[ZANG] - [Chuanjie] - ITCAssignment

Student ID: 699382953

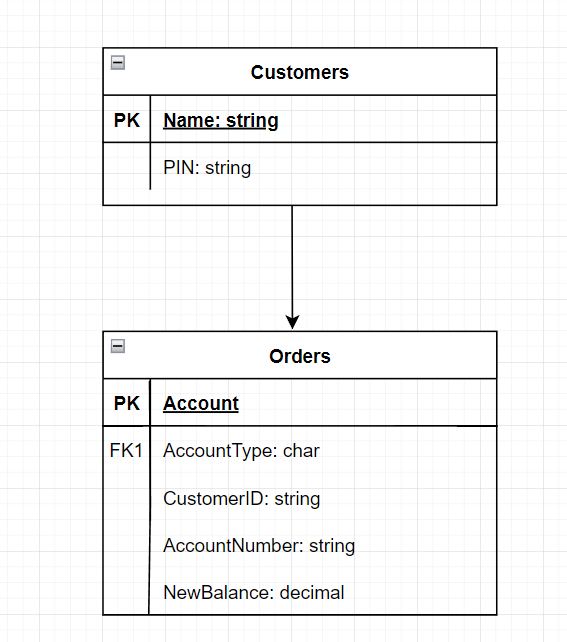
[chuanjiezang@gmail.com](mailto:chuanjiezang@gmail.com)

Part 1

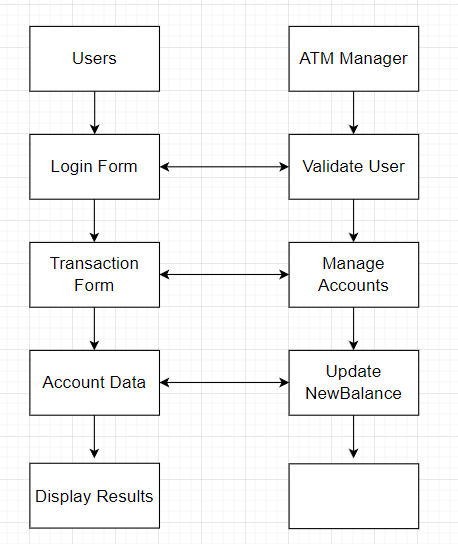
data dictionary

| **Data Element Name** | **data type** | **descriptive** |
| --- | --- | --- |
| Name | String | Customer's name |
| PIN | String | Customer's PIN, 4 characters in length |
| AccountType | Char | Account type, 'C' for checking account, 'S' for savings account |
| CustomerID | String | Customer's unique identifier |
| AccountNumber | String | Unique identifier for the account, 5 characters in length |
| AccountBalance | Decimal | Account balance |
| newBalance | Decimal | Current balance at the ATM |

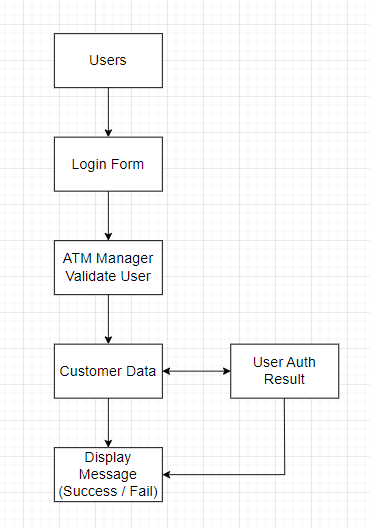
ER diagrams



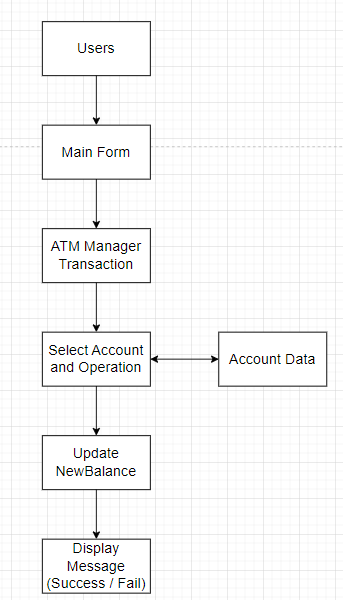
Level 0 DFD



Level 1 DFD - User Login Data Flow Diagram



Level 1 DFD - Account Operation Data Flow Diagram



Part 2

Use Visio to create flowcharts or pseudocode (at least one instance of each) to describe processes in an application.

Class Identification and Definition

Based on the project requirements and existing code, we can identify the following main classes: Customer

Customer

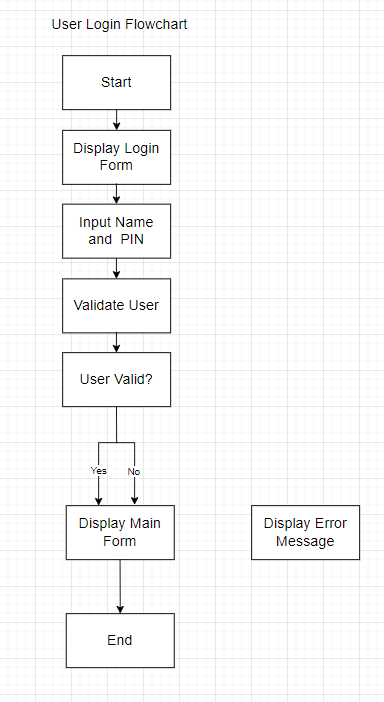
Account (base class)

Checking Account (derived from Account)

Savings Account (derived from Account)

ATM Manager

Form Classes (LoginForm, MainForm, SupervisorForm)



User Login Pseudo-Code

START

DISPLAY "Login Form"

INPUT name, pin

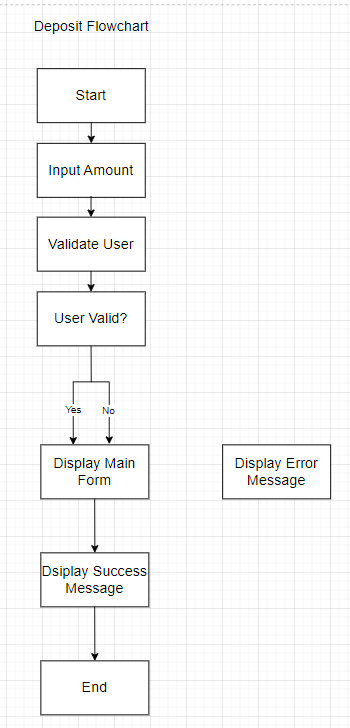
IF ValidateUser(name, pin) IS NOT NULL THEN

DISPLAY "Main Form"

ELSE

DISPLAY "Error: Invalid credentials"

END



Deposit Pseudo-Code

START

INPUT amount

IF ValidateUser(userName, userPIN) IS NOT NULL THEN

PERFORM Deposit(amount)

