UNIVERSITY OF ECONOMICS AND LAW

**FACULTY OF INFORMATION SYSTEMS**

****

**FINAL PROJECT REPORT**

**COURSE: DEEP LEARNING**

**Topic:**

**WEB AND MOBILE APPLICATION FOR ENHANCED GROCERY SHOPPING: SMART RECOMMENDATIONS AND BARCODE-BASED INVENTORY MANAGEMENT**

*Ho Chi Minh City, November 17, 2023*

UNIVERSITY OF ECONOMICS AND LAW

**FACULTY OF INFORMATION SYSTEMS**

****

**FINAL PROJECT REPORT**

**COURSE: DEEP LEARNING**

**Topic:**

**WEB AND MOBILE APPLICATION FOR ENHANCED GROCERY SHOPPING: SMART RECOMMENDATIONS AND BARCODE-BASED INVENTORY MANAGEMENT**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Group 5: UPgrade** | | | | | | | |
| **NO.** | **Name** | **ID** | | **Faculty** | **Role** | **Phone** | **Email** |
| 1 | Vu Van Son | | K204160669 | FIS | Leader | 0944244857 | sonvv20416c  @st.uel.edu.vn |
| 2 | Chau Nguyen Huong Thao | | K204162000 | FIS | Member | 0919772802 | thaocnh20416c  @st.uel.edu.vn |
| 3 | Nguyen Tuong Vy | | K204162009 | FIS | Member | 0764187037 | vynt20416c  @st.uel.edu.vn |

Lecturer: **Ph.D. Tran Duy Thanh.**

*Ho Chi Minh City, November 17, 2023*

# ACKNOWLEDGEMENTS

During this project, we received a significant amount of dedicated assistance and guidance from our lecturer - Ph.D. Tran Duy Thanh and classmates. Thanks to their contributions, we were able to successfully complete our final project.

Despite our best efforts, mistakes may have been made, and we welcome any feedback or constructive criticism you may have to help improve the quality of our project. Your input will serve as a tremendous source of motivation as we continue to develop our project in the future.

# COMMITMENT

We assure that our final project is original and based on research developed by the entire team. In the event that any provided information is found to be inaccurate or misleading, we take full responsibility for any potential consequences that may arise, and we will collaborate closely with our lecturer to rectify the situation.

Ho Chi Minh City, November 17, 2023

**Committed by**

**Team Upgrade**

# TABLE OF CONTENT

[ACKNOWLEDGEMENTS I](#_Toc151158960)

[COMMITMENT II](#_Toc151158961)

[TABLE OF CONTENT III](#_Toc151158962)

[LIST OF FIGURES V](#_Toc151158963)

[LIST OF ACRONYMS VII](#_Toc151158964)

[CHAPTER 1: OVERVIEW 1](#_Toc151158965)

[1.1 Objectives of the project 1](#_Toc151158966)

[1.2 Structure of the project report 1](#_Toc151158967)

[CHAPTER 2: THEORETICAL BASIS 2](#_Toc151158968)

[2.1 APS.NET Framework 2](#_Toc151158969)

[2.2 ML.NET Framework 3](#_Toc151158970)

[2.3 Recommendation system 3](#_Toc151158971)

[2.4 Android Studio 4](#_Toc151158972)

[2.5 Firebase 5](#_Toc151158973)

[2.6. Barcode recognition system 6](#_Toc151158974)

[CHAPTER 3: GROCERY SHOPPING DATABASE 8](#_Toc151158975)

[3.1 Database structure for Webssite 8](#_Toc151158976)

[3.1.1 Database schema 8](#_Toc151158977)

[3.1.2 The relationships between the tables 8](#_Toc151158978)

[3.2 Firebase Real-time database 9](#_Toc151158979)

[CHAPTER 4: EXPERIMENTAL APPLICATION 10](#_Toc151158980)

[4.1 Website application using Asp.net 10](#_Toc151158981)

[4.1.1 Customer 10](#_Toc151158982)

[4.1.2 Employee 22](#_Toc151158983)

[4.2 Mobile application base on Android 26](#_Toc151158984)

[4.2.1 Login and Register 28](#_Toc151158985)

[4.2.2 Product management 30](#_Toc151158986)

[4.2.3 Other functions 35](#_Toc151158987)

[CHAPTER 5: CONCLUSIONS AND FUTURE WORKS 41](#_Toc151158988)

[5.1 Results 41](#_Toc151158989)

[5.2 Limitations 41](#_Toc151158990)

[5.3 Future works 41](#_Toc151158991)

[REFERENCES 42](#_Toc151158992)

[APPENDIX 43](#_Toc151158993)

# LIST OF FIGURES

[Figure 1: ASP.NET (source: https://forums.asp.net/myforums.aspx ) 2](#_Toc151158810)

[Figure 2: : ML.NET Framework (source: https://en.wikipedia.org/wiki/ML.NET) 3](#_Toc151158811)

[Figure 3: Database diagram 7](#_Toc151158812)

[Figure 4: Firebase Real-time database 8](#_Toc151158813)

[Figure 5: Use case of Grocery shopping application 9](#_Toc151158814)

[Figure 6: Customer Login screen 10](#_Toc151158815)

[Figure 7: Customer Logion screen 11](#_Toc151158816)

[Figure 8: Homepage screen 12](#_Toc151158817)

[Figure 9: Product 13](#_Toc151158818)

[Figure 10: Product details screen 14](#_Toc151158819)

[Figure 11: Recommended products 15](#_Toc151158820)

[Figure 12: The steps to builder a recommendation system. 16](#_Toc151158821)

[Figure 13: Product recommendation ratings 18](#_Toc151158822)

[Figure 14: Customers cart 19](#_Toc151158823)

[Figure 15: Cart Total price 20](#_Toc151158824)

[Figure 16: Bill details 21](#_Toc151158825)

[Figure 17: Employee Login screen 22](#_Toc151158826)

[Figure 18: Admin Homepage 23](#_Toc151158827)

[Figure 19: Create a product screen 24](#_Toc151158828)

[Figure 20: Edit product screen 25](#_Toc151158829)

[Figure 21: Task flow of Grocery store’s mobile application 26](#_Toc151158830)

[Figure 22: Login screen 27](#_Toc151158831)

[Figure 23: Reset Password screen 28](#_Toc151158832)

[Figure 24: Register screen 29](#_Toc151158833)

[Figure 25: Splash of Product management 30](#_Toc151158834)

[Figure 26: Add product functionality 31](#_Toc151158835)

[Figure 27: Search products by scanning barcode 32](#_Toc151158836)

[Figure 28: Search products by product’s name 32](#_Toc151158837)

[Figure 29; Filter by products’ categories 33](#_Toc151158838)

[Figure 30: Product’s edit information screen. 33](#_Toc151158839)

[Figure 31; Delete a product 34](#_Toc151158840)

[Figure 32: Transaction Management 35](#_Toc151158841)

