VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**FINAL REPORT OF DESIGN PATTERN**

**APPLY DESIGN PATTERN TO PROJECT**

*Instructor*: **NGUYEN THANH PHUOC**

*Implementer*: **NGUYEN TRUNG TIN – 520H0589**

**LE MINH NHAT – 520H0560**

*Class***: 20H50202**

*Course***: 24**

**HO CHI MINH CITY, 2023**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**

****

**FINAL REPORT OF DESIGN PATTERN**

**APPLY DESIGN PATTERN TO PROJECT**

Instructor: **NGUYEN THANH PHUOC**

Implementer: **NGUYEN TRUNG TIN – 520H0589**

**LE MINH NHAT – 520H0560**

Class**: 20H50202**

Course**: 24**

**HO CHI MINH CITY, 2023**

ACKNOWLEDGEMENT

First of all, I would like to thank Ton Duc Thang University for Software Engineering in the curriculum. In particular, I would like to express my deep and sincere gratitude to Mr. **Nguyen Thanh Phuoc** who wholeheartedly guided and guided me during the study and writing of this essay. Thank you for your kind words and enthusiasm, which helped me to complete my thesis.

Thank you to my friends who have accompanied and encouraged me throughout the research process.

Thank you very much.

**PROJECT COMPLETED**

**AT TON ĐUC THANG UNIVERSITY**

I hereby declare that this is our own project and is guided by Mr. **Nguyen Thanh Phuoc**. The research contents and results on this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data from other authors, other agencies, and organizations, with citations and source annotations.

**If I find any fraud, I will take full responsibility for the content of my project.** Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

*Ho Chi Minh City, April 1st 2023*

*Author*

*Nguyen Trung Tin*

*Le Minh Nhat*

TEACHER'S CONFIRMATION AND ASSESSMENT SECTION

**TEACHER'S CONFIRMATION**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh City, April 1st 2023*

(sign and write name)

**TEACHER'S ASSESSMENT**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh City, April 1st 2023*

(sign and write name)

TABLE OF CONTENT

[ACKNOWLEDGEMENT 1](#_Toc134043816)

[TEACHER'S CONFIRMATION AND ASSESSMENT SECTION 3](#_Toc134043817)

[TABLE OF CONTENT 5](#_Toc134043818)

[LIST OF TABLES, PICTURES, GRAPHS 6](#_Toc134043819)

[CONTRIBUTION 7](#_Toc134043820)

[CHAPTER 1: REQUIRE FUNCTION 9](#_Toc134043821)

[CHAPTER 2:SELF-ASSESSMENT TABLE 18](#_Toc133785800)

LIST OF TABLES, PICTURES, GRAPHS

**LIST OF PICTURES**

[Picture 1. 1 Login form 10](#_Toc134043903)

[Picture 1. 2 Menu form 10](#_Toc134043904)

[Picture 1. 3 Manage Employee form 11](#_Toc134043905)

[Picture 1. 4 Singleton pattern structure 12](#_Toc134043906)

[Picture 1. 5 Singleton pattern class diagram 13](#_Toc134043907)

[Picture 1. 6 Strategy pattern structure 14](#_Toc134043908)

[Picture 1. 7 Strategy pattern class diagram 14](#_Toc134043909)

[Picture 1. 8 Command pattern structure 15](#_Toc134043910)

[Picture 1. 9 Command pattern class diagram 16](#_Toc134043911)

[Picture 1. 10 Repository pattern class diagram 17](#_Toc134043912)

CONTRIBUTION

|  |  |  |
| --- | --- | --- |
| Member | Assignment | Completion |
| 1. 520H0560 - Le Minh Nhat | Code, C#, Word | 100% |
| 1. 520H0589 – Nguyen Trung Tin | Code C#, Word | 100% |

