

EPSILON NET

Coding School

Το πρώτο **Epsilon Net Coding School**
σε τεχνολογίες **.NET** είναι γεγονός!




EPSILONNET
GROUP OF COMPANIES


EPSILONHR
MEMBER OF EPSILONNET - GROUP OF COMPANIES

 **data**
communication
MEMBER OF
EPSILONNET GROUP OF COMPANIES


EPSILON SINGULARLOGIC
MEMBER OF EPSILONNET GROUP OF COMPANIES

FINAL

Implement a **Fuel Station Solution** following these guidelines:

1. **A Fuel Station** is a system consisting of 3 applications:
 - **Web API** (Backend)
 - **Web Client** (Blazor Wasm)
 - **Windows Client** (WinForms)
2. **The Fuel Station system** requires the following **entities**:
 - **Customer**: ID, Name, Surname, CardNumber
 - **Item**: *ID*, Code, Description, **ItemType (enum)**, Price, Cost
 - **Employee**: ID, Name, Surname, HireDateStart, HireDateEnd, SalaryPerMonth, **EmployeeType (enum)**
 - **Transaction**: ID, Date, EmployeeID, CustomerID, **PaymentMethod (enum)**, TotalValue
 - **TransactionLine**: ID, TransactionID, ItemID, Quantity, ItemPrice, NetValue, DiscountPercent, DiscountValue, TotalValue.
 - **Ledger**: Year, Month, Income, Expenses, Total
3. **The backend** should be an ASP.NET Core Web API, used for both Web and Windows Client using http calls.
4. **Data storage** should be implemented using **Entity Framework**.

5. **Web Client** (Blazor Web Assembly) should include the following

UI Screens supporting the **CRUD** operations for:

- Customers
- Items
- Employees
- Ledger (read-only)

6. **Windows Client** should include the following UI Screens

supporting the **CRUD** operations for:

- Customers
- Items
- Transactions and Transaction Lines.

7. **Business Rules**

- **Managers** can create and view all Entities.
- **Cashiers** can create and view only **Customers, Transactions and Transaction Lines**.
- **Staff** can create and view only **Items**.
- A Customer Card Number should be Unique and should be starting with the letter **"A"**.
- An Item Code should be Unique.
- **Item Type** Enum has 3 values: **Fuel, Product, Service**
- **Employee Type** Enum has 3 values: **Manager, Staff, Cashier**
- **Payment Method** Enum has 2 values: **Credit Card, Cash**

- A Transaction begins by entering the **CardNumber** of the customer, then it should automatically find the customer. If a customer is not found, then we should be able to create one.
- A **Transaction** should have **only one** Customer but can have **multiple Items**. Only **one** of those items can be **Fuel** type per transaction.
- **Net Value** is calculated by multiplying the quantity with the item price.
- **Discount Value** is calculated by multiplying Net Value by Discount Percent.
- **Total Value** is calculated by subtracting Discount Value from the Net Value.
- If a Transaction Line includes an item of the type **Fuel** and its **NetValue is above 20 euros**, then we calculate a **10% discount** on that transaction line.
- If the **TotalValue** of the transaction is above **50 Euros**, the only acceptable payment method is **Cash**.
- In the **Monthly ledger** screen, we need to input Year and Month and calculate income, expenses and total.
- The Renting Cost of the Fuel Station will be set by the **Manager** and the initial value is **5000** Euros per Month.