Advanced Object Programming

420-P34-AS

**Hi-Tech Management System**

**Report**

PRESENTED TO:

QUANG HOANG CAO

PRESENTED BY:

AO HU

XIAO SU

**Nov 29TH, 2015**

**LASALLE COLLEGE**

TABLE OF CONTENTS

[1. Project Description 1](#_Toc436686033)

[1.1. Customer Preliminary Requirement 1](#_Toc436686034)

[1.2. Users and Operations 1](#_Toc436686035)

[2. Project Development Process 2](#_Toc436686036)

[2.1. Analysis 2](#_Toc436686037)

[2.2. Design 2](#_Toc436686038)

[2.3. Implementation 4](#_Toc436686039)

[2.4. Testing 6](#_Toc436686040)

[2.5. Deploying the Application 6](#_Toc436686041)

[3. Conclusion 6](#_Toc436686042)

[3.1. Design Problem 6](#_Toc436686043)

[3.2. Three-tier model Problem 9](#_Toc436686044)

# Project Description

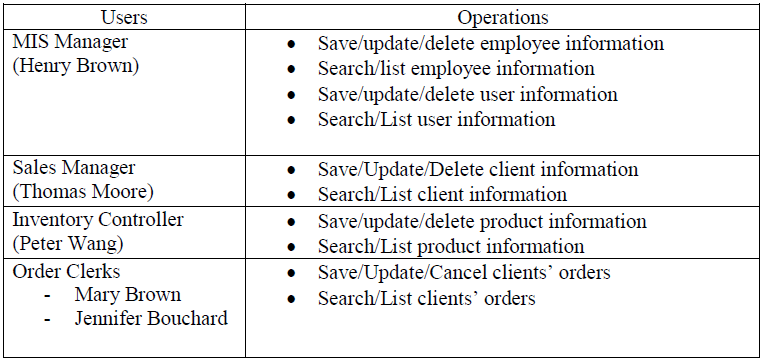
**Hi-Tech Distribution Inc.** is supplying products (computer science books and software) to nearly all the colleges and universities in Quebec. We are required to design and implement a windows-based application named **Hi-Tech Management System** using Visual Studio 2013/2012, C#.

## Customer Preliminary Requirement

1. Each book record should contain the fields: ISBN, Title, UnitPrice, YearPublished and QOH (quantity on hand). Each book must be categorized. Each software must be categorized as well.
2. Each author record should contain the following pieces of information, for example, authorID (for identification in case authors have the same name), first name, last name.
3. Hi-Tech receives products from different publishers (suppliers): Premier Press, Wrox, Murach, Prentice Hall and more.
4. Hi-Tech’s clients: Colleges and Universities in Quebec. Each college/university should contain the information such as name, street, city, postal code, phone number, fax number and credit limit (offered by Hi-Tech in the client’s contract).
5. Order clerks can take clients ‘orders (by Phone, Fax, or Email) and order payments will be made by direct withdrawal from the college/university‘s bank account as specified in the contract between Hi-Tech and the client. The order shipping date will be based on the client’s required date (e.g. one day before the required date).
6. At present, Hi-Tech has two order clerks who are responsible for taking the clients’ orders.

## Users and Operations

Hi-Tech has following clerks who are responsible for different operations.



# Project Development Process

## Analysis

In order to satisfy all the requirement of the company, we need to do:

