiple sellers offering similar items at slightly varying prices. These difficulties are exacerbated in physical retail settings, where locating the most competitively priced product in nearby stores becomes a time-intensive task. The iProVis system is specifically designed to mitigate these issues by integrating sophisticated artificial intelligence models that analyze and compare data from multiple sources, including physical store inventories and online platforms.

This document offers an extensive and detailed examination of the iProVis system's design, encompassing all essential aspects to provide a comprehensive understanding of its development and implementation. The critical components covered in this document are systematically outlined to ensure clarity and cohesion in presenting the system’s capabilities, architecture, and functionality. These elements are as follows: The User-Centric Features of the iProVis system are meticulously designed to address the needs and expectations of modern consumers. These features include real-time price comparison capabilities, allowing users to instantly identify the most affordable product options across various platforms and locations. The system also incorporates a nearest store locator, which provides users with the exact location of physical retail outlets where desired products are available. Additionally, the platform integrates recommendations for the most suitable online marketplaces in

cases where products are unavailable locally. Furthermore, the system leverages camera-based product recognition to enable effortless and intuitive interaction, allowing users to capture product images for instant information retrieval and comparison.

The Technological Framework of the iProVis system is built on advanced and robust methodologies to ensure seamless functionality and superior performance. This framework includes the deployment of state-of-the-art artificial intelligence (AI) models that facilitate accurate predictions and real-time data analysis. The use of application programming interfaces (APIs) ensures effective communication between system components and interoperability with external platforms, enhancing the system's adaptability and efficiency. Additionally, the system relies on well-structured database management to organize and store critical information, including user profiles, comprehensive product catalogs, and transaction histories. These databases are optimized for performance, ensuring rapid and reliable access to the information required for informed decision-making. The Hardware Interfaces of the system are designed to enhance usability and ensure smooth interaction with the physical components. Key elements include smartphone cameras and sensors that allow users to capture high-quality images of products for identification and analysis. These interfaces are seamlessly integrated into the system, ensuring that users can leverage the full potential of the platform without encountering technical complexities. By incorporating these components, the iProVis system guarantees a user-friendly and efficient experience that aligns with the practical needs of its audience.

Finally, the System Constraints and Assumptions are thoroughly analyzed to provide a realistic foundation for the system's design and implementation. This includes identifying limitations such as data availability, hardware dependencies, and potential scalability challenges. Assumptions regarding user behavior, market trends, and technological advancements are carefully considered to guide the design process and ensure that the system remains adaptable and relevant in a dynamic retail environment. By addressing these factors, the iProVis system is poised to deliver optimal performance while maintaining flexibility for future enhancements.

Together, these elements form the cornerstone of the iProVis system, ensuring its capacity to deliver a transformative shopping experience for users while addressing the complexities of modern retail practices. This document aims to provide a clear and detailed blueprint for the system's development and serves as a critical resource for all stakeholders involved in the project.

## 1.2 Scope

Finally, the System Constraints and Assumptions are thoroughly analyzed to provide a realistic foundation for the system's design and implementation. This includes identifying limitations such as data availability, hardware dependencies, and potential scalability challenges. Assumptions regarding user behavior, market trends, and technological advancements are carefully considered to guide the design process and ensure that the system remains adaptable and relevant in a dynamic retail environment. By addressing these factors, the iProVis system is poised to deliver optimal performance while maintaining flexibility for future enhancements.

Together, these elements form the cornerstone of the iProVis system, ensuring its capacity to deliver a transformative shopping experience for users while addressing the complexities of modern retail practices. This document aims to provide a clear and detailed blueprint for the system's development and serves as a critical resource for all stakeholders involved in the project.

## 1.3 Glossary

The following glossary provides definitions for key terms used throughout this Software Design Document (SDD), offering clarity on the concepts and components involved in the development and implementation of the Intelligent Product Vision System (iProViS).

| Term | Definition |
|---|---|
| End User | The person or group who interacts with the system at the final stage, utilizing its functionalities to make purchasing decisions. |
| E-commerce | The buying and selling of products or services over the internet through online platforms and marketplaces. |
| Physical Stores | Brick-and-mortar retail locations where products are sold directly to customers in a face-to-face environment. |
| Artificial Intelligence (AI) | Computer systems and software designed to replicate human cognitive functions such as learning, problem-solving, and decision-making. |
| AI Model | A machine learning algorithm that has been trained to perform specific tasks, such as price prediction, recommendation, or image recognition. |
| Smartphone | A mobile device that combines basic telecommunication functions with advanced features such as internet connectivity, applications, and cameras. |
| API (Application Programming Interface) | A set of protocols and tools that enable communication between different software applications, allowing them to interact and exchange data. |
| Camera Interface | A feature in the system that enables users to capture images of products, which are then analyzed for product recognition and price comparison. |
| Database | A structured system for storing and managing data, including user profiles, product information, and transaction histories, ensuring efficient data retrieval and management. |
| Hardware Interfaces | Physical components such as cameras, sensors, or other devices that allow the system to interact with the real world to gather data and provide functionalities. |
| Scrum | An Agile methodology used for iterative project management, consisting of sprints (short development cycles) to manage tasks and deliver continuous improvements. |
| Sprint | A fixed-duration development cycle in Scrum, typically lasting between 2 to 4 weeks, where a set of tasks or features is developed and delivered as a potentially shippable product increment. |

## 1.4 Overview of Document

This document is structured to provide a comprehensive understanding of the Intelligent Product Vision System (iProVis) project, detailing its design, implementation, and technical framework. The following sections outline the content and objectives of each chapter within this document:

**Section2 : ArchitecturalDesign** This section focuses on the development phase of the iProVis system. It presents the system's overall architecture, including a class diagram that illustrates the core components and their relationships. Additionally, the architectural design addresses key elements such as actors, exceptions, basic sequences, priorities, pre-conditions, and post-conditions. An activity diagram of the scenario generator is also included, showing how different scenarios are triggered and managed within the system.

**Section3 : UseCaseRealization** In this section, we explore the realization of use cases identified in the Software Requirements Specification (SRS) document. A block diagram is provided, which visually represents the system architecture based on these use cases. This diagram helps to clarify how different components of the system interact to fulfill user requirements, showcasing the practical application of each use case in the system's operation.

**Section4 : Environment** This section describes the environmental context in which the iProVis system operates. It includes sample frames from the prototype, illustrating the user interface and interactions within the system. Additionally, this section outlines the various scenarios in which the system is used, demonstrating its functionality and responsiveness in different real-world environments. Each section contributes to a detailed and technical understanding of the iProVis system, providing the necessary insights for both development and future enhancements.

## 1.5 Motivation

We are a group of enthusiastic Computer Engineering students with a strong interest in Artificial Intelligence (AI) and Computer Vision. Throughout our academic journey, we have developed a deep passion for exploring how AI can be leveraged to solve real-world problems, and how computer vision technologies can enhance user experiences and streamline processes. This project, iProVis: Intelligent Product Vision System, represents a perfect opportunity for us to apply our knowledge, skills, and creativity in a practical context.Our motivation for taking on this project stems from our desire to tackle the challenges faced by consumers in the modern shopping landscape. We recognize the growing complexity of online and physical shopping, where customers struggle to identify the best products at the most affordable prices amidst overwhelming choices. The application of AI and computer vision to simplify this process aligns with our interests and provides us with an exciting avenue to delve deeper into these technologies.

# 2. ARCHITECTURE DESIGN

## 2.1 Software Design Approach

In this graduation project for which we have applied to Tübitak, everything must be very professional. AGILE was decided on for this, which includes huge companies and firms being able to control the processes they manage for the product they are going to develop when developing any product. So we have divided this long-term process of our project into equal days and weeks. The methodology we used had many features.