[Figure 33: Splash of Person Management 36](#_Toc151158842)

[Figure 34: Add customers in Person management 37](#_Toc151158843)

[Figure 35: Performance function 39](#_Toc151158844)

# LIST OF ACRONYMS

|  |  |  |
| --- | --- | --- |
| **NO.** | **ACRONYMS** | **MEANINGS** |
| 1 | MVC | Model-View-Controller |
| 2 | MAE | Mean Absolute Error |
| 3 | XML | eXtensible Markup Language |
| 4 | RMSE | Root Mean Squared Error |
| 5 | CRUD | Create, Read, Update, Delete |
| 6 | MSE | Mean Squared Error |
| 7 | ZXing | Zebra Crossing |
| 8 | OCR | Optical Character Recognition |
| 9 | IDE | Integrated Development Environment |
| 10 | BaaS | Backend-as-a-Service |

# CHAPTER 1: OVERVIEW

## 1.1 Objectives of the project

The main objectives of the project is to be able to apply the recommended model in order to develop a simple website and mobile application for the Android platform, thereby suggesting options to customers that align with their preferences. The subsequent objective is to comprehend and effectively utilize various technologies and platforms, such as ASP.NET, ML.NET for web development and Firebase for the mobile app. The ultimate aim is to gain a more comprehensive understanding of the diverse current platforms like websites, mobile apps, and desktop apps, to further develop and apply them in practice in the future.

## 1.2 Structure of the project report

The structure of the report consists of five chapters. **Chapter 1** provides an overview of the project, while the theoretical basis will be illustrated in **Chapter 2**. **Chapter 3** shows details of the database structure for website and Firebase real-time database and the methodology for the recommendation model. Subsequently, the main functionalities of both the website and application will be discussed in **Chapter 4**, and finally, **Chapter 5** is the conclusion and future work development directions.

# CHAPTER 2: THEORETICAL BASIS

## 2.1 APS.NET Framework

ASP.NET [[1](#m1)] is a web application framework developed by Microsoft. It allows developers to build web applications and services using a variety of programming languages, including C#, Visual Basic, and F#. ASP.NET is based on the .NET Framework, which is a general-purpose programming platform that provides a wide range of services, such as memory management, threading, and security.

ASP.NET is a popular choice for developing web applications because it is easy to use, powerful, and scalable. It is also a relatively mature framework, with a large community of developers and a wealth of resources available. Object-oriented programming: ASP.NET is based on object-oriented programming principles, which means that applications are built using reusable objects that encapsulate data and behavior.



Figure 1: ASP.NET (source: https://forums.asp.net/myforums.aspx )

Object-oriented programming: ASP.NET is based on object-oriented programming principles, which means that applications are built using reusable objects that encapsulate data and behavior.

Model-View-Controller (MVC) pattern: ASP.NET applications are typically structured using the MVC pattern, which separates the application's data model, presentation layer, and control logic.

Event-driven programming: ASP.NET applications are event-driven, which means that they respond to user interactions and other events.

State management: ASP.NET provides a variety of mechanisms for managing the state of web applications, such as session state and view state.

Security: ASP.NET provides a variety of security features, such as authentication and authorization.

## 2.2 ML.NET Framework

ML.NET [[2](#m2)] an open-source framework by Microsoft, facilitates the development of machine learning applications using the C# programming language and the .NET platform. ML.NET provides a comprehensive suite of tools, libraries, and APIs that simplify the integration of machine learning capabilities into existing applications. This enables developers to create and deploy machine learning models for tasks such as classification, prediction, clustering, and various other machine learning-related tasks.

A significant advantage of ML.NET is its seamless integration with existing .NET applications, eliminating the need for developers to learn a new language or go through complex procedures to incorporate machine learning into their applications.

ML.NET offers a variety of tools and classes to assist in data preprocessing, model building, training, and evaluation. Additionally, it supports the integration of trained models into real-world applications for direct prediction and classification tasks.



Figure 2: : ML.NET Framework (source: https://en.wikipedia.org/wiki/ML.NET)

## 2.3 Recommendation system

A Recommendation System, also known as a Recommender System, is a subset of information filtering and data mining techniques used in various applications to provide personalized suggestions, recommendations, or predictions to users. The main goal of a recommendation system is to assist users in discovering items or content that they are likely to be interested in, based on their past behaviors, preferences, or similar patterns exhibited by other users.

* Recommendation systems are extensively used across a wide range of industries and contexts, including e-commerce, online streaming services, social media platforms, news websites, and many more. They enhance user experiences by helping individuals discover relevant content or products, consequently increasing engagement and user satisfaction. Several approaches exist for building recommendation systems:
* **Collaborative Filtering:** This approach generates recommendations by analyzing the historical behavior and preferences of users, identifying patterns and similarities between users or items. Collaborative filtering can be user-based (recommending items based on similar users' preferences) or item-based (recommending items similar to those the user has already interacted with).
* **Content-Based Filtering:** Content-based filtering focuses on the attributes of items or content and makes recommendations based on the similarity between the attributes of items and users' preferences. For instance, recommending movies based on genres, actors, or directors.
* **Deep Learning:** Recent advancements in deep learning techniques have been applied to recommendation systems, utilizing neural networks to capture intricate patterns and relationships in user-item interactions.
* **Hybrid Methods:** Hybrid methods combine multiple recommendation techniques to enhance accuracy and address the limitations of individual methods.

## 2.4 Android Studio

Android Studio is a software development environment (IDE) that is used to develop Android applications. It is based on the IntelliJ IDEA platform and is developed by Google. Android Studio provides a variety of tools and features that make it easy to develop Android applications, including:

* A code editor with syntax highlighting and code completion
* A debugger that allows to step through your code and see what is happening
* A layout editor that allows to design the user interface of your application
* A resource manager that allows to manage the resources of your application, such as images and sounds.
* A testing framework that allows to test your application.
* Android Studio is a powerful tool that can be used to develop a wide variety of Android applications. It is a popular choice for developers of all levels of experience.

One of the key theories of Android Studio is that it is based on the Model-View-Controller (MVC) pattern. This pattern separates the application's data model, presentation layer, and control logic. This makes it easier to develop and maintain Android applications.

Another key theory of Android Studio is that it is based on the Android Jetpack library. This library provides a set of components that can be used to develop Android applications. Jetpack components are designed to be easy to use and to work together well. They can help you to develop more robust and reliable Android applications.

## 2.5 Firebase

Firebase [[3](#m3)] is a Backend-as-a-Service (BaaS) platform that provides a suite of cloud-based services for developing and managing mobile and web applications. It offers a variety of features, including:

* **Real-time database:** A cloud-based database that allows you to store and sync data in real time.
* **Cloud Firestore:** A NoSQL document database that provides flexible and scalable data storage.
* **Authentication:** A secure authentication system that allows you to manage user accounts and permissions.
* **Cloud Storage:** A cloud-based file storage service that allows you to store and manage files, such as images and videos.
* **Hosting:** A web hosting service that allows you to deploy and host your web applications.
* **Remote Config:** A cloud-based configuration service that allows you to manage app settings without having to release new versions of your app.
* **Test Lab:** A cloud-based testing service that allows you to test your apps on a variety of devices and configurations.
* **Analytics:** A cloud-based analytics service that allows you to track user behavior and app performance.

