

WareHouse Project - Nir Banchik, HackerU Project.

This project came to show the ability to create a Backend System for WareHouse use.
The ability of the system is to manage employees,customers,items and inventory.

This Project Uses:

C# as a programming language.

MS-sql as DataBase

WPF as UI.

In this project I use the “Repository” design pattern with the “Unit of work” design pattern :

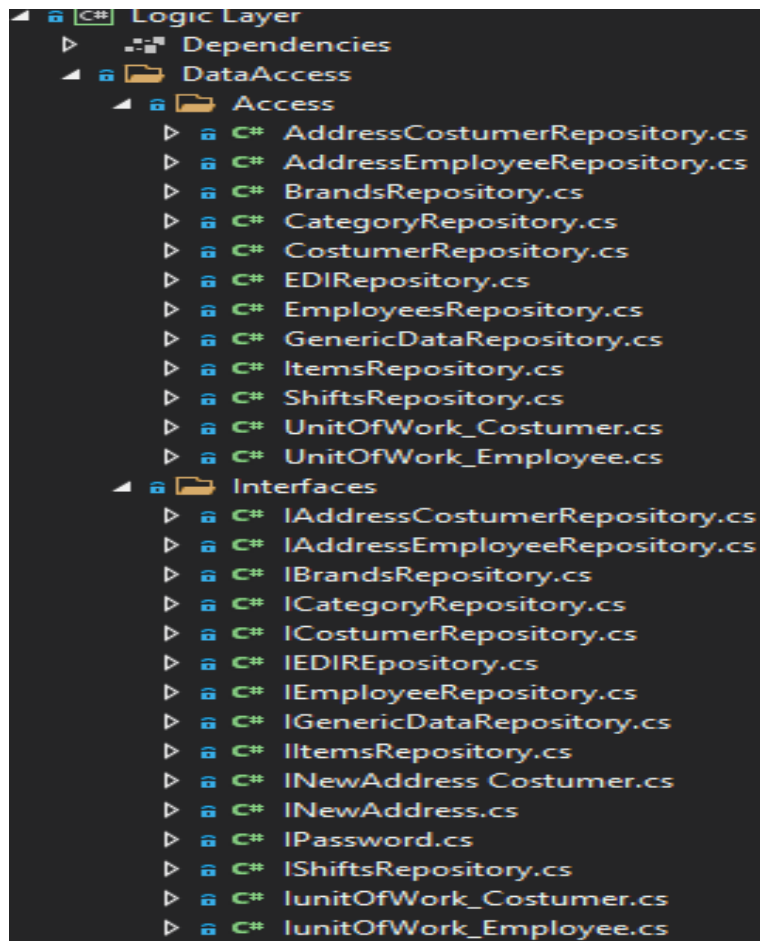
I chose to use this pattern for its versatility and generic capabilities.

In my design for every ability and table the system has, I've created a specific repository that inherits from a generic repository.

Each repository holds methods for accessing the database to decouple the Database access and the rest of the program. The repository is used for methods that are frequently used, such as get all records, add new records or complex querying the DB.

The Unit of work comes to hand as a buffer between each user of the system, holds all the repositories and the “CompleteAsync” “and ”DisposeAsync” methods for saving data in the DB and disposing of all the resources of the context.

I have used two types of units, one for employees and one for customers. (they work the same with the difference that the customer's unit holds less access to the db.



The repository:

```
public class GenericDataRepository<T> : DbContext, IGenericDataRepository<T> where T : class
{
    protected DbContext context;
    internal DbSet<T> dbSet;

    9 references
    public GenericDataRepository(DbContext context)
    {
        this.context = context;
        this.dbSet = context.Set<T>();
    }

    7 references
    public virtual async Task<bool> Add(T entity)
    {
        await dbSet.AddAsync(entity);
        return true;
    }

    3 references
    public virtual async Task Delete(int id)
    {
        dbSet.Remove(await dbSet.FindAsync(id));
    }

    21 references
    public virtual async Task<ICollection<T>> GetAllAsync()
    {
        return await dbSet.ToListAsync();
    }

    10 references
    public virtual async Task<ICollection<T>> GetByCondition(Expression<Func<T, bool>> predicate)
    {
        return await dbSet.Where(predicate).ToListAsync();
    }
}
```