Among these features, the first one is the harmony of the creation process of the product within the framework of this method. If it needs further explication, in order to enable satisfaction of customers and competitive advantages, adaptation to the shifting requirements must be ensured by quick and frequent delivery of working software. The principle we implemented in our venture in this section is to continuously upgrade the features of the project to the next level and manage the process accordingly. Collaboration between developers and business people should never be overlooked; motivated teams with confidence have to work in the right environment with the necessary support. We in our project had been committed to this very principle and were in constant collaboration. In the academic environment, outdoors, and in any other social framework, placing our capability of being a team among the main parameters that motivated the process took first place, and most of all, us.

Firstly, we discussed how our development environments were in line with the purpose of our project and how we would adapt to these environments as a team. The most suitable working method for our business was to support our project as a team through remote connection where we can use Agile and Scrum methods. We had meetings 2 days a week, especially with specific sprint durations. These meetings took 2 days a week and at least 30 minutes. We distributed Agile-specific roles to each other as a team and held our meetings accordingly.

| Month | Sprint | Goals and Tasks | Deliverables |
|---|---|---|---|
| Month 1 | Sprint 1 (2 weeks) | Planning and Research:<br>- Finalize requirements.<br>- Write user stories.<br>- Research Computer Vision libraries (e.g., OpenCV, TensorFlow) and price comparison APIs. | Documents: Requirements, user stories, research report. |
| | Sprint 2 (2 weeks) | Technical Design:<br>- Prepare system architecture and data flow diagrams.<br>- Design initial structure for Computer Vision module and price comparison API. | Documents: System design, module designs. |
| Month 2 | Sprint 3 (2 weeks) | Computer Vision Development Start:<br>- Develop a basic model to recognize product photos.<br>- Design labeling and classification processes. | Code: Working basic photo recognition model. |
| | Sprint 4 (2 weeks) | Price Comparison API:<br>- Integrate price comparison API.<br>- Develop functionality to retrieve nearby store details using test data. | Code: Working price comparison module. |
| Month 3 | Sprint 5 (2 weeks) | Integration and Testing:<br>- Integrate the Computer Vision module with the API.<br>- Test data flow between modules. | Code: Integrated version of core modules. |
| | Sprint 6 (2 weeks) | Start Mobile or Web Application Development:<br>- Design user interface (UI).<br>- Choose frontend technologies (e.g., React Native, Flutter). | Prototype: Simple user interface. |
| Month 4 | Sprint 7 (2 weeks) | Comprehensive Testing and Advanced Features:<br>- Improve the model with a larger dataset.<br>- Develop mechanisms to collect real-time feedback from users. | Code: Optimized Computer Vision model. |
| | Sprint 8 (2 weeks) | Performance Optimization:<br>- Optimize for faster results. | Code: Performance-optimized system. |

| Month | Sprint | Goals and Tasks | Deliverables |
|---|---|---|---|
| | | - Reduce latency in API and backend integration. | |
| Month 5 | Sprint 9 (2 weeks) | Beta Testing:<br>- Conduct beta testing with users.<br>- Gather user feedback. | Beta Deliverables: Beta version and test reports. |
| | Sprint 10 (2 weeks) | Bug Fixes and UX Improvements:<br>- Fix bugs identified during beta testing.<br>- Improve UI/UX. | Code: More stable version. |
| Month 6 | Sprint 11 (2 weeks) | Final Testing:<br>- Conduct application security checks.<br>- Perform system stability tests. | Test Deliverables: Security and stability reports. |
| | Sprint 12 (2 weeks) | Launch and Promotion:<br>- Release the app to stores or users.<br>- Promote on social media and online platforms. | Final Deliverables: Live product and promotional materials. |

*Figure 1 All Sprints of the Project on the Board*

Those meetings we held, they meant so much to us. The most important keyword for our project was quality. To make quality software, the process must be effective as well. In this context, we adapted to the whole process according to a visual table and we acted according to this table, and our meetings and work were always in order. (Figure 1)

Developing our software, we considered the charts that we had seen in our previous courses and which all software and sectoral companies use. In this respect, we wanted to follow a hierarchy and flow in the development of our product. We made a Class diagram which shows the technical information of our software, how it is going to be developed, what is going to be used.

### 2.1.1 Class Diagram

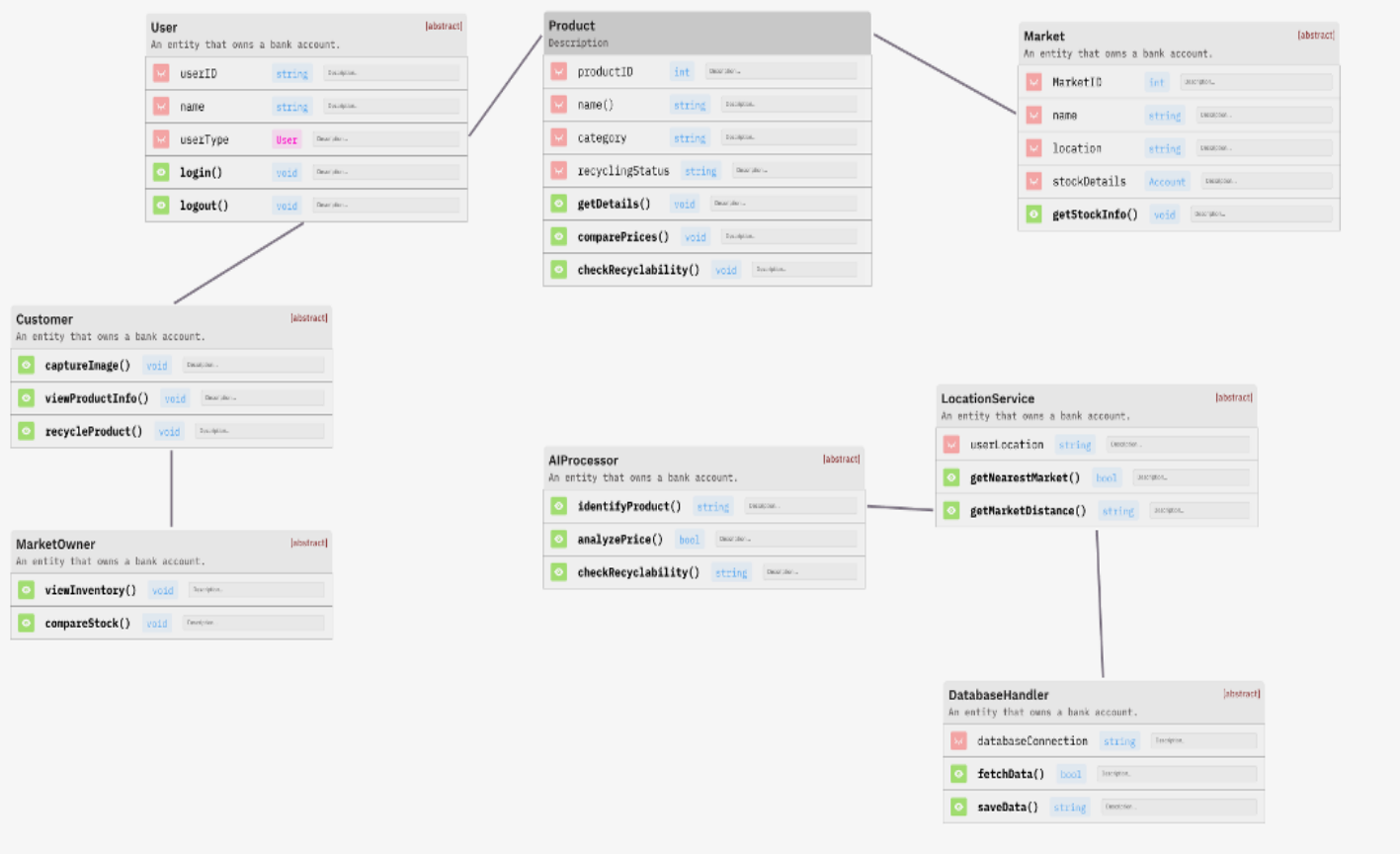



*Figure 4 Class Diagram of iProViS project*

*Figure 4* displays information about connections between the systems within the out project.

