**FINAL PROJECT DESCRIPTION FOR PRN231 COURSE**

**SPRING2024**

**Group name: 08 – Supermarket Management**

**Members of Group:**

**1. Trần Đại Nghĩa – HE163119**

**2. Nguyễn Việt Anh - HE163837**

**3. Chu Thành Duy - HE161399**

Table of Contents

[I. Introduction 3](#_Toc162297346)

[**1.1** **Purpose** 3](#_Toc162297347)

[**1.2** **Scope** 3](#_Toc162297348)

[**1.3** **Stakeholders** 4](#_Toc162297349)

[**1.4** **User interface** 4](#_Toc162297350)

[**1.5** **Platform** 5](#_Toc162297351)

[**1.6** **Technology** 6](#_Toc162297352)

[II. Project Objectives 6](#_Toc162297353)

[III. Database Design 9](#_Toc162297354)

[3.1 **Tables in your database** 9](#_Toc162297355)

[**3.2 Relationship between the tables…** 11](#_Toc162297356)

[IV. Main Functions of Application 12](#_Toc162297357)

[V. Role of members 14](#_Toc162297358)

[VI. Plan for Project (3 weeks) 14](#_Toc162297359)

[TEST CASE RESULTS 16](#_Toc162297360)

[REFERENCE MATERIALS 20](#_Toc162297361)

# I. Introduction

* 1. **Purpose**

The purpose of this document defines the requirements for a supermarket management software system. This document will be used by stakeholders, including customers, developers, and testers, to ensure that the system meets their needs.

* 1. **Scope**
* The supermarket management software system will provide the following functionalities:
* Product management: The system will allow users to manage product information, including product Id, product name, product description, price, etc.
* Customer management: The system will allow users to manage customer information, including customer Id, customer name, customer phone, etc.
* Category management: The system will allow users to manage category information, including category Id, category name, category description.
* Supplier management: The system will allow users to manage supplier information, including supplier Id, supplier name, supplier address, etc.
* Sale Transaction management: The system will allow users to manage customer orders, including customer information, product information, payment information, etc.
* Employee management: The system will allow users to manage employee information, including employee personal information, account information, and permission information, etc.
* Warehouse management: The system will allow users to manage warehouse information, including product information in the warehouse, inventory information, etc.
* Statistical reporting: The system will allow users to generate statistical reports on the supermarket's operations, including revenue, expenses, etc.
  1. **Stakeholders**

The key stakeholders of the system include:

* Customer: The supermarket is the primary customer of the system. They will use the system to manage their operations.
* Developer: The developer will be responsible for building the system. They will need to understand the customer's requirements to build a system that meets their needs.
* Tester: The tester will be responsible for testing the system to ensure that the system meets the requirements.
  1. **User interface**
* User-Friendly Interface:
  + The user interface of the supermarket management system should be designed to be intuitive and easy to navigate without the need for extensive instructions.
  + Functions and features should be organized and presented logically and clearly to optimize the user experience.
* Responsive Interface:
  + The user interface should be able to adapt flexibly to different screen sizes, including desktops, tablets, and mobile phones.
  + Interface components should automatically adjust to fit and remain user-friendly across various screen sizes.
  1. **Platform**
* Programming Languages:
  + The primary programming languages used will be C# and ASP.NET for server-side development.
  + HTML, CSS, and JavaScript will be utilized for client-side development to enhance user interactivity and presentation.
* Framework and Technologies:
  + The application will utilize the ASP.NET MVC (Model-View-Controller) framework for structuring and organizing codebase.
  + Entity Framework will be employed for database access and management.
  + AJAX (Asynchronous JavaScript and XML) will be used to implement asynchronous communication between the client and server, enhancing the responsiveness of the application.
  + Bootstrap framework will be utilized for front-end styling and responsive design.
* Database:
  + Microsoft SQL Server will be the chosen database management system for storing and managing data related to the supermarket management system.
* Development Tools:
  + Visual Studio IDE (Integrated Development Environment) will be used as the primary development tool for coding, debugging, and testing the application.
  + Git will be used for version control to manage codebase changes and collaborate among development team members.
* Deployment Environment:
  + The application will be deployed on a web server running Windows operating system, with IIS (Internet Information Services) as the web server software.
  1. **Technology**

The web application will use the following technologies:

* ASP.NET Core
* SQL Server
* Microsoft Web Code Generation
* Microsoft Extensions
* Microsoft Code Analysis
* Authentication
* Authorization
* Odata
* Identity
* Jwt
* Ajax

# II. Project Objectives

In this collaborative project, Group 08 is tasked with developing a comprehensive system for inventory management and dashboard. The project scope includes the implementation of CRUD (Create, Read, Update, Delete) functionality, robust inventory tracking features, dashboard visualization, and user management.