Firebase is a powerful platform that can be used to develop a wide variety of mobile and web applications. It is a popular choice for developers of all levels of experience.

One of the key theories of Firebase is that it is a serverless platform. This means that you do not need to manage or maintain any servers. Firebase takes care of all of the infrastructure for you, so you can focus on developing your applications.

Another key theory of Firebase is that it is a real-time platform. This means that data is synced in real-time across all devices and users. This makes Firebase a great choice for applications that require real-time data, such as chat applications and multiplayer games.

## 2.6. Barcode recognition system

A Barcode Recognition System is a crucial component in the field of inventory management and product services. It falls under the domain of Optical Character Recognition (OCR), which is a branch of Machine Learning. OCR focuses on the recognition of printed or written text characters inside digital images.

The primary objectives of implementing a barcode recognition system in a grocery store's inventory and product management system include:

* Enhanced Data Accuracy: Eliminate errors associated with manual data entry and ensure the accuracy of product information.
* Streamlined Inventory Management: Automate inventory tracking and provide real-time visibility into product stock levels, enabling proactive replenishment and reducing stockouts.
* Improved Product Management: Facilitate efficient product identification, categorization, and tracking, enabling better product planning and decision-making.
* Reduced Operational Costs: Minimize labor expenses associated with manual data entry and inventory checks, streamlining operations and reducing costs.

Barcode recognition technology has emerged as a valuable tool for grocery stores to streamline inventory management, enhance product management, and improve operational efficiency. By automating data collection and providing real-time insights, barcode recognition systems can contribute to reducing costs, improving customer satisfaction, and increasing profitability.

There are several types of barcode recognition systems, each with its unique characteristics and use cases. The most common types include:

* 1D (Linear) Barcodes: These are the traditional barcodes that represent data in the widths and spacings of parallel lines. Examples include EAN, Code 128, and UPC.
* 2D Barcodes: These barcodes store information horizontally and vertically; examples include QR codes and Data Matrix.
* 3D Barcodes: These are relatively new and involve barcodes that are embossed or engraved on a product and require special equipment to be read.

In this project, the ZXing (Zebra Crossing) library is used for barcode scanning. ZXing is an open-source, multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages. It supports a variety of barcode formats, including 1D product, 1D industrial, and 2D barcodes.

# CHAPTER 3: GROCERY SHOPPING DATABASE

## 3.1 Database structure for Webssite

We designed the grocery shopping database including Products, Employees, Categories, Orders, Customer and OrderDetails like Figure below.

A screenshot of a computer

Description automatically generated

Figure 3: Database diagram

### 3.1.1 Database schema

* **Products**: This table stores information about products, such as product name, category ID, unit price, quantity, and barcode.
* **Categories**: This table stores information about product categories, such as category name and description.
* **Orders**: This table stores information about customer orders, such as order ID, customer ID, employee ID, order date, ship date, and status.
* **OrderDetails**: This table stores information about the individual line items in an order, such as order ID, product ID, quantity, unit price, and discount.
* **Customers**: This table stores information about customers, such as customer ID, contact name, address, city, and username.
* **Employees**: This table stores information about employees, such as employee ID, last name, first name, role, username, and password.

### 3.1.2 The relationships between the tables

* A product can belong to one category, but a category can have many products.
* An order can have many order details, but an order detail can only belong to one order.
* An order is placed by one customer, and a customer can place many orders.
* An order is processed by one employee, and an employee can process many orders.

## 3.2 Firebase Real-time database

The Firebase Realtime Database stores data in a JSON tree structure and offers real-time synchronization. When data changes, it updates instantly across all connected devices. Applications can listen for data change events and work offline, syncing data when the connection is restored. Firebase Realtime Database also provides security rules to control access to the data.