The Unit
of work:

```
public class UnitOfWork_Employee : IUnitOfWork_Employee, IAsyncDisposable
{
    readonly DbContext context;

    2 references
    public UnitOfWork_Employee()
    {
        this.context = new FactoryDbContext();
        this.addressEmployee = new AddressEmployeeRepository(context);
        this.brands = new BrandsRepository(context);
        this.category = new CategoryRepository(context);
        this.employee = new EmployeeRepository(context);
        this.items = new ItemsRepository(context);
        this.shifts = new ShiftsRepository(context);
        this.customers = new customersRepository(context);
        this.addresscustomers = new AddresscustomersRepository(context);
        this.EDI = new EDIRepository(context);
    }

    4 references
    public IAddressEmployeeRepository addressEmployee { get; private set; }
    11 references
    public IBrandsRepository brands { get; private set; }
    11 references
    public ICategoryRepository category { get; private set; }
    10 references
    public IEmployeeRepository employee { get; private set; }
    8 references
    public IItemsRepository items { get; private set; }
    9 references
    public IShiftsRepository shifts { get; private set; }
    3 references
    public IcustomersRepository customers { get; private set; }
    3 references
    public IAddresscustomersRepository addresscustomers { get; private set; }
    9 references
    public IEDIREpository EDI { get; private set; }

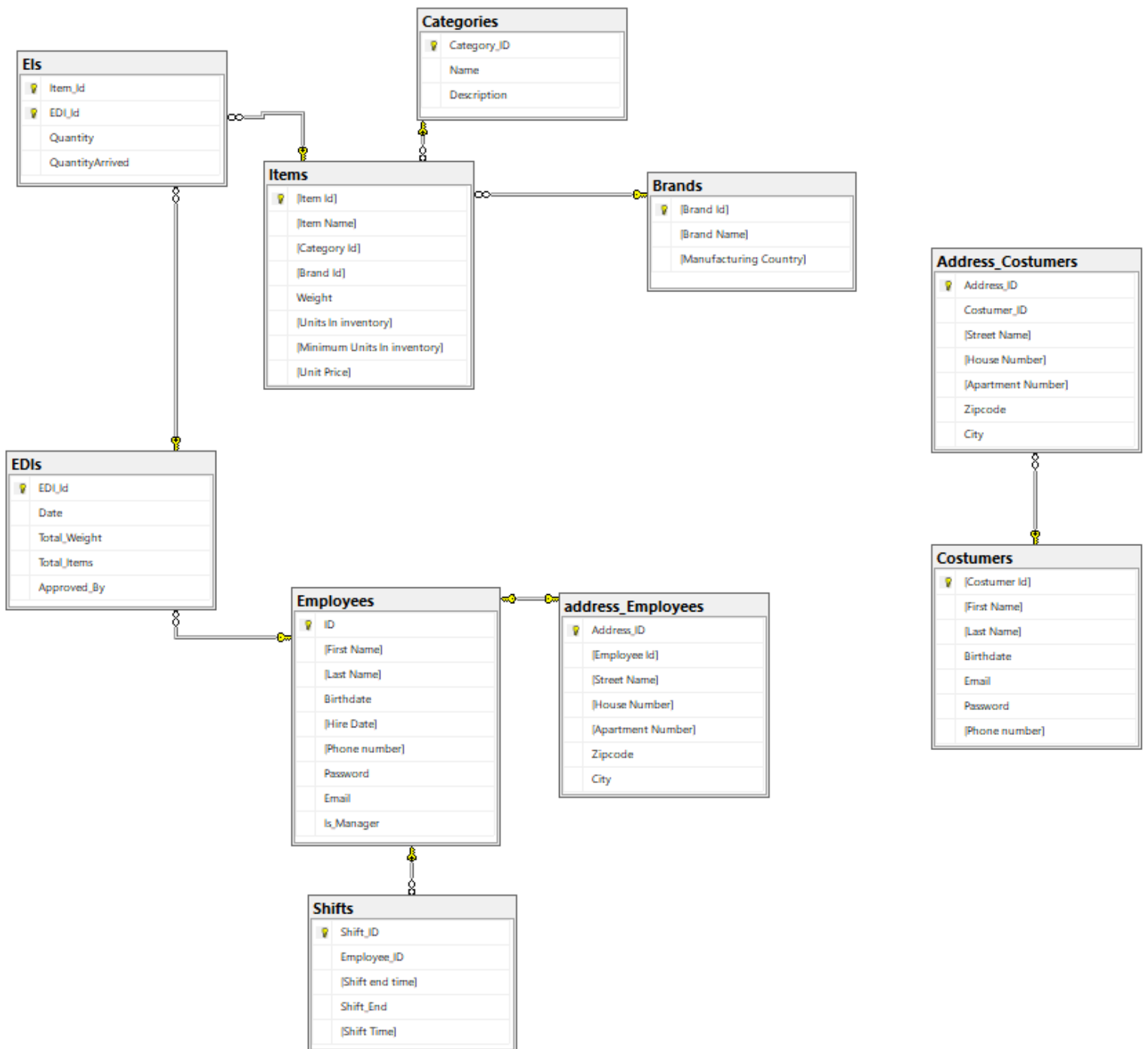
    18 references
    public async Task CompleteAsync()
    {
        await context.SaveChangesAsync();
    }

    1 reference
    public async ValueTask DisposeAsync()
    {