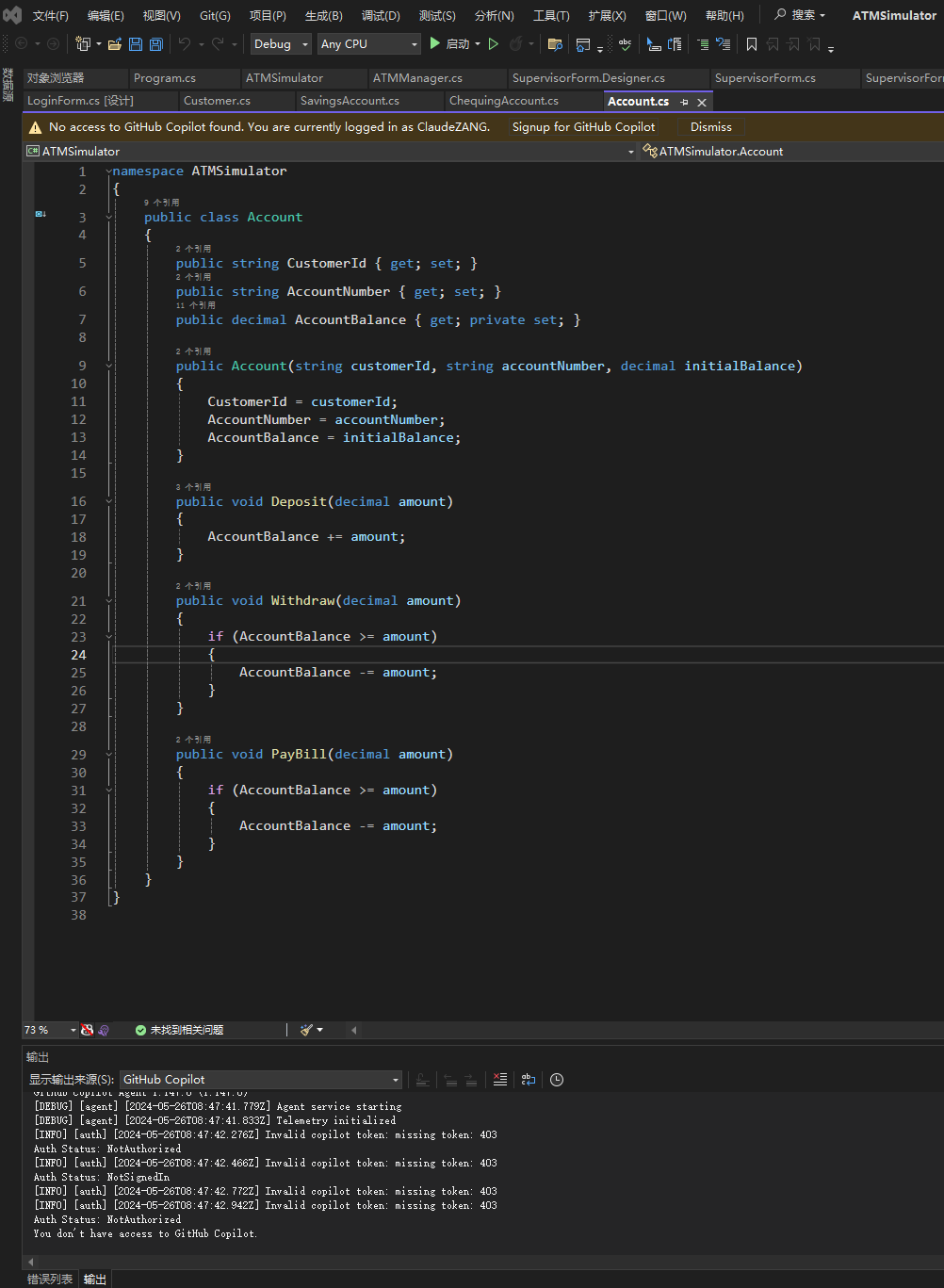
DISPLAY "Deposit successful"

ELSE

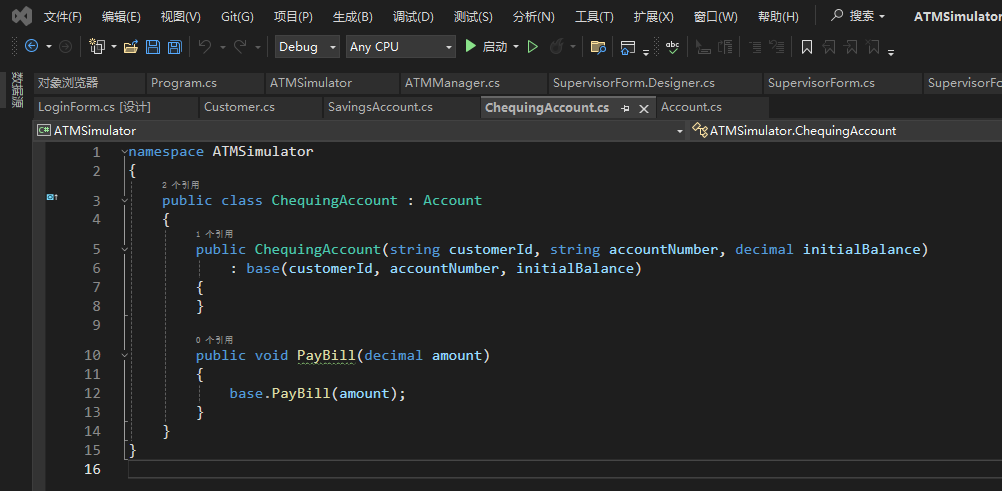
DISPLAY "Error: Invalid credentials"

END

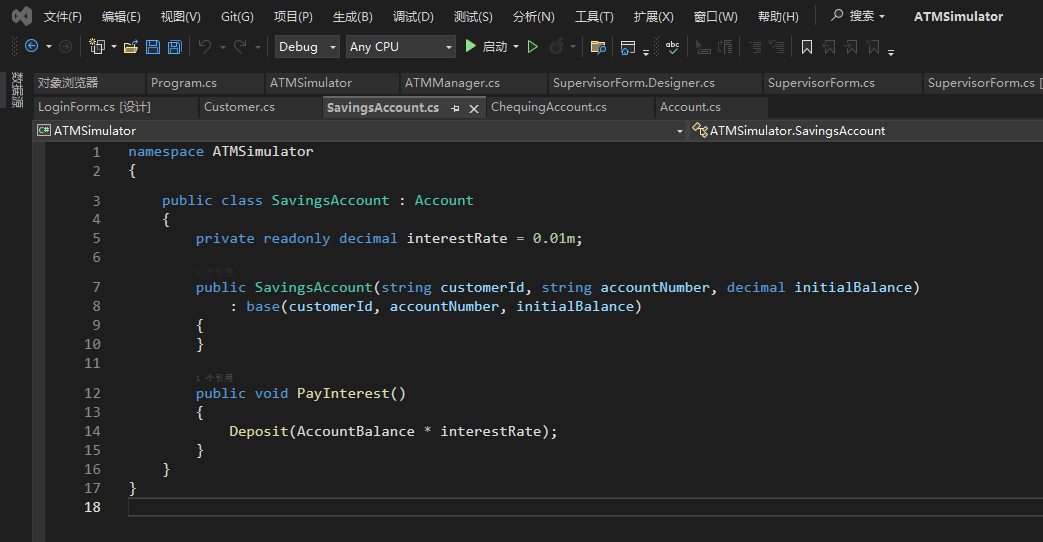
Account (base class)



