System Requirements Specification Index

For

Groceries Management Application (Console)

Version 4.0

IIHT Pvt. Ltd.
IIHT Ltd, No: 15, 2nd Floor, Sri Lakshmi Complex, Off MG Road, Near SBI LHO,
Bangalore, Karnataka – 560001, India
fullstack@iiht.com

GROCERIES MANAGEMENT APPLICATION SYSTEM

System Requirements Specification

1.Business-Requirement:

1.1 Problem Statement:

Groceries Management System: Develop Groceries Management System console application using C#, .NET Core 3.1 . Implements all the checks so that there are no errors when Groceries is added, sort from collections and use File/IO. Use proper coding standards with proper class name and method names as per given functionalities.

1.2 Following is the requirement specification:

	Groceries Management Application
	Implement the following Functionalities using a menu driven program.
1	Add at least 5 Groceries details using List generic collection. <user input>
2	Save all Groceries details in excel file.
3	Serialize Groceries List object in Binary format and save it in text file.
4	Fetch all Groceries details from the saved text file and deserialize it.
5	Show details of Groceries in descending order of name.

2.Resources available:

2.1 Package: GroceriesManagement

Names	Resource	Remarks	Status
Package Structure/Project			
Assets	Test.txt ResultSheet.xlsx	Test file contains serialized object in JSON format, ResultSheet file will be created automatically after running the application which contains groceries details.	Already Implemented
	Program.cs	Program.cs is an important class that is the entry point of application.	Partially Implemented
	Grocery.cs	Grocery class is an entity class that consists of field properties of groceries.	Already Implemented

2.2 Package: GroceriesManagement.Tests

Resources

Note: - Under the GroceriesManagement.Tests contain all test cases for code evaluation, please don't try to alter and edit it.

3.Assumptions, Dependencies, Risks / Constraints

3.1 Common Constraints:

- Develop application Groceries Management System using Collections, Classes, Exception handling, C# new features File I/O in C#, .NET.
- Implement all Functionalities using a menu driven program.
- Include C# New Features wherever required.

3.2 Exception Handling Specifications:

- Handle Exceptions in the main function by adding try/catch blocks.
- Implement FileNotFoundException, NullReferenceException built-in exception class for all scenarios with separate catch blocks.
- Implement custom exception DuplicateIdException to handle duplicate Grocery id.

4. Business Validations

4.1 Entity Class Specifications:

Create a class Grocery that consists of field properties given below. Implement IComparable<> generic interface to sort the Grocery details by name.

Class name: Grocery

Data Member (Properties):

- int GroceryId
- String GroceryName
- String Description
- int price
- Date ExpiryDate (can be null for non-packaged item)

5. Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**

2. **To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) Terminal → New Terminal.**

3. **Install the .NET SDK 6.0 by running:**
   sudo apt install dotnet-sdk-6.0

   ***If it asks for the password, provide password : pass@word1**

   If it asks: Do you want to continue? [Y/n]

   Type Y and press Enter.

4. **On command prompt, cd into your project folder (cd <Your-Project-folder>).**

5. **To build your project use command:**
   (GroceriesManagement /dotnet build)

6. **To launch your application, Run the following command to run the application:**
   (GroceriesManagement / dotnet run)

7. **This editor Auto Saves the code.**

8. **To run the test cases in CMD, Run the following command to test the application:**
   (GroceriesManagement /dotnet test --logger "console;verbosity=detailed")
   (You can run this command multiple times to identify the test case status,
   and refactor code to make maximum test cases passed before final submission)

9. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