1. Create classes and fields

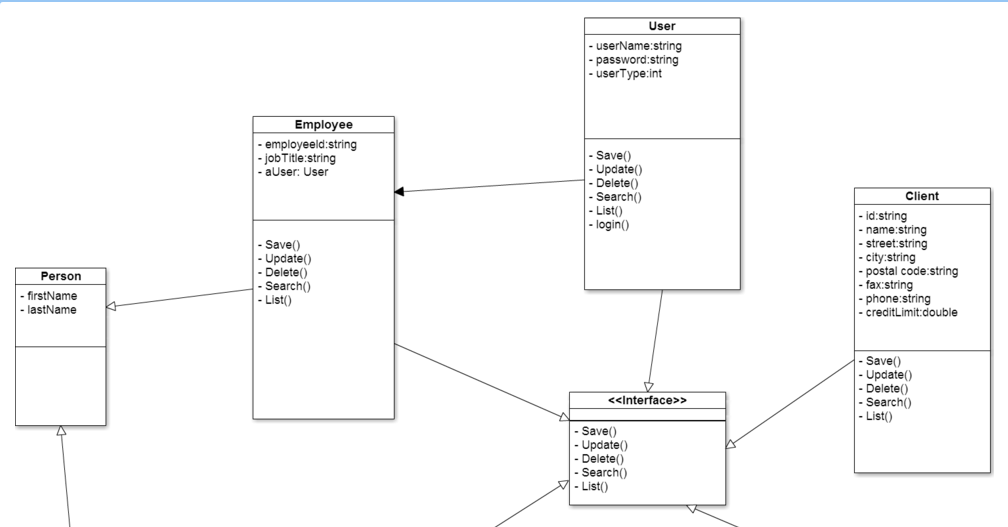
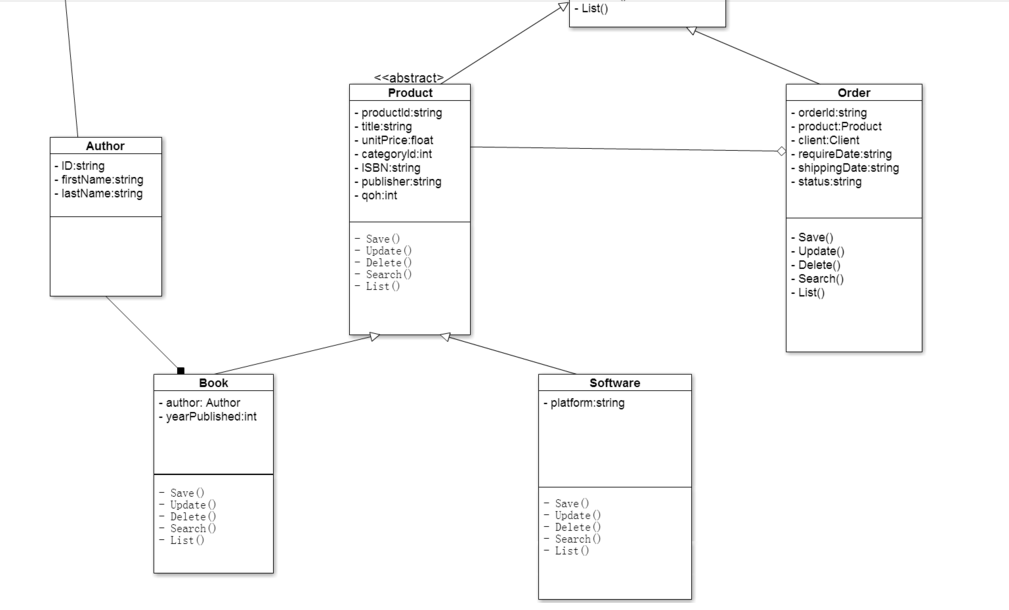
We create classes: Product, Client, Order, Employee, User, Person, Book, Software, and Author. And we define some attributes for different classes.

1. Define class relationship

* Inheritance: Employee and Author classes inherit from Person class. Book and Software classes inherit from Product class.
* Implement: Product, Client, Order, Employee and User classes are implementing an interface which include save, update, delete, search, list methods.
* Composition: Author class is composited in Book class. User class is composited in Employee class.
* Aggregation: Product class and Client class aggregate in Order class.