The iProVis project represents an innovative software system focused on two kinds of users: the End User and the Market Owner. The proposed system, based on the use of computer vision techniques combined with artificial intelligence techniques, allows users to be informed about products, make price comparisons, and investigate the possibility of recycling. The project is built around the following core components and the relationships between them.

The User class is the basic unit in the system, representing all users; it is divided into two subclasses: EndUser and MarketOwner. The methods in the User class let the users log on or log off the system. The EndUser class will enable the individual users to view the product information, price comparison, and recycling eligibility of the products. The MarketOwner class will provide the facility to the market owners to view their respective stock information and compare the stock status of other markets.

Product class maintains all the information of the product. Each product has a unique ID number (productID), name, category, price and recycling eligibility.

This class includes functions like providing the details of the product, price comparison and checking recycling eligibility. The Market class models market information and stock details. Grocery stores are located at a specific location and keep track of the availability of products. The Market class includes methods to provide information about the stock in grocery stores and has an "aggregation" relationship with the Product class.

AIProcessor: This class is used to run the computer vision and AI operations of the system. It recognizes the products in the photos and checks prices and recycling eligibility of each product. This class has a function of using location to find closest grocery stores to the user, it calculates distances in between different grocery stores as well. This class for user experience enhancement is linked to User and Market class.

The DatabaseHandler class, finally, handles all the system's data storage and retrieval operations. This class is representative of the database connection and ensures that data utilized by other components of the system is retained.

## 2.2 Architecture Design of iProViS

### 2.2.1 Profile Management

Summary: This module allows the user to access product information by taking a photo of a product. The system provides the price, availability, recyclability and multilingual support options.

Actor: User

Precondition: The user must have run the application and the product must be in the camera's field of view.

Basic Sequence:

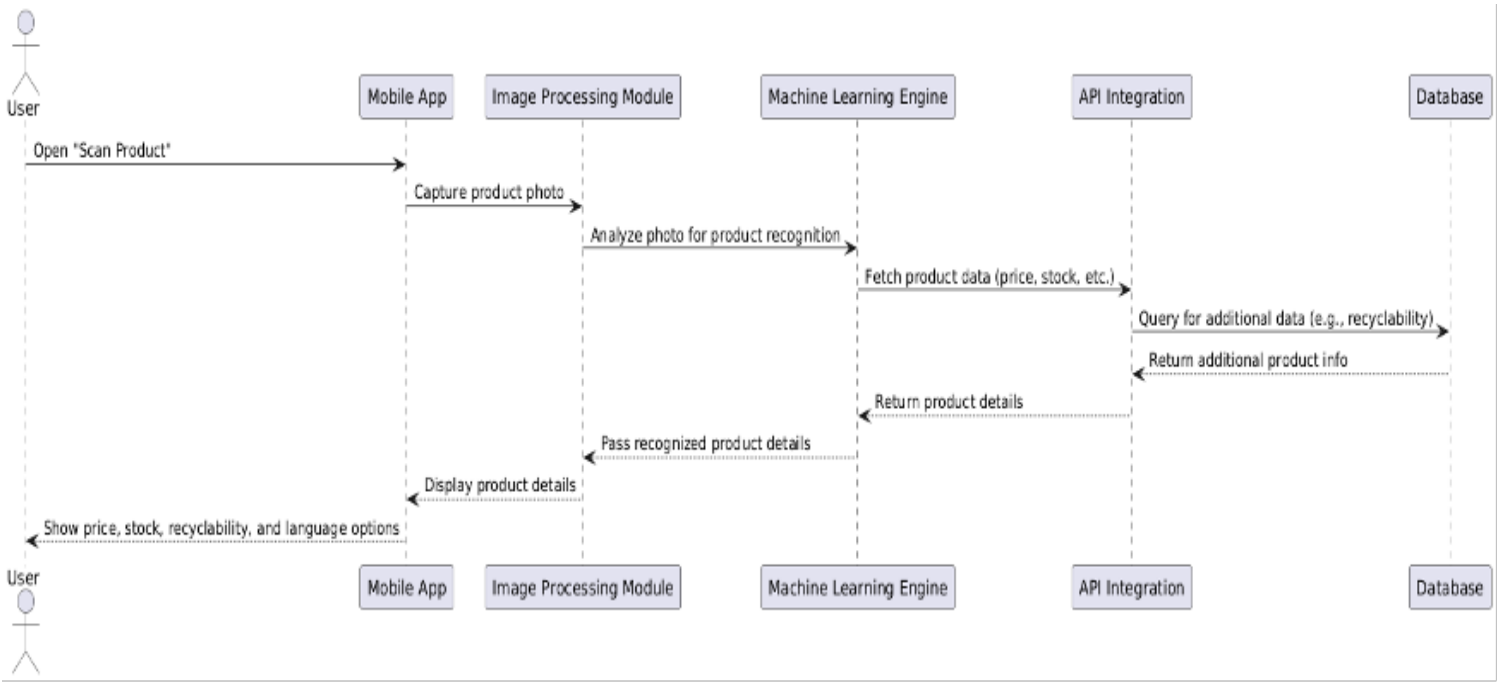
1. The user selects “Capture Image” or “Take Photo” to take a product photo.
2. he system optimizes the camera for the user to focus on the product.
3. The user takes a photo of the product.
4. The system analyzes the captured image and recognizes the product information.
5. Product information is shown on the screen:
6. Price information
7. Stock status
8. Price comparison options
9. Recyclability information
10. Multilingual support
11. The user can review product information or take another photo for another product

Exception:

1. If the system fails to recognize the product, the user will be notified and given the option to take another photo or log in manually.
2. If there is no internet connection, the system uses a limited database in offline mode.

Post Conditions: The user received detailed information about the identified product.

Priority: High



### 2.2.2 Price Comparison Menu

Summary: This module compares the prices of the defined product in different stores and allows the user to find the best price.

Actor: User

Precondition: The product must be identified by the system.

Basic Sequence:

1. The system pulls the prices of the defined product in different stores from the server.
2. The prices are displayed to the user in the form of a comparison table.
3. The user can select a store from the list and get detailed information.
4. The user is redirected to the website of the selected store or receives the store's location information.

Exception:

1. If no price information is found, the system suggests alternative products to the user.
2. If the API connection fails, the user is shown an error message.

Post Conditions: The user accessed price information from different stores and found the most suitable option.

Priority: High

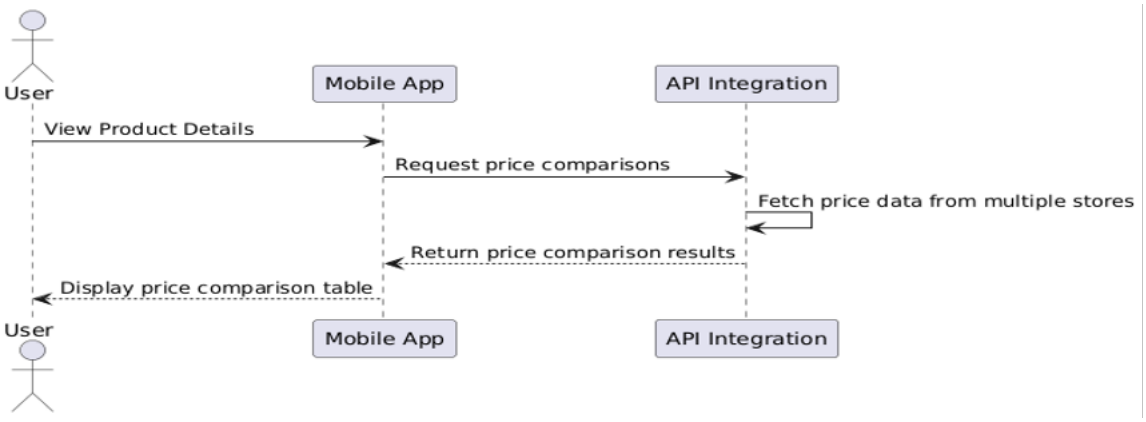



*Figure 2: Price Comparison Sequence Diagram*

### 2.2.3 User Profile Management

Summary: This module manages user profiles. The user can view and edit account information or set personal preferences.

