**Stock Management App**

There are 5 basic concepts used in this application. These are kept as tables in the database. These concepts;

- Warehouse

- Company

- Product

- Stock

- StockTransfer (Stock Transfer)

**Purpose of application**

In this application, information on the stocks of the products in different warehouses is kept.

Stock transfer can be made between warehouses.

Companies are defined in the system. After this identification, the Product can be registered depending on this company.

For example: To keep an Apple branded phone in the system, the Apple company must first be registered in the system. Subsequently, the product can be created by selecting the Apple company from the Product add page.

As an example, the IPhone product will have different stock information in different warehouses.

If warehouses such as A, B, C, D are defined in the system, separate stock information of the IPhone product can be found in each warehouse.

A product in warehouse A can be transferred to warehouse B. In this case, the stock in warehouse A will be dropped and added to warehouse B.

**Sample Scenario:**

1 - The user comes to the company creation screen.

2 - The user enters the required information and clicks the company creation button. This action will trigger a Create () method in the application.

3 - This action by the user is transmitted to the controller.

4 - The controller looks at the user's request and realizes that it must perform the Create operation.

5 - Subsequently, creating a new company in the database triggers (Controller / Model relationship).

6 - After the company is created, the user will be returned to the companies screen.

7 - The information of existing companies is taken from the database.

8 - Embeds in the .cshtml code to represent the screen

9 - View (View) is transmitted to the user.

ORM (Object Relational Mapper)

The ORM concept is used to easily create the application side Objects in the database. For example:

Staff

{

public int StaffID {...};

public string Name {...};

}

When ORM is run for creating on this object, a row with the ID and Name values ​​we have given to the object is created under the Personnel table in the database.

ORM allows us to easily switch between the objects in the application and the database. It acts as a bridge. This makes it very easy to perform CRUD operations on the database.

Business rules

1- Not being able to exceed warehouse's total capacity when adding stock

2- Control, inability to go below 0 when falling off stock,

3- When adding a new product, it does not list the companies whose isActive field is False.

4- Failure to perform transactions in storages with False availability while performing stock transactions.

5- Compulsory company name is written, the same product cannot be registered twice in the same warehouse (But the same product can be registered in different warehouses.)

**Models**

\* **Warehouse**

- Warehouse ID

- Name

- AvailableForTransfers

- StorageSpace

\* **Company**

- CompanyID

- Name

- IsActiveProvider

\* **Product**

- ProductID

- CompanyID

- Name

- SpaceOccupied

\* **Stock**

- StockID

- WarehouseID

- ProductID

\* **StockTransfer**

- StockTransferID

- ProductID

- FromStockID

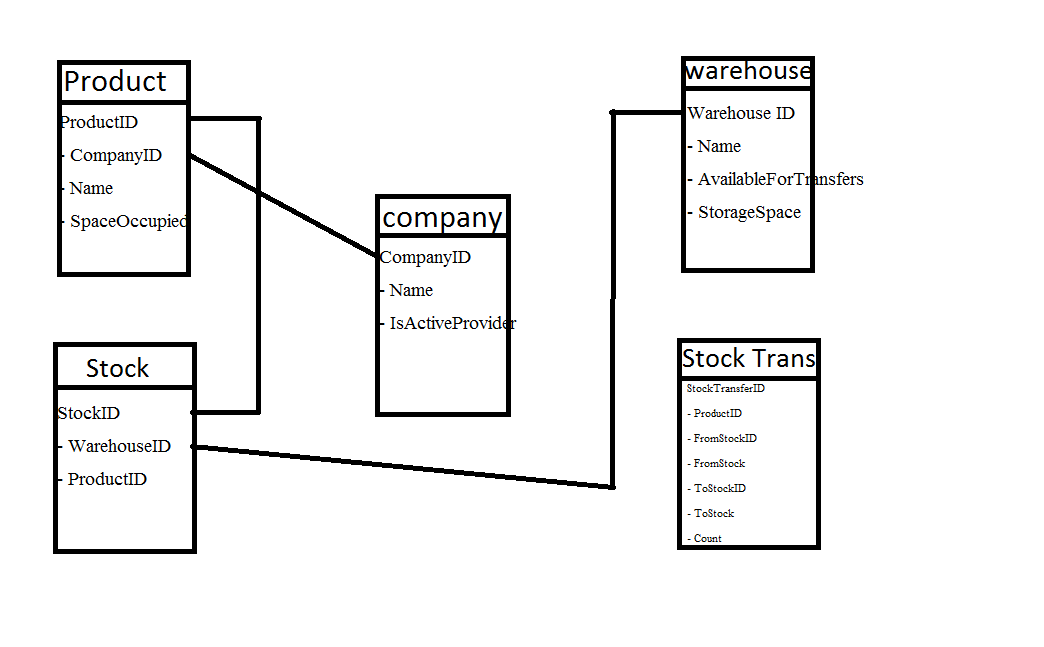
- FromStock

- ToStockID

- ToStock

- Count

**Data Model**

****

Oğuz Kaan Satan

1621221026