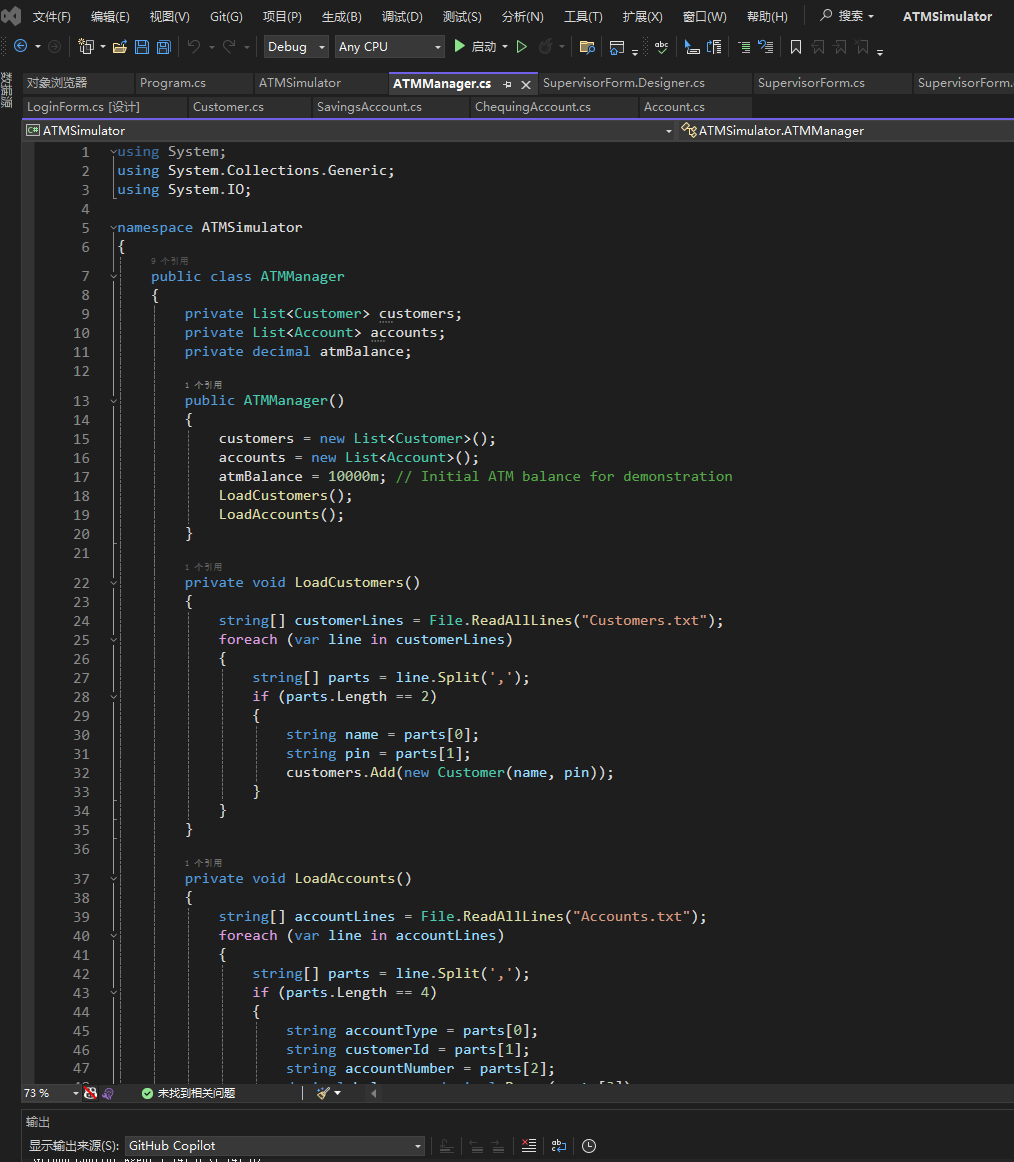
Checking Account (derived from Account)

