ASSIGNMENT 2

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 1: Programming		
Assignment title	Application development with IDE		
Academic Year	2022		
Unit Tutor			
Issue Date		**Submission date**	23/12/2022
Student Name	Tran Cong Hoang	**Student ID**	BH00317
Class	IT0601	**Assessor name**	

Submission Format:

Format: The submission is in the form of an individual written report. This should be written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs, and subsections as appropriate, and all work must be supported with research and referenced using the Harvard referencing system. Please also provide a bibliography using the Harvard referencing system.

Submission Students are compulsory to submit the assignment on the due date and in a way requested by the Tutors. The form of submission will be a soft copy in PDF posted on the corresponding course of http://cms.greenwich.edu.vn/

Note: The Assignment *must* be your work, and not copied by or from another student or from Books, etc. If you use ideas, quotes, or data (such as diagrams) from books, journals, or other sources, you must reference your sources, using the Harvard style. Make sure that you know how to reference properly and understand the guidelines on plagiarism. *If you do not, you get to fail*

Assignment Brief and Guidance:

Scenario: You have applied for a post as a trainee with a software development company and have been invited for an interview. You have passed the presentation to demonstrate your problem-solving and basic programming skills. Now you are given a more challenging task to create a fully working, secure application that has been developed using an IDE and adheres to coding standards for a detailed business problem.

Tasks

You will discuss a suitable problem with your mentor and list the user requirements before designing, implementing (coding), and testing a solution. You will create a report that should include:

- Introduction to your program (list of requirements and screenshots of the program)
- Explain some of the programming paradigms. Evaluate why and how your program uses these (or some of) paradigms.
- Explain the common features an IDE should have and evidence of how the IDE was used to manage the development of your code.
- An evaluation of developing applications using an IDE versus developing an application without using an IDE.
- An explanation and evaluation of the debugging process in the IDE used and how it helped with development.

An explanation and evaluation of coding standards used in your program and the benefits to organizations of using them.

The working application produced must also be demonstrated together with the presentation.

Learning Outcomes and Assessment Criteria		
Pass	**Merit**	**Distinction**
LO2 Explain the characteristics of procedural, object-oriented, and event-driven programming, analyze a suitable Integrated Development Environment (IDE)		
P2 Give explanations of what procedural, object-oriented, and event-driven paradigms are; their characteristics and the relationship between them.	**M2** Analyse the common features that a developer has access to in an IDE.	**D2** Critically evaluate the source code of an application that implements the programming paradigms, in terms of the code structure and characteristics.
LO3 Implement basic algorithms in code using an IDE **LO4** Determine the debugging process and explain the importance of a coding standard		
P3 Write a program that implements an algorithm using an IDE.	**M3** Use the IDE to manage the development process of the program.	**D3** Evaluate the use of an IDE for the development of applications contrasted with not using an IDE.

P4 Explain the debugging process and explain the debugging facilities available in the IDE. **P5** Outline the coding standard you have used in your code.	**M4** Evaluate how the debugging process can be used to help develop more secure, robust applications.	**D4** Critically evaluate why a coding standard is necessary for a team as well as for the individual.

Table of Contents

I) Introduction

I will discuss a suitable problem with creating a fully working, secure application that has been developed using an IDE. Coding standards in a detailed problem. I will list before designing, implementing (coding), and testing a solution. My report includes:

- Explain some of the programming paradigms and Evaluate it
- Explain the common features of IDE should have and how the IDE was used to manage the development of code.
- Introduction to the program
- An evaluation of developing applications using an IDE versus developing an application without using an IDE.
- An explanation and evaluation of the debugging process in the IDE
- Valuation of coding standards used in the program and its benefits.

II) Contents

1) Definitions, Characteristics of programming paradigms (P2).

A. Procedure-oriented programming.

1. Definition

It is a programming paradigm that divides a program into functions or subroutines. Each program can also be divided into many other subroutines to simplify their work. A program in procedural programming consists of data, modules, and procedures that manipulate the data. The two are dealt with as separate entities. A program is constructed from objects. An object consisting of the class is a collection of data (referred to as fields) and the operations (referred to as methods) that manipulate them.

2. Characteristics

- ✓ Focus on the work to be done (algorithms)
- ✓ A large program is divided into subroutines, each of which can be called one or more times in any order.
- ✓ Most functions use common data
- ✓ Data in the system is moved from one function to another.
- ✓ Function that converts data from one form to another
- ✓ Using a top-down approach in program design

3. Advantages

- ✓ Easy to deploy
- ✓ The large variety of books and course material
- ✓ Portable source code
- ✓ Less memory
- ✓ Easy tracking
- ✓ Code can be reused

4. Disadvantages

- ✓ Program code is harder to write
- ✓ Difficulty relating to real-world objects
- ✓ The data is displayed in its entirety, making it less secure

B. Object-oriented programming

1. Definition

Is a programming model that organizes software design around data or objects, rather than function and logic. An object can be defined as a data field with unique properties and behaviors.

2. Characteristics

- ✓ **Encapsulation**: All important information is contained within an object and only selected information is displayed. The implementation and state of each object are kept private within a defined class. Other objects do not have permission to access or make

changes. Only a list of public functions or methods can be called. This data hiding feature makes the program more secure

- ✓ **Abstraction:** Abstraction is a characteristic of object-oriented programming that "shows" only the necessary properties and "hides" unnecessary information. The main purpose of abstraction is to hide unnecessary details from the user. Abstraction is selecting data from a larger pool to show only relevant details of the object to the user. It helps to reduce the complexity of the program.
- ✓ **Inheritance:** Classes can reuse code from other classes. Allows developers to reuse common logic while still maintaining a single hierarchy.Binding
- ✓ **Polymorphism**: Objects are designed to share behaviors and they can take many forms. The program determines what meaning or usage is needed for each execution of that object from the parent class, reducing the need for code duplication. A subclass is created to extend the functionality of the parent class. Polymorphism allows different types of objects to pass through the same interface.

3. Advantages

- ✓ Modularity for simpler troubleshooting
- ✓ Reusing code by inheriting it
- ✓ Adaptability via polymorphism
- ✓ Problem-solving that works

4. Disadvantages

- ✓ Large program size
- ✓ Many projects lead to slow progress
- ✓ Unsuitable for a variety of problems

C. Event-driven programming

1. Definition

Event-driven programming is a programming paradigm in which the flow of program execution is determined by events. Event-driven programming focuses on events. Ultimately, the flow of the program depends on the events.

2. Characticsic

- ✓ Time-Driven

- ✓ Trigger Function
- ✓ Events
- ✓ The simplicity and Easy development
- ✓ Service- Oriented

3. Advantages

- ✓ Flexibility
- ✓ Compatible with the graphical interface
- ✓ Simplicity and Understandable
- ✓ Purely Procedural and Purely Imperative
- ✓ Allows interaction with multiple programs