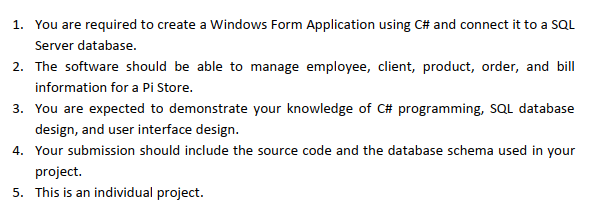
SUMMARY

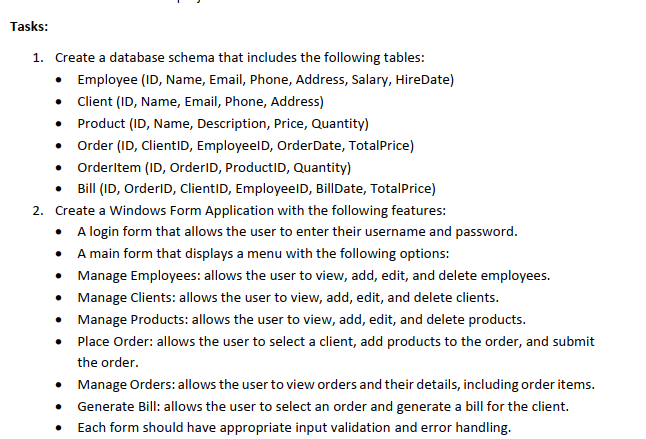
In this report, we will apply some design pattern into my project. We have completed the report and demo in C# . There is source code demo at git under the following links:

<https://github.com/nhatlenlh/FinalProjectDesignPattern.git>

CHAPTER 1: REQUIRE FUNCTION

1.1 The problem of the project

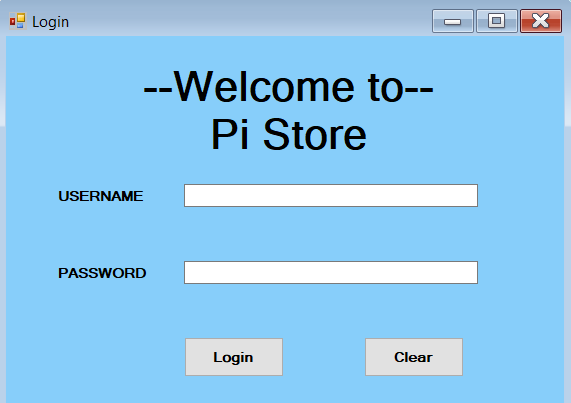




While implementing the project design pattern this time, we applied it to the EmployeeManager subsystem

