# Inventory Management System in C# – Project Report

## 1. Project Purpose

This console-based C# application helps users manage product inventories efficiently. It allows users to add, update, list, and remove products using a simple command-line interface. The project demonstrates fundamental programming concepts and back-end logic as part of the Microsoft Back-End Developer curriculum.

## 2. Project Goals

* - Develop a simple, functional, and user-friendly inventory management application.
* - Apply CRUD operations using control structures, loops, and methods in C#.
* - Demonstrate understanding of back-end concepts covered in the course.
* - Deliver a professional-quality project suitable for a portfolio or GitHub.

## 3. Requirements

### 3.1 Functional Requirements

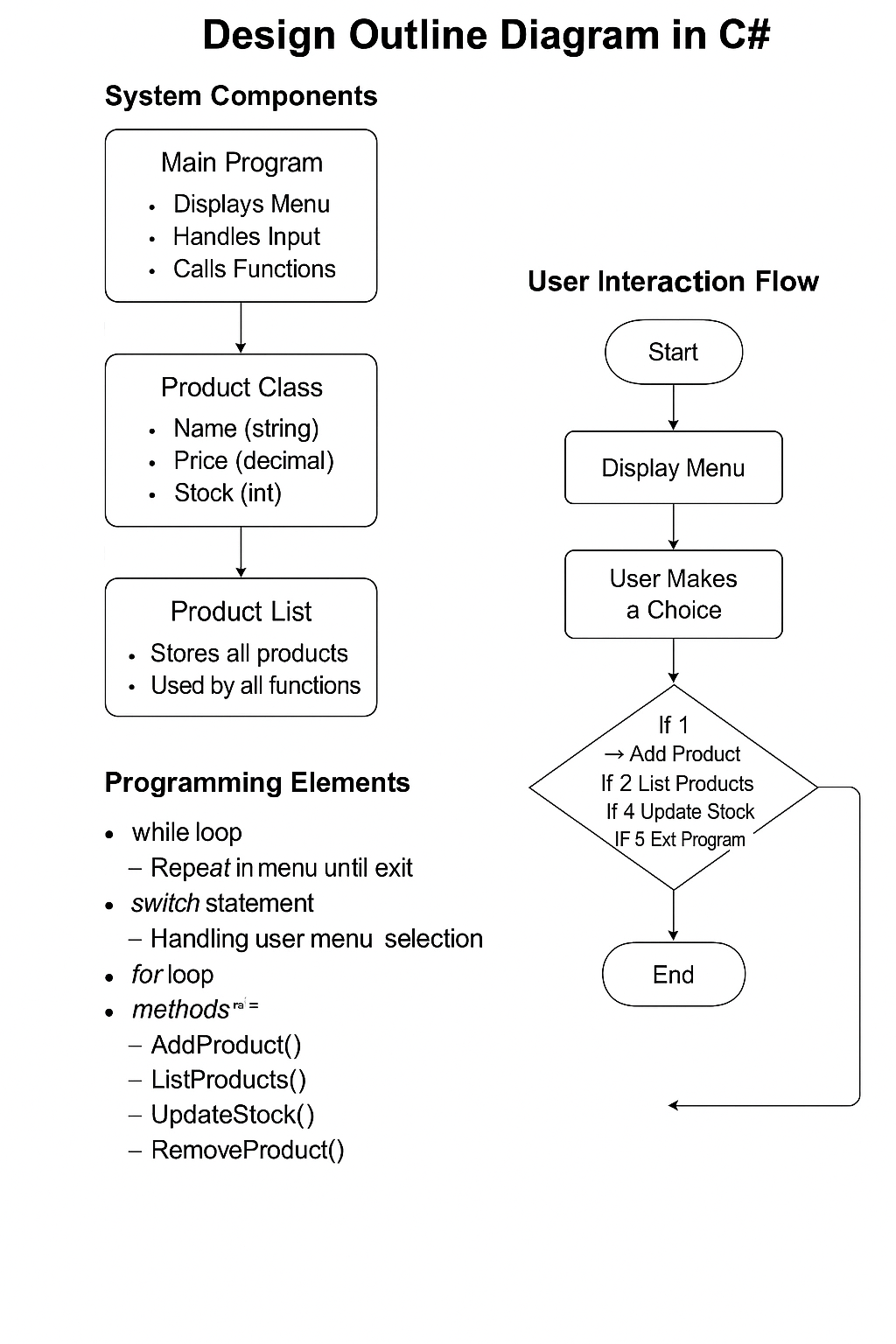
* - Users must be able to add products with name, price, and stock quantity.
* - Users must be able to update existing product stock.
* - Users must be able to view all products.
* - Users must be able to remove a product from the system.
* - The program must react accordingly to the user's menu selection.