```

Model:

In my model I have 10 tables.



To start the program:

Run the Wearehouse.sql file to populate and create the Database.(no need for any migrations)
In my system there are three types of users- Manager,Employee and customer.

You can choose which type of user to log in with.

Manager :

Email:Toto@AMG.com

Password:123123123

Employee:

Email:TScott@aol.com

Password:123123123

Customer:

Email: Mchen@Yahoo.com

Password: 123123123

The login process checks if there is a user with the email/pass combination.

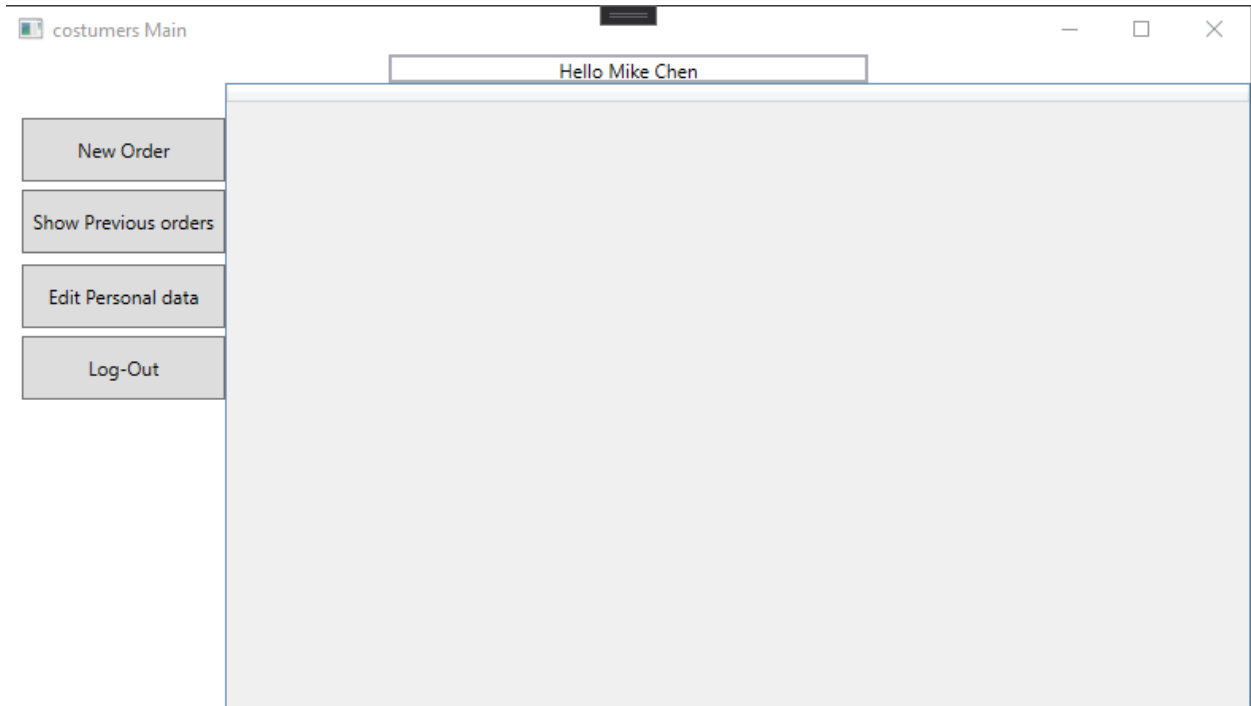
Every check of a password is encrypted.