## Design

1. Design class diagram

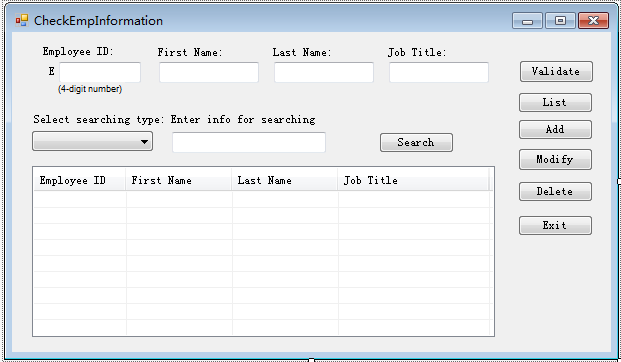


1. Design Three-tier model

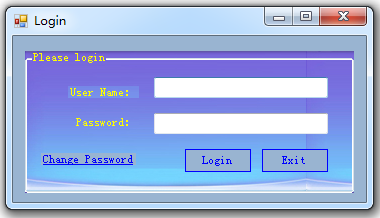
We design three-tier model of GUI, Application Domain (Business) and Data Access. GUI tier only can communicate with Business tier, and Business tier only can communicate with Data Access tier. For Business tier and Data Access tier, we creating and using the Class Library Project to implement, generate DLL file and reference in final project.

1. Design GUI

This is an example of our GUI, normally we design a list view to show all the results of user action. And we design some buttons to do save, update, delete, search and list methods. We design some text box to let user enter or modify information.



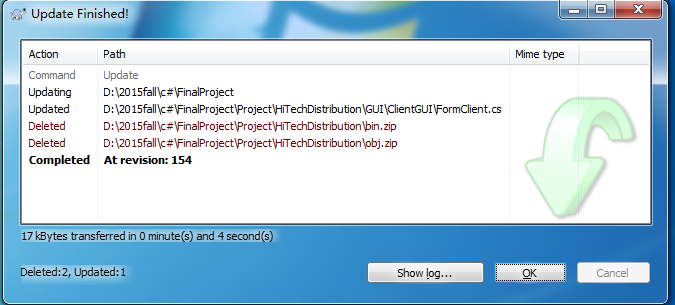
The company also has the security requirement, that when accessing the system, each user is required to enter his/her valid username and password. The user can change the password when necessary. So we design a link named “Change Password”, to let the user change his own password by himself.



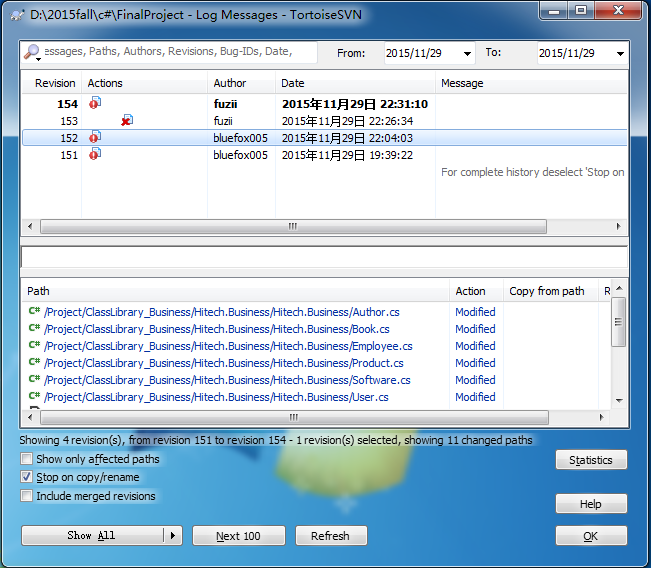
## Implementation

1. Development tools

We use Visual Studio 2012, C# to develop this application. And we use TortoiseSVN software to do version control. Up to now, we have revision 154.



If we find the latest version is not good, we can revert to any previous version as we required.



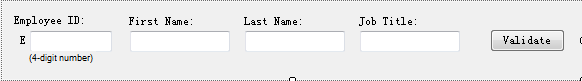
1. Division and individual responsibility

Our divisions are:

* Xiao Su is responsibility to develop Order, Client relevant issues, and initial implement of this project.
* Ao Hu is responsibility to develop Product, Employee, User relevant issues.

## Testing

We create and Use Custom Windows Controls in GUI classes to do validation.