CRUD Functionality:

* Develop a user-friendly interface allowing seamless creation, viewing, updating, and deletion of inventory items.
* Implement a responsive design to ensure optimal user experience across various devices.
* Ensure secure and efficient handling of inventory data through CRUD operations.

Inventory Management:

* Design and implement inventory tracking features to monitor stock levels, track movement, and manage inventory across multiple locations.
* Integrate barcode scanning functionality for quick and accurate inventory management.
* Implement alerts and notifications for low stock levels, expiration dates, and inventory discrepancies.

Dashboard Visualization:

* Develop a dashboard to provide users with visual insights into key inventory metrics, such as stock levels, turnover rates, and revenue.
* Utilize charts, graphs, and tables to present data in an intuitive and easy-to-understand format.
* Customize dashboard views and allow users to filter and drill down into specific inventory data.

User Management:

* Create a user authentication and authorization system for managing user access and permissions.
* Implement user profiles with the ability to update personal information and preferences.
* Ensure account security through secure authentication practices, such as password encryption.

Testing and Quality Assurance:

* Conduct thorough testing of all features to identify and resolve any bugs or issues.
* Perform performance testing to ensure the system can handle large volumes of inventory data efficiently.
* Regularly update and maintain the system to address evolving needs and potential vulnerabilities.

Documentation:

* Provide comprehensive documentation for the system architecture, database schema, and API usage.
* Document the user guide to assist end-users in navigating and utilizing the inventory management system effectively.

Business rule:

|  |  |
| --- | --- |
| **ID** | **Business Rule Definition** |
| BR-01 | Email is a unique and valid format. |
| BR-02 | Phone number is not empty, unique, and has ten digits. |
| BR-03 | Passwords must have at least eight characters and be alphanumeric. |
| BR-04 | The new password must match the confirmed password. |
| BR-05 | Name is not empty and is less than 51 characters. |
| BR-06 | The web can only be active when it is connected to the internet and server. |

# III. Database Design

## 3.1 **Tables in your database**

Table Name: Products

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| ProductID | int | Primary Key | Mã sản phẩm |
| ProductName | String(200 chars) |  | Tên sản phẩm |
| SupplierID | Int | Foreign Key | Mã nhà cung cấp |
| CategoryID | Int | Foreign Key | Mã loại hàng |
| Description | String(450 chars) | Allow Nulls | Mô tả sản phẩm |
| Discontinued | boolean |  | Trạng thái sản phẩm |
| TotalQuantity | int |  | Tổng số lượng sản phẩm |
| Price | Double |  | Giá sản phẩm |

Table Name: Categories

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| CategoryID | Int | Primary Key | Mã loại hàng |
| CategoryName | String(200 chars) |  | Loại hàng |
| Description | String(450 chars) | Allow Nulls | Mô tả loại hàng |
| Discontinued | Boolean |  | Trạng thái loại hàng |

Table Name: Suppliers

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| SupplierID | Int | Primary Key | Mã nhà cung cấp |
| CompanyName | String(450 chars) |  | Tên công ty cung cấp |
| Address | String(450 chars) |  | Địa chỉa nhà cung cấp |
| Phone | String(11 chars) |  | SĐT nhà cung cấp |
| Fax | String(11 chars) |  | Số Fax nhà cung cấp |
| HomePage | String(450 chars) |  | Trang chủ nhà cung cấp |
| Discontinued | Boolean |  | Trạng thái |

Table Name: Inventory

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| InventoryID | Int | Primary Key | Mã kho hàng |
| ProductID | Int | Foreign Key | Mã Sản phẩm |
| Quantity | Int |  | Số lượng |
| PurchasePrice | Double |  | Giá nhập |
| EntryDate | Date |  | Ngày nhập |
| EmployeeID | String | Foreign Key | ID nhân viên |

Table Name: Customers

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| CustomerID | Int | Primary Key | ID khách hàng |
| LastName | String(50 chars) |  | Tên khách hang |
| FirstName | String(50 chars) |  | Họ khách hàng |
| Address | String(450 chars) |  | Địa chỉ khách hàng |
| Phone | String(11 chars) |  | SĐT khách hàng |
| Email | String(50 chars) |  | Email khách hàng |