Actor: User

Precondition: The user must be logged in to the system.

Basic Sequence:

1. The user selects “Profile” to view their profile.
2. The system displays the user information on the screen.
3. The user can edit their information by selecting “Edit”.
4. The user can change language or notification preferences.
5. The user can exit the profile by saving the changes.

Exception: If the database connection fails, the user will be shown an error message.

Post Conditions: The user profile has been successfully updated. Priority: Medium

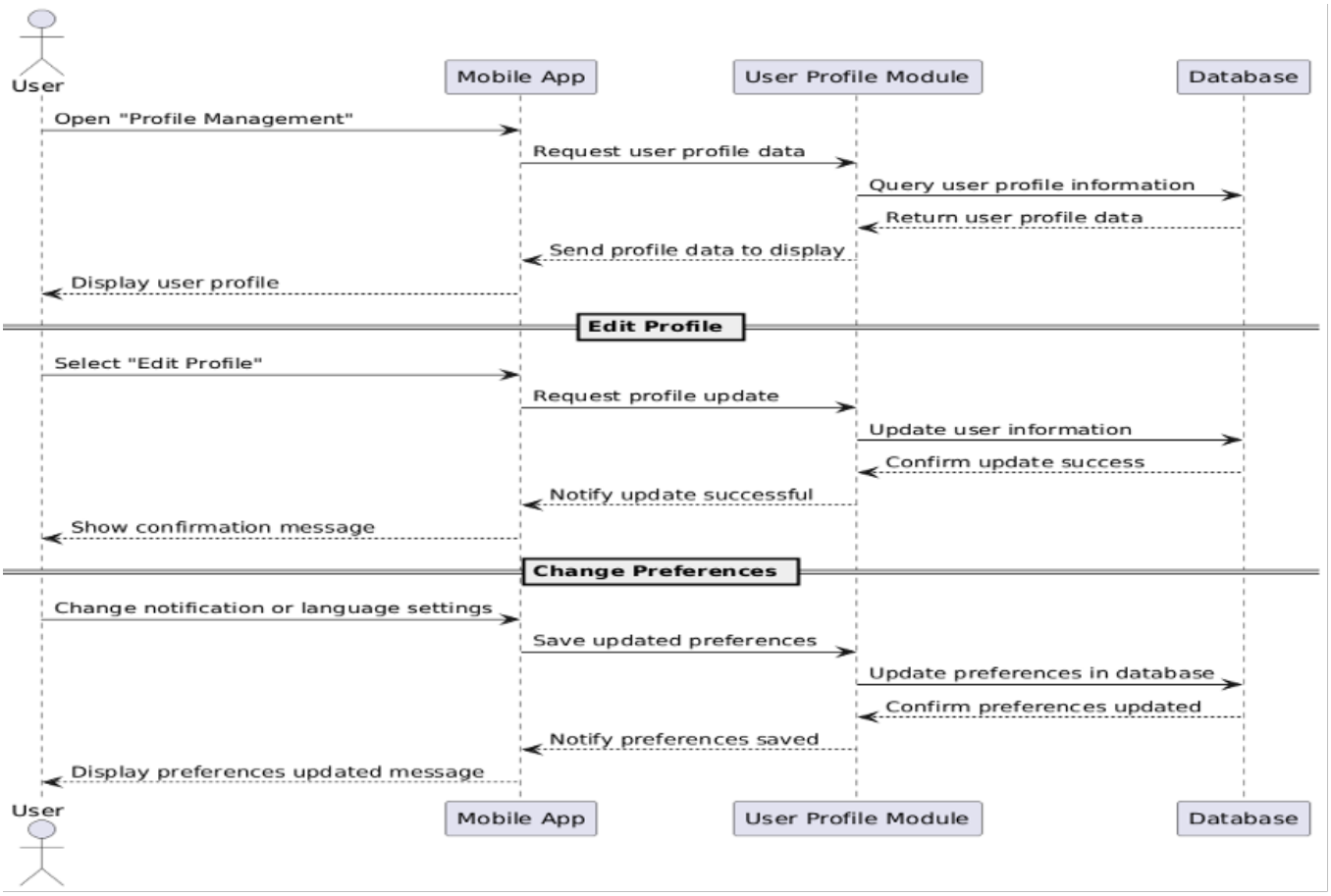



*Figure 3: User Profile Management Sequence Diagram*

### 2.2.4 Notification System

Summary: This module sends notifications to users about discounts and promotions on defined products.

Actor: User

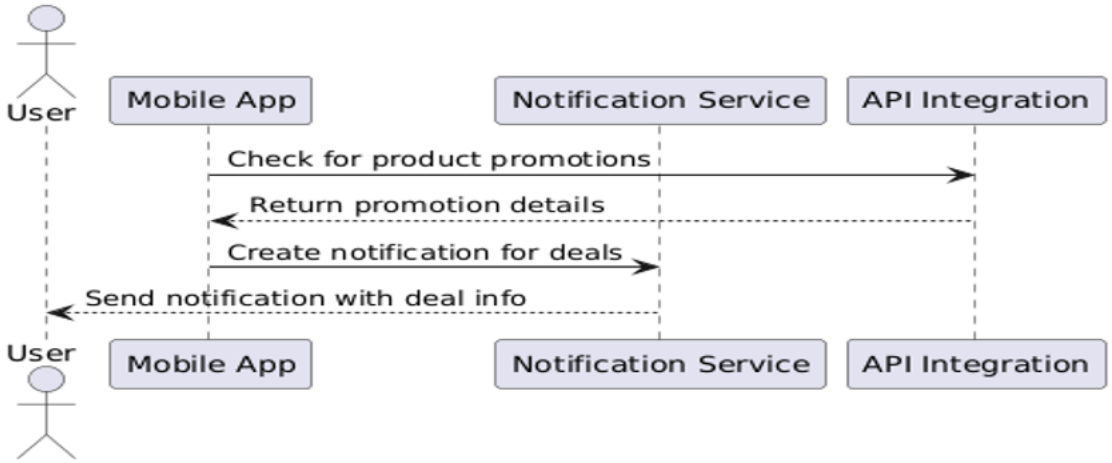
Precondition: The user must have taken a product photo and the system must have identified the product.

Basic Sequence:

1. The system checks whether the identified product is on sale.
2. Notification containing discount or promotion information is sent to the user.
3. The user can get detailed information by clicking on the notification.
4. The user is directed to the store to apply the discount.

Exception: If no discount or promotion is found, the user will be notified that no offer is available. Post Conditions: The user received information about discount opportunities for the relevant product.