## Deploying the Application

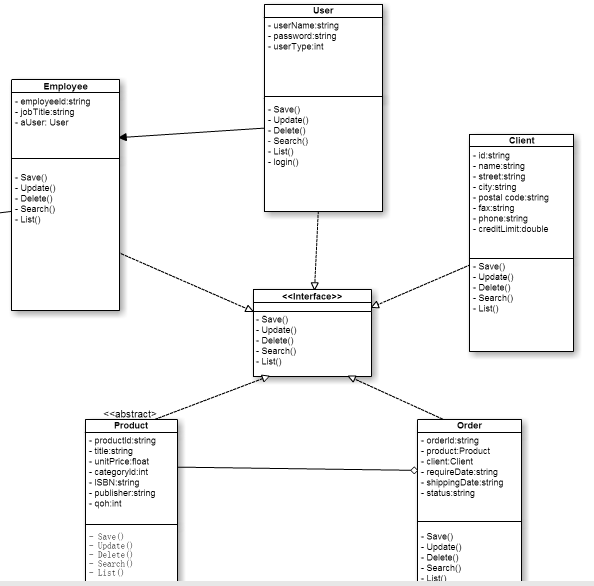
Our application can run in windows 7 platform.

# Conclusion

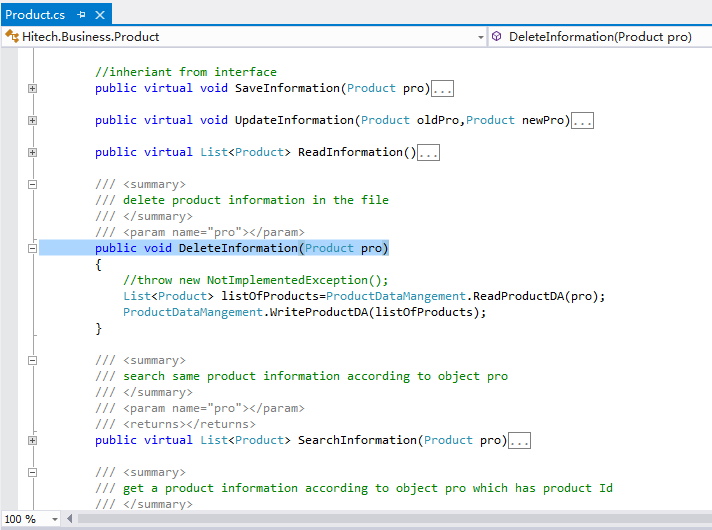
## Design Problem

1. Abstract class should not implement Interface class

Product class is a abstract class, Book and Software classes are inheritance from product.