Passwords must be between 8-12 characters.

The image shows two overlapping web browser windows. The left window, titled 'Log In Page', contains the text 'Are you a Costumer or Employee?' and two buttons labeled 'Employee' and 'Costumer'. The right window, titled 'Employee Login', displays a login form. It has an 'Email:' label with a text input field containing 'Toto@AMG.com', and a 'Password:' label with a password input field showing eight asterisks. Below the password field is a 'Log-In' button. A validation message box is visible, stating 'Password Cannot Must be between 8-12 characters'.

Customer Menu:

In the customer menu the customer has the ability to create orders from the warehouse(not implemented),see past orders(not implemented) and edit personal information.



Employee Menu:

In the employee menu, the employee can see their own shifts, edit personal data and change their password, and the main ability of the employee is accepting EDI to the warehouse.

Employee Main

Hello Tom

Close and end Shift

Fulfill next EDI

Edit personal info

See all my shifts

Shift start	ShiftEnd	TotalTime
12/12/2021 10:33:07 AM		0

Personal Info

First Name:

Tom

Last Name:

Scott

Email:

TScott@aol.com

Phone number:

4498752524

Address:

Street: TScott@aol.com

House number:

32

Apartment number:

9

Zipcode:

65879541

City:

Ramat Hasharon

Enter Password

Re-Enter Password

Update password

Update

When the employee presses the “fulfill next EDI” , the employee will get the next unfulfilled in-EDI.

The employee needs to “ enter the number of items in the “real” shipment and accept the number.

EDI Accept

Employee Name:

EDU ID:

Time Placed:

Tom Scott

1000

12/12/2021 10:04:49

Submit Done

Accept Arrival	Item Id	Name	Quantity in EDI	Quantity Arrived	
Confirm Arrived	3	Soft Compound	3		
Confirm Arrived	4	Wet Compound	2		

Accept Arrival	Item Id	Name	Quantity in EDI	Quantity Arrived
Confirm Arrived	3	Soft Compound	3	3
Confirm Arrived	4	Wet Compound	2	

If the quantity arrived is not the same as quantity in EDI a message Will pop and ask from the employee to accept its not an error

Confirm Arrived	4	Wet Compound	2	4
-----------------	---	--------------	---	---

Quantities not matched

Arrived quantity different from quantity on EDI is that correct?