Table Name: SalesTransactions

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| TransactionID | Int | Primary Key | Mã giao dịch |
| CustomerID | Int | Foreign Key | ID khách hàng |
| EmployeeID | String | Foreign Key | ID nhân viên |
| TransactionDate | Date |  | Ngày giao dịch |
| CashRecieved | Double |  | Tiện nhận |
| Change | Double |  | Tiền thừa |
| TotalPrice | Double |  | Tổng đơn giá |
| Discontinued | Boolean |  | Trạng thái |

Table Name: TransactionDetails

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| TransactionID | Int | Foreign Key, Primary Key | Mã giao dịch |
| ProductID | Int | Foreign Key, Primary Key | Mã sản phẩm |
| Subtotal | Double |  | Giá |
| Quantity | Int |  | Số lượng |
| Discount | Double | Allow Nulls | Giảm giá |

Table Name: Employees

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** |  |  |
| EmployeeID | String | Primary Key | Mã nhân viên |
| LastName | String(50 chars) |  | Tên nhân viên |
| FirstName | String(50 chars) |  | Họ nhân viên |
| DoB | Date |  | Ngày sinh nhân viên |
| HireDate | Date |  | Ngày nhận việc |
| Address | String(450 chars) |  | Địa chỉ nhân viên |
| Phone | String(11 chars) |  | SĐT nhân viên |
| Discontinued | Boolean |  | Trạng thái |

## **3.2 Relationship between the tables…**

* The "Product" table is linked to the "Suppliers" table through the foreign key "SupplierID."
* The "Product" table is linked to the "Category" table through the foreign key "CategoryID."
* The "Inventory" table is linked to the "Product" table through the foreign key "ProductID" and is also linked to the "Employees" table through the foreign key "EmployeeID."
* The "Batch" table is linked to the "Product" table through the foreign key "ProductID" and is also linked to the "Inventory" table through the foreign key "InventoryID."
* The "SaleTransactions" table is linked to the "Customers" table through the foreign key "CustomerID" and is also linked to the "Employees" table through the foreign key "EmployeeID."
* The "TransactionDetails" table is linked to the "SalesTransactions" table through the foreign key "TransactionID" and is also linked to the "Product" table through the foreign key "ProductID."

# IV. Main Functions of Application

The supermarket management web application will provide the following main functions for both Admin, Employees, and Warehouse:

For Admin:

* Product management:
  + Add, delete, and edit product information (product name, description, image, price, promotion, etc.).
  + Categorize and sort products by different criteria (product type, brand, supplier, etc.).
  + Check inventory and update product quantities.
* Employee management:
  + Add, delete, and edit employee information (employee name, role, permissions, etc.).
  + Track and evaluate employee performance.
* Customer management:
  + Add, delete, and edit employee information (customer name, phone, etc.).
  + Track and evaluate customer performance.
* Category management:
  + Add, delete, and edit category information (category name, description, etc.).
  + Track and evaluate category performance.
* Supplier management:
  + Add, delete, and edit supplier information (supplier name, description, etc.).
  + Track and evaluate supplier performance.
* Dashboard management:
* View and export reports on the supermarket's revenue and expenses.
* Analyze and summarize data on sales, profits, expenses, customers, and products.
* Control the supermarket's financial situation.
* Sale Transaction management:
* Add, delete, and edit sale transaction information (product name, customer name, etc.).
* Track and evaluate supplier performance.

For Employees:

* Customer checkout:
* Enter customer order information.
* Process different payment methods (cash, credit/debit cards, etc.).
* Print sales receipts.
* Customer service:
* Answer customer questions about products and services.
* Help customers find the products they need.
* Resolve customer issues related to orders or payments.

**-** Sale Transaction management:

* Payment creates invoices for customers.

For Warehouse:

* Inventory management:
  + Receive new inventory from suppliers.
  + Ship inventory to customers.
  + Check inventory and update product quantities.
* Category management:
  + Add, delete, and edit category information (category name, description, etc.).
  + Track and evaluate category performance.
* Supplier management:
  + Add, delete, and edit supplier information (supplier name, description, etc.).
  + Track and evaluate supplier performance.
* Product management:
  + Add product information (product name, description, image, price, promotion, etc.).
  + Check inventory and update product quantities.