Priority: Medium



## 2.3 Activity Diagram

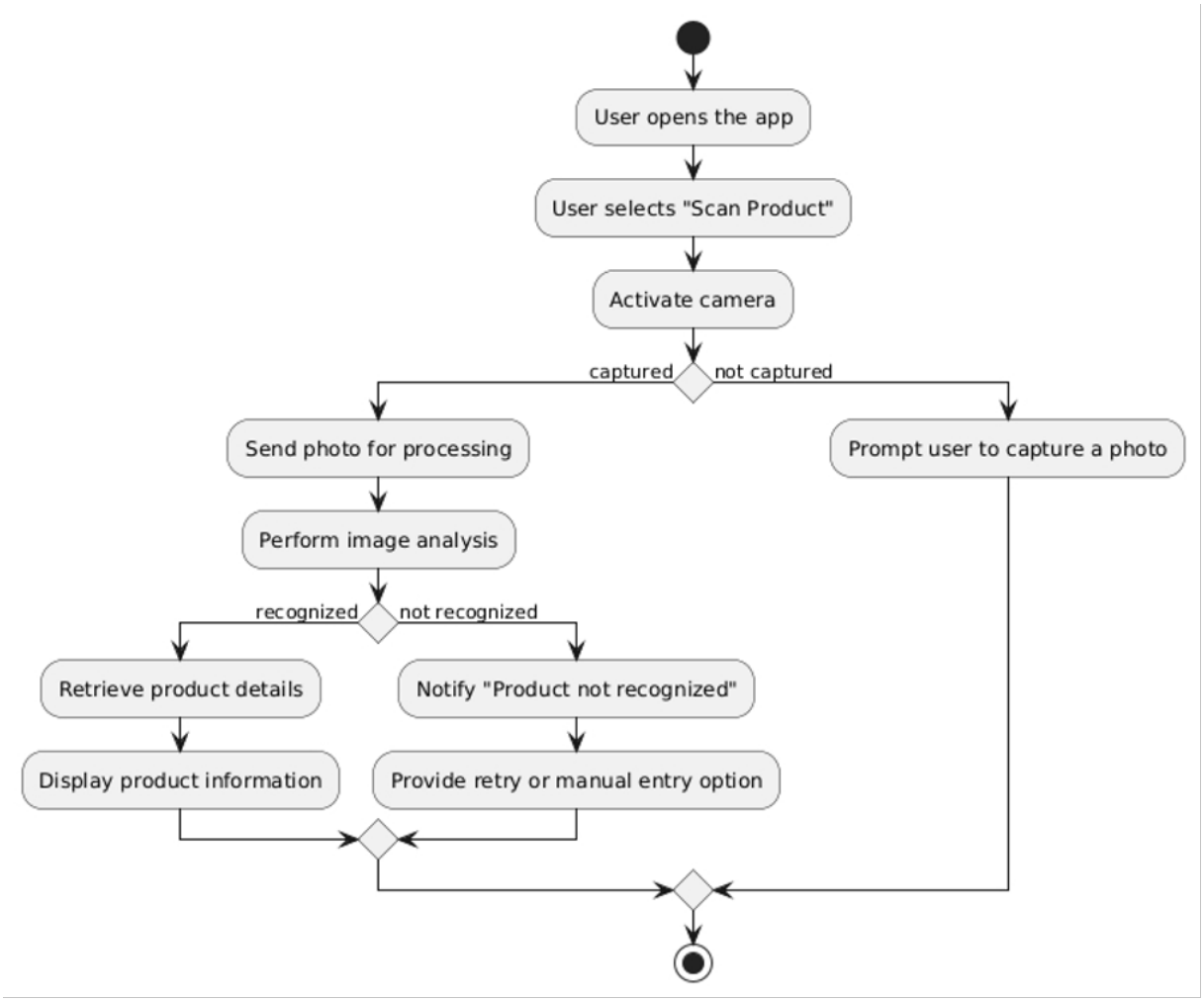



*Figure 5: Product Recognition Activity Diagram*

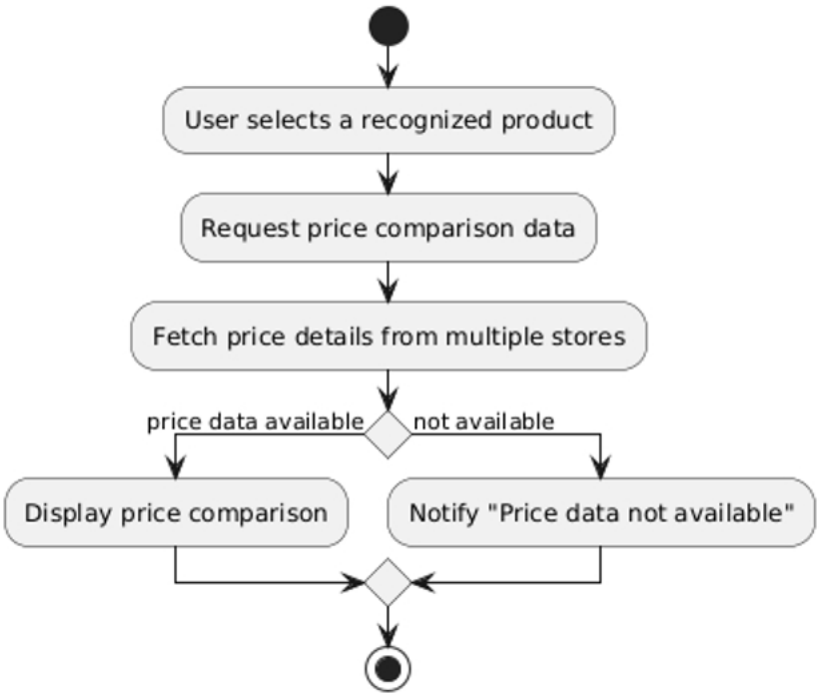



*Figure 6: Price Comparison Activity Diagram*

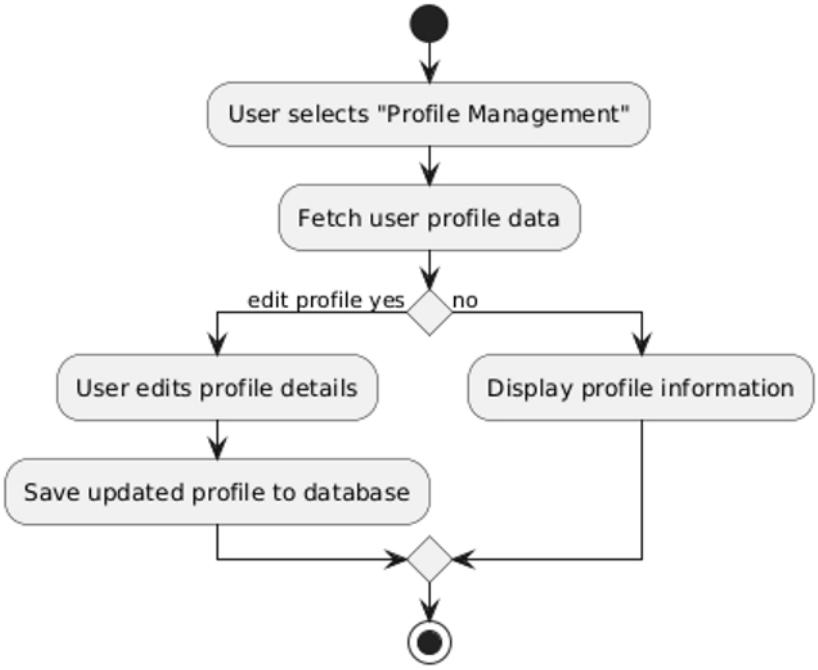



*Figure 7: Profile Management Activity Diagram*

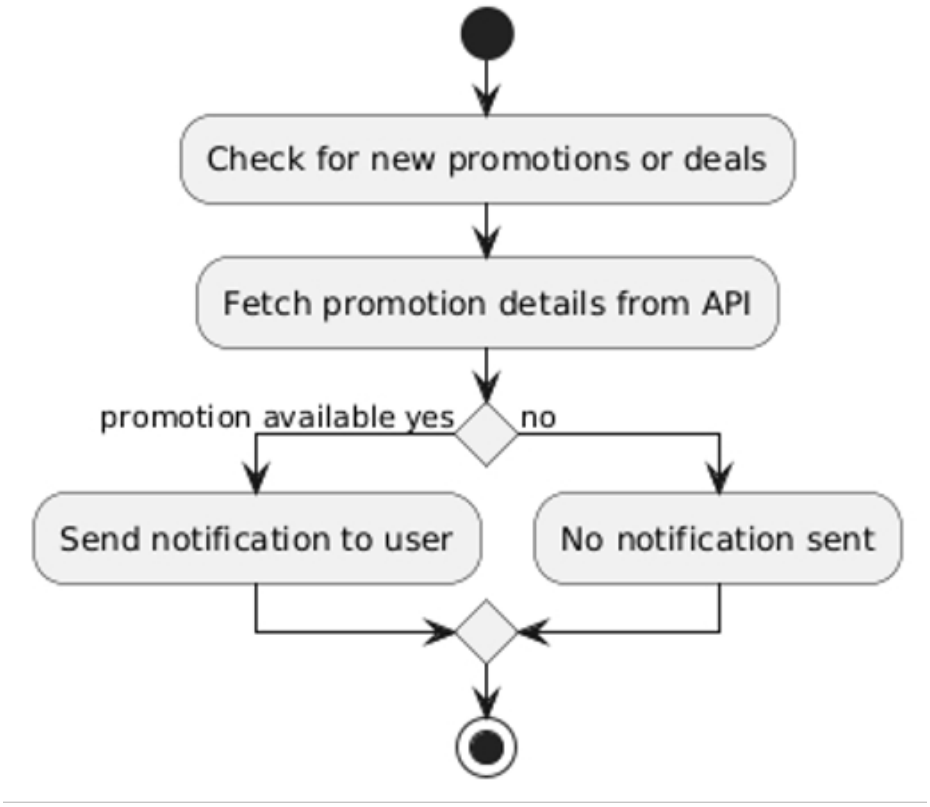



*Figure 8: Profile Management Activity Diagram*

# 3. User Interface Design

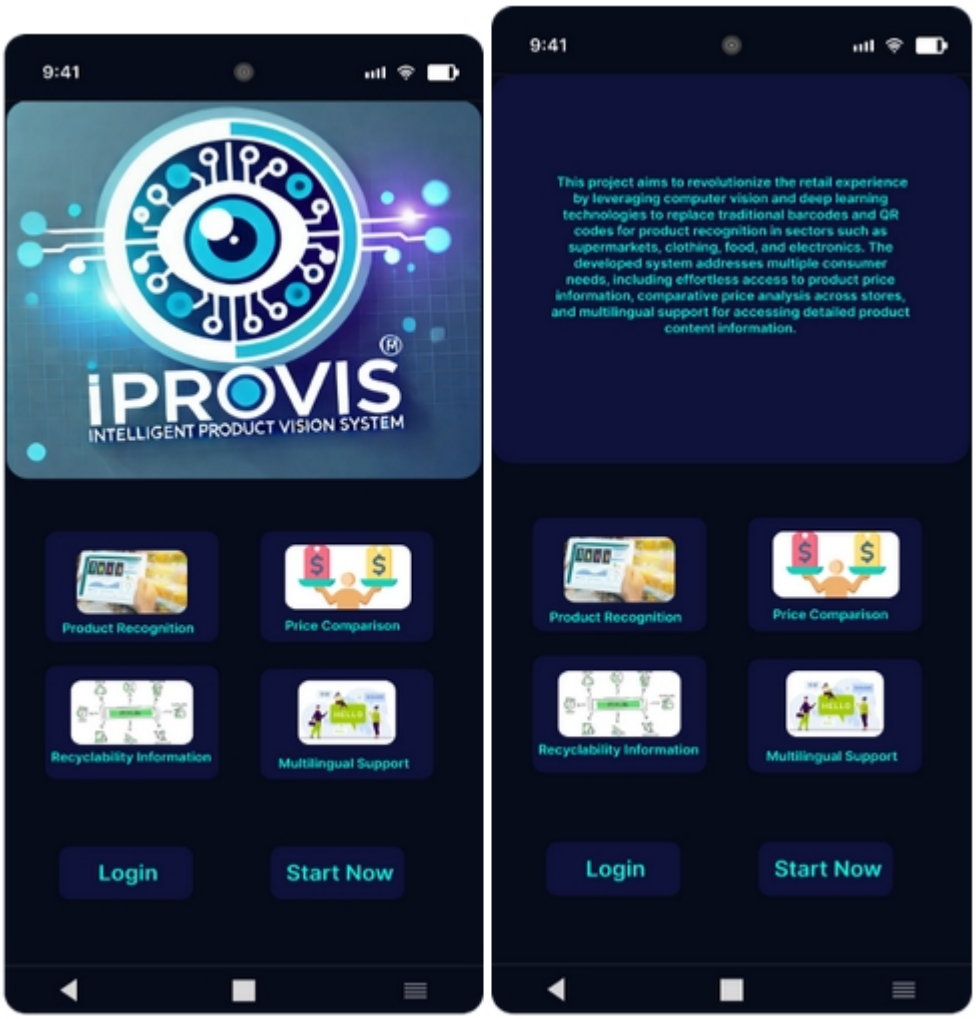
## 3.1 Overview

The user interface (UI) for the iProVis system is designed to be intuitive, user-friendly, and visually appealing, targeting a broad audience with varying levels of technical expertise.

## 3.2 Key UI Components

Home Screen

**App Overview:** A brief introduction to the app's purpose and key features (e.g., "Scan products, compare prices, and find eco-friendly recycling tips with ease!"). **Login Button:** Directs users to the login or register screen to create or access their account. **Start Button:** Allows users to proceed directly to the scanning feature without logging in (guest mode). **Navigation Menu:** A simple menu to explore additional features or app settings.



**Camera:** Real-time interface for product recognition using the device camera. Features autofocus, flash toggle etc.

**Scan Indicator: ** Visual elements like focus boxes or animation to indicate scanning progress.

**Action Buttons:**

- Manual Upload: Opens a gallery for manually product lookup.
- Capture/Scan Button: Initiates product scanning.