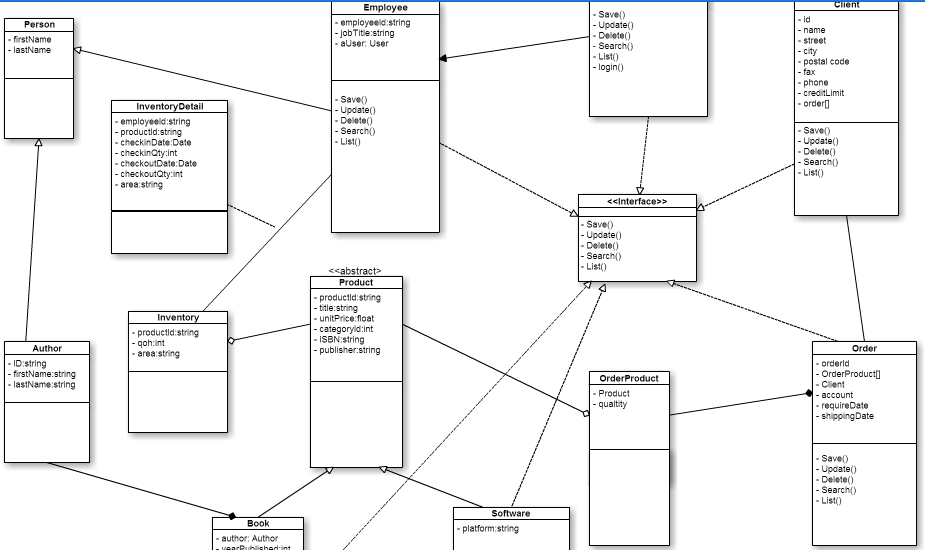


If Product class implement the interface, there are some advantages. For example, I can do delete and search methods in product, and need not to override in Book and Software classes. But the problem is, if we do like this, our partner will not understand you well. If there is only one or two partners, it is ok you can talk with them, and told them how to use your class. But if there are big project, it is impossible to do like this. So the best way to do is trying to make your code easy to read and understand your logic, and use the common logic to write your code, do not make partner confused of your implementation.



1. Quantity should not belong to Product class

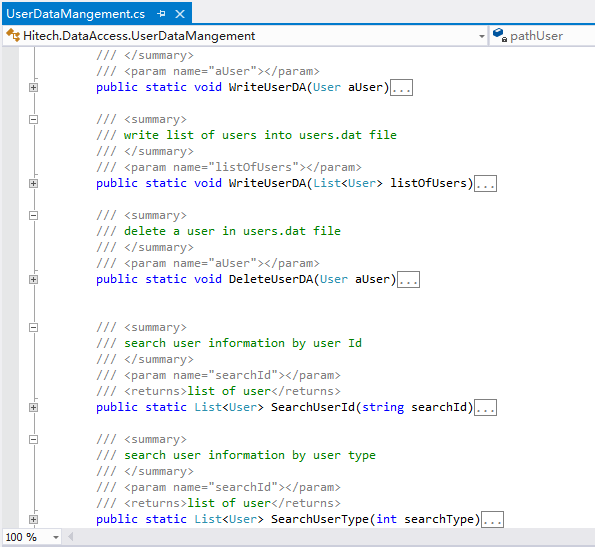
If we do a similar project, it is better let Book and Software class inherent from abstract Product class, and each implement methods from interface. And we need separate attribute qoh from Product class, create a new class named Inventory, which include product id, qoh and area attribute. And we also need to create a associated class named Inventory-Detail to record inventory history information.



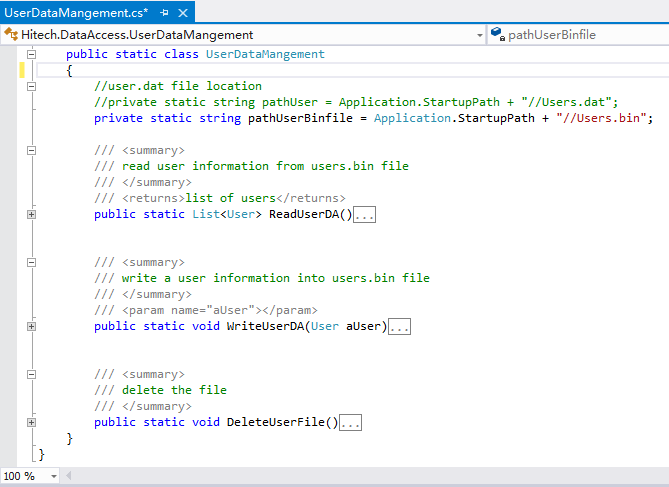
## Three-tier model Problem

1. Data Access layer should only do one simple jobs with file.

At beginning, in data access tier, I write many methods to deal with file. For example, there are a method which write file with one object and another overload method which write file with list of objects. In this way, when I want to change to use binary file, it is not easy to coding. Because there are too many method to change, it is not convenient to implement.



So I change to simplify data access tier like bellow, there is only three methods to deal with file, other methods have moved to business tier, we can get the same result just use these three methods in data access tier. If we want to use another way to save or read file , it is just change these three methods is ok.



1. Cross-reference problem

In our project , business tier and data access tier they are cross-reference in each class library project. The problem is , if I delete a method in data access tier, and this method is used in some methods in business tier. When you try to re-build it , the DLL file referenced disappears, these two projects will create many errors like bellow. It is hard to fix this problem. To make it simple, it is better to use previous version and redo your modification of your class.