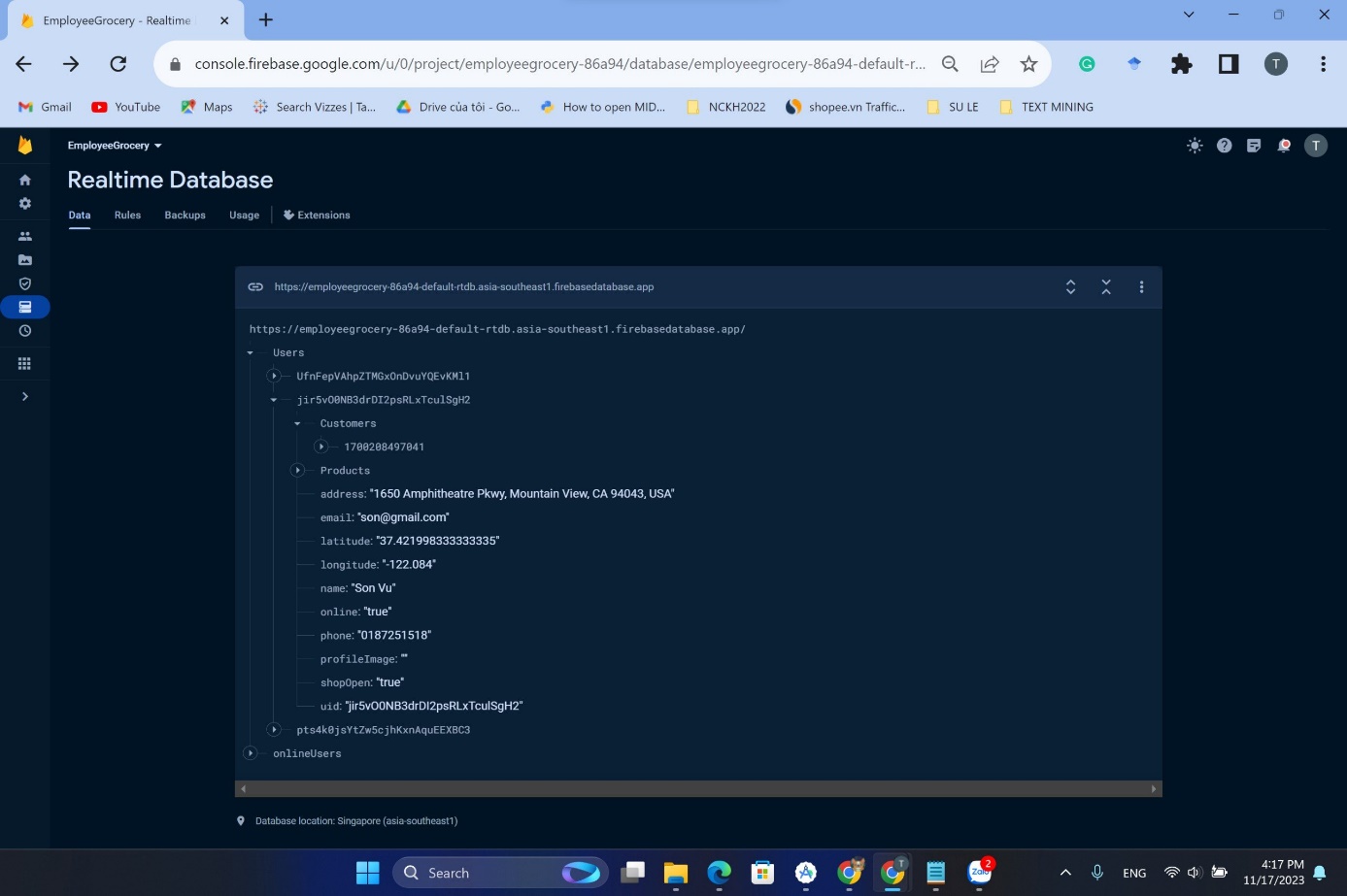


Figure 4: Firebase Real-time database

# CHAPTER 4: EXPERIMENTAL APPLICATION



Figure 5: Use case of Grocery shopping application

## 4.1 Website application using Asp.net

### 4.1.1 Customer

* **Function: Login**

A screenshot of a login form

Description automatically generated

Figure 6: Customer Login screen

The login function is a simple form-based login. The user enters their username and password and then clicks the "SIGN IN" button to submit the form. The server then checks the username and password against its database, and if they are correct, the user is logged in.

The login function is located in the center of the image and is enclosed in a white box with a purple border. The box is titled "Sign in to continue", and the login form contains two fields: username and password. The username field is located above the password field, and the password field is masked, so that the user's password is not visible to others.

Below the password field is a purple button that says "SIGN IN". When the user clicks this button, the login form is submitted to the server. If the login is successful, the user is redirected to the main page of the website. Otherwise, the user is shown an error message.

* **Function: Log On**

A screenshot of a login form

Description automatically generated

Figure 7: Customer Logion screen

The register function allows the customer to create a new account on the website. It requires you to enter your full name and the name of your current city. Once you have entered this information, you can click the "LOG ON" button to create your account

By registering for a customer account, users will be able to access features and content that are not available to unregistered users, such as making purchases, viewing personalized recommendations, and saving their progress.

The register function of customer typically collects only basic information about the user, such as name, email address, and location. This information is used to create the user's account and to provide them with a personalized experience.

* **Homepage**

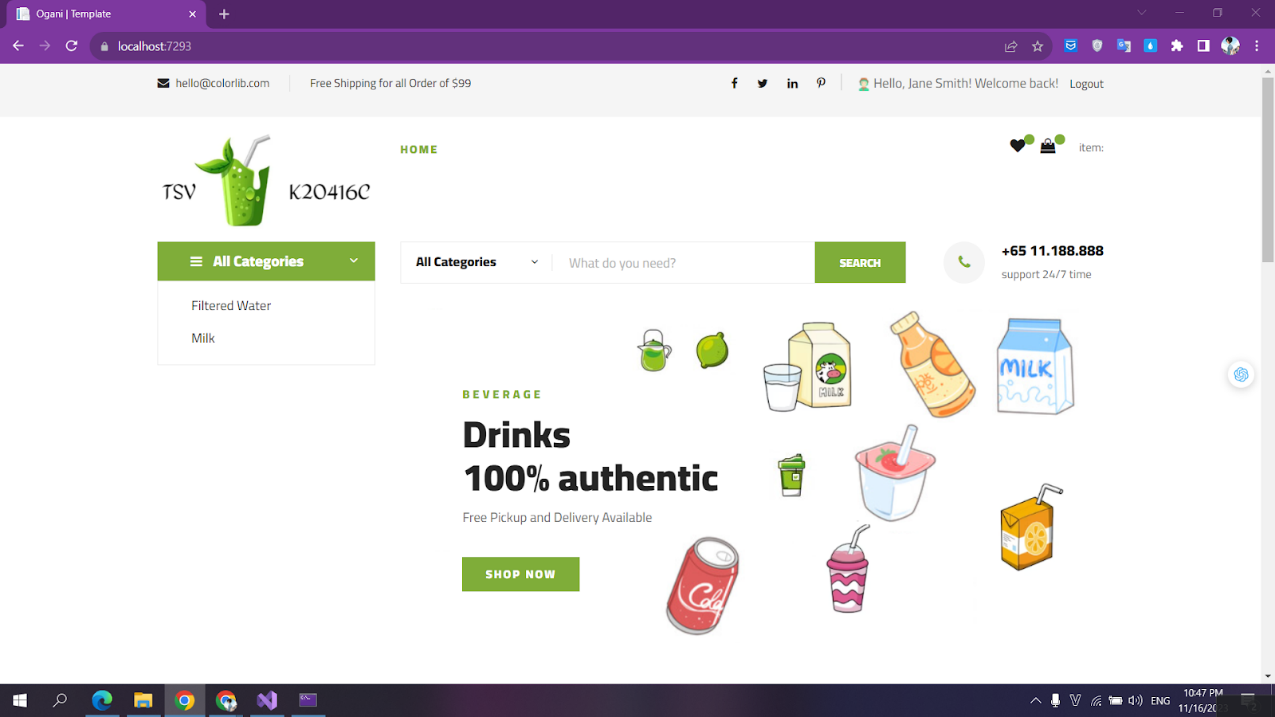


Figure 8: Homepage screen

A screenshot of a product

Description automatically generated

Figure 9: Product

The homepage function of the website is to provide users with a way to browse and purchase a variety of drinks. The homepage is divided into several sections:

* Header: The header contains the website's logo, a search bar, and a shopping cart icon.
* Logo: The logo is a visual representation of the website's brand.
* Search bar: The search bar allows users to search for specific drinks on the website. Users can type in the name of a drink or a keyword, and the search bar will return a list of relevant results.
* Shopping cart icon: The shopping cart icon allows users to view the items that they have added to their cart. Users can click on the shopping cart icon to proceed to checkout.
* Hero section: the hero section features a photo of many refreshing drink with different types of beverage.
* Categories: The categories section lists the different types of drinks that are available on the website. Users can click on a category to view all of the drinks in that category.
* Products: The products section displays a selection of drinks that are available for purchase. Users can click on a product to view more information about it, such as the ingredients, nutritional information, and customer reviews. The products are displayed in a grid format, with each product containing the following information:
* Product image: A photo of the product.
* Product name: The name of the product.
* Product price: The price of the product.
* Footer: The footer contains links to the website's policies, contact information, and social media pages.
* About us: A page about the website and the company behind it.
* Contact us: A page with the website's contact information.
* Social media icons: Links to the website's social media pages.

The homepage function of the website is to provide users with a way to browse and purchase a variety of drinks.