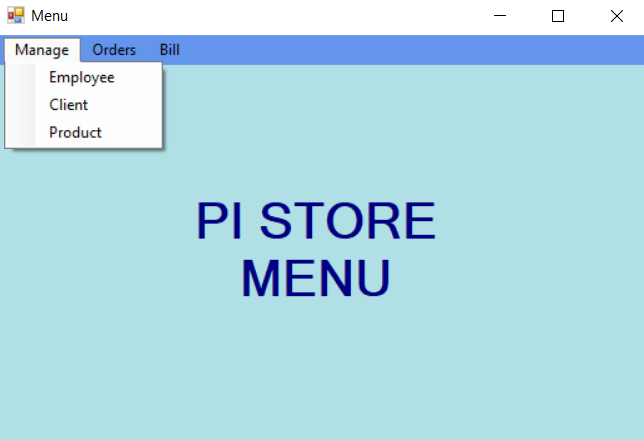
1.2 Architectural design

- Login form



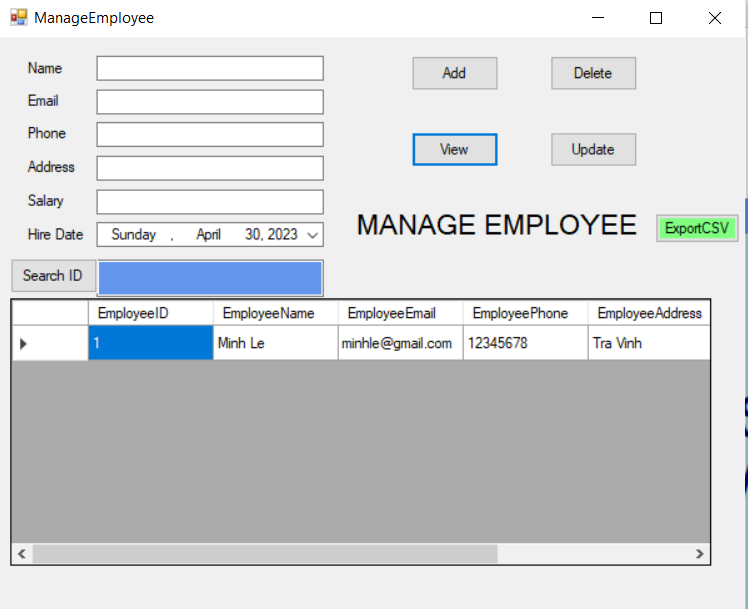
Picture 1. 1 Login form

-Menu:



Picture 1. 2 Menu form

- ManageEmployee:



Picture 1. 3 Manage Employee form

1.3 Detailed design

1.3.1 Singleton pattern

- Intent

-Singleton is a creational design pattern that lets you ensure that a class has only one instance, while providing a global access point to this instance.

- Problem

-When you want to limit resource usage or when you have to deal with

sensitive object whose data cannot be shared by all instances.

- Ensure that a class has just a single instance.

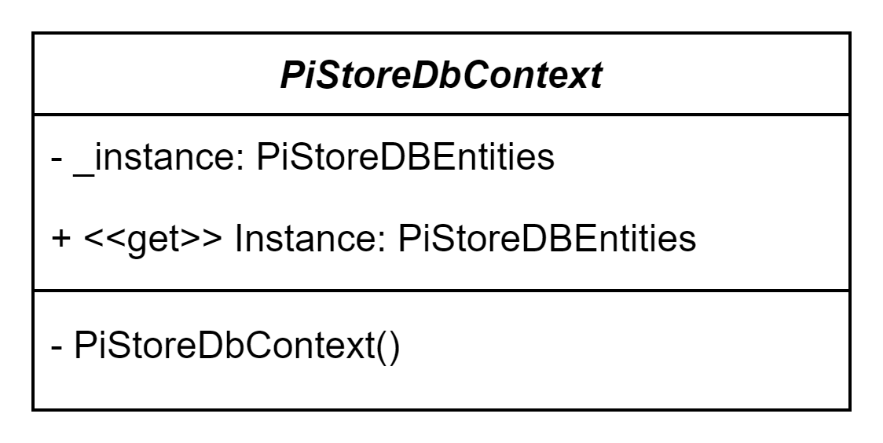
- Provide a global access point to that instance.

**- Structure**



Picture 1. 4 Singleton pattern structure

**- Class diagram**



Picture 1. 5 Singleton pattern class diagram

1.3.2 Strategy pattern

**- Intent**

Strategy is a behavioral design pattern that enables you to define a family of algorithms, separate each one into a different class, and interchangeably use their objects.

**- Problem**

The Strategy Pattern defines a set of algorithms, encapsulates each algorithm, and makes it easy to dynamically change the algorithms within the object.

The Strategy Pattern is to help decouple the handling of a particular function from the object. Then create a set of algorithms to handle that function and choose which algorithm we find most appropriate when executing the program.