4. Disadvantages

- ✓ Less logical
- ✓ Difficult to find the error
- ✓ Slower

D. The relationship between the 3 programming paradigms

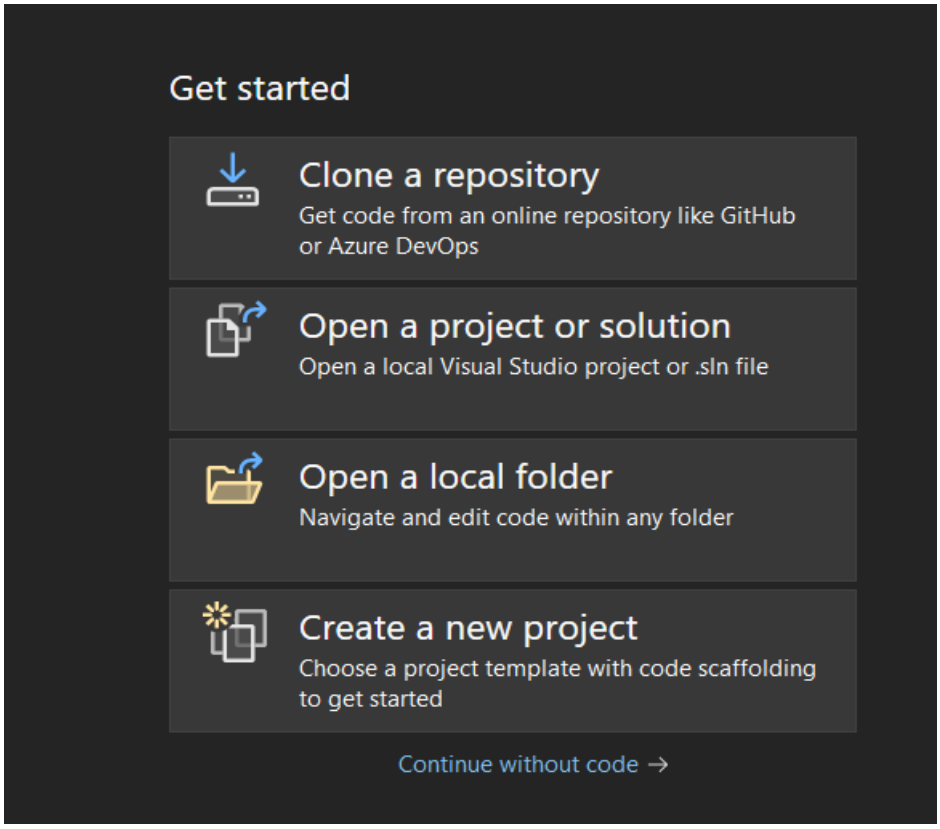
- ➢ Procedural programming is written as a set of steps that happen one after another, from start to finish. In contrast, declarative programs try to describe the result without regard to the steps taken by the computer but rather with some description or denotation of the desired result.
- ➢ Objec-oriented programming is defining an object and the functions it can perform. These are the functions that other objects can inherit to create new objects that contain earlier ones. Obje-oriented programs are usually mostly procedural.
- ➢ Event-Driven programming is about writing event-handling procedures and having the event loop provided by the basic system. You can save the trouble of writing your event loop and benefit from various libraries in this way. Event-Driven programs are very often written using an object-oriented style, but not always.
- ➢ Therefore, these three categories are not related in a hierarchical sense, but rather, in everyday usage, they are generally nested inside of one another.

2) A program that implements an algorithm using an IDE (P3).

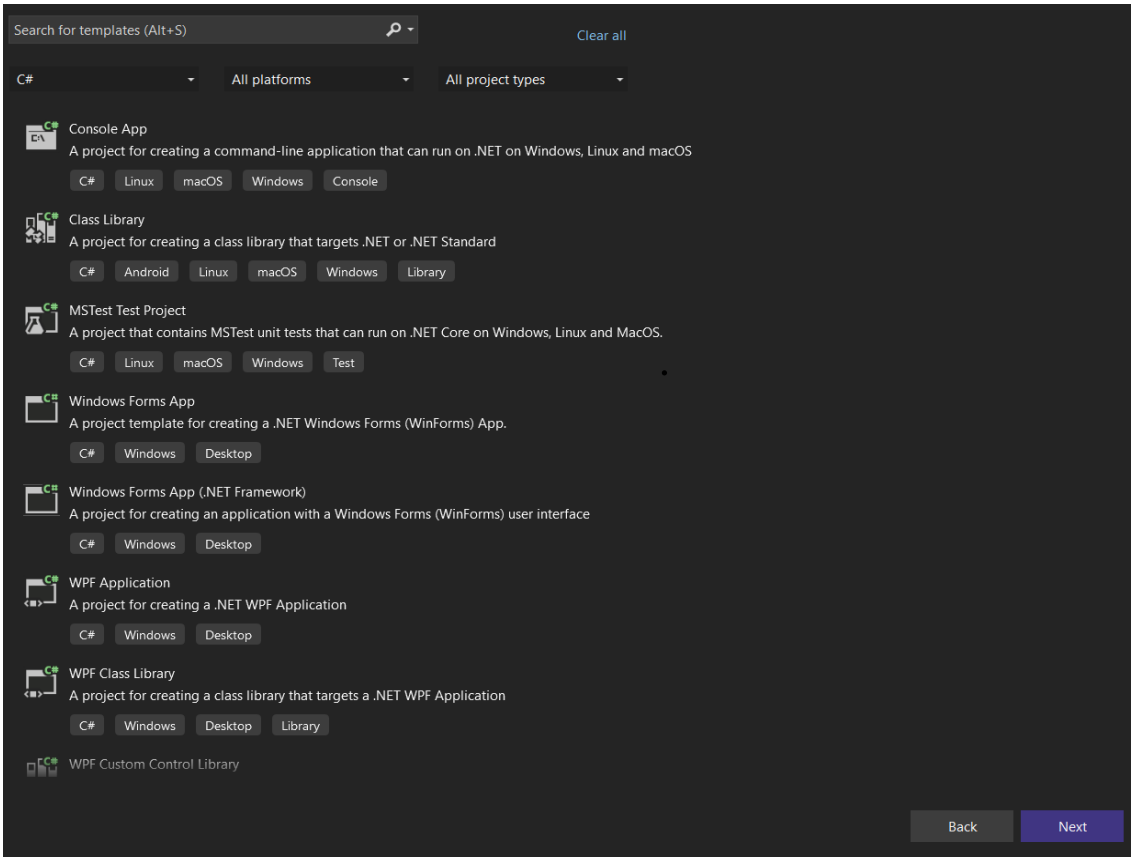
In this article I build a store management application. The application requires users to be logged in to be able to perform operations in the program. The operations (features) that the program allows users to perform after successful login are adding, editing, deleting, searching, printing invoices, product information. In the sales section, the application allows users to search for products and buy the desired product. In the statistics category, the application supports users to calculate the number of goods, orders, and sales of the goods.

A. Process of creating a new project in Winform

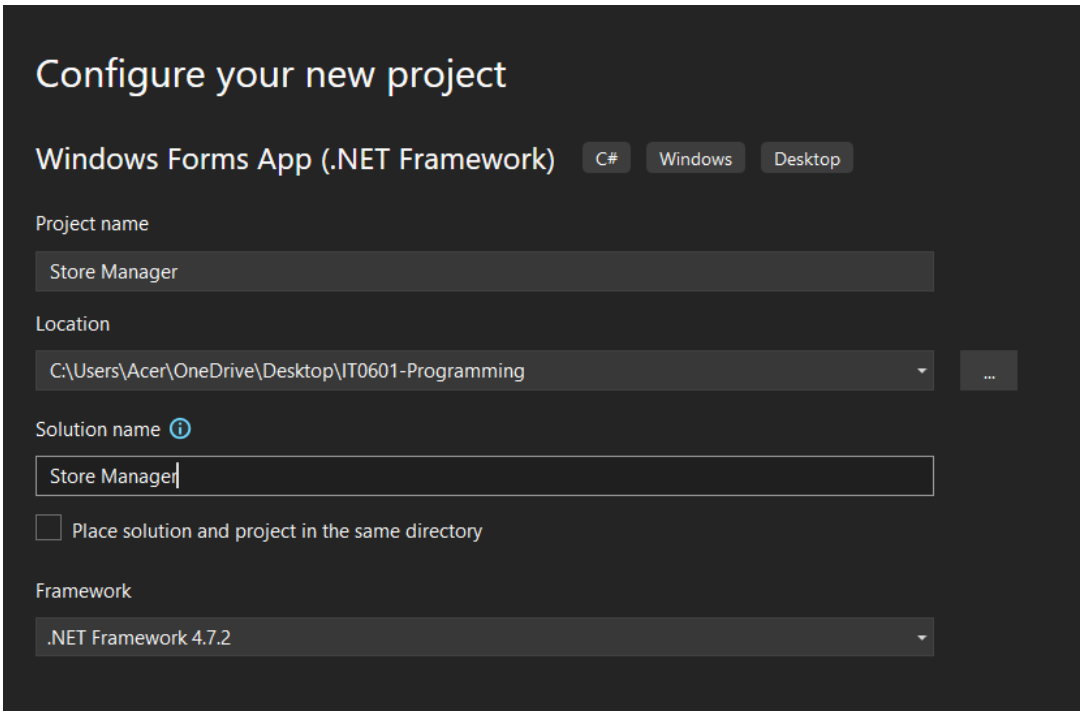
✓ Step 1: Click " **Create a new project**"