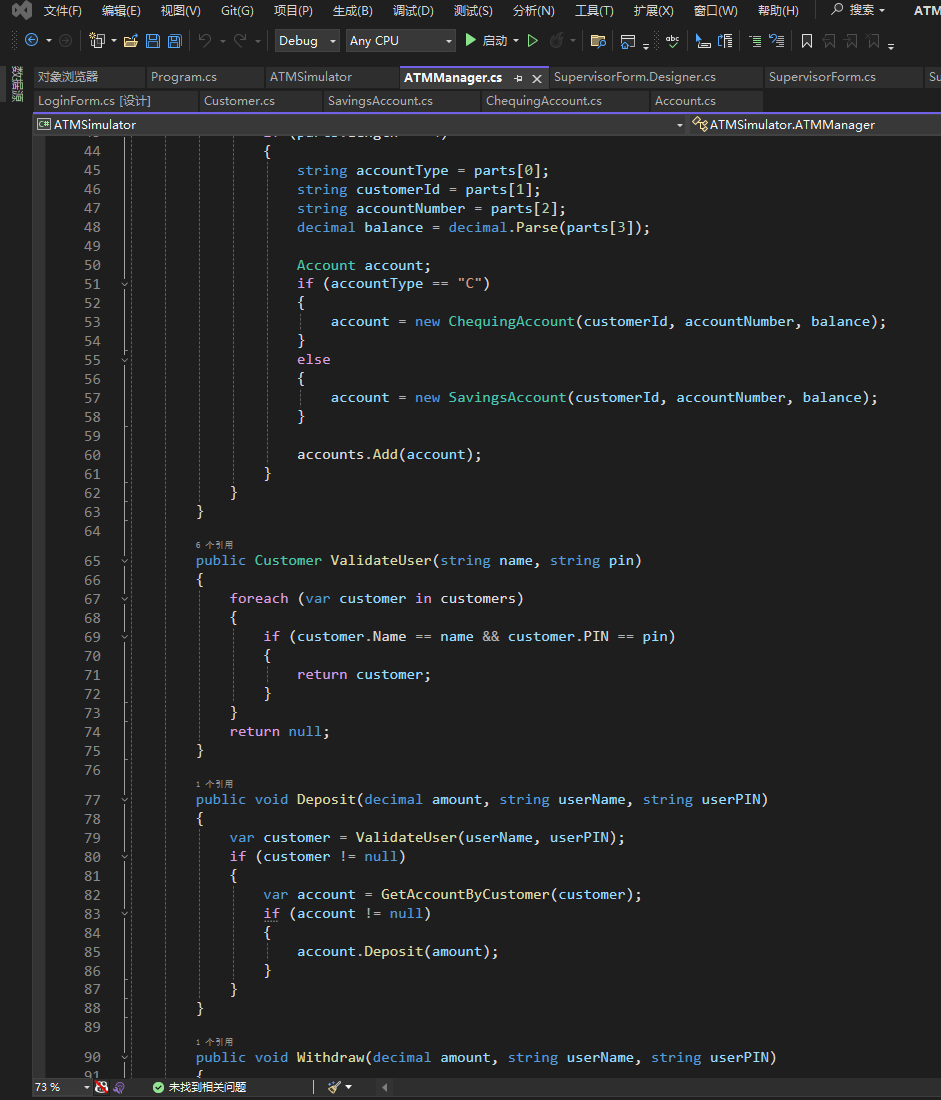


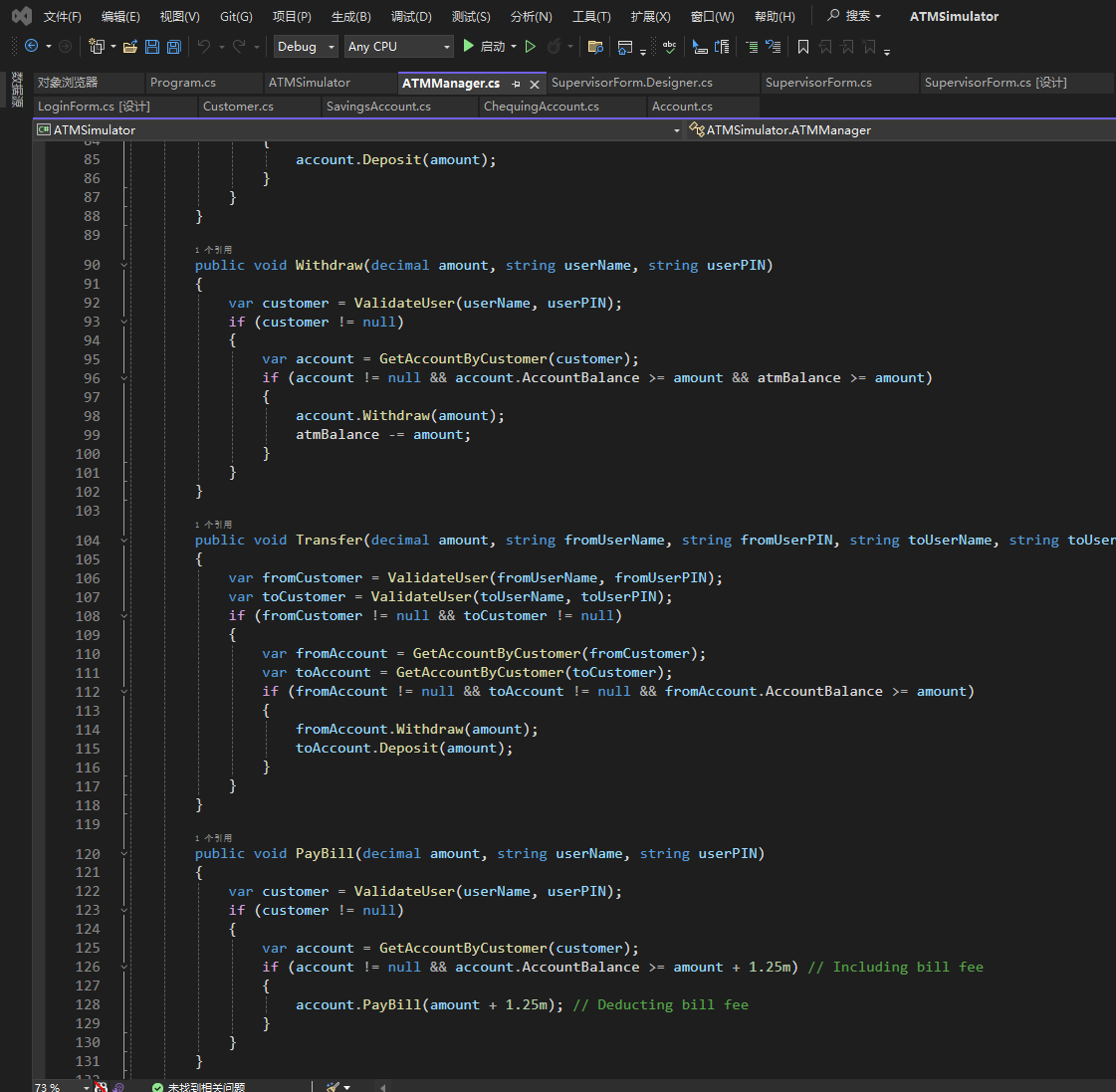
Savings Account (derived from Account)