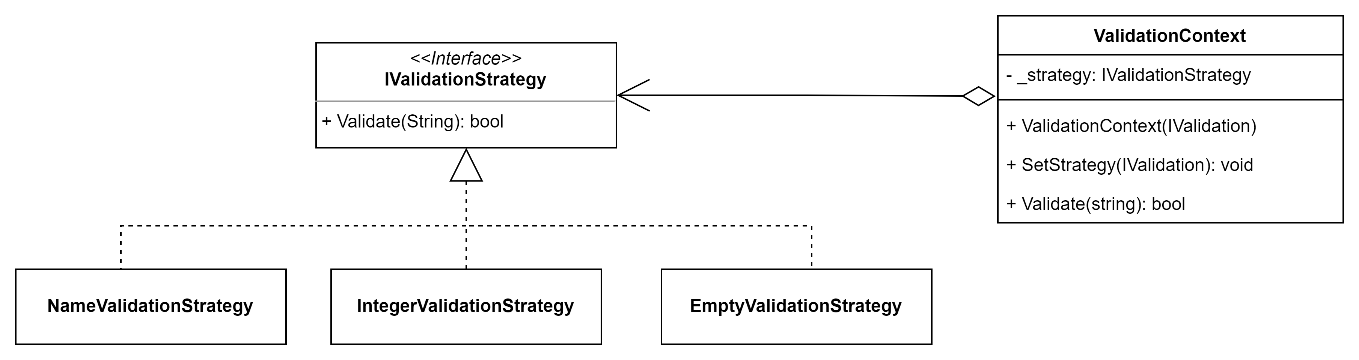
In this project, we use Strategy to validate in login, manage form

**- Structure**



Picture 1. 6 Strategy pattern structure

- **Class diagram**



Picture 1. 7 Strategy pattern class diagram

1.3.3 Command pattern

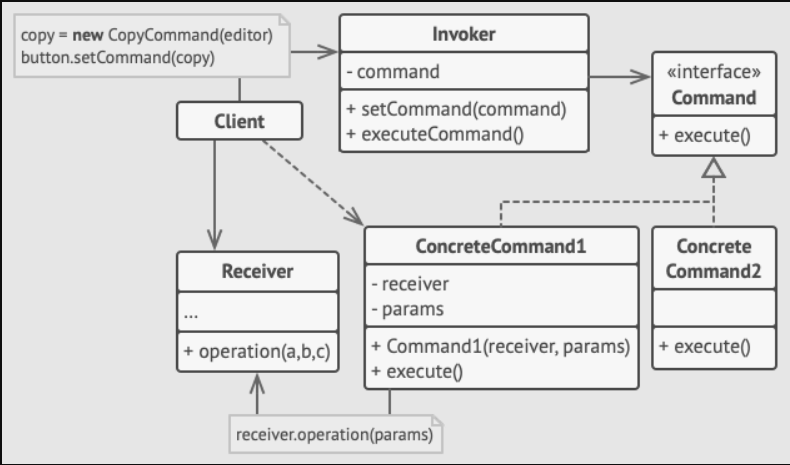
**- Intent**

The command is a behavioral design pattern that turns a request into a stand-alone object that contains all information about the request. With this transformation, you can support undoable operations, delay or queue a request's execution, and pass requests as method arguments.

**-Problem**

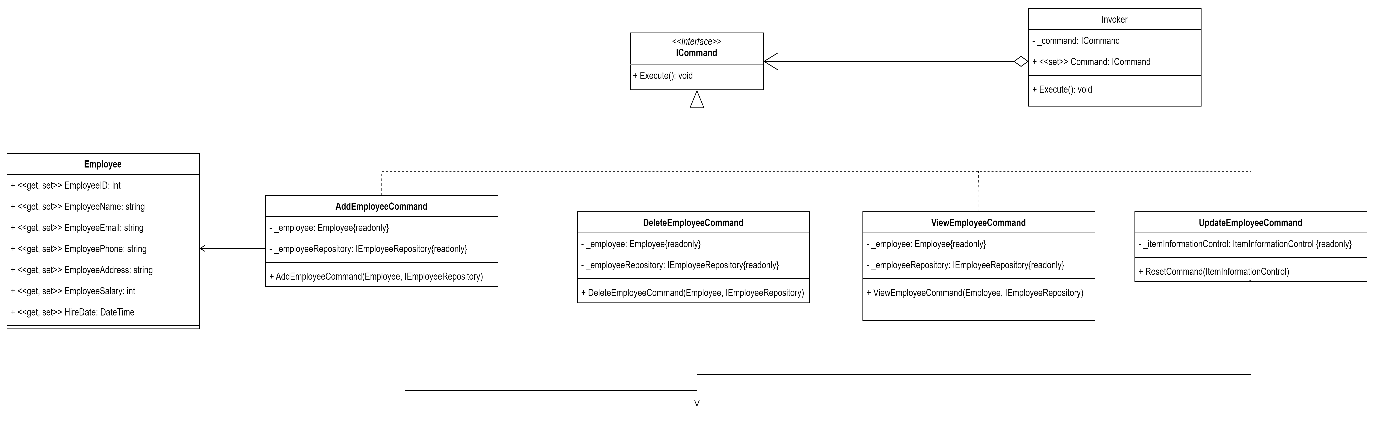
The command pattern allows turning requests into independent objects, which can be used to parameterize objects with different requirements. The Command Pattern allows all requests to an object to be stored in the object itself as a Command object.