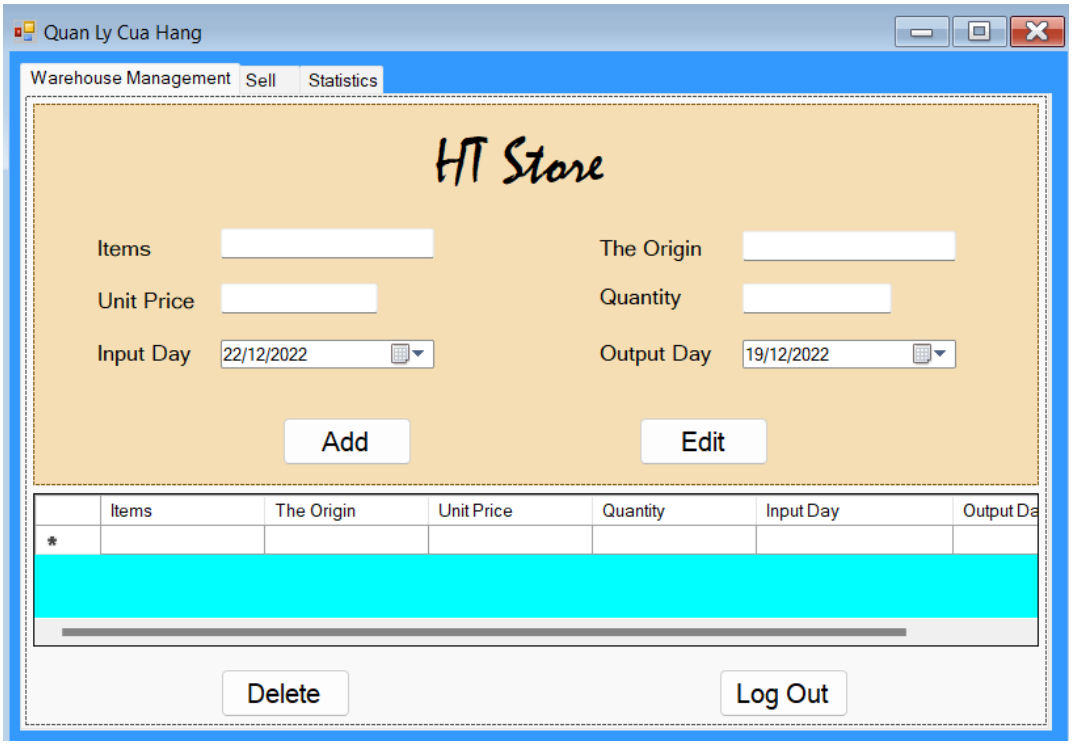
✓ Step 2: Choose " **Windows Form App (.NET Framework)**" and click " **Next**"

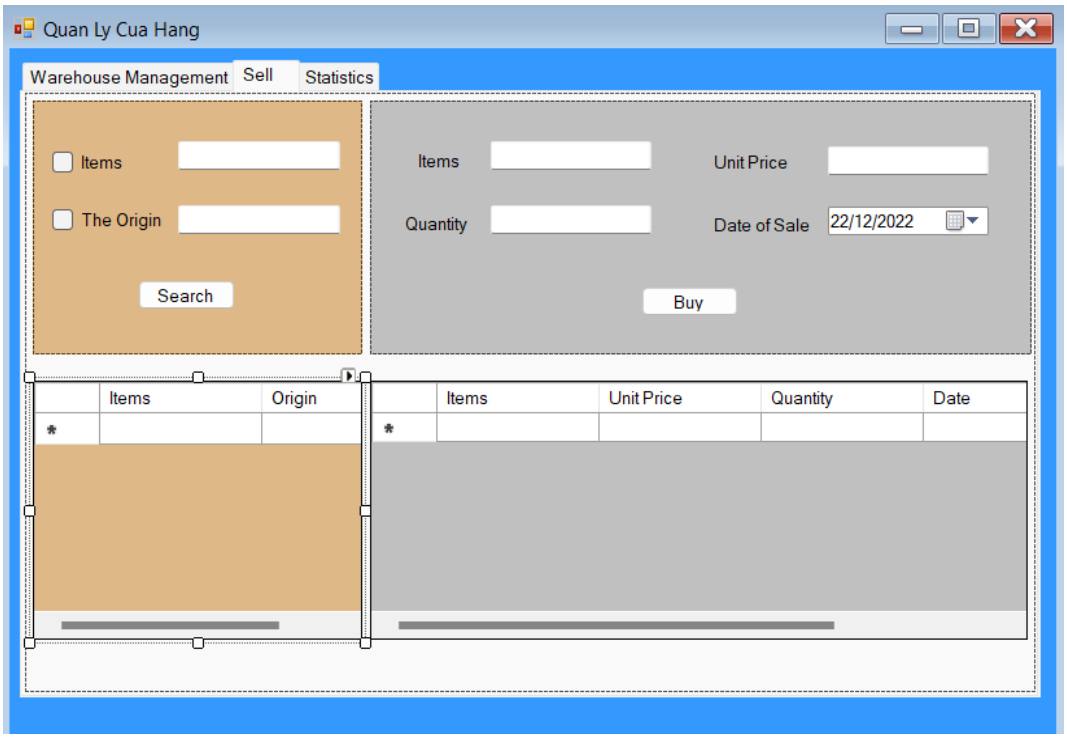


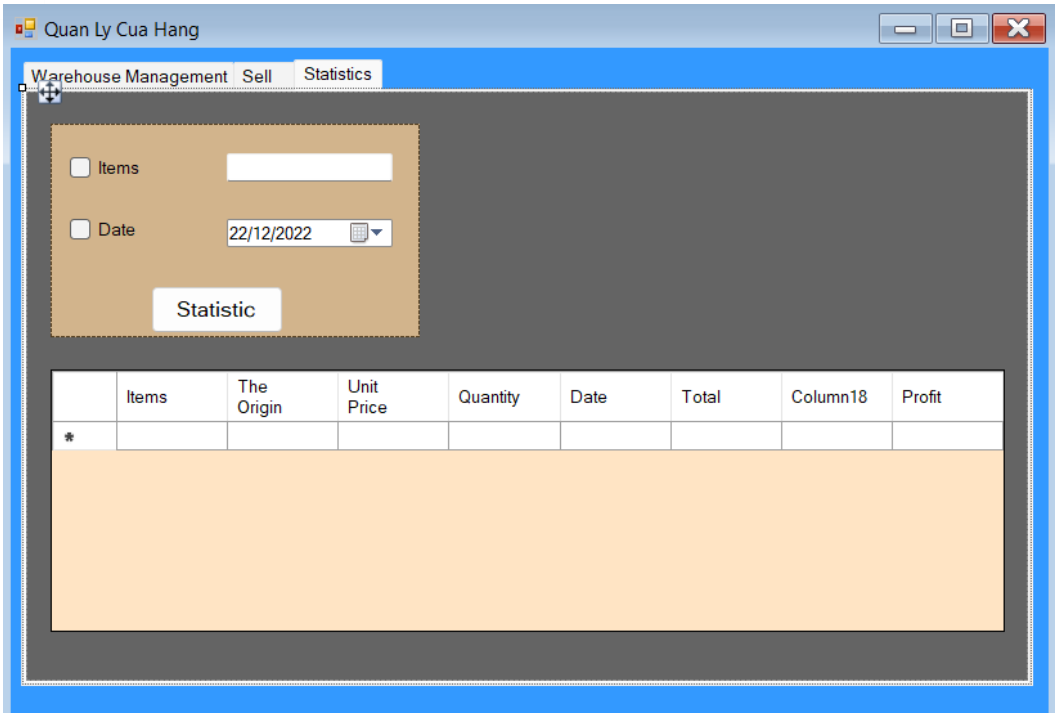
✓ Step 3: Here, Set the "**Project name**" and "**Solution name**". Then click " **Create**"