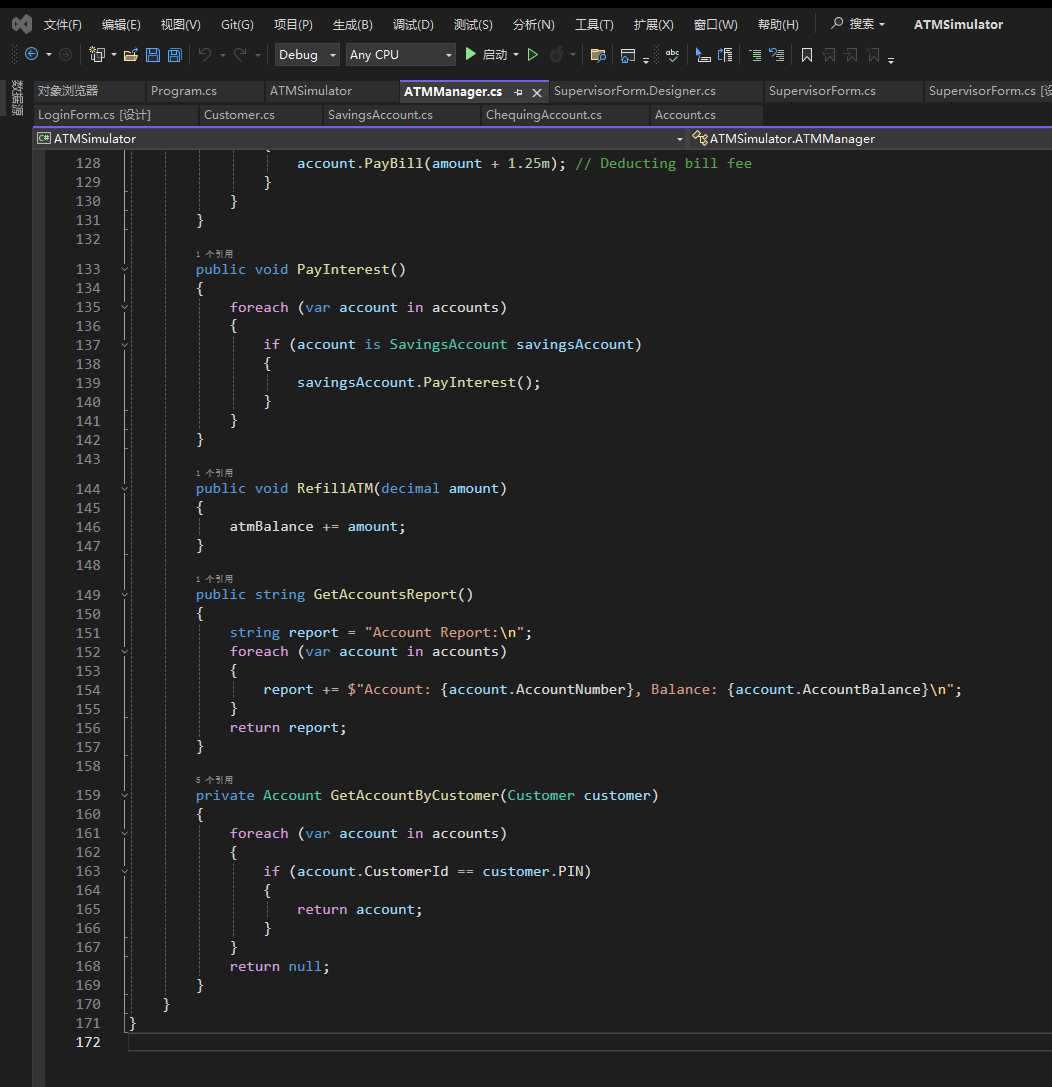


ATM Manager

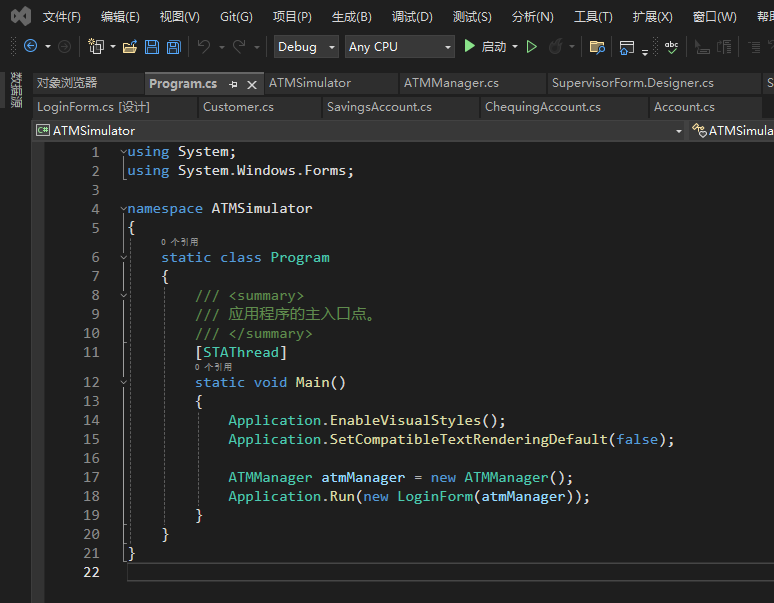


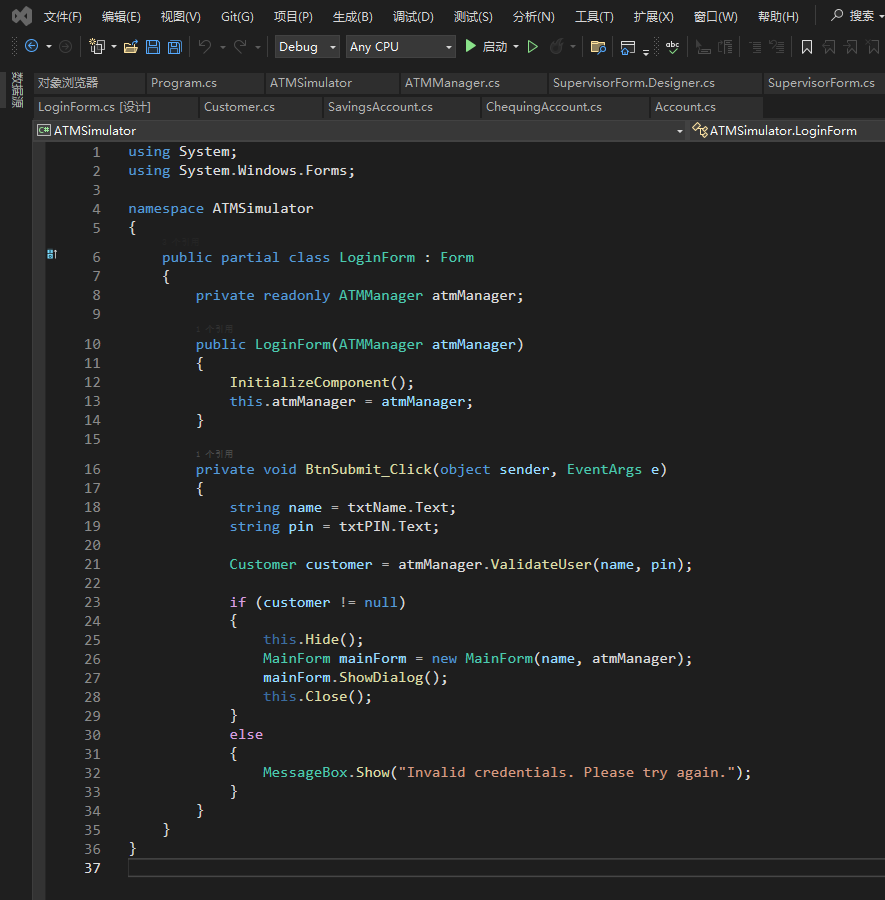


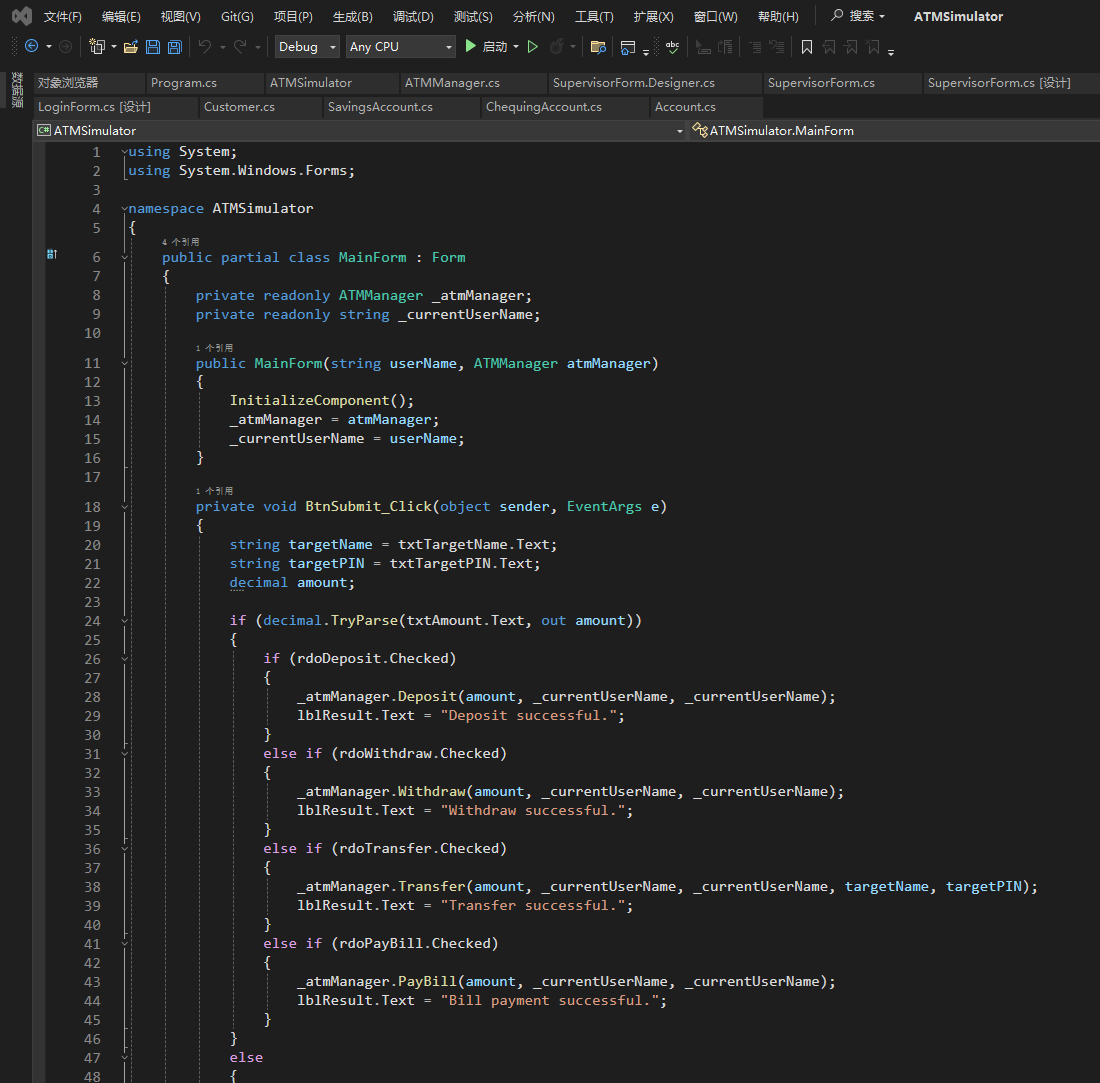