Yes

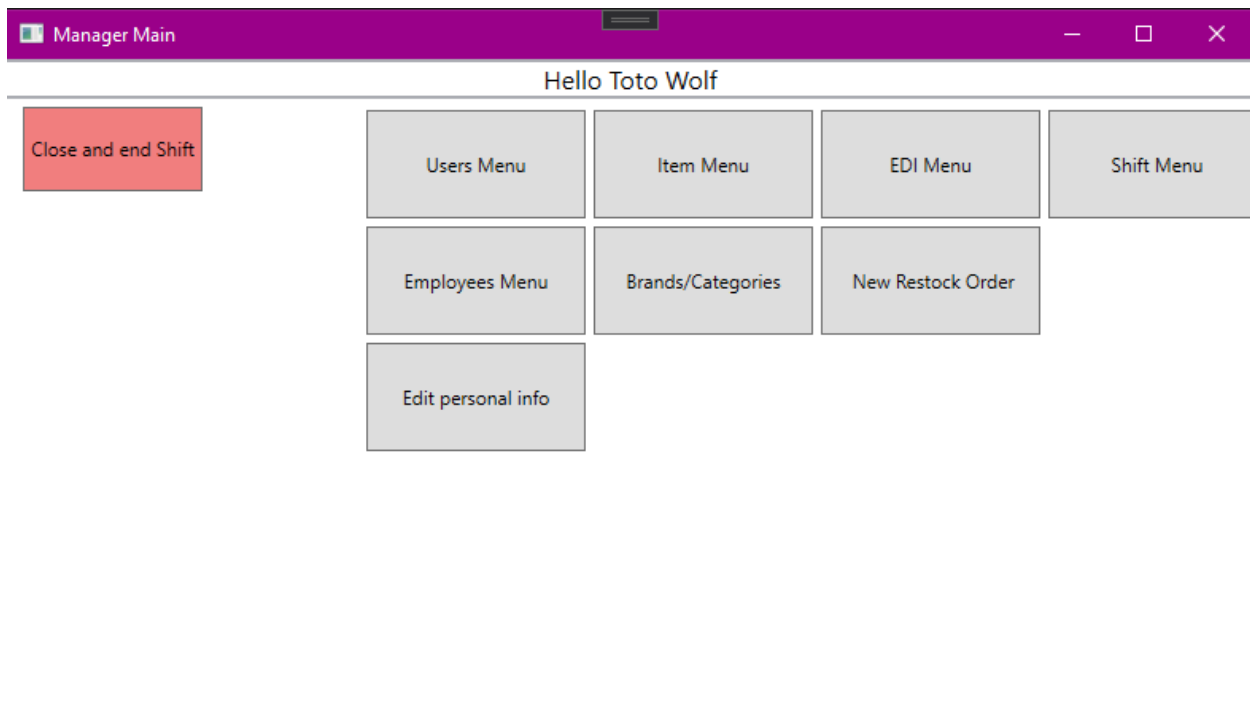
No

When the employee finishes counting all items, the items will update in inventory.

Manager Menu:

The manager can:

- 1.Add/update/delete Items in warehouse
- 2.Add/update/delete Brands and Categories
- 3.Add/edit employees and Customer
- 4.See shifts for all employees.
- 5.Create a new Restock order for an item.
6. See all EDI's that have not been fulfilled and "closed ones" and who fulfilled them.



Item Menu

Category

Brand

Item Name

Add items

search

Item ID	item name	Category name	Brand	Price	Units in Inventory	Minimum in Inventory	Action	
1	Hard compound	Race Tyres	Beta-Juliet	1500	4	2	UpdateDelete	
2	Mideum Compound	Race Tyres	Beta-Juliet	1700	6	6	UpdateDelete	
3	Soft Compound	Race Tyres	Fair-rari	1770	8	8	UpdateDelete	
4	Wet Compound	Race Tyres	Nonda	1250	8	10	UpdateDelete	
5	Plastic Dice set	Play_Dice	Fair-rari	15	45	100	UpdateDelete	
6	Metal Dice set	Play_Dice	Beta-Juliet	25	10	15	UpdateDelete	
7	Special D20 Die	Play_Dice	Beta-Juliet	15	5	3	UpdateDelete	
8	Blue Bycycle deck	Playing-Cards	Nonda	10	35	25	UpdateDelete	
9	Red Bycycle deck	Playing-Cards	Fair-rari	12	10	25	UpdateDelete	
10	Crazy Bycycle deck	Playing-Cards	Nonda	105	15	25	UpdateDelete	

Category

Brand

Playing-Cards

search

Nonda

Fair-rari

Beta-Juliet

Update Item

Item Name

Price

Hard compound

1500

Items in inventory

Minimum items in inventory

4

2

Weight

2

Category

Brand

Race Tyres

New

Beta-Juliet

New

Update Item

Add Item

Item Name

Price

Items in inventory

Minimum items in inventory

Weight

Category

Brand

On the item menu you can search by Brand or category and item name.