* **Function: Product details**

A product on the screen

Description automatically generated

Figure 10: Product details screen

The product detail function allows users to view more information about a specific product. This information includes:

* Product name: The name of the product.
* Product description: A detailed description of the product, including its features, benefits, and specifications.
* Product images: Multiple images of the product, from different angles.
* Product reviews: Reviews from other customers who have purchased the product.
* Add to cart button: A button that allows users to add the product to their shopping cart.

The product detail function is an important part of e-commerce website, as it allows users to learn more about the products they are interested in before making a purchase.

* **Function: Recommended Products**

A screenshot of a product

Description automatically generated

Figure 11: Recommended products

We recommend products utilizing a trained machine learning model to generates personalized product recommendations based on a user's purchase history and machine learning predictions. Here's a breakdown of the details:

* Retrieving User ID: The code starts by extracting the user ID (CustomerID) from the HTTP context's session variable.
* Creating DbContext and Model Loading: It then creates an instance of the BeverageRetailContext class, which represents the database connection. Next, it loads the trained machine learning model from the specified path (pathTrainedmodel) into a ITransformer object (trainedModel).
* Creating Prediction Engine and Model Schema: A prediction engine (predictionengine) is created using the loaded model (trainedModel), and the model schema (modelSchema) is obtained for further processing.
* Retrieving Products: All products from the Products table are retrieved into a list (products).
* Generating Recommendations: The code iterates through the list of products (products), creating a DataEntry object for each product, including the user ID (CustomerID) and product ID (ProductID).
* Prediction and Recommendation Filtering: For each product, the prediction engine (predictionengine) is used to predict the likelihood of purchase (result.Score). If the prediction score (result.Score) exceeds a threshold value (0.4 in this case), the product is considered a recommendation and added to the recommendedProducts list.
* Returning Recommendations: The method finally returns the list of recommended products (recommendedProducts), containing product ID (ProductID), product name (ProductName), prediction score (Score), and product image (ProductImage).

A screenshot of a computer

Description automatically generated

Figure 12: The steps to builder a recommendation system.

The system is in the process of evaluating a machine learning model. The following steps are shown:

* Create MLContext: This step creates an MLContext object, which is used to manage the machine learning pipeline.
* Read the training data: This step reads the training data into the MLContext object. The training data is a set of examples of past purchases, which is used to train the machine learning model.
* Transform data: This step transforms the training data into a format that is compatible with the machine learning model.
* Create pipeline: This step creates a machine learning pipeline. The pipeline consists of a sequence of steps that are used to train and evaluate the machine learning model.
* Train the model: This step trains the machine learning model on the training data.
* Evaluate model: This step evaluates the machine learning model on a held-out test set. The test set is a set of examples that were not used to train the model.
* Test single prediction: This step tests the machine learning model on a single example. The example is a customer ID and a product ID. The model predicts the probability that the customer will purchase the product.
* Consume: This step consumes the machine learning model. The model can be used to generate product recommendations for customers.

**The evaluation metrics are:**

* Loss function: The loss function is a measure of how well the machine learning model performs on the test set. A lower loss function indicates that the model is performing better.
* Mean Absolute Error (MAE): The MAE is a measure of the average difference between the predicted and actual values. A lower MAE indicates that the model is making more accurate predictions.
* Root Mean Squared Error (RMSE): The RMSE is a measure of the square root of the average squared difference between the predicted and actual values. A lower RMSE indicates that the model is making more accurate predictions.
* Mean Squared Error (MSE): The MSE is a measure of the average squared difference between the predicted and actual values. A lower MSE indicates that the model is making more accurate predictions.

**Here is all customers and theirs product recommendation ratings:**

A screenshot of a computer

Description automatically generated

Figure 13: Product recommendation ratings

* **Function: Customer cart**

A group of containers of liquid

Description automatically generated

Figure 14: Customers cart

A screenshot of a shopping receipt

Description automatically generated

Figure 15: Cart Total price

The customer cart page is a crucial component of any e-commerce website, serving as the virtual shopping basket where customers can review and manage the items they intend to purchase. It plays a pivotal role in the checkout process, providing a clear overview of the items, their quantities, and the overall order total before proceeding to payment.

* Product Summary:
* Product Image: A visual representation of the product, allowing customers to easily identify and recall the items they have added to their cart.
* Product Name: A clear and concise label of the product, ensuring customers can distinguish between similar items without confusion.
* Product Quantity: The number of units of each product in the cart, enabling customers to adjust quantities as needed.
* Product Price: The individual price of each product, providing customers with a breakdown of the cost of each item.
* Order Total:
* Subtotal: The total price of all items in the cart, excluding any applicable taxes or shipping costs.
* Order Total: The final price of the entire order, including all items, taxes, and shipping charges.
* **Function Billing:**

A screenshot of a receipt

Description automatically generated

Figure 16: Bill details

### 4.1.2 Employee

* **Function: Login**

A screenshot of a login page

Description automatically generated

Figure 17: Employee Login screen

The employee login function allows employees to access their company accounts. To log in, employees must enter their username and password in the corresponding fields and click the "Sign In" button. If the credentials are correct, the employee will be logged in to their account. Otherwise, an error message will be displayed.

* **Admin Homepage**

A screenshot of a computer

Description automatically generated

Figure 18: Admin Homepage

An admin home page that is specifically focused on product management. The page would be used to manage aspects of the products on a website, such as:

* Adding new products
* Editing existing products
* Deleting products
* **Function: Create a product**

A screenshot of a computer

Description automatically generated

Figure 19: Create a product screen

The function Create Product is to allow users to create a new product on the website. This can be done by entering the following information:

* Product Name: The name of the product.
* Product Description: A detailed description of the product, including its features, benefits, and specifications.
* Product Images: Multiple images of the product, from different angles.
* Product Category: The category that the product belongs to, such as clothing, electronics, or food.
* Product Price: The price of the product.
* Product Inventory: The number of units of the product that are in stock.

Once the user has entered all of the required information, they can click the Create Product button to create the new product. The product will then be added to the website and made available for purchase.

The Create Product button is an important feature for any e-commerce website, as it allows users to add new products to their store.

* **Function: Edit product**

A screenshot of a recipe

Description automatically generated

Figure 20: Edit product screen

The editing function is accessed by clicking on an "Edit" button next to a product listing.

Users can modify various product attributes, including:

* Product Name: Updating the product's name to reflect changes or clarify its identity.
* Product Description: Revising the product description to provide more accurate, up-to-date information or enhance its appeal to customers.
* Product Images: Replacing or adding new product images to showcase the product from different angles or provide a better visual representation.
* Product Category: Assigning the product to a different category if its classification has changed or to align it with customer search behavior.
* Product Price: Updating the product price to reflect cost changes, promotions, or market adjustments.
* Product Inventory: Adjusting the product inventory levels to ensure availability and prevent overselling or stockouts.

