

# Milestone 5 – Inventory Manager Application

Kelly Lamb | Professor Aiman Darwiche Ph.D. | Grand Canyon University: CST-117 Enterprise Application Programming I

## Problem / Question

Create a visually appealing, functionally superior inventory management application for any chosen business using C#.Net and Visual Studio

## Hypothesis

- Create C# classes and forms to handle data storage, model, logic, and visual representations for inventory. Utilize advanced techniques in C# to store/retrieve persistent data; Utilize libraries to create spreadsheet reports.
- Design processes incrementally – confirming results in stages

## Project Overview

1. Develop application logic and flesh it out in a wireframe
2. Design Classes to model data and logic
3. Create UML documents and define dependencies / relationships
4. Create Flowcharts, initial logic; updated version with classes
5. Write C# code – confirm that it follows flowchart logic
6. Create videos to demonstrate effective progress in each step
7. Modify code to final stage to produce visually appealing product
8. Modify UML diagram to more accurately represent final product
9. Produce a poster that highlights the process

## Classes / Research

### Model Classes

InventoryItem  
InventoryManager

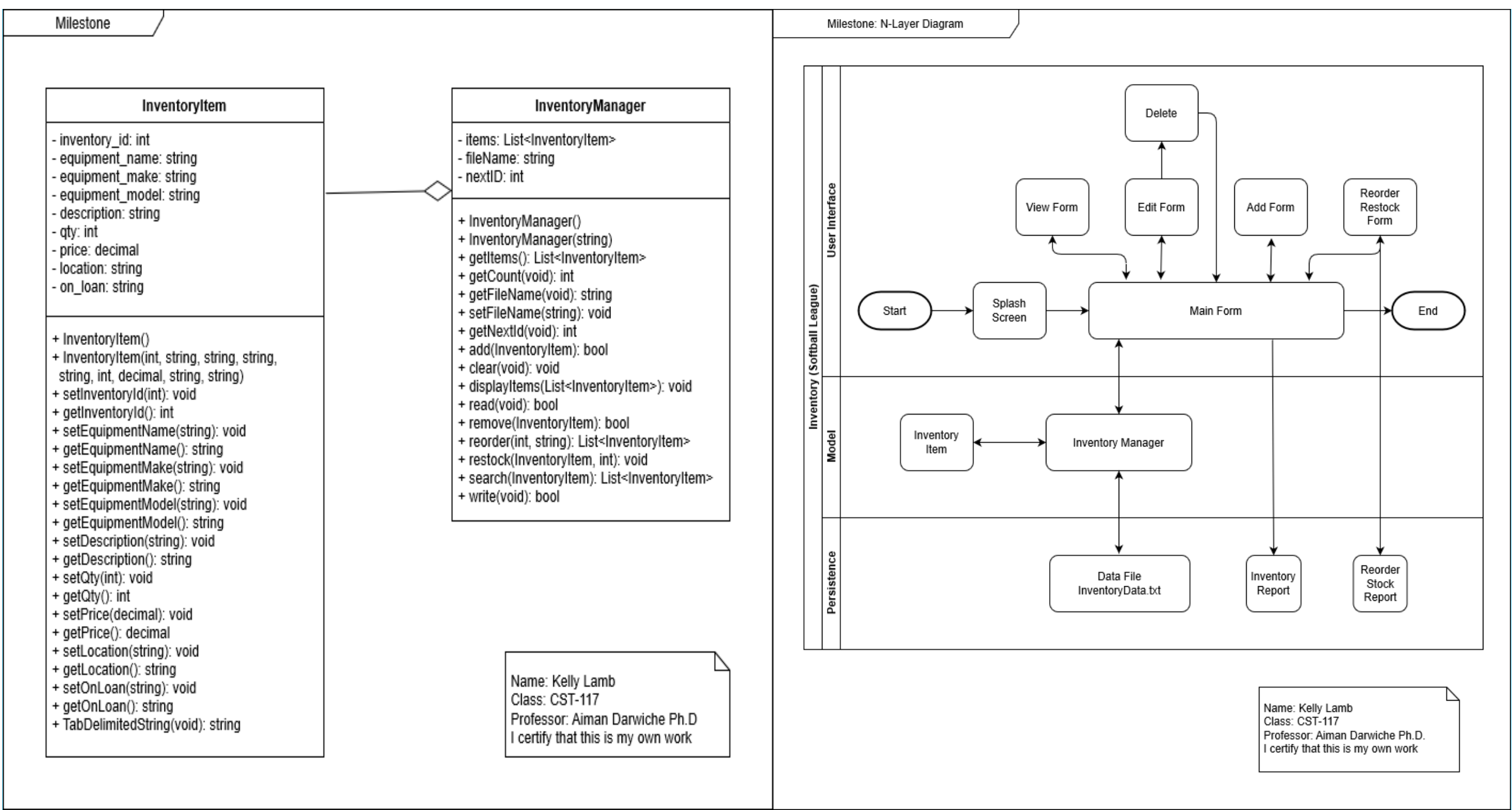
### Data Access Classes

InventoryManager  
InventoryData  
EPPlus.dll

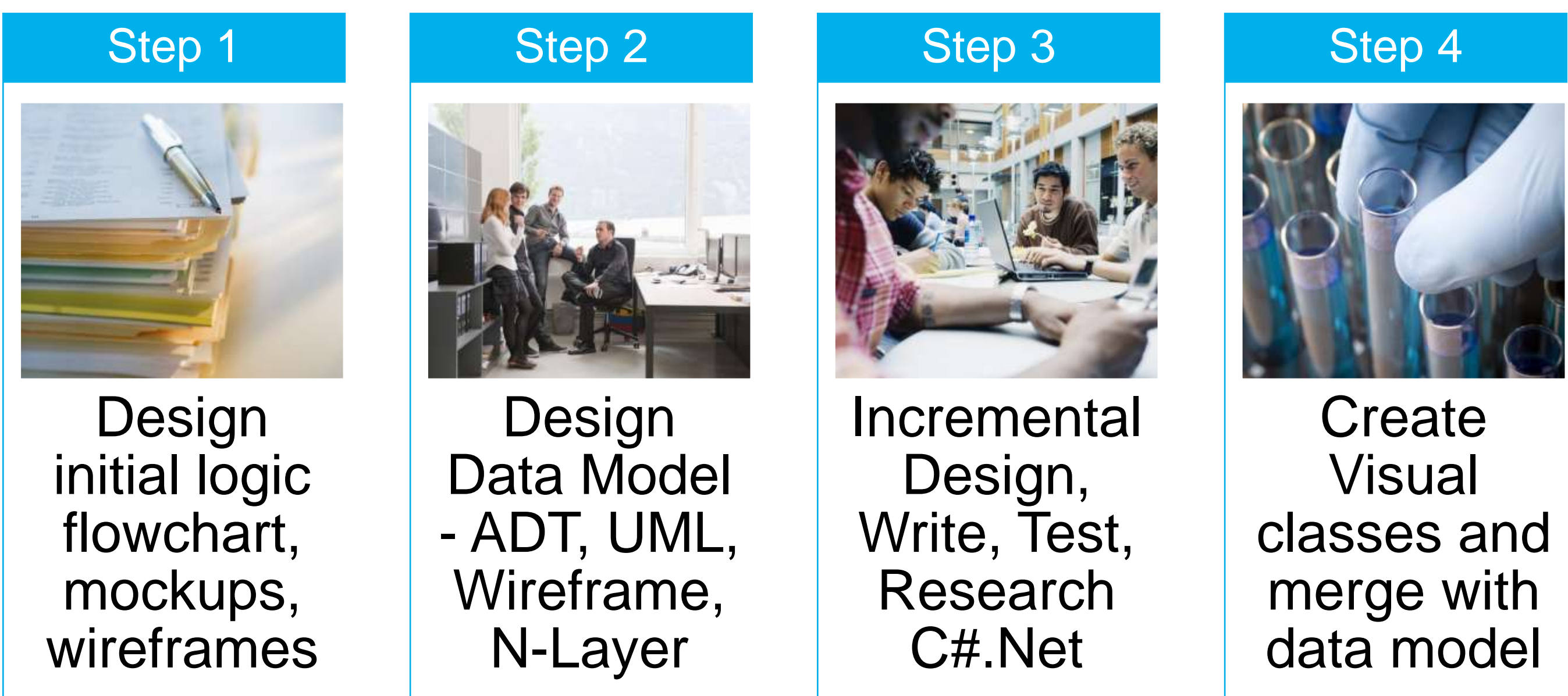
### Visual Classes

AddForm  
EditForm  
MainForm  
Program  
ReorderForm  
SplashScreen  
ViewForm

## Flow Chart / UML



## Procedure - Milestones



## Base Code Snippets

```
// InventoryManager.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using EPPlus;
using Microsoft.Office.Interop.Excel;

namespace InventoryManager
{
    public class InventoryManager
    {
        private List<InventoryItem> items;
        private string itemName;
        private int itemID;

        public InventoryManager()
        {
            items = new List<InventoryItem>();
        }

        public void AddItem(InventoryItem item)
        {
            items.Add(item);
        }

        public void EditItem(int itemID, string newName, decimal newPrice, string newLocation, string newOnLoan)
        {
            InventoryItem item = items.FirstOrDefault(i => i.Inventory_id == itemID);