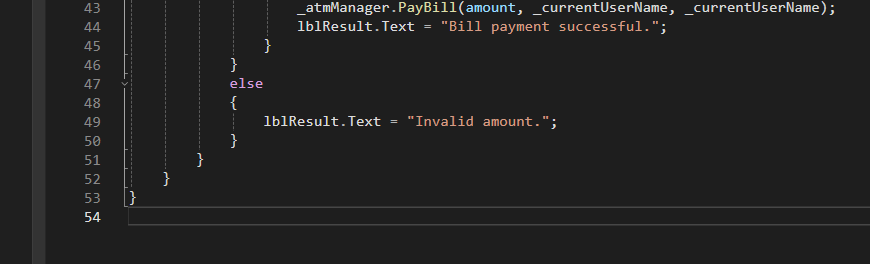


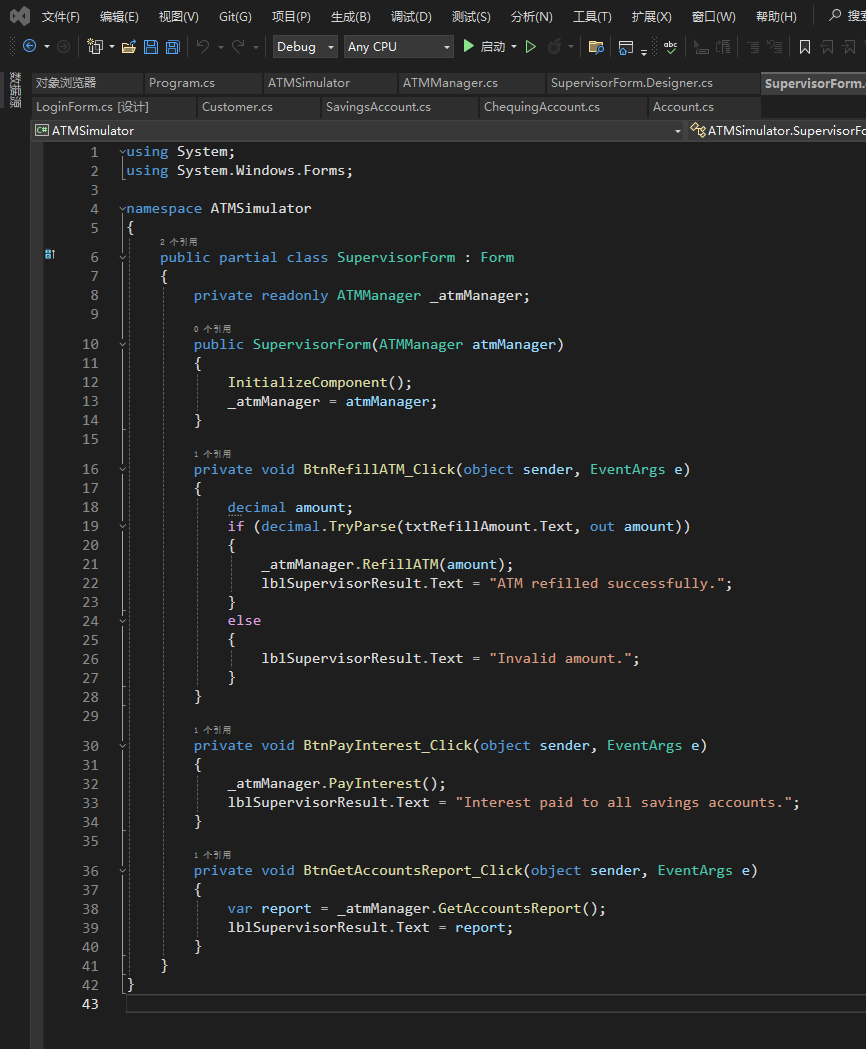
Form Classes (LoginForm, MainForm, SupervisorForm)