### 3.2 Non-Functional Requirements

* - Written in C# using Visual Studio Code.
* - Code should be clean, well-commented, and readable.
* - Feedback messages must be displayed for all actions.
* - Must include at least one control structure, one loop, and one method.
* - Program must handle incorrect input gracefully.

## 4. System Flow Diagram

The system architecture and user interaction flow are illustrated below:



## 5. Full Source Code

using System;  
using System.Collections.Generic;  
  
namespace StockManagementSystem  
{  
 class Product  
 {  
 public string Name { get; set; }  
 public decimal Price { get; set; }  
 public int Stock { get; set; }  
  
 public Product(string name, decimal price, int stock)  
 {  
 Name = name;  
 Price = price;  
 Stock = stock;  
 }  
  
 public void DisplayInfo()  
 {  
 Console.WriteLine($"📦 Product: {Name}, 💸 Price: {Price}₺, 📦 Stock: {Stock}");  
 }  
 }  
  
 class Program  
 {  
 static List<Product> productList = new List<Product>();  
  
 static void Main(string[] args)  
 {  
 bool running = true;  
  
 while (running)  
 {  
 ShowMenu();  
  
 Console.Write("👉 Your choice: ");  
 string input = Console.ReadLine();  
  
 switch (input)  
 {  
 case "1": AddProduct(); break;  
 case "2": ListProducts(); break;  
 case "3": UpdateStock(); break;  
 case "4": RemoveProduct(); break;  
 case "5":  
 Console.WriteLine("👋 Exiting program...");  
 running = false;  
 break;  
 default:  
 Console.WriteLine("⚠️ Invalid choice. Please try again.");  
 break;  
 }  
  
 Console.WriteLine();  
 }  
 }  
  
 static void ShowMenu()  
 {  
 Console.WriteLine("=== 📦 INVENTORY MANAGEMENT SYSTEM ===");  
 Console.WriteLine("[1] Add Product");  
 Console.WriteLine("[2] List Products");  
 Console.WriteLine("[3] Update Stock");  
 Console.WriteLine("[4] Remove Product");  
 Console.WriteLine("[5] Exit");  
 }  
  
 static void AddProduct()  
 {  
 Console.Write("📝 Product name: ");  
 string name = Console.ReadLine();  
  
 Console.Write("💸 Price: ");  
 decimal price = Convert.ToDecimal(Console.ReadLine());  
  
 Console.Write("📦 Stock quantity: ");  
 int stock = Convert.ToInt32(Console.ReadLine());  
  
 Product newProduct = new Product(name, price, stock);  
 productList.Add(newProduct);  
  
 Console.WriteLine("✅ Product added successfully!");  
 }  
  
 static void ListProducts()  
 {  
 Console.WriteLine("=== 📋 Product List ===");  
  
 if (productList.Count == 0)  
 {  
 Console.WriteLine("❌ No products available.");  
 return;  
 }  
  
 for (int i = 0; i < productList.Count; i++)  
 {  
 Console.Write($"[{i + 1}] ");  
 productList[i].DisplayInfo();  
 }  
 }  
  
 static void UpdateStock()  
 {  
 ListProducts();  
  
 if (productList...  
  
(Full code available in inventory\_code.cs)