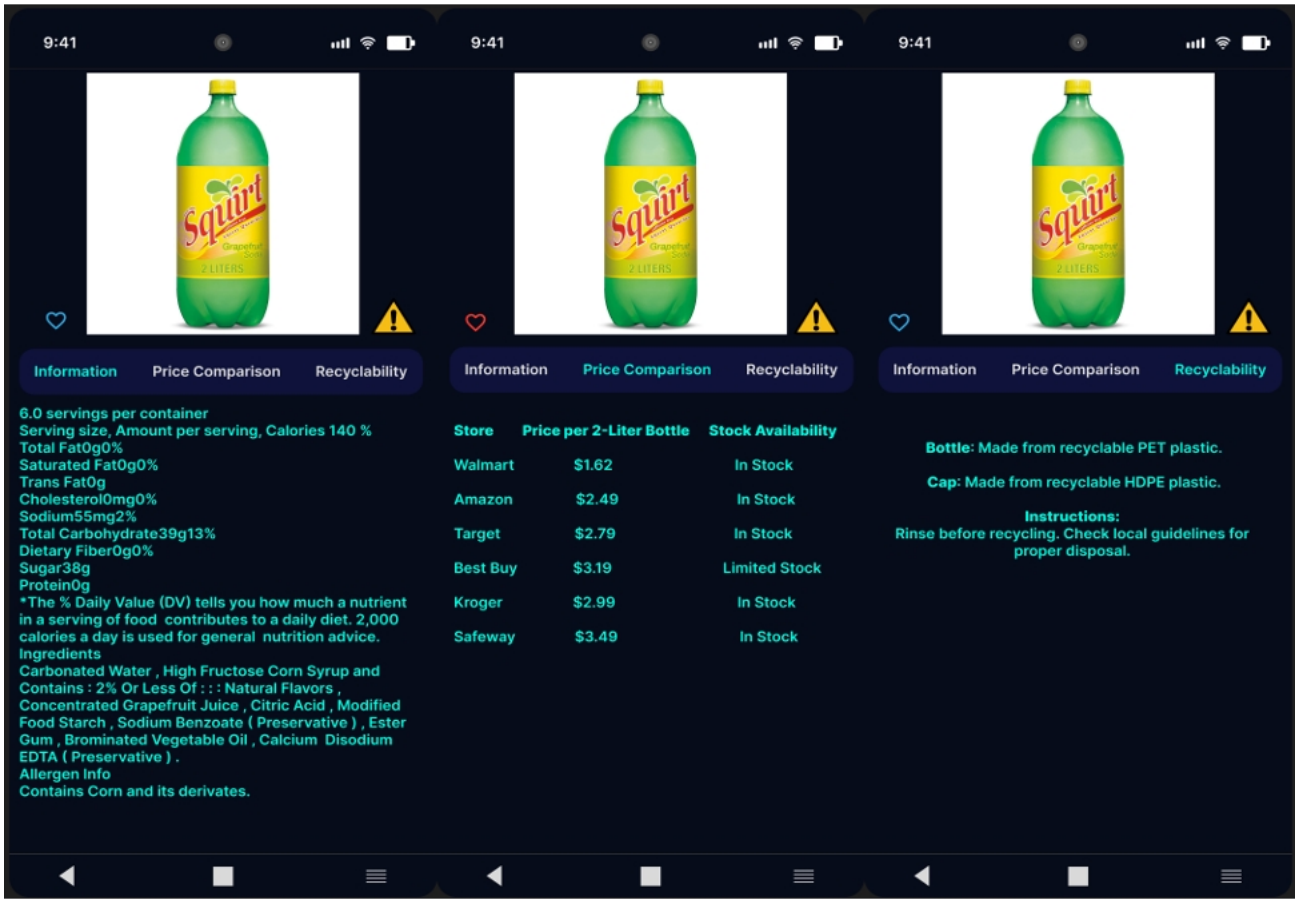
**Scan Results Screen**

**Product Information:** Displays details such as product name, manufacturer, and associated tags.

**Price Comparison:** Shows pricing options from multiple vendors with links for purchasing.

**Recycling Information:** Provides guidance for eco-friendly disposal or recycling of the product.

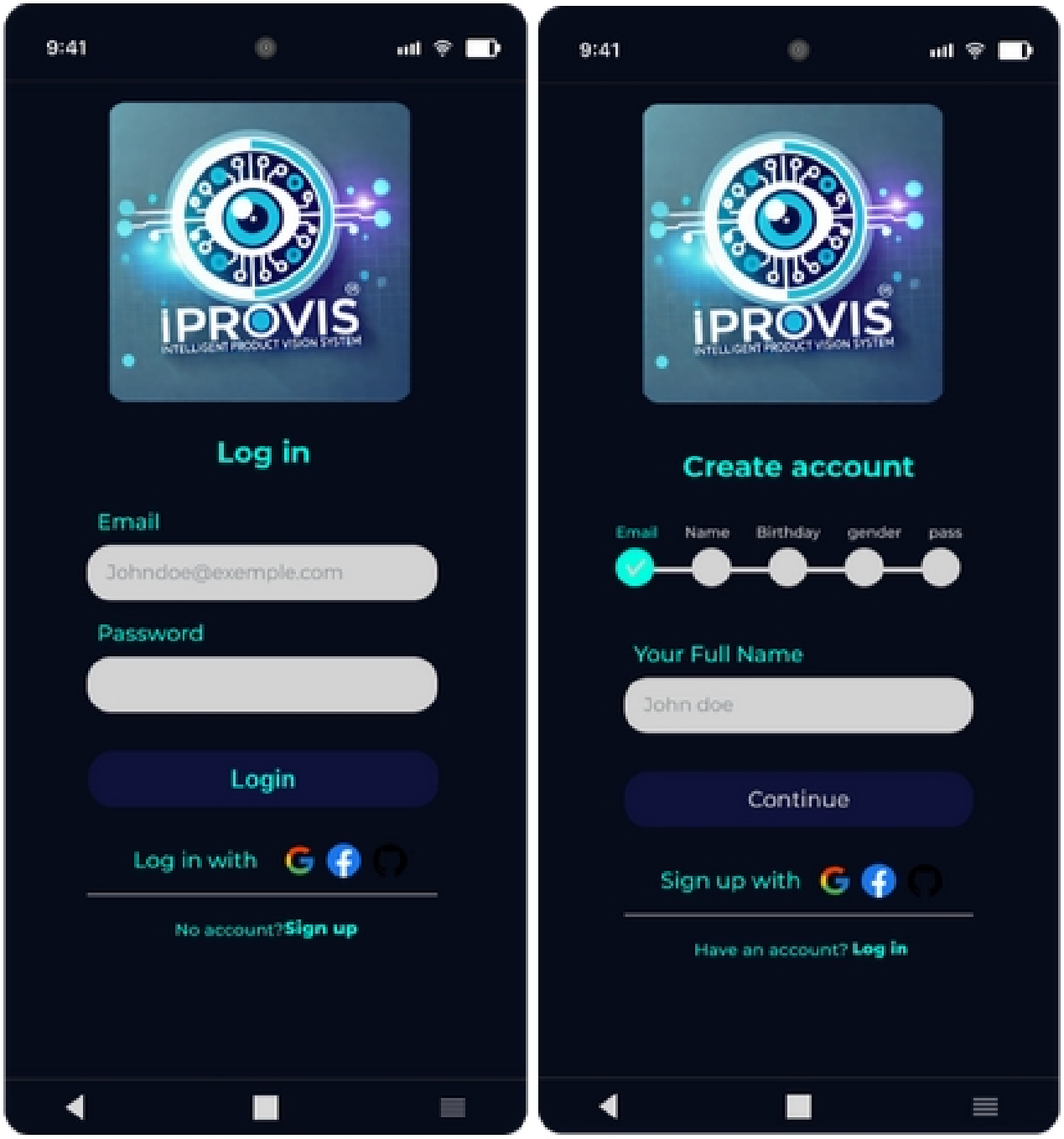
**Action Buttons:** Save to Favorites: Logs the scanned product for future reference.



**Login and Register Screens**

**Login Screen:** Fields for username/email and password, along with a "Forgot Password" option. Option for social login via Google, Facebook or Github.