Part 3

Create a new project named 'ATMSimulator'

Add class:

Account.cs

Chequing.cs

Savings.cs

Customer.cs

ATMManager.cs

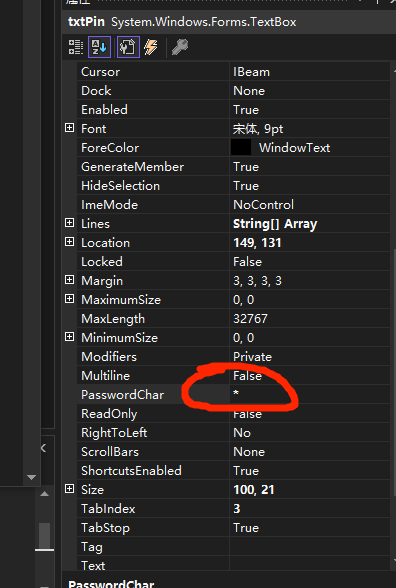
Create user interface:

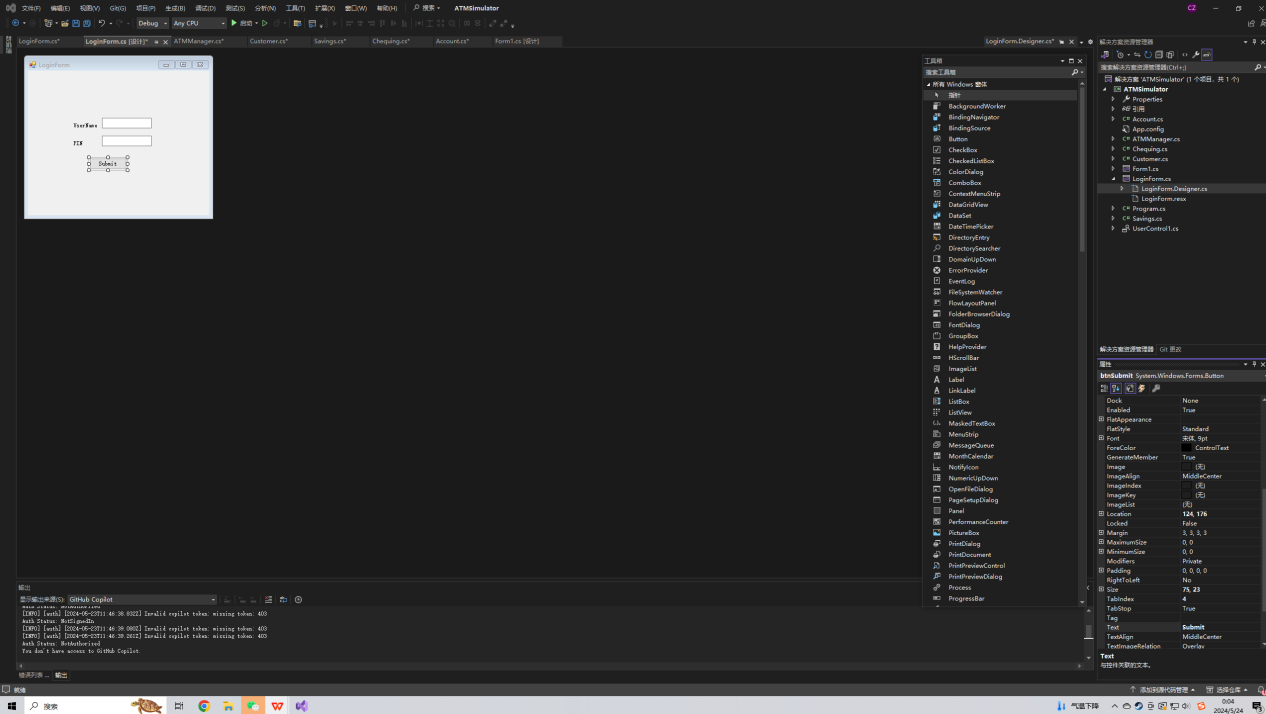
LoginForm.cs

Add two 'Label' controls to display 'UserName' and 'PIN'

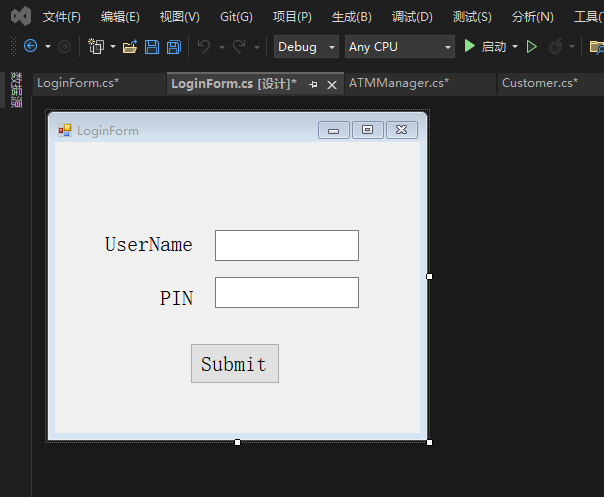
Add two 'TextBox' controls for inputting 'UserName' and 'PIN', and set the 'PasswordChar' property of the PIN input box to \* to hide the input content

Add a 'Button' control for submitting username and PIN





Adjust the position of controls to make the layout beautiful.



Double-click the Submit control button and fill in the code in LoginForm.cs

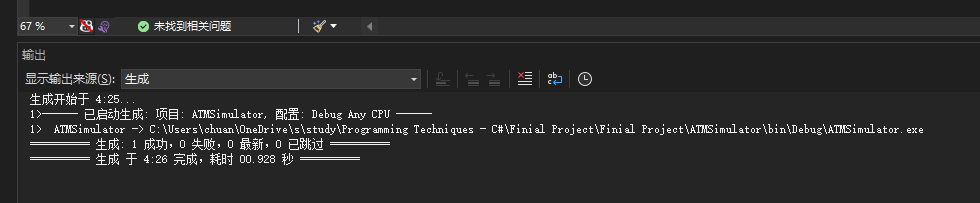
Add a windows form and name it ‘MainForm.cs’