## 4.2 Mobile application base on Android

In the future, we will continue developing our project on both website and app platforms. We will optimize existing functionalities and introduce new ones, aiming to enhance the user experience to make it more engaging. Through these efforts, our goal is to create a more versatile and feature-rich application that truly caters to our users' needs. We hope to receive positive feedback to further our development in the future.

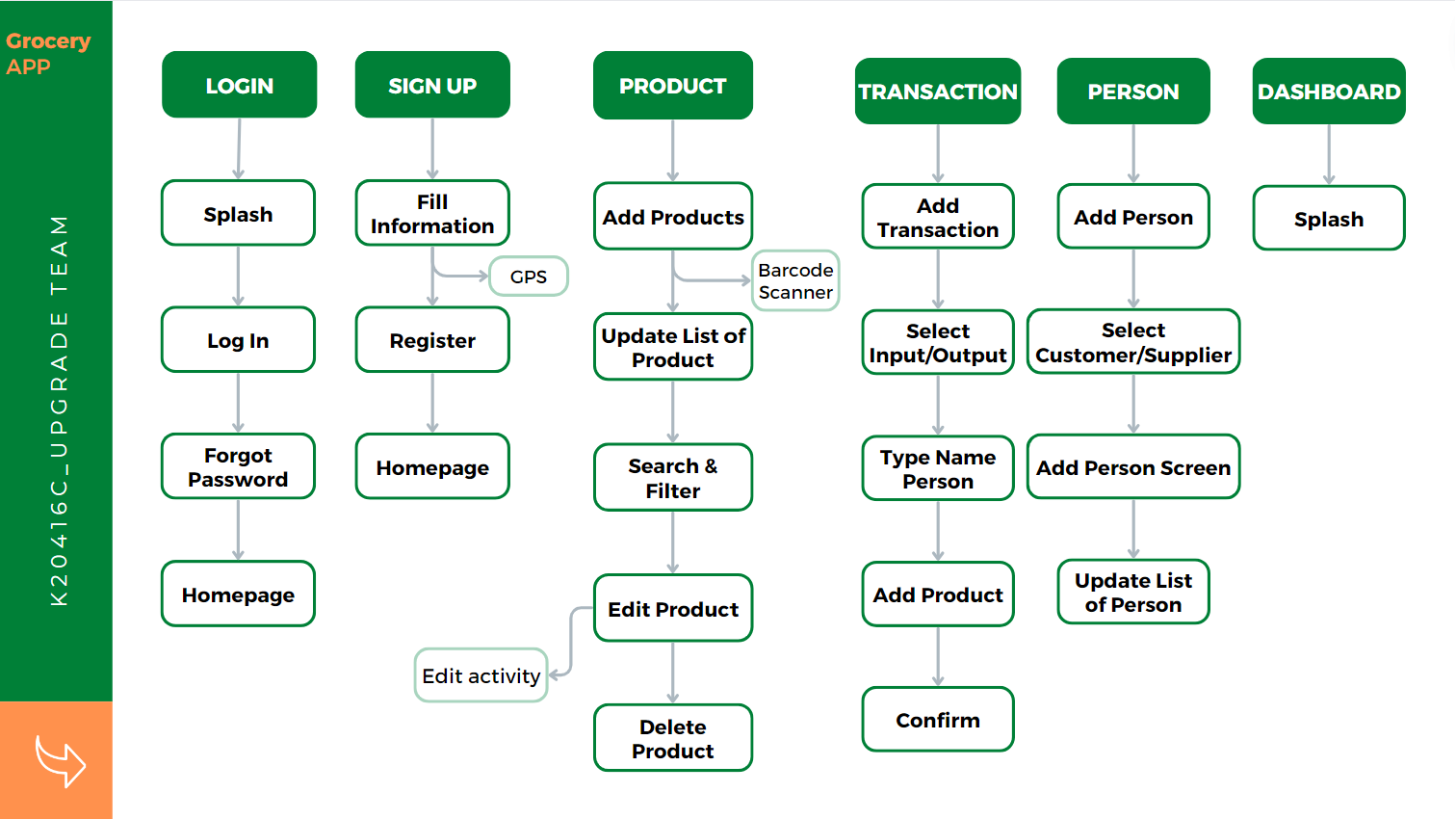


Figure 21: Task flow of Grocery store’s mobile application

Based on the task flow diagram, the application provides an overview of the operational content of each function within the application, specifically:

* Product Management: This is integrated with computer vision in barcode scanning capacity that allow users to input, edit, and search for product information quickly and accurately.
* Transaction Management: This allows users to create orders and control product inventory through operations on two tabs - the Input tab and the Output tab. For each transaction, users can scan the barcode for one or more products and enter the quantity of each type. The system will automatically store the time and other information of the transaction.
* Human Resource Management: This allows the storage of information about loyal customers and suppliers, facilitating the analysis and development of relationships between the store and users, and between the store and suppliers.
* Performance: This allows the creation of visual reports to help store management improve and develop business capabilities.

### 4.2.1 Login and Register

* **Function login**

The Login function is a crucial feature that ensures the security of our application. Users are required to fill in their registered email and password to access their accounts. In case a user forgets their password, we have incorporated a ‘Forgot Password’ option to assist in recovering their account swiftly and securely.

A screenshot of a login form

Description automatically generated

Figure 22: Login screen

A screenshot of a login form

Description automatically generated

Figure 23: Reset Password screen

As mentioned, users forgot their password can click on “Forgot password” and fill their signed up email then our system will automatically send them a electronic construction to their email address.

* **Function Register**

Our Register function is designed to provide a seamless onboarding experience for new users. During the registration process, users are requested to turn on their GPS. This requirement is aimed at capturing the exact location of the user, which in turn saves time and enhances the user experience. Once the registration is successful, users are automatically redirected to the Homepage, marking the beginning of their journey with our application.

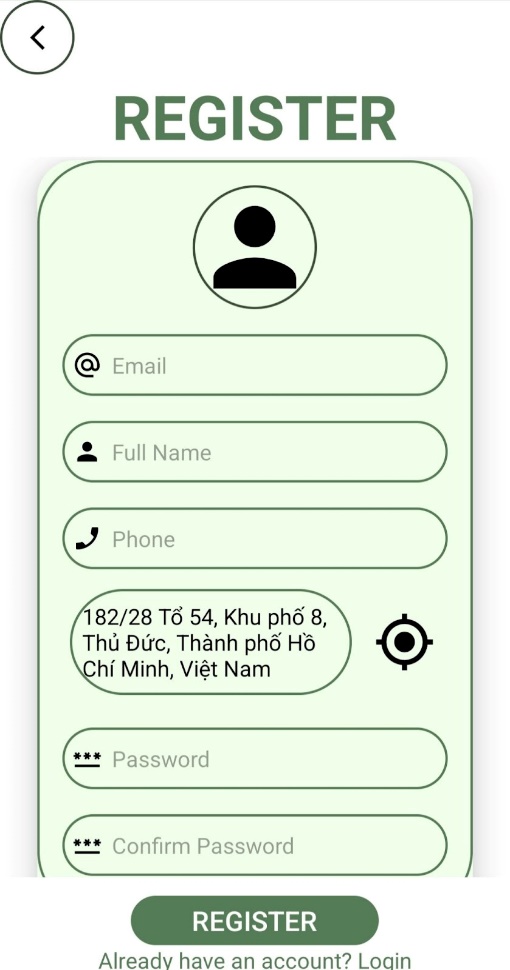


Figure 24: Register screen

### 4.2.2 Product management

The primary purpose of our mobile application is to include a barcode scanner, which is an important component in sales and product services owing to the need of keeping track of all objects in one location. Many ways have been implemented to make the barcode reading process more user-friendly. Using image processing, our program successfully constructed a barcode recognition system. By taking an image with a webcam, the system will be able to read barcodes.