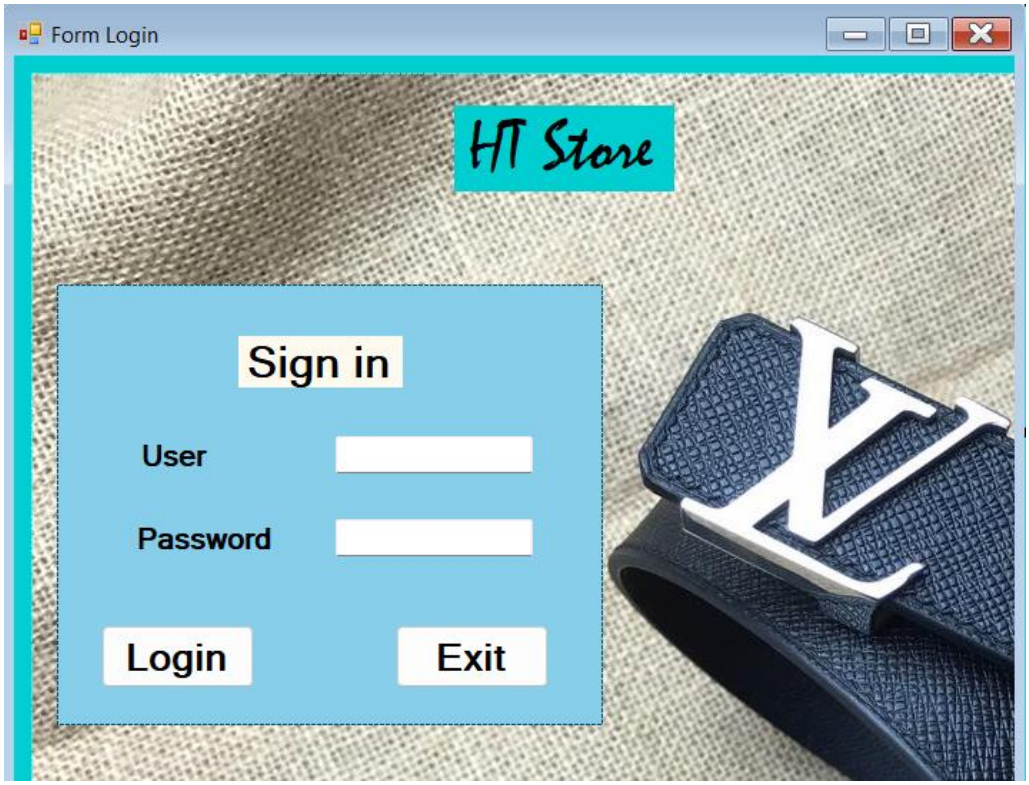
✓ Step 4: Design the form as follows







✓ To secure information, we need to set up a login form



✓ To manage products, we need to declare class " Sanpham"

```
namespace QuanLyCuaHang
{
    4 references
    internal class Sanpham
    {
        3 references
        public string Items { get; set; }
        3 references
        public string Origin { get; set; }
        3 references
        public string Price { get; set; }
        3 references
        public string Quantity { get; set; }
        4 references
        public string InputDay { get; set; }
        2 references
        public string OutputDay { get; set; }

    }
}
```

✓ Next, we declare the array " Sanpham "

```
namespace QuanLyCuaHang
{
    { } namespace QuanLyCuaHang
    4 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        // khai bao mang doi tuong sp
        Sanpham[] listSanPham = new Sanpham[100];
        //khai bao luu thu tu sp
        int n = 0;
        2 references
        private void Form1_Load(object sender, EventArgs e)
        {

        }
```

✓ Event when clicking the “Add” button and adding each object in turn

```
private void btnAdd_Click(object sender, EventArgs e)
{
    //Tao moi mot doi tuong san pham
    Sanpham sanpham = new Sanpham();
    sanpham.Items = txtTensp.Text;
    sanpham.Origin = txtXuatxu.Text;
    sanpham.Price = txtGia.Text;
    sanpham.Quantity = txtSl.Text;
    sanpham.InputDay = txtNgaynhap.Text;
    sanpham.OutputDay = txtNgayxuat.Text;

    // dua lan luot doi tuong vao
    listSanPham[n] = sanpham;
    n++;
    txtTensp.Clear();
    txtXuatxu.Clear();
    txtGia.Clear();
    txtSl.Clear();
    display();
}
```

✓ Function returns

```
1 reference
public void display()
{
    dgvCuahang.DataSource = null;
    dgvCuahang.DataSource = listSanPham;
    dgvCuahang.Refresh();
}
```

✓ Event when click on " DataGridView"

```
int index;
1 reference
private void dgvCuahang_CellClick(object sender, DataGridViewCellEventArgs e)
{
    // Lay chi muc tung dong
    index = e.RowIndex;
    //Hien thi thong tin tung dong
    txtTensp.Text = listSanPham[index].Items.ToString();
    txtXuatxu.Text = listSanPham[index].Origin.ToString();
    txtGia.Text = listSanPham[index].Price.ToString();
    txtSl.Text = listSanPham[index].Quantity.ToString();
    txtNgaynhap.Text = listSanPham[index].InputDay.ToString();
    txtNgayxuat.Text = listSanPham[index].InputDay.ToString();
}
```

✓ Event when click on " Edit" button

```
1 reference
private void btnEdit_Click(object sender, EventArgs e)
{
    // Lấy dữ liệu từ textbox đưa vào  dgv
    listSanPham[index].Items = txtTensp.Text;
    listSanPham[index].Origin = txtXuatxu.Text;
    listSanPham[index].Price = txtGia.Text;
    listSanPham[index].Quantity = txtSl.Text;
    listSanPham[index].InputDay = txtNgaynhap.Text;
    listSanPham[index].OutputDay = txtNgayxuat.Text;
    dgvCuahang.DataSource = null;
    dgvCuahang.DataSource = listSanPham;
    dgvCuahang.Refresh();

}
```

✓ Event when click " Delete" button

```
1 reference
private void btnDelete_Click(object sender, EventArgs e)
{
    bool alert;
    alert = MessageBox.Show("Do you want to Delete?", "Warning", MessageBoxButtons.OKCancel,
    if (alert)
    {
        //Khi xóa thì phần tử vị trí thứ index sẽ bị thay thể bởi phần tử thứ index+1
        while (index < n)
        {
            listSanPham[index] = listSanPham[index + 1];
            index++;
        }
    }
    listSanPham[n - 1] = null;
    n = n - 1;

    dgvCuahang.DataSource = null;
    dgvCuahang.DataSource = listSanPham;
    dgvCuahang.Refresh();
}
```

✓ Event when click " Log out" button.

```
1 reference
private void btnLogout_Click_1(object sender, EventArgs e)
{

    DialogResult l = DialogResult;
    l = MessageBox.Show("Do you want to log out?", "Warning!", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (l == DialogResult.Yes)
    {
        Application.Exit();
    }
}
```

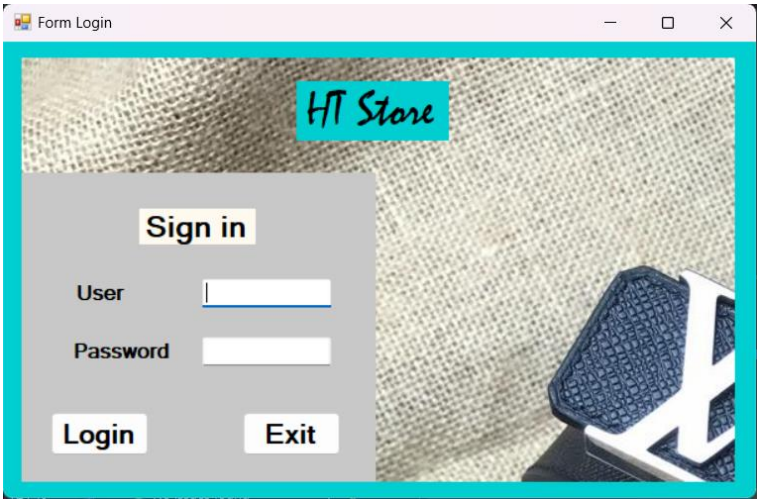
✓ Event when click " Log in" button

```
1 reference
private void btnLogin_Click_2(object sender, EventArgs e)
{
    if (CheckLogin())
    {
        string user = txtUser.Text;
        string pass = txtPass.Text;
        if (user == "admin" && pass == "123")
        {
            MessageBox.Show("Successfully Login");
            Form1 c = new Form1();
            this.Hide();
            c.ShowDialog();
            this.Dispose();
        }
        else
        {
            MessageBox.Show(this, "Incorrect account or password!!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

    }
}
```