Through the new item screen you can choose an already existing brand and category or create a new one.

You can't finish creating without choosing brand and category or empty fields

Category or Brand Must BE Chosen

Cannot leave empty fields!

Brand/Category Menu									
See Brands		Add new Brand			See categories		Add new Category		
ID	Name	Manufacturing Country	Action		ID	Name	Description	Action	
1	Nonda	China	Update		1	Playing-Cards	Regular Playing card game	Update	
2	Fair-rari	Spain	Update		2	Play_Dice	Normal play Dice set.	Update	
3	Beta-Juliet	USA	Update		3	Race Tyres	F1 Race Tyres	Update	

On the Brands/Category you can update,search and add new brands and categories.

Shifts Menu

From Date:

To Date:

Employee ID:

Select a date 15

Select a date 15

Search

Show on now

ID	First Name	Last Name	Shift Start	Shift End	Action
1	Toto	Wolf	12/12/2021 11:09:22 AM		Show Employee info

On the shift menu you can see who is on now.And search by employee id or date of shifts

UserMenu

By Id:

By Name:

By Phone number:

By Email:

Show Users

ID	First Name	Last Name	Date of Birth	Email	Phone Number	Addresses	Action
1	Mike	Chen	6/17/1992 12:00:00 AM	Mchen@Yahoo.com	8457099476	42-09 47th ave 50 3, 444021 Sunnyside	View User orders

On the user menu you can see all users or search by parameters.

Employees Menu

By Id:

By Name:

By Phone number:

By Email:

New Users

Show Users

ID	First Name	Last Name	Date of Birth	Hire Date	Email	Phone Number	Address	Action
1	Toto	Wolf	3/14/1990 12:00:00 AM	12/11/2021 3:51:00 PM	Toto@AMG.com	05455679823	Dizingof 224 3, 123456 Tel Aviv	View Employee EDI's
2	Tom	Scott	3/30/1988 12:00:00 AM	12/12/2021 10:27:00 AM	TScott@aol.com	4498752524	Sokolov 32 9, 65879541 Ramat Hasharon	View Employee EDI's

On the employee page you can see all employees and create new users.(same as register of customers)

Registrar Employee

First Name

Last Name


Email

☐ Manager?

Phone Number

Date of Birth

Password

Select a date 

Street Name

House number

Apartment

Zipcode

City

Done

Restock Menu

Item Id	Name	Quantity in Inventory	Minimum Quantity in Inventory	Quantity to order	Action
4	Wet Compound	8	10	2	<div>Insert</div>
5	Plastic Dice set	45	100	55	<div>Insert</div>
6	Metal Dice set	10	15	5	<div>Insert</div>
9	Red Bicycle deck	10	25	15	<div>Insert</div>
10	Crazy Bicycle deck	15	25	10	<div>Insert</div>

Finish order

Accept all to order

When entering the Restock menu you can see all the items in the warehouse with quantity in inventory lower than minimum quantity.

You can't see items that need to be ordered that have already been ordered but EDI has not fulfilled yet.

You can choose the amount you want to order on "Quantity to order" with the default on the difference between the amount in inventory and minimum amount.

You can accept all items to order like that or order individual items.

EDI Menu

From Date:

To Date:

EDI Id:

Select a date 15

Select a date 15

Search

Show not Approved

EDI ID	EDI Date	Handling Employee	Total Items	Total Wight	Action
1001	12/12/2021 11:42:12 AM		2	4.4	Show Items On EDI

On the EDI menu you can see all in-EDI by date and by id.
 You can show only EDI's that have not been accepted by the employee yet.
 On arrived edi you can see the Quantity arrived .

Item Id	Name	Quantity in EDI	Quantity Arrived
3	Soft Compound	3	3
4	Wet Compound	2	0

Item Id	Name	Quantity in EDI	Quantity Arrived
4	Wet Compound	2	