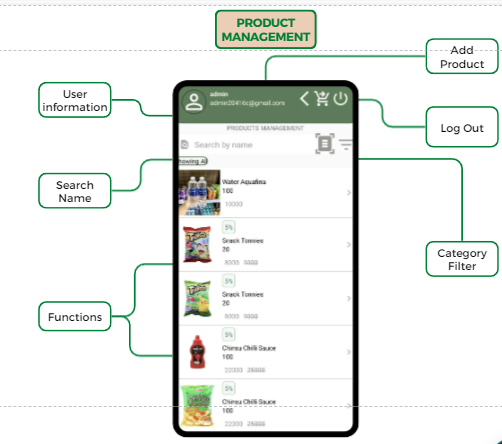


Figure 25: Splash of Product management

At first, users can create a new product by filled completely necessary fields, such as: product name, product description, quantity, original price, category, and its barcode. By the way, users can manage products’ promotion by switching on or off discount mode. If discount is available, sellers need to complete discount price and optional discount note fields. In our application, we allow users to insert products’ image to easily recognize and categorize products and this field is an optional one.

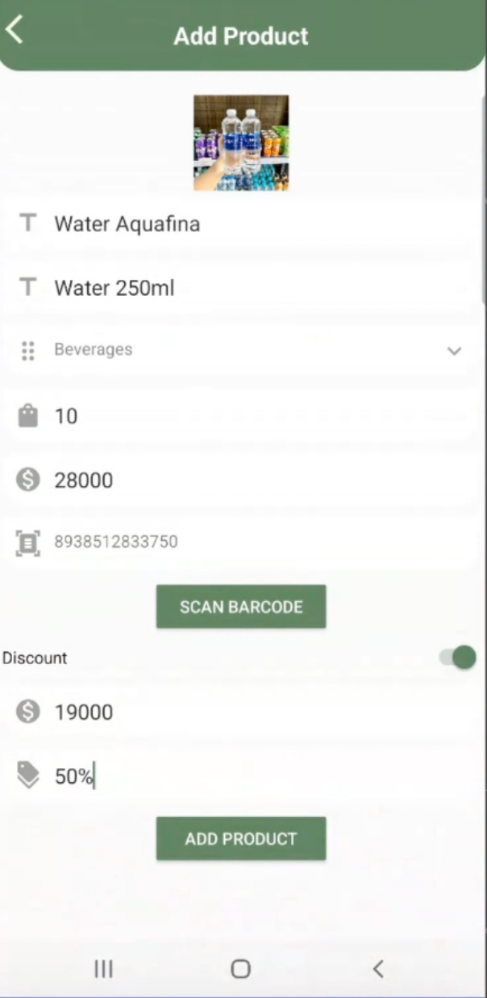


Figure 26: Add product functionality

After adding products, they will be presented in the splash screen of product management. Employees can search for products by their name and also leverage barcode scanner for faster look up. Furthermore, filtering by categories is available to used in our application.

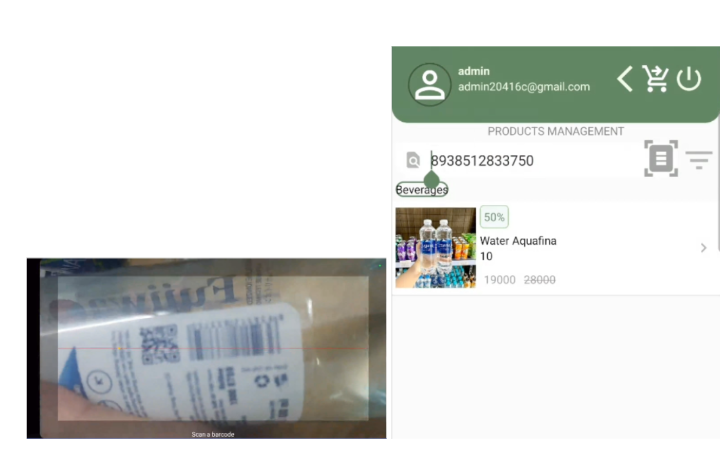


Figure 27: Search products by scanning barcode

A screenshot of a phone

Description automatically generated

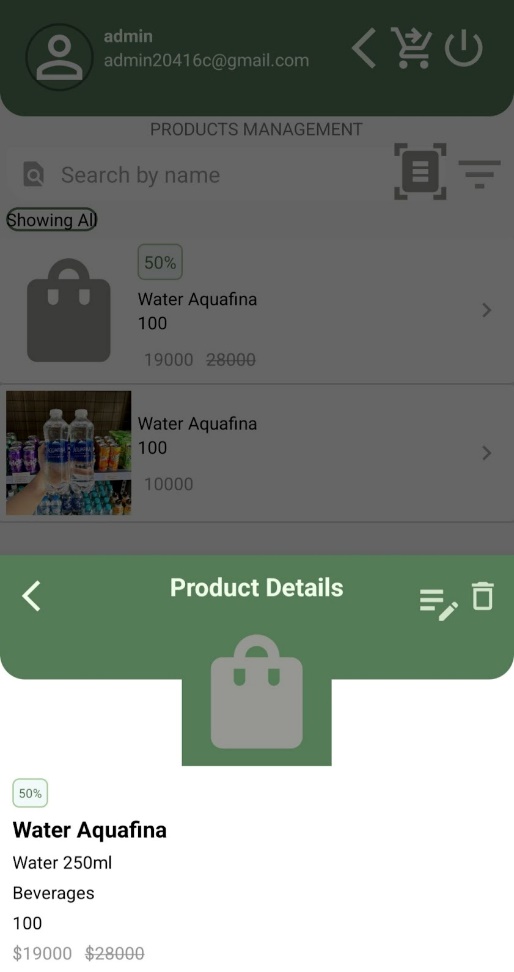
Figure 28: Search products by product’s name

A screenshot of a menu

Description automatically generated

Figure 29; Filter by products’ categories

To edit or delete a product, users can select their target items and its detailed information will be display, which allows users to validate products’ information before making any further action.