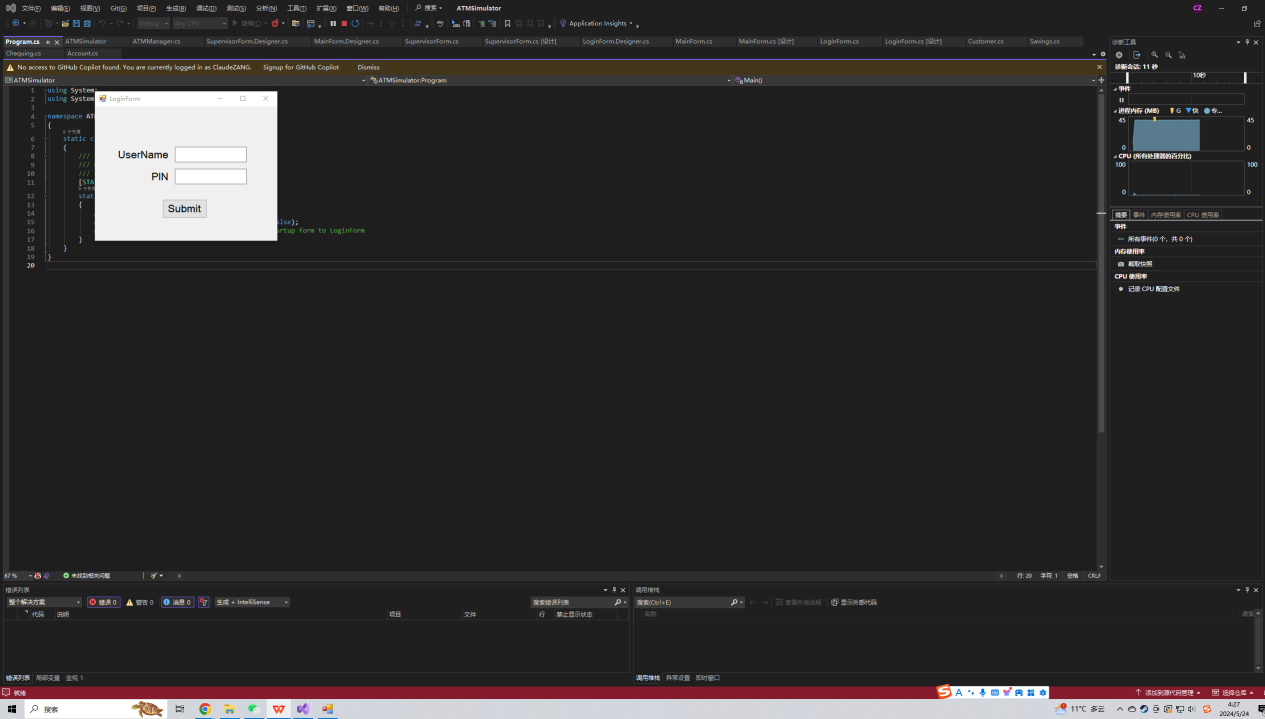
Repeat and add SupervisorForm.cs

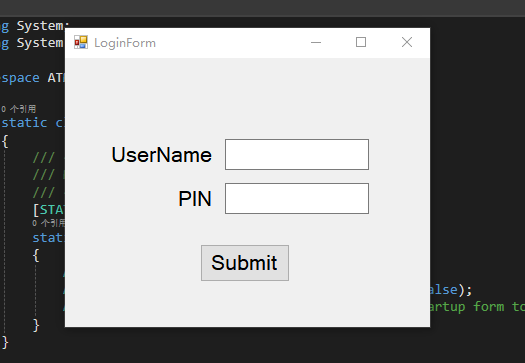
In Windows Forms applications, it is generally recommended to use PascalCase to name controls. This means that the first letter of each word should be capitalized, so I modified the previous naming.(for name and text in property)

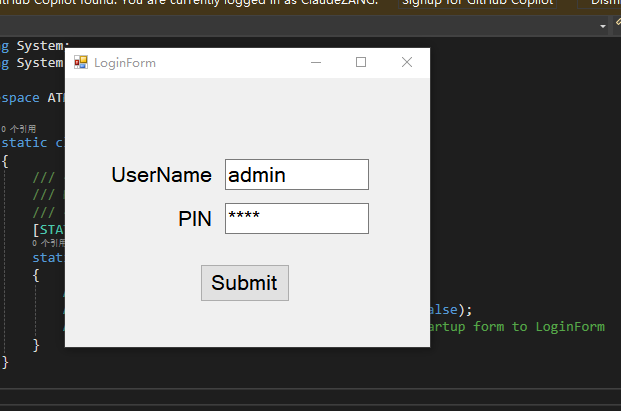
After part1 code, make sure LoginForm is the startup form.(ATMSimulator.Program)

Press Ctrl + Shift + B to compile the entire solution, making sure there are no compilation errors.

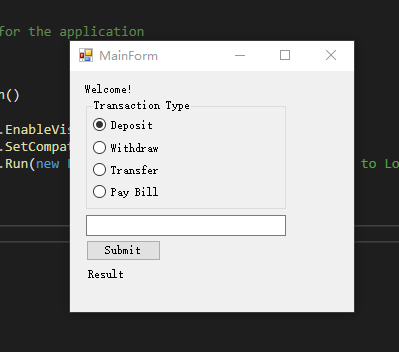


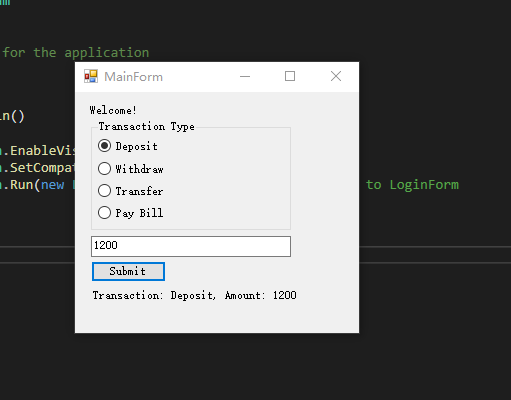




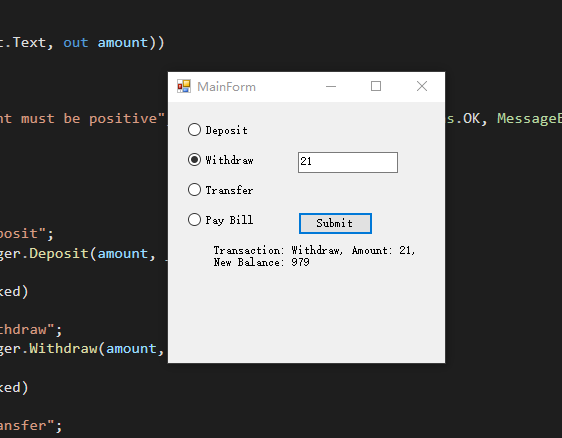


USE ‘admin’ and ‘1234’ as password





In order to solve the problem of displaying balance, I modified the code and interface. Ensure that only positive integers can be entered when depositing and withdrawing money.



After many modifications(5h!), I solved the problem that newbalance cannot be displayed correctly after withdrawing:)

I suddenly discovered that there are files such as customers.txt. I need to modify the code according to the specific requirements of this part of the project.