            if (item != null)
            {
                item.Equipment_name = newName;
                item.Price = newPrice;
                item.Location = newLocation;
                item.On_loan = newOnLoan;
            }
        }

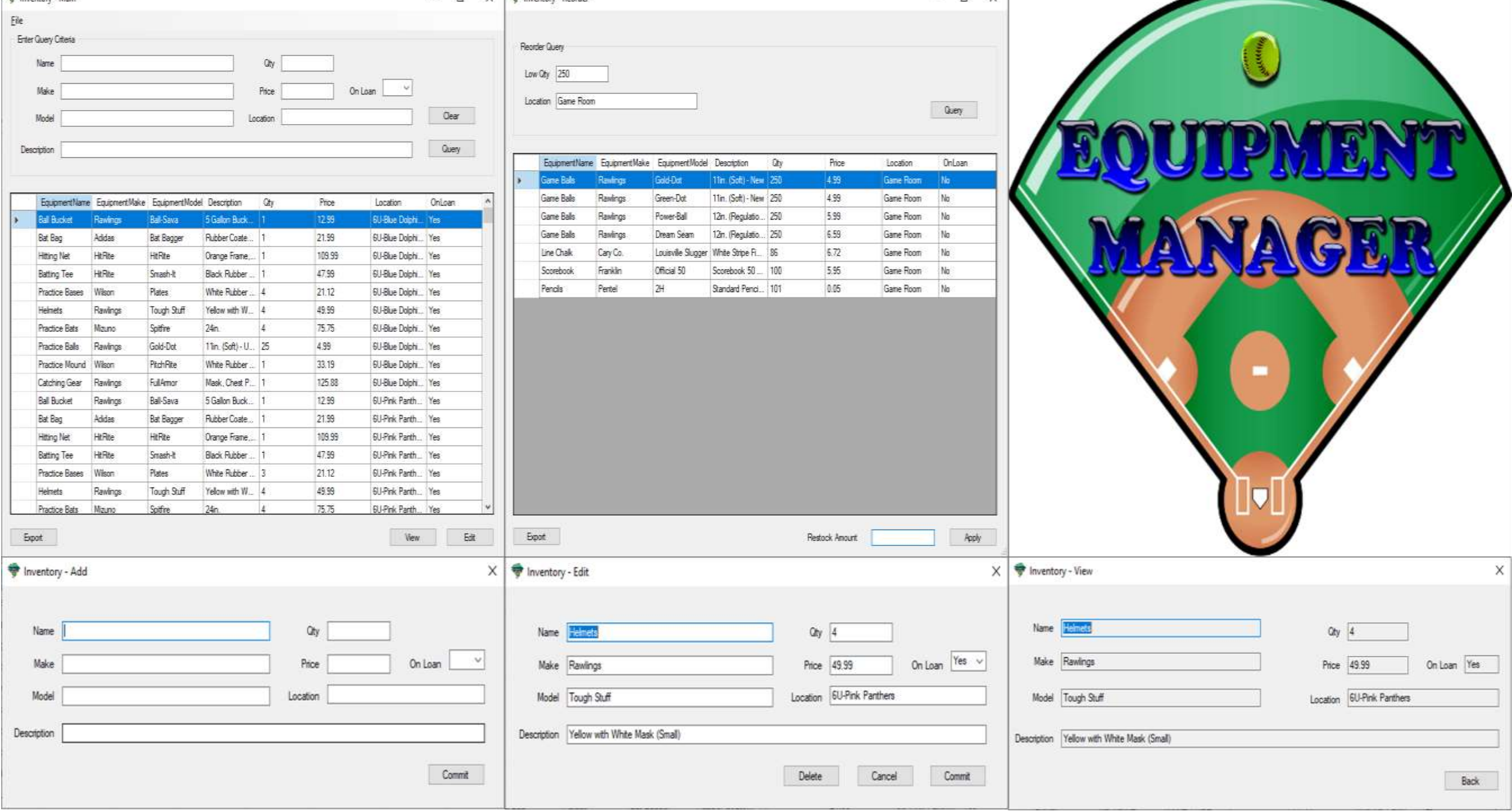
        public void DeleteItem(int itemID)
        {
            InventoryItem item = items.FirstOrDefault(i => i.Inventory_id == itemID);
            if (item != null)
            {
                items.Remove(item);
            }
        }

        public void ViewItems()
        {
            foreach (InventoryItem item in items)
            {
                Console.WriteLine($"ID: {item.Inventory_id}, Name: {item.Equipment_name}, Price: {item.Price}, Location: {item.Location}, On Loan: {item.On_loan}");
            }
        }

        public void ReorderStock(int itemID, int qty)
        {
            InventoryItem item = items.FirstOrDefault(i => i.Inventory_id == itemID);
            if (item != null)
            {
                item.Qty = qty;
            }
        }

        public void GenerateReport()
        {
            // Generate report logic
        }
    }
}
```

## Results - Screenshots



- Application initializes, loads data, handles all process requirements
- Screenshots: Represents all implemented functionality
- Logo, Inventory Listing, Add, Edit, View, Reorder Stock
- Spreadsheet Reporting – Excel Inventory / Reorder (not shown)
- Number of classes: 7 (excluding form designer)
- Number of form layouts: 5
- Number of lines: C# code: 2,838

## Conclusion

- Softball Equipment Manager – Surprisingly intuitively easy to code but provided strong logic requirements, decisions, and learning C# and Visual Studio.
- Flowchart, UML, Wireframe, and N-Layer Diagrams are great tools to help in the design process
- Visual Studio offers a robust graphics programming platform but required significant research and understanding – many tutorials online.
- Very satisfying outcome watching everything work together.

## Works Cited

- Starting Out With Visual C# 4e T. Gaddis
- docs.microsoft.com
- docs.microsoft.com/en-us/dotnet/csharp
- www.stackoverflow.com
- csharp.net-tutorials.com
- www.tutorialspoint.com
- www.c-sharpcorner.com
- www.youtube.com
- www.bitbucket.com