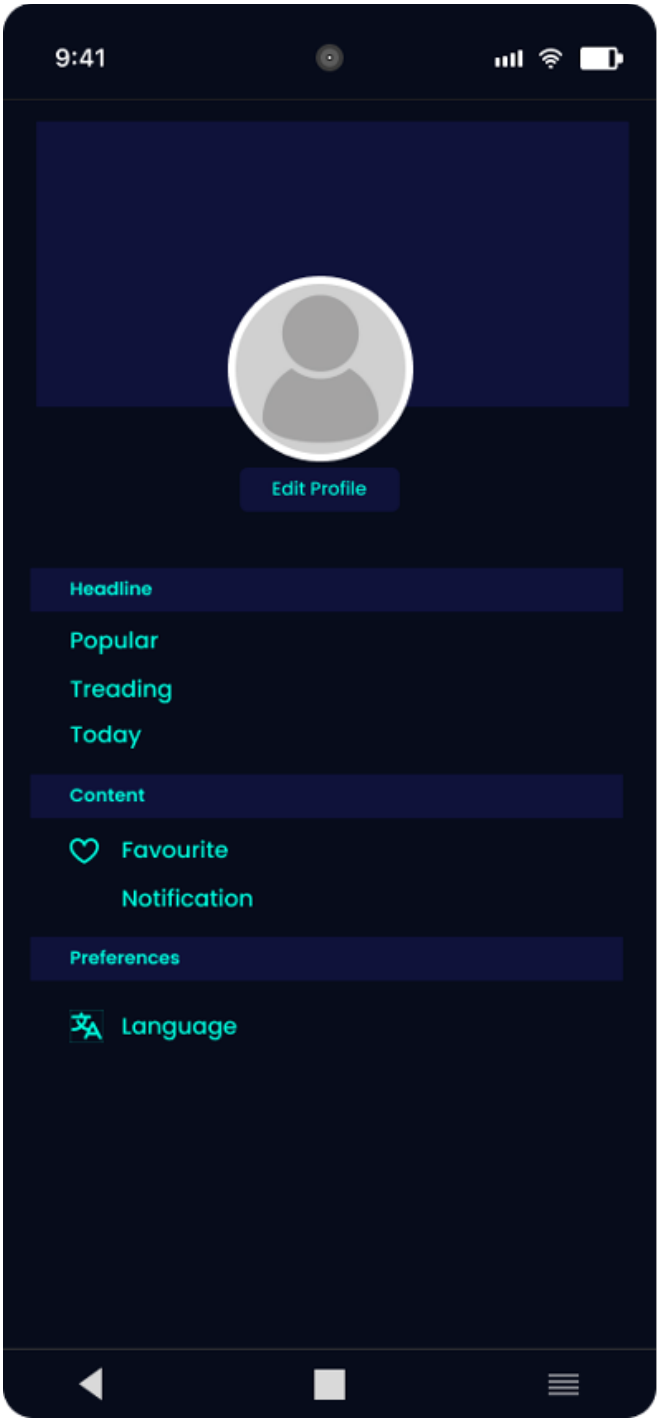
**Register Screen:** Includes fields for creating a new account: name, email, password, and password confirmation. Includes terms of service acceptance.



**Profile Management:** Options to update user details like name, email, and profile picture.

**Preferences:** Language selection settings.

**Favorites Scans:** Displays a list of previously scanned products with sorting and search options.



## 3.3 Design Principles

**Consistency:** Uniform color schemes, typography, and button styles across all screens.

**Responsiveness:** Adapts seamlessly to various device screen sizes.

**Accessibility:** Includes features like high contrast, and large fonts for improved usability.

## 3.4 Additional

## 1. Architectural Constraints

- The app will initially be developed for Android devices, with potential expansion to iOS in the future if there's enough demand and the project performs well.
- To work smoothly, the app needs a device equipped with a functioning camera, at least 2GB of RAM, and stable internet access.
- APIs for features like price comparison, translation, and product database access will follow all applicable licensing and usage rules.

## 2. Security Considerations

- User data such as login credentials, preferences, and activity history will be encrypted using strong encryption standards (AES-256) to protect sensitive information.
- All communication between the app and servers will use HTTPS, ensuring that transmitted data stays secure.
- Any user activity data will be anonymized to comply with privacy regulations.

## 3. Performance Objectives

- The app will aim to identify scanned products within 2 seconds to ensure a quick and seamless user experience.
- It should handle up to 20,000 users at the same time, leveraging load balancing to distribute traffic efficiently.
- The system's backend will automatically scale during high-traffic periods to maintain consistent performance without service interruptions.

## 4. Usability and Accessibility

- The interface will be simple and intuitive, ensuring users can navigate the app with ease.
- Accessibility features will include high-contrast themes and larger font sizes to accommodate users with visual impairments.
- The app will support multiple languages (at least 10 major ones) to make it usable for a diverse audience.

## 5.Testing and Validation

- Each part of the app will be tested with automated unit tests to catch bugs early, with a target of at least 90% test coverage.
- Integration testing will ensure smooth interactions between the app and external systems like APIs for product recognition and price comparison.
- User testing will involve real users trying the app to confirm its usability and gather feedback.
- Load testing will simulate heavy usage to confirm that the app remains stable under pressure.

## 6. Maintenance and Future Enhancements

- A bug tracking system will be in place to quickly identify and resolve any issues that arise.
- Regular updates will include new features and improvements based on user feedback.
- If there’s demand, the app will be expanded to iOS and potentially a web version.
- Recycling-related features will continue to grow, incorporating more data sources to help users identify recyclable materials.

## Conclusion

The iProViS project represents a significant step forward in utilizing artificial intelligence and computer vision for real-world applications. By providing users with a seamless way to access comprehensive product information, iProViS not only improves the shopping experience but also encourages environmentally conscious behaviors through its focus on recyclability and green practices. The modular design and scalable infrastructure ensure that the system is adaptable to future advancements and broader use cases.

This project demonstrates the potential of integrating innovative technologies to address everyday challenges faced by consumers. By enabling smarter purchasing decisions, fostering sustainability, and offering personalized features, iProViS highlights the impact technology can have on enhancing both user experience and global environmental awareness. As we look forward, iProViS has the potential to evolve further, incorporating more advanced features and expanding its reach to benefit even more users worldwide.

# References

[1] Li, Q., Peng, X., Cao, L., Du, W., Xing, H., Qiao, Y., & Peng, Q. (2020). Product image recognition with guidance learning and noisy supervision. Computer Vision and Image Understanding, 196, 102963. https://www.sciencedirect.com/science/article/abs/pii/S1077314220300436

[2] Bai, Y., Chen, Y., Yu, W., Wang, L., & Zhang, W. (2020). Products-10k: A large-scale product recognition dataset. arXiv preprint arXiv:2008.10545. https://ar5iv.labs.arxiv.org/html/2008.10545

[3] Chen, S., Liu, D., Pu, Y., & Zhong, Y. (2022). Advances in deep learning-based image recognition of product packaging. Image and Vision Computing, 128, 104571. https://www.sciencedirect.com/science/article/abs/pii/S0262885622002001

[4] Guo, X., Wu, H., Gao, Y., Rennie, S. J., & Feris, R. (2021). Fashion IQ: A New Dataset Towards Retrieving Images by Natural Language Feedback. https://www.cs.utexas.edu/~grauman/papers/fashion-iq-CVPR2021.pdf

[5] George, M., & Floerkemeier, C. (2014). Recognizing Products: A Per-exemplar Multilabel Image Classification Approach. Springer. https://link.springer.com/chapter/10.1007/978-3-319-10605-2_29

[6] Shaikh, A., Khan, R., Panokher, K., Ranjan, M. K., & Sonaje, V. (2023). E-commerce Price Comparison Website Using Web Scraping. ResearchGate. https://www.researchgate.net/publication/374386196_E-commerce_Price_Comparison_Website_Using_Web_Scraping

[7] Pangriya, R., & Jaiswal, C. (2023). AMAZON GO!!!! JUST WALKOUT. ResearchGate. https://www.researchgate.net/publication/373976515_AMAZON_GO_JUST_WALKOUT

[8] Chen, Y., Wang, L., & Zhang, W. (2021). Mobile Vision: Enhancing Product Recognition on Handheld Devices for Retail Applications. Journal of Computer Vision and Mobile Applications, 12(3), 155-169.

https://www.researchgate.net/publication/346856840_Deep_Learning_for_Retail_Product_Recognition_Challenges_and_Techniques

[9] M. O., Varghese, C., Pathak, D., & Varde, A. S. (2021). An Intelligent Food Inventory Monitoring System Using Machine Learning and Computer Vision. https://www.researchgate.net/profile/Tianyu-Li-59/publication/363685048_An_Intelligent_Food_Inventory_Monitoring_System_using_Machine_Learning_and_Computer_Vision/links/6331dd2986b22d3db4e54309/An-Intelligent-Food-Inventory-Monitoring-System-using-Machine-Learning-and-Computer-Vision.pdf

[10] Figma Intercative User Interface. https://www.figma.com/proto/XjNlPgp29jBlOt3RZ86eeS/IPROVIS?node-id=0-1&t=DLGgdtSQD1LJddPR-1