**VIET NAM GENERAL CONFEDERATION OF LABOR**

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**REPORT**

**SOFTWARE ENGINEERING**

**FINAL PROJECT**

*Instructor*: **PHẠM THÁI KÌ TRUNG**

*Students*: **NGUYỄN PHAN QUỐC HƯNG – 519H0171**

**DƯƠNG QUANG THẾ – 519H0350**

Group **: 5** -Shift : **T6C2**

**Ho Chi Minh City, 2022**

THANKS YOU

We would like to express our sincere thanks to Mr. PHAM THAI KY TRUNG - lecturer in the subject of SOFTWARE TECHNOLOGY, TON DUC THANG University for enabling us to complete this report.

**PROJECT WAS COMPLETED AT TON DUC THANG UNIVERSITY**

I hereby declare that this is our own project and is guided by teacher Pham Thai Ky Trung;. The research contents and results in 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 of other authors, other agencies and organizations, with citations and source annotations.

If we detect any fraud, we will be fully responsible for the content of our project. Ton Duc Thang University is not involved in copyright that caused by us during the implementation process (if any).

*Ho Chi Minh city, January 12, 2022*

*Author*

*(Sign and write your full name)*



Dương Quang Thế

**The evaluation part of the teacher marks the test**

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

Ho Chi Minh city

(sign and write your name)

TABLE OF CONTENTS

[LỜI CẢM ƠN i](#_Toc76097515)

MỤC LỤC 1

I. INTRODUCTION 2

1. Purpose and scope 2

2. Product overview 2

3. Structure of the document 2

II. PROJECT MANAGERMENT PLAN 2

1. Project organization 2

2. Lifecycle model used 2

3. Hardware and software resource requirements 3

4. Gannt diagram 3

5. Impact of the project on individuals and organizations 3

III. REQUIREMENT SPECIFICATIONS 3

1. Stakeholders for the system 3

2. Usecase diagrams 3

3. Usecase specification 4

IV. ARCHITECTURE 7

1. Architectural style used 7

2. Technology, software and hardware used 7

3. Rationale for architectural styles and models 7

V. DESIGN 7

1. Database design 7

2. Class diagrams 8

3. Sequence diagrams 9

VI. TEST PLAN 13

1. Basic test 13

VII. DEMO 14

1. Database 14

2. Source code 17

3. Link chạy demo 19

1. INTRODUCTION
2. Purpose and scope: This section is intended to assist learners in understanding diagram design, use case specification, customer needs assessment capabilities, software infrastructure, databases, application and system development. The scope of research and implementation for the topic of software engineering focuses on creating software that improves operational efficiency and people's quality of life.
3. Product overview: From the specifications and requirements of the topic, we can describe the class diagram, the usage diagram and understand how to create the system interface and user interface. Clarifying each function of the system, optimizing functions to limit errors to meet user requirements.
4. Structure of the document: Includes introduction, project management plan, requirements specification, architecture, design, test plan to ensure the software works properly and finally the Demo is the result of the product including the basic database, source code and test.
5. PROJECT MANAGEMENT PLAN
6. Project organization: My team will divide the work for 2 people including: analyzing project requirements, drawing diagrams, designing databases, creating main functional code interfaces, testing errors and finally running product demos.
7. Lifecycle model used: Waterfall model because of the use of waterfall model as follows: Simple, easy to understand and use. For smaller projects, the waterfall model works well and yields consistent results. The stages of waterfall model are rigid and precise, one phase is done once, it is easy to maintain.
8. Hardware and software resource requirements:

* About hardware:

Intel Core i5-6500 CPU @ 3.20GHz

Ram 8GB

* About software: SQL server management studio, Visual studio.

1. Gantt Diagram:



1. Impact of the project on individuals and organizations: For organizations, it will be easier to manage employees, have a reward and punishment policy, and spend less time. As for individuals, the work will be easier, less stressful, more professional, and less time wasted on cumbersome manipulations to work more efficiently.
2. REQUIREMENT SPECIFICATIONS
3. Stakeholders for the system: Develop, Accountant, Branch.
4. Usecase diagrams:

Diagram

Description automatically generated

1. Use case specification:

|  |  |  |
| --- | --- | --- |
| Use case ID | UC - 1 |  |
| Use case name | Create import form |  |
| Description | Tạo phiếu nhập kho cho hàng hóa |  |
| Actor | Accountant (kế toán) |  |
| Priority | Must have |  |
| Trigger | Có hàng hóa được chuyển đến |  |
| Pre – condition(s) | * Kho còn trống * Hàng hóa nằm trong danh sách kinh doanh |  |
| Post – condition(s) | * Hàng hóa nhập kho thành công * Các mặt hàng mới nhập sẽ có thông tin |  |
| Basic flow | 1. Kiểm tra hàng hóa 2. Xác thực hàng đúng tiêu chuẩn 3. Thống kê thông tin sản phẩm 4. Nhập vào phiếu xác nhận 5. Hệ thống ghi nhận đơn hàng | 1 → 2 → 3 → 4 → 5 |
| Alternative flow |  |  |
| Exception flow | 2a. Hàng không đúng tiêu chuẩn  2a1. Trả về  2a2. Không nhập phiếu | 1 → 2 → 3 → 4 → 5  2a → 2a1 → 2a2 |

|  |  |  |
| --- | --- | --- |
| Use case ID | UC - 3.1 |  |
| Use case name | Create export form |  |
| Description | Tạo phiếu xuất kho cho hàng hóa |  |
| Actor | Accountant (kế toán) |  |
| Priority | Must have |  |
| Trigger | Có hàng hóa được chuyển đi |  |
| Pre – condition(s) | * Có yêu cầu đặt/mua hàng * Hàng hóa còn đủ số lượng |  |
| Post – condition(s) | * Hàng hóa xuất kho thành công * Hệ thống cập nhật |  |
| Basic flow | 1. Kiểm chứng hàng trong kho 2. Xác thực hàng đúng tiêu chuẩn 3. Chuyển cho giao hàng 4. Nhập vào phiếu xác nhận 5. Hệ thống ghi nhận đơn hàng | 1 → 2 → 3 → 4 → 5 |
| Alternative flow | 2a. Hàng không đúng tiêu chuẩn  2a1. Chọn cái khác | 1 → 2 → 3 → 4 → 5  2a → 2a1 |
| Exception flow | 2b. Không còn hàng  2b1. Báo hết hàng  2b2. Không xuất phiếu | 1 → 2 → 3 → 4 → 5  2a → 2a1  2b → 2b1 → 2b2 |

|  |  |  |
| --- | --- | --- |
| Use case ID | UC - 2.1 |  |
| Use case name | Order |  |
| Description | Đặt mua hàng |  |
| Actor | Branch (các đại lí, nhà bán lẻ, cá nhân) |  |
| Priority | Must have |  |
| Trigger | Có nhu cầu buôn bán sử dụng |  |
| Pre – condition(s) | * Có đủ kinh tế * Có nguồn tiêu thụ |  |
| Post – condition(s) | * Nhận được hàng * Đánh giá sản phẩm |  |
| Basic flow | 1. Đăng nhập vào web công ty bán 2. Lựa chọn sản phẩm phù hợp 3. Bỏ vào giỏ hàng 4. Lựa chọn phương thức thanh toán 5. Xác nhận thanh toán | 1 → 2 → 3 → 4 → 5 |
| Alternative flow | 1a. Quên mật khẩu  1a1. Tạo mật khẩu mới  1a2. Đăng nhập lại  1b. Chưa có tài khoản  1b1. Tạo tài khoản  1b2. Đăng nhập | 1 → 2 → 3 → 4 → 5  1a → 1a1 → 1a2  1b → 1b1 → 1b2 |
| Exception flow | 2a. Không còn hàng  2a1. Đăng xuất  2a2. Không đặt nữa | 1 → 2 → 3 → 4 → 5  2a → 2a1 → 2a2 |

1. ARCHITECTURE
2. Architectural style used:

Layered architecture because

- We just need to understand the layers below the one we're working on.

- Each class can be replaced with an equivalent implementation without affecting other classes.

- A class can be used by several different top level classes.

1. Technology, software and hardware used: SQL, C#; SQL server managerment studio, Visual stuido; Intel Core i5-6500 CPU @ 3.20GHz, Ram 8GB.
2. Rationale for architectural styles and models: The application server provides access to the data to the client. It serves as an interface between the client and one or more database servers. The database server does all the rest of the query processing.
3. DESIGN
4. Database design:

Diagram, schematic

Description automatically generated

1. Class diagrams:

Diagram

Description automatically generated

1. Sequence diagrams:

Box and whisker chart

Description automatically generated with low confidence

Timeline

Description automatically generated with medium confidence

Timeline

Description automatically generated

A picture containing timeline

Description automatically generated

Box and whisker chart

Description automatically generated with medium confidence

Timeline

Description automatically generated with medium confidence

A picture containing timeline

Description automatically generated

Chart, box and whisker chart

Description automatically generated

1. TEST PLAN

Experiment with some basic test cases

1. DEMO
2. Database:



Table

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

Graphical user interface

Description automatically generated with low confidence

1. Source code:

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. Link video demo:

<https://drive.google.com/file/d/1VYDdHlcogO6nyxdAZNARpdRqr-b4n1bV/view?usp=sharing>

link git: <https://github.com/Hung21j/FinalMVC/tree/master>