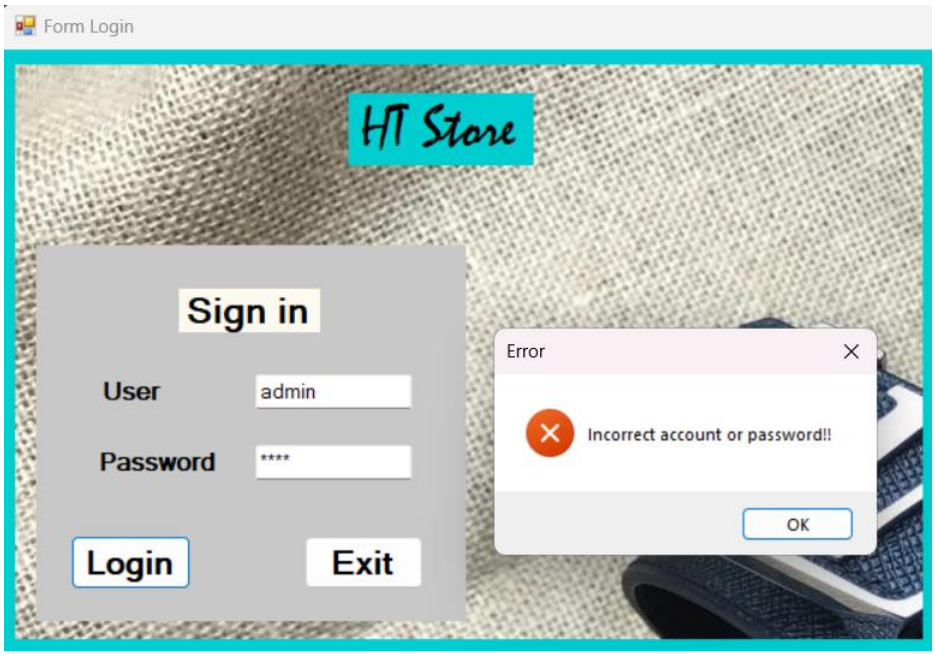
✓ Function CheckLogin

```
// ckeck login
1 reference
public bool CheckLogin()
{
    if (string.IsNullOrWhiteSpace(txtUser.Text))
    {
        MessageBox.Show(this, "Account can not blank", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtUser.Focus();
        return false;
    }
    if (string.IsNullOrWhiteSpace(txtPass.Text))
    {
        MessageBox.Show(this, "Password can not blank", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtPass.Focus();
        return false;
    }
    return true;
}

1 reference
private void btnExitlogin_Click_1(object sender, EventArgs e)
{
    Application.Exit();
}
```

- Interface at program start, user need Sign in with account available

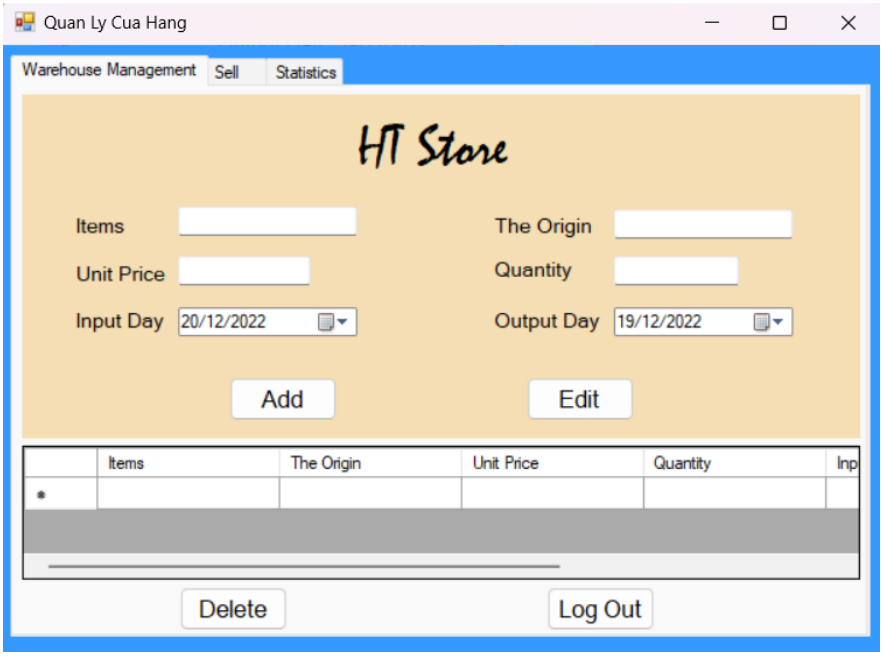


- Interface when entering the username and password if the password is incorrect

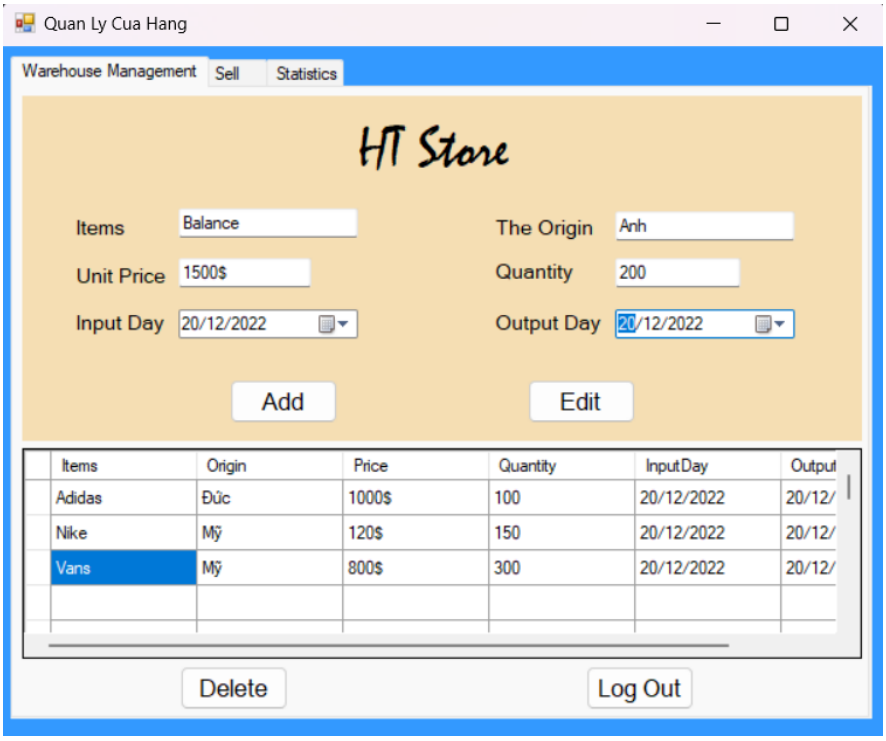


- Interface when entering the username and password if the password is correct

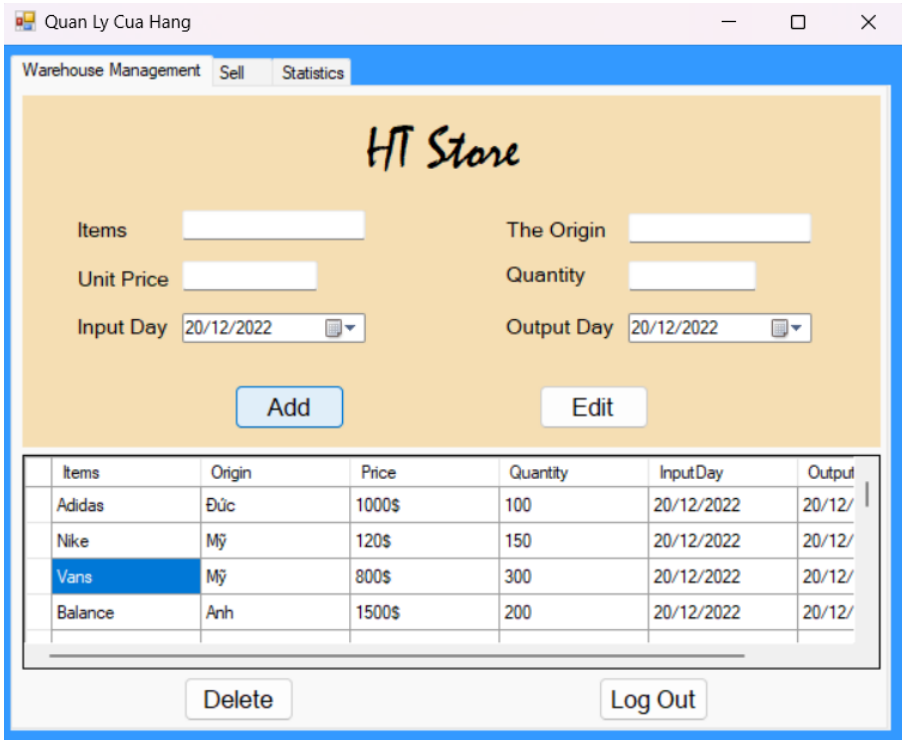
- Click Ok, Switch to the store management interface, Here, users can enter product information