 A screenshot of a phone

Description automatically generated

Figure 30: Product’s edit information screen.

*A screenshot of a phone

Description automatically generated*

Figure 31; Delete a product

All changes will be update real-time in the Firebase database.

### 4.2.3 Other functions

* **Function: Transaction management**

The Transaction Management function is a comprehensive feature designed to streamline the process of order creation and inventory control. This function operates through two distinct tabs - the Input tab and the Output tab.

* Input Tab: This tab is designed for users to add new products into the system. Users can scan the barcode of one or more products and enter the quantity of each type. This information is then added to the inventory in real-time, ensuring that the inventory data is always up-to-date.
* Output Tab: This tab is used when a product is sold or removed from the inventory. Similar to the Input tab, users can scan the barcode of the product(s) and enter the quantity being removed. This ensures that the inventory reflects the most recent transactions.

For each transaction, whether it’s adding to or removing from the inventory, the system automatically stores crucial information such as the time of the transaction, the products involved, and the quantities. This data is invaluable for tracking inventory changes over time and can aid in future decision-making processes.

This function is designed to provide a seamless and efficient method for managing transactions, thereby enhancing the overall user experience and operational efficiency.

*A screenshot of a phone

Description automatically generated A screenshot of a phone

Description automatically generated*

Figure 32: Transaction Management

* **Function Human Resource Management (Person)**

The Human Resource Management function, also referred to as the Person function, is a vital feature of our application that focuses on managing relationships with two key stakeholders - loyal customers and suppliers.

* Loyal Customers: This function allows the storage of detailed information about loyal customers, including their purchasing habits, preferences, and feedback. This data can be analyzed to understand customer behavior better, tailor services to meet their needs, and strengthen the relationship between the store and its customers.
* Suppliers: Similarly, the function also facilitates the storage of comprehensive information about suppliers. This includes details about the products they supply, their delivery schedules, and their terms and conditions. Having this information at hand helps in maintaining a smooth supply chain and fostering a strong and mutually beneficial relationship with suppliers.

By facilitating the analysis and development of relationships between the store and its users, and between the store and its suppliers, the Human Resource Management function plays a crucial role in the overall growth and success of the business. It ensures that the store can meet its customers’ needs effectively while maintaining a steady and reliable supply of products.

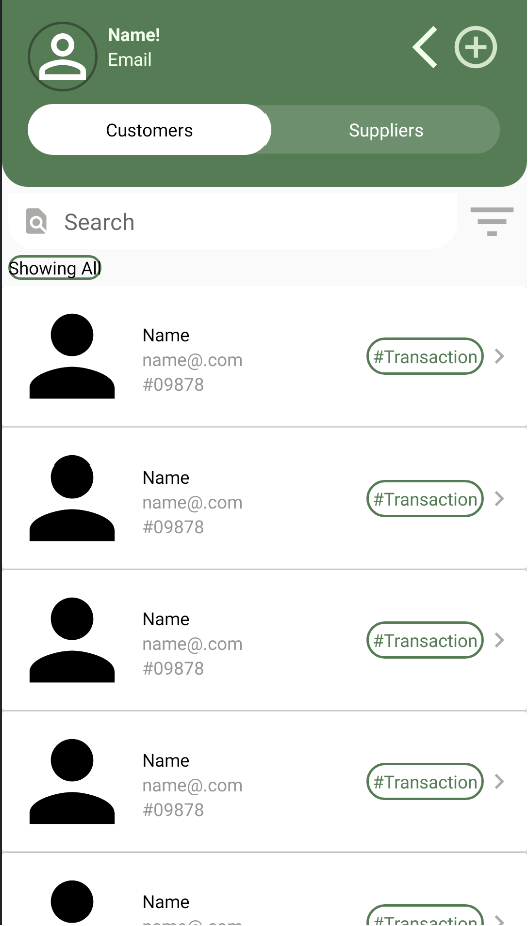


Figure 33: Splash of Person Management

A screenshot of a phone

Description automatically generated

Figure 34: Add customers in Person management

* **Perfomance management**

The Performance function is a powerful feature that enables the creation of visual reports, providing a clear and concise view of the store’s operations. This function is designed to aid store management in improving and developing their business capabilities.

* Visual Reports: The function allows the generation of visual reports based on various parameters such as sales, inventory, customer behavior, and supplier performance. These reports present data in an easy-to-understand format, making it simpler for management to identify trends, patterns, and areas of concern.
* Business Improvement: By providing a clear picture of the store’s performance, these visual reports can guide management in making informed decisions to improve various aspects of the business. This could include optimizing inventory levels, enhancing customer service, or improving supplier relationships.
* Business Development: Beyond just improvement, the insights gained from these reports can also aid in the development of new business strategies. Whether it’s identifying new market opportunities, planning promotional campaigns, or expanding the product range, the Performance function provides the data needed to drive business growth.

In essence, the Performance function serves as a valuable tool for store management, offering the insights needed to not just maintain, but also enhance and grow the business. It ensures that the store is always moving forward, adapting to changes, and striving for success.

A screenshot of a graph

Description automatically generated

Figure 35: Performance function

As mentioned above, these functions are not ready for serving due to an array of technological problems, which will be handled and updated soon in the near future.

### 

### CHAPTER 5: CONCLUSIONS AND FUTURE WORKS

## 5.1 Results

After a period of learning and hands-on practice, our team has completed our project. Through this endeavor, we've developed an application for the Android platform and a website. Within these, we've implemented a recommended system to offer suggestions to users during their shopping experience. This project has granted us some insight into platforms like mobile apps and web development. Despite being a completed project, we acknowledge areas that require improvement and modification to enhance our application for the future. However, due to limited resources and time constraints, we haven't fully realized the development of an efficient application.

## 5.2 Limitations

Despite the successful development and deployment of the project, certain limitations persist. Due to time constraints, the application hasn't been fully optimized or diversified, and its functionalities remain at a basic level. Furthermore, the application's interface lacks aesthetic appeal and falls short in complete optimization for user experience. Additionally, unforeseen errors within the application are inevitable. We will optimize and upgrade the project in the future to address these issues.

## 5.3 Future works

In the future, we will continue developing our project on both website and app platforms. We will optimize existing functionalities and introduce new ones, aiming to enhance the user experience to make it more engaging. Through these efforts, our goal is to create a more versatile and feature-rich application that truly caters to our users' needs. We hope to receive positive feedback to further our development in the future.

# REFERENCES

[1] <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core>

[2] <https://learn.microsoft.com/en-us/dotnet/machine-learning/how-does-mldotnet-work>

[3] <https://firebase.google.com/docs?hl=vi>

# APPENDIX

1. [The link of the detailed pictures in this report.](https://drive.google.com/drive/folders/1XSXBj0lDKxCW34kfETENjkH6gvjWEcy-?fbclid=IwAR1yNFDhb1wQl41rqaePkUWPtlFxRTQZA_ysY2V6OB9iLJpL5JgXQTz5QZ4)