- **Structure**



Picture 1. 8 Command pattern structure

**- Class Diagram:**



Picture 1. 9 Command pattern class diagram

1.3.4 Repository pattern

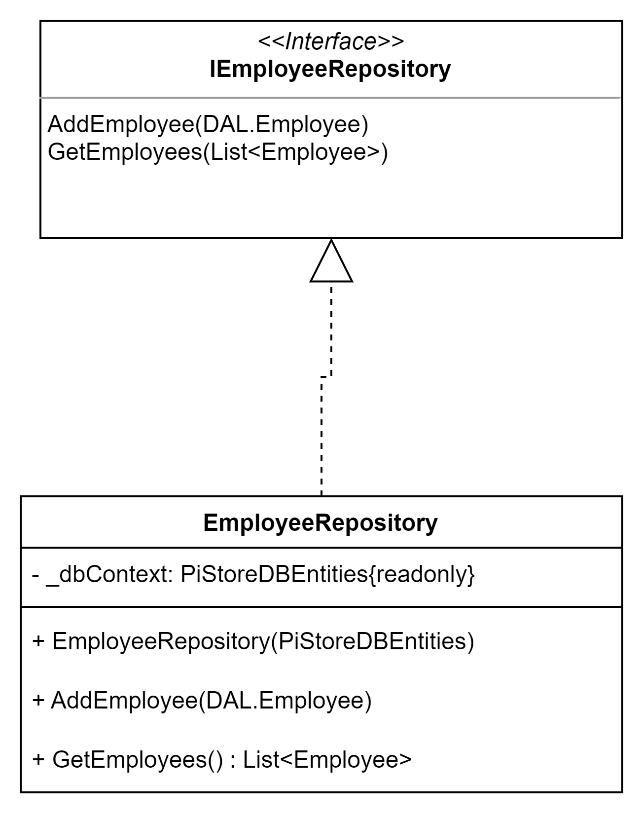
**- Intent**

The Repository pattern is a Domain-Driven Design pattern intended to keep persistence concerns outside of the system’s domain model. One or more persistence abstractions - interfaces - are defined in the domain model, and these abstractions have implementations in the form of persistence-specific adapters defined elsewhere in the application.

**- Problem**

Repository acts as a connection layer between the Business layer and the Model of the application. We need to use the repository pattern when: A single place to change data access and data processing and is responsible for mapping tables to objects.

**- Class diagram**



Picture 1. 10 Repository pattern class diagram

CHAPTER 2: SELF-ASSESSMENT TABLE

The number of pattern using in this report are 4

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Pattern** | **Score** | |
| **Singleton** | **1.5** | |
| **Strategy** | | **1.5** |
| **Command** | | **1.5** |
| **Repository** | | **1.5** |
|  |  | | **1.5** |
| **Total** |  | | **7.5** |

REFERENCE

1. <https://refactoring.guru/design-patterns/singleton>
2. <https://refactoring.guru/design-patterns/strategy>
3. <https://refactoring.guru/design-patterns/command>
4. <https://www.codeguru.com/csharp/repository-pattern-c-sharp/>