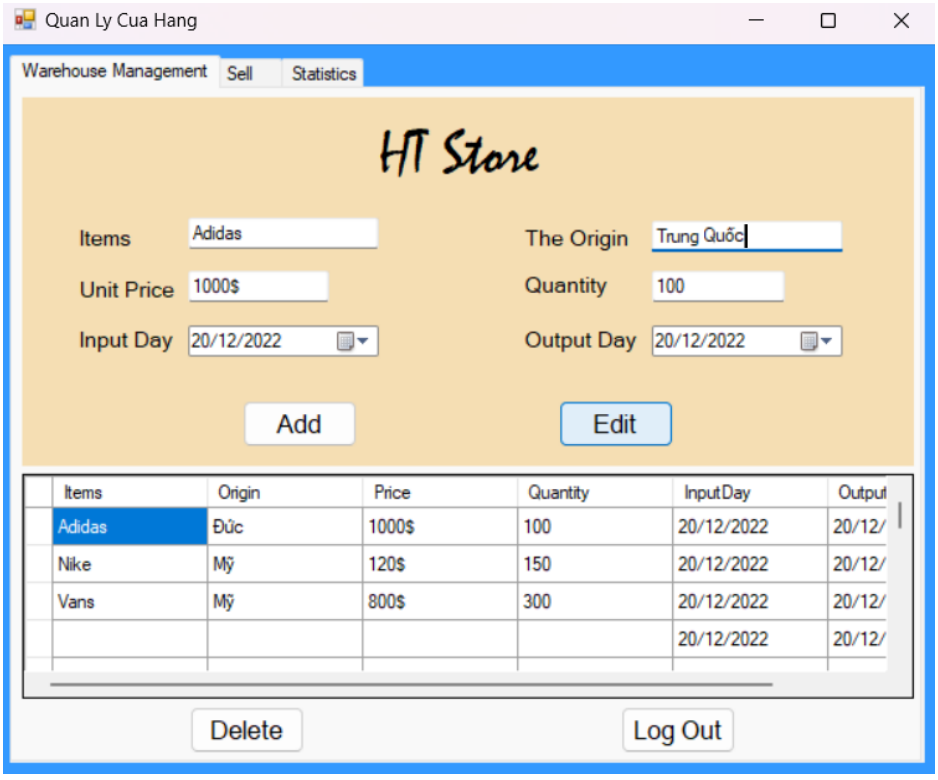
- Enter product information and press Add



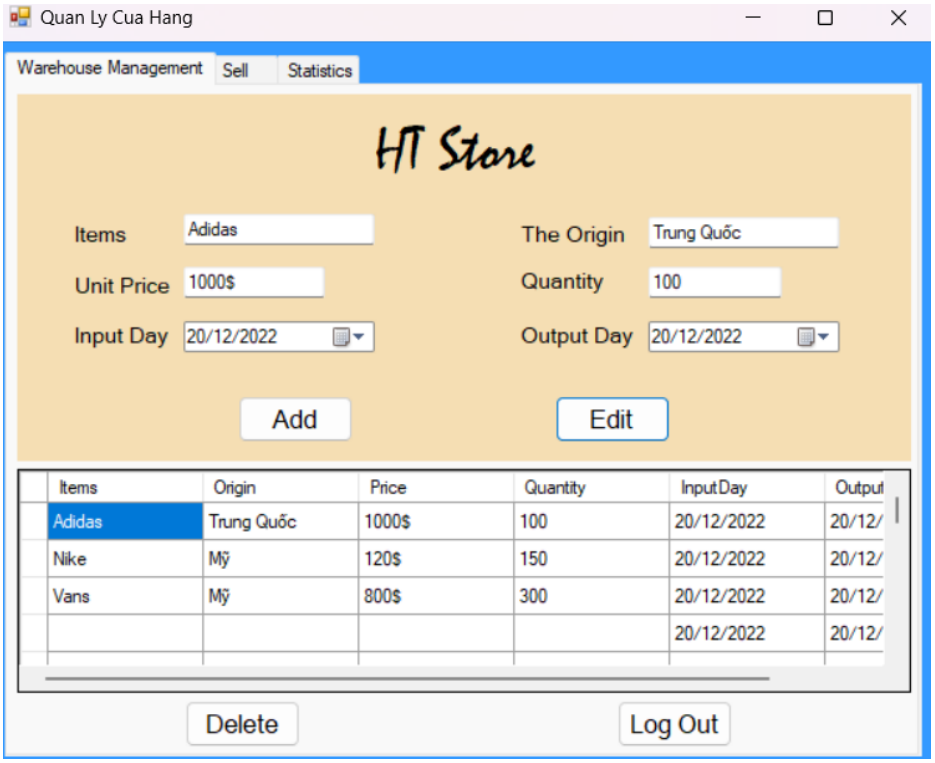
- After click Add



- To change the information, select the information to correct and click Edit



- Click Edit to finish editing



- Delete similar

3) The debugging process and the debugging facilities available in the IDE (P4).

A. Definitions

Debugging is the process of finding and fixing coding errors in a program. The goal of debugging is to identify and fix the root cause of the error. Debugging plays a crucial role in the software development process. It can take a long time to fully identify and eliminate the error. The debugging process itself includes determining the cause of the error and correcting it

B. The benefits

✓ Help report the error condition whenever it happens
✓ Make the software development process unhindered
✓ It provides information regarding data structures that make interpretation easier

- ✓ Help reduce unrealistic information
- ✓ Avoiding complicated test code reduces software build time

