Phase 2 Project Design Document

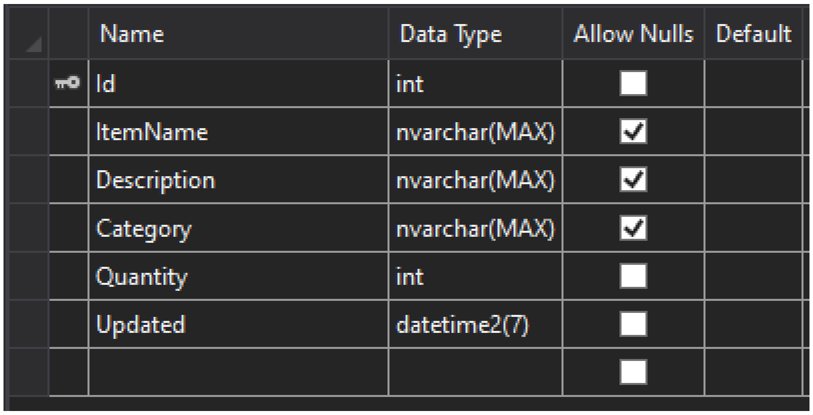
1) General Description of the Application

The web application will be a personal inventory system. It will allow the user to manage items in their inventory. The web application will have at least five web pages. Four of these web pages will be for CRUD functionality that will allow for the items in the inventory to be created, read, updated, and deleted from a database. The fifth web page will be an about page, which will describe how to use the web application. An additional feature that might be included is allowing users to sign in and manage their individual inventories.

2) Technology Overview

The web application will be a ASP.NET Core MVC web application. The CRUD functionality of the items and the database initialization will be handled by Microsoft Entity Framework Core. The database will be a LocalDB database. The web application will also utilize HTML Helper methods and enable API calls to view the data in JSON form regarding the inventory items.

3) Database Schemas



	Name	Data Type	Allow Nulls	Default
🔑	Id	int	□	
	ItemName	nvarchar(MAX)	☑	
	Description	nvarchar(MAX)	☑	
	Category	nvarchar(MAX)	☑	
	Quantity	int	□	
	Updated	datetime2(7)	□	
			□	

4) Authentication & Authorization

An additional feature that might be included in the web application is managing multiple user accounts each with their own inventory record. Authentication and authorization will be used here.

5) Controllers

The web application will have a home controller, an items controller, and an API controller. The home controller will handle the welcome and about action methods which will return the welcome and about views. The items controller will handle the index, details, create, edit, delete, and delete confirm action methods which will return their respective views. The API controller will handle API calls regarding the inventory's items.

6) Unit Testing

The unit testing will be handled through a xUnit Test Project (.NET Core). The unit tests will be used for some of the action methods within the items controller class and will be organized by arrange, act, and assert pattern. They will check that these methods return a view and the correct data to the view.

7) Logging and Exception Handling

Exception handling will occur in the Edit action method of the items controller class. It will occur here to catch any errors that might happen when saving the changes to the database in particular the DbUpdateConcurrencyException. Exception handling will also occur in the main method of the program class. It will occur here to catch any errors that happen when seeding the database. Any errors that occur in the main method when seeding the database will be logged by getting an ILogger instance from dependency injection after building the host.

8) Configurable Parameters

The web application will have a configurable parameter in the appsettings.json file. That parameter will be a connection string stored as “InventoryManagerContext.”