# V. Role of members

Table – tasks of member

|  |  |  |
| --- | --- | --- |
| **Full Name** | **StudentId** | **Role** |
| Trần Đại Nghĩa | HE163119 | Leader, BA, DEV, Tester |
| Chu Thành Duy | HE161399 | BA, DEV, Tester |
| Nguyễn Việt Anh | HE163837 | BA, DEV, Tester |

# VI. Plan for Project (3 weeks)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Member** | **Time** |
| 01 | Create group | All members | Week 1  (04/01/2024) |
| 02 | Create idea project | All members | Week 1  (04/01/2024) |
| 03 | Create maintain functions | All members | Week 1  (06/01/2024) |
| 04 | Create idea database | All members | Week 2  (08/01/2024) |
| 05 | Create Database design | All members | Week 2  (08/01/2024) |
| 06 | Create project description | All members | Week 2  (11/01/2024) |
| 07 | Create test case | All members | Week 2  (13/01/2024) |
| 08 | Create GitHub project | All members | Week 3  (15/01/2024) |
| 09 | Design Database | All members | Week 3  (18/01/2024) |
| 10 | Build Business Object | All member | Week 4  (25/01/2024) |
| 11 | Build Data Access | All member | Week 5 + 6  (23/02/2024) |
| 12 | Build Project Api | All member | Week 7 + 8 + 9  (16/03/2024) |
| 13 | Build Project Client | All member | Week 7 + 8 +9 +10  (18/03/2024) |

# TEST CASE RESULTS

**Test API**

|  |  |  |
| --- | --- | --- |
| **API** | **Response** | **Response Detail** |
| Get: /api/Products | Success | View list products Json |
| Get: /api/products/{id} | Success | View product Json |
|  | Fail | Error |
| Post:/api/Products/{id} | Success | Add new product |
|  | Fail | Error |
| Put: /api/Products/{id} | Success | Update product |
|  | Fail | Error |
| Delete: /api/Products/{id} | Success | Delete product |
|  | Fail | Error |
| Get: /api/Categories | Succes | View list categories Json |
| Get: /api/Categories/{id} | Success | View category Json |
|  | Fail | Error |
| Post: /api/Categories/{id} | Success | Create new category |
|  | Fail | Error |
| Put: /api/Categories/{id} | Success | Update category |
|  | Fail | Error |
| Delete: /api/Categories/{id} | Success | Delete category |
|  | Fail | Error |
| Get: /api/Suppliers | Succes | View list suppliers Json |
| Get: /api/Suppliers/{id} | Success | View supplier Json |
|  | Fail | Error |
| Post: /api/Suppliers/{id} | Success | Create new supplier |
|  | Fail | Error |
| Put: /api/Suppliers/{id} | Success | Update supplier |
|  | Fail | Error |
| Delete: /api/Suppliers/{id} | Success | Delete supplier |
|  | Fail | Error |
| Get: /api/Batchs/{id} | Success | View batch Json |
|  | Fail | Error |
| Post: /api/Batchs/{id} | Success | Create new batch |
|  | Fail | Error |
| Put: /api/Batchs/{id} | Success | Update batch |
|  | Fail | Error |
| Get: /api/Customers | Succes | View list customers Json |
| Get: /api/Customers/{id} | Success | View customer Json |
|  | Fail | Error |
| Post: /api/Customers/{id} | Success | Create new customer |
|  | Fail | Error |
| Put: /api/Customers/{id} | Success | Update customer |
|  | Fail | Error |
| Get: /api/Employees | Succes | View list employees Json |
| Get: /api/Employees/{id} | Success | View employee Json |
|  | Fail | Error |
| Post: /api/Employees/{id} | Success | Create new employee |
|  | Fail | Error |
| Put: /api/Employees/{id} | Success | Update employee |
|  | Fail | Error |

**Test Client**

|  |  |  |
| --- | --- | --- |
| Account | Login | Functions |
| Employee | Fail | Inform Invalid Email or Password |
| Successful | View Sale Transaction Page |
| Create Customer |
| Admin | Fail | Inform Invalid Email or Password |
| Successful | Category Management   * CRUD * Search by Name |
| Account Management   * Active/Inactive User * View List User |
| Supplier Management   * CRUD * Search by Name |
| Product Management   * CRUD * Search by Name |
| View Dashboard |
| Inventory Management   * CRUD * Search by Name |
| Warehouse | Fail | Inform Invalid Email or Password |
|  | Success | Inventory Management   * CRUD * Search by Name |
|  |  | Category Management   * CRUD * Search by Name |
|  |  | Supplier Management   * CRUD * Search by Name |
|  |  | Product Management   * CRUD * Search by Name |

# REFERENCE MATERIALS

[**https://stackoverflow.com/**](https://stackoverflow.com/)

[**https://github.com/**](https://github.com/)

[**https://learn.microsoft.com/**](https://learn.microsoft.com/)

[**https://www.udemy.com/**](https://www.udemy.com/)

**https://www.javatpoint.com/**