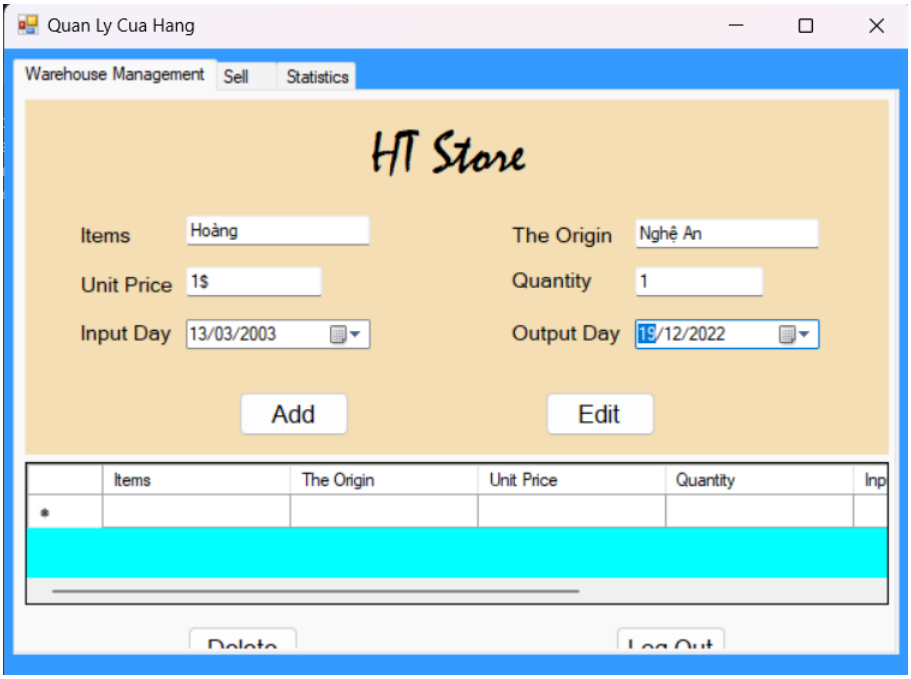
C. Debugging process

a. **Step1:** Set BreakPoint

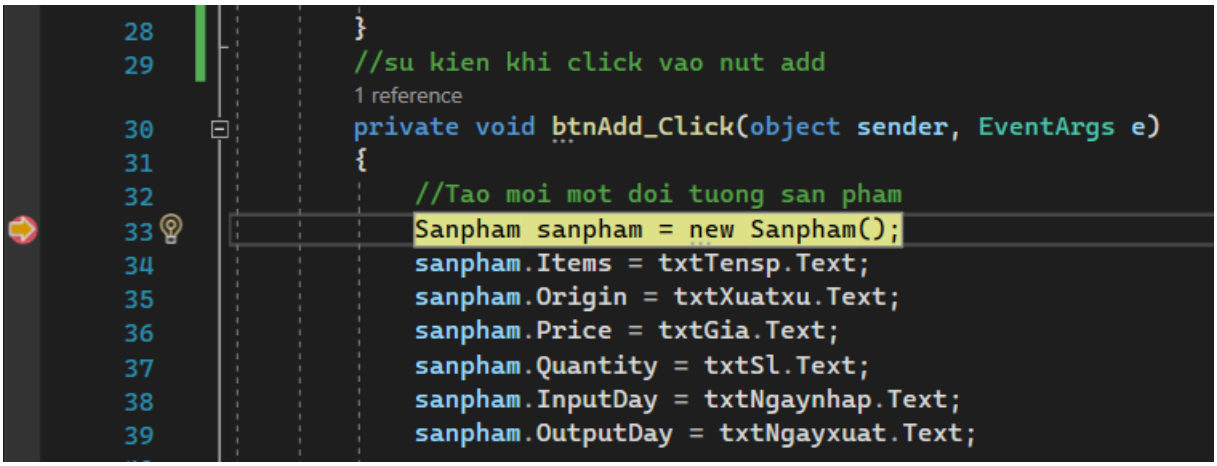
Move the red dot in the leftmost margin to the place you want to check

```
1 reference
30  private void btnAdd_Click(object sender, EventArgs e)
31  {
32      //Tao moi mot doi tuong san pham
33      Sanpham sanpham = new Sanpham();
34      sanpham.Items = txtTensp.Text;
35      sanpham.Origin = txtXuatxu.Text;
36      sanpham.Price = txtGia.Text;
37      sanpham.Quantity = txtSl.Text;
38      sanpham.InputDay = txtNgaynhap.Text;
39      sanpham.OutputDay = txtNgayxuat.Text;
40
```

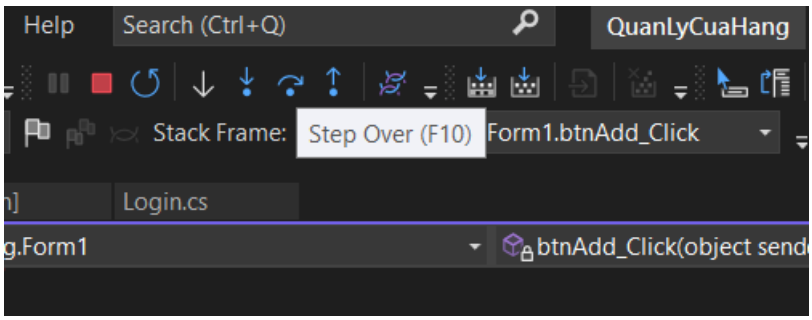
b. **Step2:** Start to debug



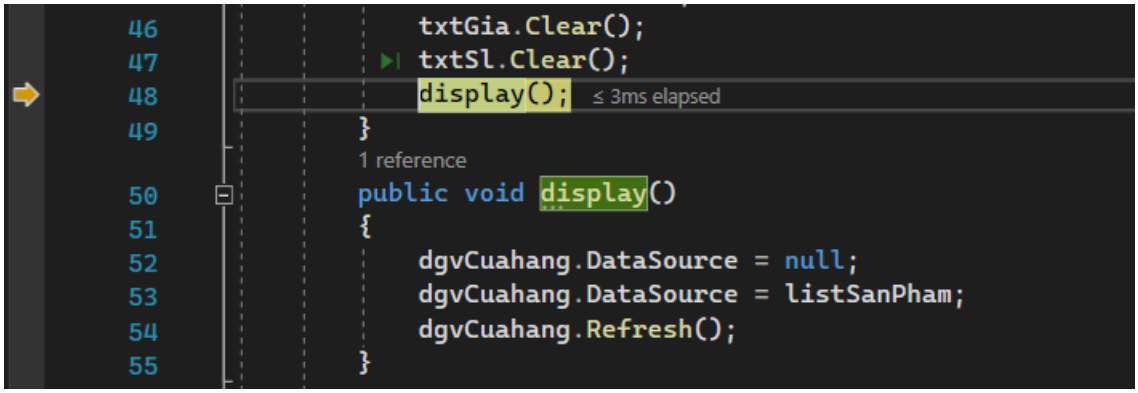
When the program runs to the set Breakpoint, the program will stop.



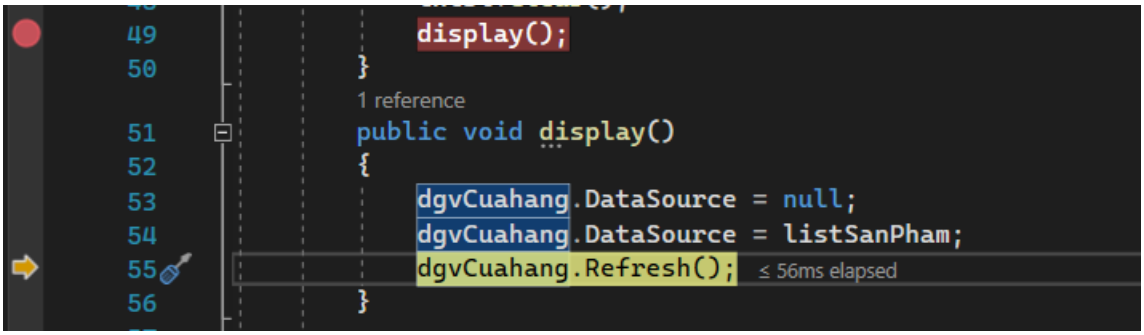
To continue, press F10 (Fn + F10) or " Step over"
step over: The program will run step by step, going through the function (only getting the return value of the function)



To a function, another function is running, the program will stop

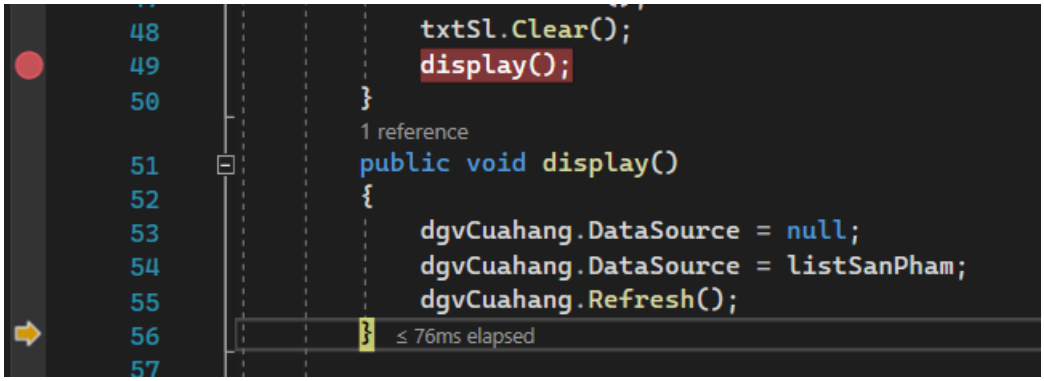


To run into the function, press "step into" or F11 (Fn + F11)



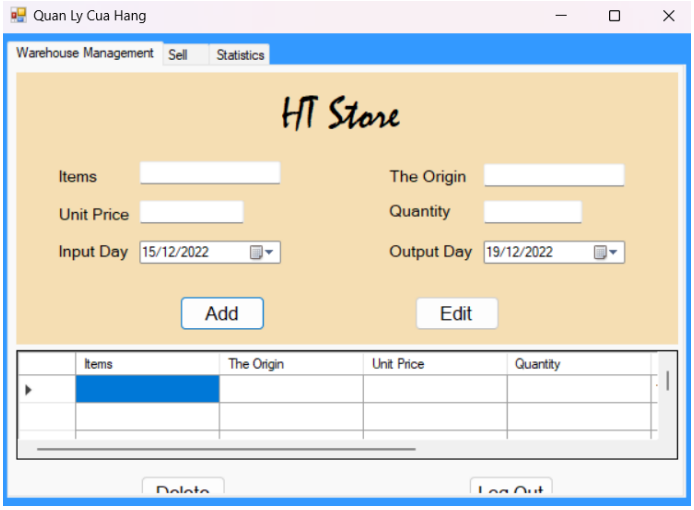
step into: The program runs step by step, going inside the subfunctions

After running in the function, press "step over" to exit



To exit the function and continue the program, press " step out" of F11(Fn+F11)

step out: When step out is pressed, the program jumps to the next break point. If there are no more breakpoints, it will end debugging.



After pressing "step out", the program continues to run normally

4) Code Standard (P5).

A. Definition

Code standards are a set of principles, best practices, programming styles, and conventions that developers follow when writing source code for a project. In other words, it is a set of principles when programming that makes code easier to read, understand, and maintain. Developers should follow these standards to keep the code maintainable, keep it transparent, sane and readable, and keep it scalable.

B. The benefits

- Provides a unified appearance for code written by different engineers.
- Improves the readability and maintainability of code reducing complexity.
- Helps reuse code and makes error detection easy
- Promote sound programming practice and increase efficiency

C. Code standards used in the program

a. Naming standards

A naming convention is a set of rules for choosing the character sequence to be used for identifiers which denote variables, types, functions, and other entities in source code and documentation

- **Variable names**: lowercase letters, meaningful names...

```
// khai bao mang doi tuong sp
Sanpham[] listSanPham = new Sanpham[100];
//khai bao luu thu tu sp
int n = 0;
2 references
private void Form1_Load(object sender, EventArgs e)
{

}
```

In the above code I used standard code to name variable int

❖ **Function name**: Capitalize the first letter...

```
if (CheckLogin())
{
    string user = txtUser.Text;
    string pass = txtPass.Text;
    if (user == "admin" && pass == "123")
    {
        MessageBox.Show("Successfully Login");
        Form1 c = new Form1();
        this.Hide();
        c.ShowDialog();
        this.Dispose();
    }
    else
    {
        MessageBox.Show(this, "Incorrect account or password!!",
    }

}
```

In my above code, the functions declared in the class are capitalized so that it makes sense

❖ **Class name**: The class name must be a capitalized letter noun, the class name should be simple and descriptive, avoid abbreviations

```
4 references
internal class Sanpham
{
    3 references
    public string Items { get; set; }
    3 references
    public string Origin { get; set; }
    3 references
    public string Price { get; set; }
    3 references
    public string Quantity { get; set; }
    4 references
    public string InputDay { get; set; }
    2 references
    public string OutputDay { get; set; }

}
```

b. Standard layout

Clear, logical, easy to understand layout

```
// ckeck login
1 reference
public bool CheckLogin()
{
    if (string.IsNullOrWhiteSpace(txtUser.Text))
    {
        MessageBox.Show(this, "Account can not blank", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtUser.Focus();
        return false;
    }
    if (string.IsNullOrWhiteSpace(txtPass.Text))
    {
        MessageBox.Show(this, "Password can not blank", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtPass.Focus();
        return false;
    }
    return true;
}
```

D. Standard code annotation

```
    }
    listSanPham[n - 1] = null;
    n = n - 1;

    dgvCuahang.DataSource = null;
    dgvCuahang.DataSource = listSanPham;
    dgvCuahang.Refresh();
}
1 reference
private void btnLogout_Click_1(object sender, EventArgs e)
{

    DialogResult l = DialogResult;
    l = MessageBox.Show("Do you want to log out?", "Warning!", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (l == DialogResult.Yes)
    {
        Application.Exit();
    }
}
```

Some code annotation standards that I use: null, Exit, Refresh

5) Common features of the IDE(M2).

A. What is IDE?

An IDE (integrated development environment) is a software application that combines, in one place, all the necessary tools for a software development project. At a more basic

level, the IDE provides an interface for users to write code, organize groups of text, and automate programming fallbacks. But instead of a basic code editor, the IDE combines the functionality of multiple programming processes into one. The "built-in" component of the IDE is really what sets it apart from simple code editing tools. Debuggers, compilers, and automation capabilities also contribute. But one of the biggest reasons why IDEs can be beneficial over other tools is the ability to customize the environment with plugins and integrations. Plugins help customize workflows and add functionality.

B. Common features in IDE

IDE has been around for decades. From merely being a platform for debugging and testing purposes to integrated software that can be a developer extension, IDEs continue to evolve and change over time. Here are some standard features that developers have access to in the IDE:

- **Text editor:**

A text editor created to write and alter source code will be present in almost every IDE. Some tools may have components for dragging and dropping user interface elements, but most have a simple interface that highlights language-specific syntax.

- **Debugger**:

Debugging tools that assist users in identifying and correcting errors in source code. They frequently simulate real-world actual situations to evaluate functionality and performance. Programmers and software engineers can often test different pieces of code and identify bugs before the application is released.

- **Compilers**:

Compilers are components that translate a programming language into a machine-processable form, such as binary code. Machine code is analyzed to ensure its correctness. After that, pars and optimized by the compiler the code to improve performance.

- **Code completion:**

Code completion features assist programmers by intelligently identifying and inserting common code elements. These features help developers save coding time and reduce the chance of typos and errors.

- **Programming language support:**

IDEs are usually specific to a single programming language, although some also offer multilingual support. Therefore, the first step is to figure out what languages you will code in and narrow down your list of potential IDEs accordingly.

- **Integrations and plug-ins**:

With the name integrated development environment, it is not surprising that integrations need to be considered when looking at IDEs. Your IDE is your development portal, so incorporating all your other development tools improves the development process and productivity.

6) The IDE in the development process(M3).

7) Evaluate the debugging process(M4).

A. Introduction debugging process

Debugging is a core and important feature of Visual Studio Code. It is a powerful tool. Debugging means removing bugs from your code. It is a very specialized tool for programmers. It allows you to test your code.

B. Advantages

- ✓ **Save time**: Performing debugging at an early stage saves software developers time as they can avoid using complex codes during software development.
- ✓ **Report Errors**: It gives error reports as soon as they occur. This allows bugs to be detected at an early stage and makes the software development process hassle-free.
- ✓ **Easy Interpretation**: It provides easy interpretation by providing more information about the data structure.
- ✓ **Release software without bugs**: By finding bugs in software, it allows developers to fix them before release and provide bug-free software to users.

C. Disadvantages

Not running in real-time, so may not expose all problems.

D. Evaluate

In the process of writing programs, bugs are inevitable. So the debugger plays a very important role in the project development. Debugging is the process of diagnosing errors in a program and then trying to fix them. It allows the programmer to correct errors in the program to turn it into a more complete program. It's a process that complements testing, which involves learning how one bug affects the entire program. I find it very important to learn how to debug a program so that you can create a safer and more efficient application. Debugging will definitely help you and break down complex problems into simple problems. If you examine each part of the code, you can systematically discover which variables and functions need tuning. Debugging can also improve the quality of the project, thereby increasing the security and safety of the program. It doesn't matter how well you code, if you write the wrong program it won't help anyone. If you write the right program, but the user can't use it, then you haven't written the code. So a good debugger will tell you where the problem lies. Some basic troubleshooting steps:

- "Recognize if an error occurs".
- "Isolate the source of the error".
- "Determine the cause of the error".
- "Determine how to fix the error".
- "Repair and Inspection".

In this section I have shown you how to use the debugging tools in Visual Studio 2017 and their debugging facilities in detail, they are very powerful features and have a close debugging step. . In short, debugging helps programmers develop a more robust, secure application program. So this will be an important skill that a programmer should have. Debugging will be the key so that we can package the program into a perfect project.

III) Conclusion

In the above report, I discussed the issue of creating a fully functional, secure application developed with an IDE. Coding standards in a detailed issue. I have listed the user requirements The report explained some programming models common IDE features, introduced the program I designed myself, Explained and evaluated the debugging process in IDE, and evaluated the coding standards used in the program and its benefits. The above content is well worth your time to explore

IV) Reference material

- Procedure-oriented programming

https://www.techopedia.com/definition/21481/procedural-programming

- Advantages and Disadvantages of Object-Oriented Programming

https://hackr.io/blog/procedural-programming

- Object-oriented programming

https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP

https://www.roberthalf.com/blog/salaries-and-skills/4-advantages-of-object-oriented-programming

- Even-Driven Programming

https://studybay.com/blog/event-driven-development-features/

- Advantages and Disadvantages of Even-driven programming

https://benchpartner.com/advantages-and-disadvantages-of-event-driven-programming

- What Are IDE and Common Features

https://www.g2.com/articles/ide#why-is-an-ide-important

- Code Standard

https://www.geeksforgeeks.org/coding-standards-and-guidelines/

- Debugging

https://www.javatpoint.com/debugging